

Modellering av Populasjonsdynamikk

Introduksjon

I dette opplegget skal vi se på hvordan vi kan *modellere* hvordan populasjonen av en art i et miljø vil endre seg over tid. Med begrepet *modellere* mener vi at vi skal utvikle en forklaring på hvorfor og hvordan populasjonen endrer seg. Modellering av bestander er et viktig verktøy i økologi, hvor man ønsker å forstå seg på samspill mellom organismer og deres økosystem. Man kan også bruke modeller til å gjøre forutsigelser for å bedre forvalte og ivareta naturen. Samtidig kan slike modeller være viktig for å forstå effekten av endringer til gitte økosystem over tid, for eksempel fra klimaforandringer eller på grunn av migrasjon og evolusjon.

Ingress: Om fisk og hai i Middelhavet

I midten av 1920 årene forsket den italienske biologen Umberto D'Ancona bestandene av fisk i middelhavet. Det var ikke mulig å direkte måle mengden fisk i havet, så man måtte gjøre anslag basert på den informasjonen tilgjengelig. En viktig kilde til informasjon var mengden fisk som ble fanget i komersiell fiske og hvordan disse mengden endret seg over tid.

Et dataset som var av spesiell interesse for D'Ancona dreier seg om mengden av det som ble fisket som var matfisk, og hvor mye som var større rovfisk som hai. Hai og lignende arter var gjerne ikke like lønnsomt å fiske for fiskere, og det ble sett på som et biprodukt. Det ble holdt telling på andelen av denne "upopulære" fangsten fiskerne fikk hvert år.

D'Ancona var spesielt interessert i hvordan disse tallene endret seg iløpet av første verdenskrig. Under krigen var komersiell fiske satt på vent, og det ble fisket drastisk mindre fisk fra middelhavet under årene krigen pågikk. Det viste seg at dette ikke bare påvirket mengden fiske som ble gjort, men også forholdet mellom "matfisk" og hai. Tallene er som følger:

År	Andelen hai og lignende
1914	11.9 %
1915	21.4 %
1916	22.1 %
1917	21.2 %
1918	36.4 %
1919	27.3 %
1920	16.0 %
1921	15.9 %
1922	14.8 %
1923	10.7 %

Før og etter krigen holdt andelen seg forholdsvis stabil, men under krigen økte mengden hai som ble fanget drastisk. D'Ancona var spesielt interessert i disse tallene fordi man skulle tro at mengden fisk i havet økte betraktelig når mengden komersiell fiske gikk ned. Men disse datene, og andre data han fant, indikerte at mengden fisk ikke økte så mye, men at mengden hai økte drastisk.

Dette var et meget overraskende funn, og førte til utviklingen av den første rovdyr-bytte *modellen*, som forklarte disse observasjonene. Dette kommer vi tilbake til i slutten av opplegget.

Om dette opplegget

Målet med dette opplegget er å vise hvordan man kan bruke matematikk til å lage slike populasjonsmodeller. Vi kommer til å starte med veldig enkle modeller, og bygge dem opp gradvis så de har mer komplisert oppførsel. På den måten blir modellene mer realistiske.

Før vi setter igang med selve modelleringen dekker vi litt matematikk og programmering, så det blir lettere å forstå matematikken som ligger bak modellene og å diskutere denne. Derimot er det ikke matematikken, eller programmeringen, som er viktig, men forståelsen den kan gi oss.

Etterhvert som vi jobber oss igjennom dette opplegget er det derfor viktig å stoppe opp, reflektere og diskutere med andre rundt modellene og hva de forteller oss. Jobb derfor aller helst i par med dette opplegget.

1) Kort om tallfølger

Tall som kommer etter hverandre i en bestemt rekkefølge, kaller vi en tallfølge. Et eksempel kan være de fem første oddetallene i stigende rekkefølge:

$$1, 3, 5, 7, 9$$

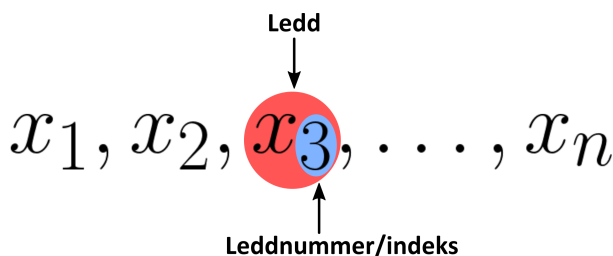
Dette er en tallfølge som består av fem *ledd*. Vi kaller også en slik tallfølge for *endelig*, i motsetning til en uendelig en. Om en tallfølge er veldig lang skriver vi gjerne ... for å slippe å skrive alle leddene, for eksempel:

$$1, 3, 5, \dots, 99,$$

være tallfølgen av alle oddetall under 100. Hvis vi ønsker å lage en uendelig tallfølge av alle oddetall skriver vi det gjerne som

$$1, 3, 5, 7, 9, \dots$$

Når vi skal snakke om tallfølger er det vanlig å kalle hvert enkelt tall i rekka for et ledd. Vi gir gjerne et navn til hele følget, for eksempel x , og da kan vi kalle første ledd for x_1 , andre ledd for x_2 og så videre. Tallet under kaller vi for *leddnummeret*, eller for *indeksen*.



