FMSF80 - Datorlaboration 2

Syfte

Syftet med dagens laborationen är att du skall:

- Få förståelse för diskreta, bivariata och betingade fördelningar.
- Förstå hur simuleringar kan användas för att illustrera komplexa fördelningar.
- Bli bekant med summor av stokastiska variabler.
- Få förståelse för hur och när centrala gränsvärdessatsen kan användas.

Bakgrund

Laborationen består av två delar. Först studerar vi hur en bivariat diskret fördelning kan konstrueras från enklare delkomponenter och undersöker de resulterande marginal och betingade fördelningar. Därefter undersöker vi summor av stokastiska variabler och centrala gränsvärdessatsen.

En modell för skördeutfall

I första delen av laborationen kommer vi att studera en enkel modell för skördeutfall. Frågan är hur stor skörd man kan förvänta sig om man planterar n st frö. För att modellera skördeutfallet kan vi dela upp problemet i två steg:

1. Först konstruerar vi en modell för antalet av de planterade fröna som gror.
2. Därefter funderar vi på hur stort skördeutfallet (antal nya frö) blir om precis k st frö gror.

Den resulterande modellen består nu av en fördelning för antalet frö $p_X(k)$ och en betingad fördelning för skördeutfallet, $p_{Y|X=k}(l|k)$. Bayessats och satsen om total sannolikhet ger oss nu den gemensam fördelningen för antalet frö som gror och skördeutfallet samt marginal fördelningen för skördeutfallet.

$p_{Y|X=k}(l|k) = p_X(k) \cdot p_{Y|X=k}(l)$ och $p_Y(l) = \sum_k p_X(k) \cdot p_{Y|X=k}(l)$

Utöver dessa fördelningar är även den betingade fördelningen för X givet Y intressant, $p_{X|Y=l}(k|l)$. Om vi enbart observerar den totala skörden y så kan denna fördelning användas för att säga något om hur många frö som faktiskt grott.

Förberedelseuppgifter

1. Förvissa dig om att du förstår vad en sannolikhetsfunktion är
2. **Mozquizto:** Vi planterar 7 frö med grobarhet 75%. Ange fördelningen för antalet frön som kommer att gro (om de gror oberoende av varandra) samt fördelningens väntevärde och varians.
3. **Mozquizto:** Om $X_i \in \mathrm{Po}(\mu_i)$ och oberoende vilken fördelning har då summan $Y = \sum_{i=1}^n X_i$?
4. Förvissa dig om att du förstår hur total sannolikhet fungerar för väntevärde, d.v.s. hur man kan beräkna $\mathrm{E}(Y) = \mathrm{E}(\mathrm{E}(Y|X))$.
5. **Mozquizto:** Förvissa dig om att du förstår vad Centrala gränsvärdessatsen innebär och när den kan användas.
6. Vi beräknar medelvärdet $\bar{X}$ av oberoende s.v. $X_i \in \mathrm{Po}(3), i = 1, \ldots, n$ (samma väntevärde för alla X_i). Ange väntevärde och varians för $\bar{X}$. Vilken fördelning får $\bar{X}$ (approximativt) när n är stort? Ungefär hur stort måste n vara för att approximationen ska bli bra?

Importera moduler och ladda upp filer till Colab

Kör koden nedan för att hämta de väsentliga modulerna vi kommer att använda i laborationen.

```
# Importerar moduler
import numpy as np
import scipy.stats as stats
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Utöver modulerna ovan använder laborationen funktionen **harvest**. För att harvest ska importeras måste du:

1. Ladda ner **harvest.py** från kurshemsidan
2. Klicka på mappen *Filer* till vänster i *google colab* menyn
3. Ladda upp **harvest.py** genom att klicka på *Ladda upp till sessionens lagringsutrymme* (eller drag-n-drop filen)
4. Om du får ett **FileNotFoundError** när du försöker importera **harvest** högerklicka på **harvest.py** i google colab och *Kopiera sökväg*. Klistra sedan katalog namnet på platsen för **/content** (om sökvägen är **folder/harvest.py** ska du klistra in **folder/**)