Rekursive Formler for Tallfølger

En *rekursiv* formel spesifiserer hvordan vi kan regne ut et ledd i en tallrekke ved hjelp av de tidligere leddene. Ordet "rekursiv" noe som gjentar seg, eller som inneholder seg selv. La oss for eksempel se på oddetallene igjen. Hvordan lager vi tallfølget med oddetall? En måte å gjøre det på er for hvert nye ledd i rekka, ta det forrige, og pluss på 2. Dette kan vi skrive som den rekursive formelen:

$$x_{i+1} = x_i + 2.$$

Altså. Om vi kjenner til ledd x_i finner vi *neste ledd*, x_{i+1} , ved å pluss på to.

Om vi blir gitt en slik rekursiv formel trenger vi også en startbetingelse for å faktisk finne en tallfølge, fordi vi må starte ett eller annet sted. Vi må altså bli gitt første ledd i følget. For å lage oddetallene kan vi derfor si at $x_1 = 1$. Deretter kan vi bruke formelen til å regne oss oppover:

$$x_2 = x_1 + 2 = 1 + 2 = 3,$$

$$x_3 = x_2 + 2 = 3 + 2 = 5,$$

$$x_4 = x_3 + 2 = 5 + 2 = 7,$$

OSV.

Merk at for å finne tallfølget er vi nødt til å regne oss oppover ledd for ledd, vi kan ikke "hoppe" over noen ledd, fordi den rekursive formelen krevet at vi hele tiden kjenner forrige ledd. Altså bygger vi opp rekka ledd for ledd.

I dette eksempelet startet vi rekka på x_1 , altså det første leddet. Men i resten av dette opplegget vil vi istedet si at det første leddet i rekka er x_0 . Dette gjør vi fordi leddet x_i skal representere en populasjon ved år i . Altså er x_0 bestanden ved "år 0". Samtidig er det vanlig å begynne å telle fra 0 i programmering, så x_0 er et naturlig startpunkt.

Programmere Rekursive formeler

La oss nå se hvordan vi kan programmere en rekursiv formel. Først må vi importere all funksjonalitet vi trenger, dette gjør vi med kommandoen under:

In []:

```
# Konfigurer Python så vi har alt vi trenger
from pylab import *
```

Vi skal nå lage et kort program som kan regne ut tallfølget vårt. Vi viser koden først, og så forklarer vi den

In []:

```
N = 5
x = zeros(N)

x[0] = 1

for i in range(N-1):
    x[i+1] = x[i] + 2

print(x)
```

Her definerer vi først to variabler, N er antall ledd vi ønsker å finne. Deretter definerer vi selve tallfølgen x , som består av N ledd, dette gjør vi med kommandoen `zeros(N)`. Når vi laget følgen på denne måten blir alle ledd satt til 0.

Deretter setter vi startleddet x_0 . For å referere til et gitt ledd kan vi bruke firkantparenteser, så `x[0]` blir det samme som x_0 .

Deretter bruker vi en løkke for å regne oss oppover i tallfølget. Ettersom at vi har N ledd totalt, og vi har allerede satt start-leddet, må vi løkke over de resterende $N - 1$ leddene. Løkken `for i in range(N-1):` gjør dette for oss, i vil nå øke for hver gang løkka kjører, så først vil $i = 0$, så $i = 1$, så $i = 2$ og så videre.

Oppgave 1a) Tolke kode

- Gå igjennom koden linje for linje og forklar hva hver linje gjør og hvorfor vi trenger den
- Prøv å endre startleddet til $x_0 = 0$. Hva skjer? Forklar.

Oppgave 1b) Doblinger

Se på den rekursive formelen

$$x_{i+1} = 2x_i.$$

- Hva forteller denne formelen oss?
- Hvis vi skal lage et tallfølge fra formelen med startledd $x_0 = 1$, hvordan vil dette følget se ut?
- Fyll inn kode i cellen under og sjekk om du hadde rett. (Fyll inn kode alle steder det står . . .)

In []:

```
N = 10
x = zeros(N)

x[0] = ...

for i in range(N-1):
    x[i+1] = ...

print(x)
```

Oppgave 1c) Renteberegninger

Si at du ønsker å spare penger i banken. Ved starten av hvert år gjør du et innskudd på 2000 kroner. I tillegg får du 2.5 årlig rente. Hvor mye penger vil du ha spart opp etter 10 år?

- Om vi lar x_i være saldo på konto etter i år. Forklar hvorfor problemstillingen kan skrives som den rekursive formelen

$$x_{i+1} = 1.025 \cdot (x_i + 2000).$$

- Utifra denne formelen, hva bør x_0 være?
- Fyll inn koden under for å regne ut saldoen på konto over 10 år (Fyll inn der det står . . .)

In []:

```
N = ...
x = zeros(N)

x[0] = ...

for i in range(N-1):
    x[i+1] = ...

for i in range(N):
    print(f"Etter {i:2} år er det {x[i]:5.0f} kroner på konto.")
```

- Etter du har fått til koden over kan du kjøre denne koden for å plote saldoen over tid

In []:

```
plot(x, 'o-')

# Pynte på plott
xlabel('Antall år')
ylabel('Saldo')
grid()
axis((0, 10, 0, 25000))

show()
```

2) Modellering av Populasjonsvekst

Vi skal nå begynne å se på hvordan vi kan modellere veksten av en populasjon av organismer. Vi kommer til å begynne veldig enkelt, med modeller som er altfor enkle til å være nyttige, men etterhvert skal vi komme frem til en modell som er mer realistisk.

For å gjøre diskusjonen rundt modelleringen litt mer konkret lager vi oss et lite scenario vi skal prøve å modellere. Anta at vi har en liten øy ute i fjorden der det er rikelig med planteliv, men ikke så mye dyr. Vi slepper ut en liten bestand av harer på øya, f.eks 10 stykk. Harene har altså rikelig med mat, og ingen konkurranse. Under disse forholdene er det naturlig å tenke seg at populasjonen vil vokse. La oss prøve å lage en modell av denne veksten.

Konstant Vekst

Ettersom at vi tror populasjonen vil vokse kan vi begynne med å lage en rekursiv formel som inkluderer et vekstledd. Si for eksempel at vi tror det iløpet av det første året blir født 5 ny harer, da vil formelen:

$$x_{i+1} = x_i + 5,$$

beskrive dette.

Dette er en veldig enkel model, men la oss se hvordan den oppfører seg før vi gjør noe mer avansert. De første 5 årene blir utviklingen som følger:

In []:

```
N = 6
x = zeros(N)

x[0] = 10

for i in range(N-1):
    x[i+1] = x[i] + 5

for i in range(N):
    print(f"Etter {i} år er det {x[i]:.0f} harer på øya")
```

For å svare på om dette er en *god* modell eller ikke må vi tenke på hva målet med modellen er, nemlig å forutsi hvordan bestanden endrer seg over tid. For å faktisk svare på om det er en god modell eller ikke må man derfor sammenligne med virkelighete, og *etterprøve* forutsigelsene. Derimot er det jo ikke slik at vi faktisk har en øy hvor vi har sluppet ut harer på, men vi kan jo innbille oss at vi hadde gjort det.

Si at vi drar tilbake til øya etter 1 år, da finner vi at det nå er 17 harer på øya. Altså bommet vi med 2 harer. Vi kan nå tilpasse eller oppdatere modellen vår utifra denne ekstra informasjonen. Vi sier derfor at det hvert år blir 7 flere harer, istedenfor 5. Dette endrer egentlig ikke stort i selve modellen, vi kan derfor gå et steg lengre å skrive den på en *generell form*:

$$x_{n+1} = x_n + c.$$

Her er c en variabel som vi kan endre avhengig av hvor stor den årlige veksten skal være. Vi kaller en slik variabel gjerne for en *parameter*, og vi kan justere denne opp og ned for å få modellen vår til å passe bedre.

Modellen vi har lagd nå kan vi kalle for en *konstant vekst* modell, fordi veksten er den samme hvert år. Dette fungerer bra det første året, kanskje de første 2-3 årene tilogmed. Men etterhvert som populasjonen øker forventer vi jo at veksten skal øke og, 100 harer føder jo flere barn enn 10 harer.

Oppgave 2a) Diskusjon

Diskuter med sidemannen

- Hvilke antagelser ligger til grunn for denne modellen?
- Hvilke begrensninger har modellen?
- Har du noen idéer for hvordan vi kan forbedre modellen?

Populasjonsavhengig vekst

Det er naturlig å tenke seg at antall nye harer som fødes avhenger av hvor mange harer som allerede lever. Det er jo tross alt harene som får barn.

For å gjøre modellen vår bedre bør vi derfor endre fra et konstant vekstledd, til et vekstledd som avhenger av populasjonen selv. Den enkleste formen for et slikt ledd blir som følger:

$$x_{n+1} = x_n + b \cdot x_n.$$

Her er leddet $b \cdot x_n$ et vekstledd som skalerer med bestanden. Hvis bestanden er stor, så er x_n stor, og det blir født flere harer, hvis bestanden er lav, så er x_n lav, og det blir født færre. Her er b en konstant som sier hvor mange harer det blir født i gjennomsnitt per hare, vi kan kalle denne parameteren for *fødselsraten*.

Oppgave 2b) Modellere

- Fyll inn koden under så den modellerer veksten av harer over 10 år, om vi antar at de har en fødselsrate på 22%

In []:

```
N = ...
x = zeros(N)

x[0] = 10
b = ...

for i in range(N-1):
    x[i+1] = ...

plot(x, 'o-')
xlabel('År')
ylabel('Harebestand')
axis((0, N, 0, 80))
show()
```

- Hva slags kurve får vi denne gangen. Forklar forskjeller og likheter fra den forrige modellen.
- La oss si vi drar tilbake til øya etter 10 år og estimerer at det nå er 95 harer på øya. Hvilken fødselsrate b må vi bruke i modellen vår for å tilpasse modellen best mulig?

Dødsfall

Når vi snakker om populasjonsutvikling over mange år er det urimlig å bare snakke om harer som blir født, faktumet er jo at noen vil død også. La oss derfor videreutvikle modellen vår ved å legge til et ledd til

$$x_{n+1} = x_n + \underbrace{b \cdot x_n}_{\text{vekstterm}} - \underbrace{d \cdot x_n}_{\text{dødsterm}}.$$

Her har vi lagt til leddet $d \cdot x_n$, og merk at vi trekker fra dette leddet, ergo er det et ledd som vil få populasjonen til å synke. Vi har her kalt dette et dødsledd. Ettersom at vi tenker på harene som på en øy er dødsfall den eneste måten at bestanden synker. For et annet økosystem kan derimot dette leddet også inkludere migrasjon ut av omerådet.