```
import sys
#Addera content till sökvägen för python
sys.path.append('/content')  #Här kan du behöva uppdater sökvägen.
#importera harvest
from harvest import harvest
```

```
#för icke colabs (antar att harvest.py ligger i samma katalog)
from harvest import harvest
```

Modell för skördeutfall

Diskret variabel: Antal frö som gror

Vi vill simulera antalet frön som kommer att gro bland de sju planterade fröna. Det kan vi göra på två sätt. Det mest rättframma är att simulera 7 frön och räkna antalet som gror. Funktionen

```
stats.uniform.rvs(size=?)
```

ger en array med **size** rektangelfördelade slumptal, U, mellan 0 och 1. För att sannolikheten att ett frö kommer att gro skall bli p kan vi helt enkelt se efter om $U \leq p$. I så fall kommer fröet att gro. Om $U > p$ så kommer det inte att gro. För att få reda på antalet frön som kommer att gro bland de 7 summerar vi den resulterande 0/1-variabeln:

```
p = ?
U = stats.uniform.rvs(size=?)    #n sampel från en U(0,1)
print('U :', U )
print('U<=',p, ':', U<=p )
X = sum(U<=p)
print('Antal frö som gror', X)
```

Uppgift: Jämför resultatet av **U = stats.uniform.rvs(size=?)** och **U<=p** och förvissa dig om att du förstår vad som hände.

För att illustrera vad som händer så kan vi också plotta slumptalen och den sannolikhet som vi jämför med.

```
# Plot
plt.stem(U)
plt.axline((0,p), slope=0) #linje från punkten (0,p) med lutning 0
```

Mozquizto: Hur många frön grodde?

Ett smidigare sätt är att utnyttja att vi vet att antalet frön som kommer att gro är $\mathrm{Bin}(n,p)$-fördelat. Vilket vi kan simulera med **stats.binom**, först undersöker vi hjälp texten för att förstå funktionen

```
help(stats.binom)
```

Sen simulerar vi ett tal från en $\mathrm{Bin}(n,p)$-fördelning.

```
X = stats.binom.rvs(?,?)
print(X)
```

Uppgift: Gör om simuleringen några gånger om. Hur många frön brukar gro?

Antalet frön som kommer att gro varierar uppenbarligen från gång till gång. För att se hur vanligt det är med olika antal frön som kommer att gro simulerar vi $N = 100$ planterings tillfällen.

```
n = ?
p = ?
N = ?;  #antal sampel
#simulera från binomalfördelningen
X = stats.binom.rvs(n, p, size=N)
print(X)
```

Vi kan beräkna antal värden av varje typ (dvs antal 0:or, antal 1:or, etc.)

```
#notera att vi vill ha n+1 lådor (0,1,2,...,n)
(antal,varden) = np.histogram(X, bins=n+1, range=(0,n+1))

#antal samplade 3:or
print('antal 3:or:', sum( X==? ))
#Värdena finns nu i vektorn varden och i vektorn antalet finns i antal av repsketive värde
print('Antal värden =', varden[?], 'enligt np.histogram:', antal[?])
```

Vi kan illustrera fördelningen med ett stolpdiagram (vi har ju en diskret variabel). I **sns.histplot** anger vi gränser för indelningen och bredd på varje låda, t.ex. så vill vi ha värdena $x = 0$ i intervallet $(-0.5,\, 0.5)$.

```
sns.histplot(X, binwidth=1, binrange=(-0.5,n+0.5))
plt.ylabel('Antal frön som gror')
plt.xlabel('Antal försök')
```

Uppgift: Var det någon av planteringstillfällen som inte hade några groende frön alls?

Uppgift: Hur många av planteringstillfällen gav 5 groende frön? Hur många gav högst 2 groende frön?

Vi vill nu jämföra våra 100 påsar med den teoretiska sannolikhetsfunktionen. För att göra det måste vi skala om y-axeln till andelar (Notera att **res.plot** automatisk adderar en förklaring av färgerna.).