Merk at vi nå kan slå sammen veksttermen og dødstermen i modellen vår:

$$x_{n+1} = x_n + (b - d)x_n.$$

Her er b og d konstante parametere, så vi kan slå dem sammen til en ny parameter r , som vi kan kalle *netto vekst*.

$$x_{n+1} = x_n + r \cdot x_n.$$

Merk at modellen vår dermed egentlig ikke har endret seg, vi har bare endrer tolkningen av parametrene i modellen vår. Merk også at dersom det er flere dødsfall en fødsler vil netto veksten r bli negativ. Selv om vi skriver det som en addisjon kan altså denne modellen fange både populasjonsvekst og fall.

Oppgave 2c) Diskusjon

- Hva slags type vekstkurve får vi ut av denne modellen?
- Er denne typen vekst rimelig?
- Foreslå noen tilfeller der modellen kan være gyldig, og andre tilfeller der den ikke er gyldig

Oppgave 2d) Sulten Ørn

La oss si at det etterhvert kommer en sulten ørn som oppdager øya og velger å bli boende der. Hvert år spiser ørna 15 harer.

- Hvordan kan du inkludere ørnas effekt på modellen?
- Fyll inn koden under så vi modellerer veksten både før og etter ørna introduseres. Merk at vi bruker en *if-test* for å sjekke om ørna er tilstede eller ikke. Anta en fødselsrate på $b = 0.22$, en dødsrate på $d = 0.03$. Ørna kommer til øya etter 7 år.

In []:

```
N = 21
x = zeros(N)

# Parametere
x[0] = 10
b = ... # Fødselsrate
d = ... # Dødsrrate
r = b - d # Netto populasjonsvekst
T = ... # Året ørna dukker opp

for i in range(N-1):
    if i < T:
        x[i+1] = ...
    else:
        x[i+1] = ...

plot(x, 'o-')
xlabel('År')
ylabel('Harebestand')
axis((0, N, 0, 140))
axvline(T, color='darkred')
show()
```

- Hva skjer når ørna oppdager øya med smakfulle harer?
- Endre så ørna istedet dukker opp 13 år etter vi satt ut harene. Hva skjer nå?
- (Utfordrende) Om du prøver å tilpasse verdiene i modellen så den havner i en slags likevekt oppdager du kanskje at dette er veldig vanskelig. Diskuter med sidemannen og se om du kan forstå hvorfor.

Oppgave 2e) Bakterievekst

Så langt har vi snakket om harer på en øy, men et veldig vanlig system å bruke modellene vi har sett på er bakteriekulturer. I et laboratorium kan vi dyrke frem bakterier i petriskåler, og se på hvor raskt dette går.

Bakterier formerer seg ved en spesiell form for celledeling, hvor en bakterie deler seg i to. I modellen svarer dette altså til at bestanden dobblen seg for hver celledeling. Det er vanlig i biologien å snakke om tiden det tar for bakteriene å doble seg, fremfor en vekstrate. Derimot kan vi knytte de to sammen ved at vekstraten kan skrives som

$$r = \frac{\ln 2}{\lambda},$$

der λ er tiden det tar for bakteriene å doble seg.

Hvor ofte bakterier deler seg avhenger av arten, og forhold som temperatur og pH. Men i praksis er det langt raskere enn hvert år. For å tilpasse modellen vår endrer vi derfor fra å regne antall hvert år til antall hver time. La oss anta at en gitt type E. Coli dobler seg hver 37.5 time.

- Fyll inn koden under så vi finner bakterieveksten over en uke, altså 7 døgn dersom vi starter med 100 tusen celler.

In []:

```
N = ... # Antall ledd (hvert ledd svarer til én time)
x = zeros(N)

x[0] = ...
doblingstid = ...
r = log(2)/doblingstid

for i in range(N-1):
    x[i+1] = ...

plot(x)
xlabel('Timer')
ylabel('Bakterier (i antall tusen)')
grid()
show()
```

- Om du leser av figuren, hvor mange timer tar det før det har blitt mer enn én million bakterier i kulturen?

3) Bærekraftig Vekst

En av antagelsene vi har gjort i alle modellene vi har sett på så langt er at det alltid er rikelig tilgjengelig med mat for organismene. Dermed tillater vi bestanden i modellen å vokse mot uendelig. Dette er jo åpenbart en dårlig antagelse om vi lar modellen gå over veldig lang tid, men på kort tidsskala er det mer rimelig.

La oss derimot si at vi ønsker å si noe om hvordan bestanden vil se ut på lengre sikt, da må vi bygge inn et element av ressurser inn i modeller vår, vi begynner nå å snakke om *bærekraftig vekst*.

Modellen vi har lagd så langt passer fint så lenge det er godt med ressurser. Men etterhvert som bestanden øker vil det til slutt gå tomt for mat. I tillegg er det andre faktorer som mangel på ly, spredning av sykdomer, konkurranse mellom individer og så videre. Kort fortalt, når en bestand vokser seg stor vil det begynne å oppstå problemer.

For å modellere dette vil vi anta at et gitt økosystem har en gitt bæreevne M . Bæreevnen er et mål på antall individer økosystemet maksimalt kan takle over lengre tid. Om antall individer begynner å nærme seg bæreevnen M vil der ikke lenger være nok ressurser til alle, og populasjonsveksten vil avta. Måten vi viser dette matematisk er ved å si at vekstraten r ikke lenger er konstant, men skalerer med populasjonen som følger:

$$r(x_i) = R \cdot \left(1 - \frac{x_i}{M}\right).$$

Vi har nå parameteren R , som er et mål på vekstraten dersom det er rikelig med ressurser, mens den faktiske vekstraten r blir skalert med populasjon.

Oppgave 3a) Diskusjon

- Hva blir vekstraten (r) hvis populasjonen er mye mindre enn bæreevnen (M)?
- Hva blir vekstraten (r) hvis populasjonen er veldig nærme bæreevnen (M)?
- Hva blir vekstraten hvis populasjonen er halvparten av bæreevnen?

Løsningsforslag

Funksjonen går fra R ved $x_i = 0$, og synker lineært til den er 0 ved $x_i = M$. Om populasjonen er halvparten av bæreevnen vil derfor vekstraten være halvparten av maks.

Oppgave 3b) Tilbake til hareøya

Vi går nå tilbake til hareøya. Modeller veksten av harer med vår nye bærekraftige vekst. Anta som før at vi slepper ut 10 harer, med en fødselsrate på 0.22 og dødsrate på 0.03. Si at øya har en bæreevne på 120 harer. Modeller bestanden over 60 år.

In []:

```
N = ...
x = zeros(N)

x[0] = 10
b = ...
d = ...
R = (b - d)
M = ...

for i in range(N-1):
    x[i+1] = ...

plot(x, 'o-')
xlabel('År')
ylabel('Harebestand')
axis((0, N, 0, 140))
axhline(M, linestyle='--', color='darkred')
show()
```

Diskuter

- Beskriv utviklingen av bestanden over tid
- Hvor lang tid tar det før bestanden har nådd en likevekt?
- Hvordan tror du modellen fortsetter over de neste 40 årene?
- Hvor realistisk tror du denne modellen? Kan du komme på noen viktige momenter som mangler?
- Hva tror du skjer om vi starter modellen med fler individer enn det er bæreevne til på øya? For eksempel 300 harer. Bruk koden til å sjekke om du hadde rett

Logistisk vekst

Vi har her kallt det for "bærekraftig" vekst, men det er vanligere å kalle denne typen vekstmodell for *logistisk vekst*. Dette er fordi kurven vi får ut av modellen er et eksempel på en logistisk kurve. Dette er i sammenligning til vår forrige modell, som gir en *eksponentiell vekst*.

"Overskyting" av bæreevnen

Ved å legge til bæreevnen til økosystemet på øya har vi sørget for at populasjonen ikke vokser uten noen grenser, men stabiliserer seg på det systemet faktisk kan opprettholde. Derimot forutsier modellen at veksten gradvis avtar etterhvert som bestanden vokser, slik at bestanden får en "myk" innføring til likevektsverdien.

Dette er kanskje ikke alltid en god beskrivelse. Harene på øya vet jo ikke at de har en maksgrense, de bare lever livene sine. Kort foralt er det enkelt å se for seg at populasjonsveksten er så stor at øya blir overpopulert, altså at bestanden overskyter bæreevnen. Det som da skjer er at vi har en større bestand enn det er ressurser til, og bestanden vil derfor begynne å dø.

Det finnes mange måter vi kan inkludere en slik "overskyting" i modellen vår. Men for å holde ting enkelt kan vi rett og slett si at vekstraten ikke avhenger av populasjonen akkurat nå, men litt lenger tilbake i tid. På den måten er det rett og slett litt "treghet" i systemet.

Oppgave 3c) Overskytende modell

Kopier svaret ditt fra oppgave (3b), men endre veksttermen slik at vi bruker formelen

$$r = R \left(1 - \frac{x_{i-5}}{M} \right),$$

for vekstraten. Kort fortalt har vi bare byttet ut x_i med x_{i-5} , altså avhenger vekstraten av populasjonen for 5 år siden.

(Du lurer kanskje på hvordan dette fungerer for de første par stegene, tross alt har vi ingen x_{-5} , x_{-4} osv. Men dette problemet løser seg selv på "magisk" vis i koden, ettersom at disse leddene vil automatisk tolkes som 0.)

In []:

```
N = 61
x = zeros(N)

x[0] = 10
b = 0.22
d = 0.03
R = (b - d)
M = 120

for i in range(N-1):
    x[i+1] = x[i] + R*(1-x[i-5]/M)*x[i]

plot(x, 'o-')
xlabel('År')
ylabel('Harebestand')
axis((0, N, 0, 160))
axhline(M, linestyle='--', color='darkred')
show()
```

Kjør koden, og diskuter

- Hvordan er kurva forskjellig fra den forrige?
- Prøv å forklar hva som skjer i de ulike "stadiene" i utviklingen.
- Hva skjer på lang sikt?
- Hva skjer om du øker eller minker "forskyvningen" i vekstraten? (Altså, fra x_{i-5} til x_{i-10} for eksempel).

Skade til økosystemet

Vi har nå sett hvordan vi kan inkludere overpopulasjon, som fører til svingninger i bestanden. Derimot er bæreevnen til økosystemet konstant i modellen. Om populasjonen er langt større enn bæreevnen er det kanskje mulig at det blir gjort varig skade til økosystemet. Igjen er dette mulig å inkludere på mange måter, men vi går for en enkel løsning.