```
#binomial sannolikhets funktion beräknad för samma lådor som ovan
pmf = stats.binom.pmf(varden[0:n+1], ?, ?)
#Det är enklast att lägga samman alla data i en pandas data.frame
res = pd.DataFrame({'x':varden[0:n+1],
                    'Simulering': antal/sum(antal), #dela med totalen för att få andel
                    'Teoretisk': pmf})
#och sen plotta
res.plot(x='x',y=['Simulering','Teoretisk'], kind='bar')
```

Mozquizto: Hur stämmer andelen av de simulerade planterings tillfällena som hade precis 5 groende frön eller högst 2 groende frön med motsvarande sannolikheter? (Jämför med resultatet från **stats.binom.pmf** och **stats.binom.cdf**).

```
stats.binom.pmf(?,?,?)
```

```
stats.binom.cdf(?,?,?)
```

Mozquizto Experimentera med att ändra grobarheten från $p = 0.75$ och antalet frön från $n = 7$. Hur ändrar sig fördelningen när n eller p minskar eller ökar?

```
n = ?
p = ?
#simulera från binomalfördelningen
X = stats.binom.rvs(?, ?, size=?)
#räkna antal
(antal,varden) = np.histogram(X, bins=n+1, range=(0,n+1))
#Läg samman alla data i en pandas data.frame
res = pd.DataFrame({'x':varden[0:n+1],
                    'Simulering': antal/sum(antal), #dela med totalen för att få andel
                    'Teoretisk': stats.binom.pmf(varden[0:n+1], ?, ?) })
#och sen plotta
res.plot(x='x',y=['Simulering','Teoretisk'], kind='bar')
```

Centrala gränsvärdessatsen för binomialfördelning

Om $np(1-p) > 10$ kan binomialfördelningen approximeras med en normalfördelning. Vi kan jämföra fördelningsfunktionerna och se hur bra det blir:

```
n = ?
p = ?
E = ?             # väntevärde i binomialfördelning (använd n och p)
V = ?              # varians i binomialfördelning (använd n och p)

#Skappa en vektor för den kontinuerliga variablen som är mu+/-4*sigma, 100 values
x1 = np.linspace(E-4*np.sqrt(V),E+4*np.sqrt(V), 100)
#och en vektor [0,1,2,...,n] för den diskreta tätheten
x2 = np.arange(n+1)

plt.step(x2, stats.binom.cdf(x2, ?, ?), where='post')
plt.plot(x1, stats.norm.cdf(x1, ?, ?))      #tips: np.sqrt ger roten ur
```

Uppgift: Pröva med lite olika värden på n och p. Testa både när det går bra att normalapproximera och när det inte går.

Mozquizto: Beräkna sannolikheten att högst 2 frön gror, både exakt och med normalapproximation.

```
stats.norm.cdf(?,?,?)
```

```
stats.binom.cdf(?,?,?)
```

Simulering med hjälp av betingad fördelning: Skördeutfall

Vi tänker oss nu att varje frö som gror ger upphov till ett Poissonfördelat antal nya frön, i medeltal 10 frön per groende ursprungligt frö. Frön som inte gror ger naturligtvis inga nya frön. Vi är intresserade av fördelningen för det totala antalet nya frön som erhålls om vi planterar 7 frön med 75% grobarhet.

Sedan tidigare har vi att X = ”antal frön som gror” $\in \mathrm{Bin}(7, 0.75)$. Om exakt $X = k$ frön grodde blir Y = ”antal nya frön” en summa över antalet frö från k st oberoende plantor. D.v.s. Summan av k stycken oberoende $\mathrm{Po}(10)$-fördelade variabler, en för varje groende frö:

$Y = \sum_{i=0}^{k} Z_i$ där $Z_i \in \mathrm{Po}(10)$

Från förberedelseuppgifterna har vi då att fördelningen för $Y|X=k \in \text{Po}(10 \cdot k)$ där $k = 0, \ldots, 7$. Fördelningen för Y ges då av (Satsen om Total Sannolikhet)

$$p_Y(l) = \sum_{k=0}^{7} p_{Y|X=k}(l) \cdot p_X(k) = \sum_{k=0}^{7} \frac{(10 \cdot k)^l}{l!} \cdot e^{-10 \cdot k} \cdot \binom{7}{k} \cdot 0.75^k \cdot 0.25^{7-k}$$

För att ta reda på hur denna fördelning ser ut studerar vi först det enklare fallet med enbart $n = 2$ planterade frön. Först illustrerar vi sannolikheten att 0, 1 eller 2 frö gror.