Vi setter bæreevnen til 120 ved start. Om populasjonen er mindre eller lik denne bæreevnen så blir det ikke gjort noen skade. Alt harene spiser vokser tilbake til neste år. Om populasjonen går over bæreevnen derimot, så sier vi at bæreevnen synker litt for å reflektere at harene gjør skade til økosystemet. Altså er

$$M_{i+1} = \begin{cases} M_i & \text{hvis } x_i \leq M, \\ M_i - 1 & \text{hvis } x_i > M. \end{cases}$$

Vi kan kode dette som følger:

In []:

```
N = 61
x = zeros(N)
M = zeros(N)

x[0] = 10
b = 0.22
d = 0.03
R = (b - d)
M[0] = 120

for i in range(N-1):
    x[i+1] = x[i] + R*(1-x[i-5]/M[i])*x[i]

    if x[i] <= M[i]:
        M[i+1] = M[i]
    else:
        M[i+1] = M[i] - 1

plot(x, 'o-')
plot(M, '--')
xlabel('År')
ylabel('Harebestand')
axis((0, N, 0, 160))
show()
```

Oppgave 3d) Skadet økosystem

- Kjør koden
- Hvordan er modellen endret?
- Vi ser at skaden til økosystemet er relativt liten. Øk den årlige skaden som skjer til -5 , hva skjer med modellen nå?
- (Utfordrende) Skaden til økosystemet er nå konstant, uavhengig av hvor overpopulert økosystemet er. Klarer du å finne en formel som gjør slik at skaden skalerer med hvor overpopulert systemet er?

4) En Bytte-Rovdyr modell

Så langt har vi kun diskutert modeller med én art. Vi i en oppgave introduserte vi et rovdyr, men vi modellerte ikke en populasjon av rovdyr. I mange økosystem er det derimot samhandling mellom ulike arter, og spesielt de hvor vi har både rovdyr og byttedyr er av interesse.

Vi skal kun se på én slik modell for rovdyr-byttedyr, og dette er den første slike modellen som ble utviklet. Den er oppkalt etter de to matematikerene som jobbet på den og heter derfor Lotka-Volterra modellen.

Ettersom at vi nå skal diskutere to populasjoner trenger vi to tallfølger. Den ene som representerer byttedyrene, vi kaller denne x , og den andre som representerer rovdirene, vi kaller denne for y .

For å si hvordan byttedyrene utvikler seg begynner vi med modellen der vi antok rikelig med mat og at veksten var proporsjonal med bestanden, altså

$$x_{i+1} = x_i + \underbrace{a \cdot x_i}_{\text{Vekstterm}}$$

Her har vi omdøpt vekstraten til a , vi forklarer hvorfor snart. Merk at vi lar være å ta med bæreevnen til økosystemet, for befolkningen vil nå uansett holdes i sjakk av rovdirene.

For å legge til effekten av rovdirene legger vi til et ledd som beskriver de dørene som blir spist. Dette leddet må være proporsjonalt med antall rovdyr, jo fler rovdyr det er, jo fler er det som jakter og spiser byttedyrene. Leddet må også være proporsjonalt med antall byttedyr, for jo fler byttedyr det er, jo lettere er det for rovdirene å jakte på dem. Derfor får vi:

$$x_{i+1} = x_i + \underbrace{a \cdot x_i}_{\text{Vekstterm byttedyr}} - \underbrace{b \cdot x_i \cdot y_i}_{\text{byttedyr som blir spist}}.$$

Nå må vi finne en formel for rovdirene. Nå kan vi ikke si at veksten av rovdirene kun avhenger av rovdypopulasjonen, fordi de må ha nok tilgang på mat (det vil si byttedyr) for å vokse. Veksttermen for rovdyr avhenger altså av hvor mye de får spist, og vil derfor avhenge av begge bestandene. Leddet vi nå har laget sier bare at bestanden vokser når det er mye mat tilgjengelig. Om det *ikke* er mat tilgjengelig må rovdyr bestanden dø ut. Vi legger derfor til et ledd for dødsfall:

$$y_{i+1} = y_i + \underbrace{c \cdot x_i \cdot y_i}_{\text{Vekstterm rovdyr}} - \underbrace{d \cdot y_i}_{\text{Naturlig død}}.$$

Programmere modellen

Å programmere denne modellen er veldig likt som før, men vi kan ikke lenger regne ett og ett år av gangen, av matematiske årsaker. Istedet må vi regne med en mye finere "tidsoppløsning". Vi ønsker ikke å gå inn for mye på matematikken nå, vi bare viser koden som løser dette problemet under. Du kan godt lese denne koden, og se om du ser hva vi har endret fra tidligere:

In []:

```
T = 40      # Antall År
dt = 0.001  # Tidssteg
N = int(T//dt)  # Antall simuleringspunkter

t = zeros(N+1)
x = zeros(N+1)
y = zeros(N+1)

x[0] = 40
y[0] = 10

# Parametere
a = 0.5 # Fødselsrate byttedyr
b = 0.01 # Dødsrate byttedyr
c = 0.01 # Fødselsrate rovdyr
d = 0.6 # Dødsrate rovdyr

for i in range(N):
    t[i+1] = t[i] + dt
    x[i+1] = x[i] + (a*x[i] - b*x[i]*y[i])*dt
    y[i+1] = y[i] + (c*x[i]*y[i] - d*y[i])*dt

plot(t, x)
plot(t, y)
legend(['Byttedyr', 'Rovdyr'])
show()
```

Oppgave 4a) Diskusjon

- Kjør koden
- Forklar oppførselen til rovdyr-bytte modellen.
- Klarer du å trekke noen paralleller fra denne modellen til de tidligere modellene vi har diskutert?

Denne rovdyr-bytte modellen er egentlig en veldig enkel matematisk modell, som tar hensyn til veldig lite kontekst. Men den har en veldig interessant forutsigelse, nemlig at populasjonen av disse to artene vil svinge periodisk, altså at de gjentar seg selv med jevne mellomrom.