```
#parametrar för binomial fördelningen
n = 2
p = ?

#och x-vektor för binomialfördelningen (vi har ju att k=0,1,...,n)
x = np.arange(?)

#Plotta binomialfördelningen
plt.bar(?, stats.binom.pmf(?,?,?))
plt.title('Antal frö som gror')
plt.ylabel('p(x)')
```

Därefter illustrerar vi de tre olika varianterna av betingade fördelningar: $\text{Po}(0), \text{Po}(10), \text{Po}(20)$.

```
#parametrar och x-vektor för binomial fördelningen
n = 2
p = ?
x = np.arange(?)

# parametrar och y-vektor för poisson
mu = 10
y = np.arange(4*mu+1)

## plottar
#Först binomial fördelning med n=2
ax = plt.subplot(2,1,1) #första ploten ska vara hela övre raden
ax.bar(x, stats.binom.pmf(?,?,?))
ax.set_title('Antal frö som gror')
ax.set_ylabel('p(x)')

#Sen poisson fördelning för x=0, 1 och 2. Vi gör beräkningarna genom att loopa över i=0,1,2
for i in range(3):
    ax = plt.subplot(2,3,4+i)
    ax.bar(y, stats.poisson.pmf(y, ?*mu))
    ax.set_title('Skörd om ' + str(?) + ' frö gror')
    ax.set_ylabel('p(y|x=' + str(?) + ')')

#justera plottar så att texten inte överlappar
plt.subplots_adjust(wspace=0.7, hspace=0.4)
```

Mozquizto: Hur ändrar sig den betingade fördelningen för Y givet X när antalet groende frön ändrar sig?

Uppgift: Tänk efter hur fördelningen för Y borde se ut, när vi viktat ihop dessa 3 fördelningar med vikter enligt binomialfördelningen för antalet groende frön.

För att undersöka hur Y borde se ut kan vi simulera från fördelning. Först använder vi **stats.binom.rvs** för att dra 1000 sampel som illustrerar hur många av våra 2 frö som gror i olika försök.

```
# Simulera N fall
N = 1000
n = 2
p = ?
X = stats.binom.rvs(?,?,size=?)

#antal frö som gror
sns.histplot(X, binwidth=1, binrange=(-0.5,n+0.5), stat='density')
```

Givet att $X = k$ frö gror så vet vi att det betingade antalet nya frö nu är $Y|X = k \in \text{Po}(10 \cdot k)$. Eftersom vi simulerat antalet frö som gror ovan så kan vi för varje fall simulera det betingade antalet nya frö.

```
#N fall
N = 1000
#simulera antal frö som gror
n = 2
p = ?
X = stats.binom.rvs(?,?,size=?)
# simulerad skörd
mu = ?
Y = stats.poisson.rvs(?)