Som vi forklarte tidligere må modeller etterprøves mot virkeligheten. Om man studerer populasjoner i naturen er de av og til veldig stabile over tid, mens i andre tilfeller ser man en klar svingning over tid, slik som i modellen vår her. I hvert tilfelle er det ulike faktorer som virker inn, som gjør at ulike modeller kan være nyttige. Derimot kan man ofte i rovdyr-bytte system se nettopp det fenomenet som vår enkle rovdyr-bytte modell fanger opp her.

Tilbake til haiene i middelhavet

Vi begynte å motivere hele dette opplegget ved å se på bestanden av fisk og hai i middelhavet under krigstidene. Her var det biologen D'Ancona som observerte at når mengden kommersiell fiskeaktivitet sank under krigen, steg haibestanden betraktelig.

D'Ancona sin hypotese var at rovdyr bestanden økte nettopp fordi mennesker fisket mindre. Det var altså ikke andel byttedyr som økte, men snarere rovdirene (dvs haiene) som hadde fortjenste av dette. Men dette var bare en hypotese. D'Ancona gikk derfor til den italienske matematikeren Vito Volterra som studerte denne

hypotesen nærmere.

Det var nettopp ut av studien av fiske- og haibestandene i middelhavet på 1920-tallet at den første rovdyr-bytte modellen oppstod. For å etterprøve om D'Ancona sin hypotese var rimelig la Volterra også inn effekten av fiske i denne modellen, og fant da at D'Ancona sin hypotese virket meget rimelig, ihvertfall fra et matematisk perspektiv.

Oppgave 4b) Diskusjon

Vi skal ikke gå noe særlig lenger med å legge til mer kompleksitet i modellene våres, for det begynner å bli litt vel matematisk. Men diskuter gjerne med sidemannen hvordan rovdyr-bytte modellen vi har laget kan forklare at haibestanden økte under krigen, i motsetning til fiskebestanden.

5) Case: Hudson Bay's Harer og Gauper

Så langt har vi bare utforsket modeller og tenkt oss frem til om de er rimelige. Men i praksis når man jobber med naturvitenskaplige modeller er det viktig å sammenligne dem med virkeligheten og etterprøve dem. Den store utfordringen i denne jobben er å innhente nok data. Spesielt når det kommer til populasjonsdata er det store utfordringer med å måle antall individer av en gitt art det er over tid.

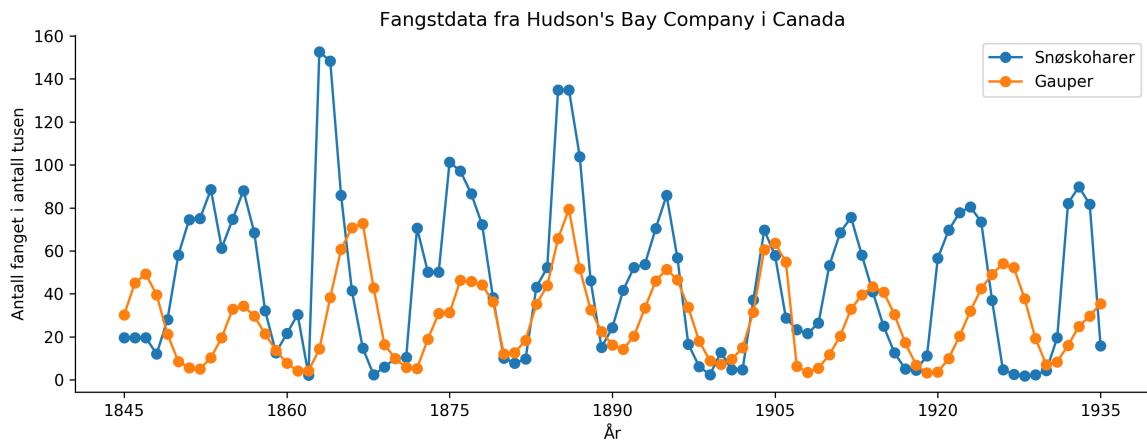
Når matematisk modellering av populasjoner begynte i 1920 var det derfor stor diskusjon om disse modellene var virkelighetsnære eller ikke. Et viktig og berømt datasett historisk var derfor data fra bedriften *Hudson's Bay Company* i Canada. Dette store selskapet kjøpte pelser fra jegere som de solgte videre til resten av verden. Dette var stor buisness og selskapet holdt telling på hvor mange pelser de fikk inn av ulike arter hvert år helt fra 1845 frem til 1935. Dette representerte altså et 90 år langt datasett om populasjonsdata fra ett gitt økosystem, nemlig Hudson Bay i Canada.

Vi skal nå ta en titt på noe av denne informasjonen. For å holde ting på et lett nivå holder vi diskusjon til å fokusere på et gitt rovdyr og et gitt byttedyr. To av artene selskapet fanget var Snøskoharer og Kanadisk gaupe. Det som er interessant er at disse gaupene jakter på disse harene.



Figur: Bilde av Kanadisk Gaupe og Snøskohare. Bildene er tatt av henholdsvis [Ken Williams](https://en.wikipedia.org/wiki/File:Canadian_lynx_by_Keith_Williams.jpg) (https://en.wikipedia.org/wiki/File:Canadian_lynx_by_Keith_Williams.jpg) (CC BY 2.0 (<https://creativecommons.org/licenses/by/2.0/deed.en>)) og [D. Gordon E. Robertson](https://commons.wikimedia.org/wiki/File:Snowshoe_Hare,_Shirleys_Bay.jpg) (https://commons.wikimedia.org/wiki/File:Snowshoe_Hare,_Shirleys_Bay.jpg) (CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>))) og brukt under Creative Commons lisensiering.

Figuren under viser hvor mange pels fra snøskoharer og gauper som ble levert inn til selskapet i tidsperioden. Det er ingen garanti for at antall pelser svarer nøyaktig til størrelsen av populasjonen, men det er ihvertfall en god indikator.



Figur: Denne figuren viser ekte data funnet i årene fra 1845 til 1935 av Hudson's Bay Company i Canada. Figuren er lagd i Python, med verdier funnet fra [Whitman College](http://people.whitman.edu/~hundlehr/courses/M250F03/M250.html). (<http://people.whitman.edu/~hundlehr/courses/M250F03/M250.html>)

Oppgave 5a)

Studer figuren over og diskuter følgende med partner:

- På hvilke måter ligner denne figuren på den enkle rovdyr-byttedyr modellen vår?
- På hvilke måter er den annerledes?
- Kan du tenke på faktorer som kanskje kan forklare hvorfor modellen vår ikke passer perfekt?

Vi ser at figuren inneholder periodiske svingninger av de to artene, slik som i vår rovdyr-bytte modell. Vi ser også at rovdyr-populasjonen ser ut til å følge byttedyrene. Når det er mye byttedyr vokser rovdynene frem, dermed faller byttedyrbestanden, rovdynene får mindre å spise og faller tilbake - og slik gjentar det seg.

Derimot ser vi også at i denne ekte datene er det mye mer "støy", det er ikke en helt jevn svinging. I starten ser vi for eksempel at rovdyrbestanden faller og holder seg lav selv om det er mye byttedyr. Det er ulikt hvor stor bestanden blir for hver svingning, eller hvor lenge bestanden holder seg stor osv.

Det er ikke vanskelig å komme på faktorer vi ikke har inkludert med i modellen vår. Innvirkning fra andre arter for eksempel, det finnes jo andre dyr som spiser harer, og gaupa jakter også på andre ting enn harer. Vi har ikke inkludert vær og klima, migrasjon av dyr, sykdomsutbrudd, osv. Lista er lang med kompliserte faktorer vi ikke har inkludert i modellen.

Tilpasse modellen etter data

La oss se om vi klarer å tilpasse rovdyr-bytte modellen vår så den passer til dataene fra harer og gauper. For å forenkle dette litt velger vi å fokusere på årene 1900 til 1920, hvor vi ser på et litt annet datasett, som vi har tatt fra [her](https://github.com/stan-dev/example-models/blob/master/knitr/lotka-volterra/hudson-bay-lynx-hare.csv) (<https://github.com/stan-dev/example-models/blob/master/knitr/lotka-volterra/hudson-bay-lynx-hare.csv>)

Oppgave 5b) Plott data

Plott populasjonsdataene av harer og gauper for hvert år mot hverandre ved å kjøre koden under:

In []:

```
år = range(1900, 1921)
gauper = [4.0, 6.1, 9.8, 35.2, 59.4, 41.7, 19.0, 13.0, 8.3, 9.1, 7.4, 8.0, 12.3, 19.0, 13.0, 8.3, 9.1, 7.4, 8.0, 12.3, 19.0]
harer = [30.0, 47.2, 70.2, 77.4, 36.3, 20.6, 18.1, 21.4, 22.0, 25.4, 27.1, 40.3, 57.0, 47.2, 70.2, 77.4, 36.3, 20.6, 18.1, 21.4, 22.0, 25.4, 27.1, 40.3, 57.0]

plot(år, harer, 'o-')
plot(år, gauper, 'o-')
xlabel('År')
ylabel('Antall')
legend(['Gauper', 'Harer'])
xticks(range(1900, 1921, 5))
axis((1900, 1920, 0, 80))
grid()
show()
```

(Utfordrende) Oppgave 5c) Tilpass parametre

Prøv nå å endre de 4 parametrene i modellen (a , b , c og d) slik at modellen ligger så tett opp til de ekte dataene som mulig. Endre koden under og kjør på nytt til du klarer å peile deg inn på rimelige parametre.

Merk at dette kan være ganske vanskelig, da samspillet mellom de 4 gjør ting kompliserte, og ting endrer seg voldsomt. Prøv å gjør små endringer av gangen, da feil valg av parametre gjerne gir matematiske løsninger som er urimelige.

Samarbeid gjerne på tvers av grupper for å finne de beste parametrene du klarer.

In []:

```
T = 20          # Antall År
dt = 0.001      # Tidssteg
N = int(T//dt)   # Antall simuleringspunkter

tid = zeros(N+1)
gauper = zeros(N+1)
harer = zeros(N+1)

tid[0] = 1900
harer[0] = 30
gauper[0] = 4

# Parametere
a = ...
b = ...
c = ...
d = ...

for i in range(N):
    tid[i+1] = tid[i] + dt
    harer[i+1] = harer[i] + (a*harer[i] - b*harer[i]*gauper[i])*dt
    gauper[i+1] = gauper[i] + (c*harer[i]*gauper[i] - d*gauper[i])*dt

plot(tid, harer)
plot(tid, gauper)
legend(['Harer', 'Gauper'])
xticks(range(1900, 1925, 5))
show()
```