#antal frö som gror och simulerad skörd
ax = plt.subplot(211)
sns.histplot(X, binwidth=1, binrange=(-0.5,n+0.5), stat='density')
ax = plt.subplot(212)
sns.histplot(Y, binwidth=1, binrange=(-0.5,max(Y)+0.5), stat='density')
```

I det här fallet kan vi faktiskt använda satsen om total sannolikhet för att låta `python` räkna ut sannolikhetsfunktionen för Y, men i mer komplicerade fall är simulering ibland det bästa (enda) vi kan göra:

```
# Teoretisk fördelning med satsen om total slh.
n = 2
p = ?
mu = ?
y = np.arange(4*mu+1)

pY = stats.poisson.pmf(y,?) * stats.binom.pmf(?,?,?)        #fallet X=0
pY = pY + stats.poisson.pmf(y,?) * stats.binom.pmf(?,?,?)   #fallet X=1
pY = pY + stats.poisson.pmf(y,?) * stats.binom.pmf(?,?,?)   #fallet X=2

#%% Plot för n=2
plt.bar(y, pY)
plt.xlabel('antal nya frön')
plt.title('Fallet med n = '+ str(n))
```

Uppgift: Ser fördelningen ut som du hade väntat dig? Stämmer det med simuleringarna?

För det allmänna fallet, exempelvis $n = 7$, kan vi använda en **for**-sats för att beräkna summan över k:

```
# Allmänt n med for-loop
n = 7
y_n = np.arange(n*mu+1)
pY_n = 0
for k in range(n+1):  #range är heltal 0<=k<n+1
    pY_n = pY_n + stats.poisson.pmf(y_n,?) * stats.binom.pmf(?,?,?)

# Plot för båda fallen
ax = plt.subplot(211)
ax.bar(y, pY)
ax.set_xlabel('antal nya frön')
ax.set_title('Fallet med n=2')

ax = plt.subplot(212)
ax.bar(y_n, pY_n)
ax.set_xlabel('antal nya frön')
ax.set_title('Fallet med n =' + str(n))

#justera plottar så att texten inte överlappar
plt.subplots_adjust(hspace=0.6)
```

Bivariat och marginal fördelning

Funktionen **harvest.py** ritar upp sannolikhetsfunktionen för Y (dvs marginalfördelningen) och den gemensamma fördelningen för X och Y i en modell där

$Y|X = x \in \text{Po}(\mu \cdot x)$ och $X \in \text{Bin}(n, p)$

för valfria värden på n, p och μ. Och den gemensamma fördelningen för antalet frön som gror, X, och antalet nya frö som skördas, Y. är

$$p_{X,Y}(k, l) = p_{Y|X=k}(l) \cdot p_X(k), \qquad k = 0, 1, \ldots, n;\ l = 0, 1, 2, 3, \ldots$$

```
help(harvest)
```

```
fig = harvest(?,?,?)
```

Mozquizto: Experimentera med olika värden på n, p och μ. Vad händer om antalet planterade frö, n, minskar eller ökar? Om grobarheten, p, minskar eller ökar? Om medelantalet nya frön per frö som gror, μ, minskar eller ökar?

Uppgift: Vad händer om grobarheten är 100%?

Uppgift: Kan du få marginalfördelningen att se normalfördelad ut?

Mozquizto: Experimentera med olika värden på n, p och μ. Hur hänger den bivariata sannolikhetsfunktionen, $p_{X|Y}(k, l)$, ihop med marginalsannolikheten, $p_Y(l)$?

Bivariat fördelning och betingad fördelning

Givet den gemensamma sannolikhetsfunktionen kan vi också räkna ut den betingade fördelningen för hur många frön som grott givet att vi vet skördeutfallet

$$p_{X|Y=l}(k) = \frac{p_{X,Y}(k,l)}{p_Y(l)} = \frac{p_{Y|X=k}(l) \cdot p_X(k)}{p_Y(l)}, \qquad k = 0, 1, \ldots, n$$

Funktionen kan också illustrera den betingade sannolikhetsfunktionen om vi har skördat $y = 25$ frö om den anropas med en extra parameter (vad vi betingar på)

```
fig = harvest(?, ?, ?, ?)
```

Mozquizto: Experimentera med olika värden på n, p, μ och observerat skördeutfall y. Hur hänger den bivariata sannolikhetsfunktionen, $p_{X,Y}(k,l)$, ihop med den betingade sannolikheten, $p_{X|Y=l}(k)$?

Mozquizto: Vad är det troligaste antalet frö som grott (högst betingad sannolikhet) givet att man skördade 50 frön?

Centrala gränsvärdessatsen

Vi skall nu titta lite närmare på Centrala Gränsvärdessatsen (CGS).

Summa av Poisson

Vi börjar med en liten simulering från en känd fördelning, två slumpmässiga obervationer x_1, x_2 från $X \in \mathrm{Po}(\mu)$ där $\mu = 3$. Vi ska sedan beräkna medelvärdet $\bar{x}$ och se hur nära väntevärdet μ det hamnar.

```
mu = ?                              # det sanna mu-värdet
x = stats.poisson.rvs(?, size=?)    # simulera 10 Po(mu)-slumptal
x = x.reshape((?,?))                # organisera som en 2x5-matris
print(x)
xmedel = x.mean(axis=0)            # 5 medelvärden av 2 Po-variabler var
print(xmedel)
```

Uppgift: Gör om simuleringen och medelvärdesberäkningen några gånger. Verkar medelvärdet variera mindre än de enskilda observationerna? Borde den det? I så fall, hur mycket mindre?

Låt oss göra om simuleringarna ett stort antal gånger så vi får bättre uppfattning om hur medelvärdet beter sig:

```
mu = ?
n = ?        # antal termer i medelvärdet
M = ?        # antal simuleringar
x = stats.poisson.rvs(mu, size=?)   # simulera n*m Po(mu)-slumptal
x = x.reshape((?,?))                # organisera som en nxM-matris
xmedel = x.mean(axis=0)             # M st medelvärden

#definiera en gemensam binrange för jämförelse
binrange = (-0.5,5*mu+0.5)
#subplots
fig, axs = plt.subplots(2, 1, constrained_layout=True)
# histogram över de Mst x[0,:]-värdena (Poisson)
sns.histplot(x[0,:], binwidth=1, binrange=binrange, ax=axs[0])
axs[0].set_title('Poisson variabler')
# histogram över de Mst x-medelvärdena
sns.histplot(xmedel, binwidth=0.5, binrange=binrange, ax=axs[1])
axs[1].set_title('Medelvärde av ' + str(n))
```

Uppgift: Experimentera med lite olika värden på n och se vad som händer med medelvärdet. Du kan behöva ändra klassbredden, **binwidth** i det undre histogrammet för att se något.

Mozquizto: Jämför variationen hos de enskilda observationerna i den övre figuren och variationen för skattningarna i den undre figuren. Hur ändrar sig variationen hos observationernavnär vi ändrar n?

Uppgift: Experimentera med $\mathrm{Po}(3)$-fördelningen. Hur ser approximationen ut för olika n (och μ)?

Uppgift: För små n (eller μ) kan normalapproximationen uppenbarligen bli negativ. Blir $\bar{X}_n$ någonsin negativ, om X_i är Poisson-fördelade?

Medel av andra fördelningar

Enligt centrala gränsvärdessatsen vet vi att $\sum X_i$ och därmed också medelvärdet $\bar{X}$ blir normalfördelat om vi summerar tillräckligt många variabler; **oavsett** vilken fördelning X_i har. Vi undersöker nu hur CGS fungerar för medelvärde av

exponentialfördekade variabler.

```
n = ?        # antal termer i medelvärdet
M = ?        # antal simuleringar
x = stats.expon.rvs(scale=?, size=?)   # simulera exponential
x = x.reshape((?,?))                   # n x M-matris. x1 i första raden, xn i sista.
xmedel = x.mean(axis=0)                # M st medelvärden

#definiera en gemensam binrange för jämförelse
binrange = (0, np.max(x))
#subplots
fig, axs = plt.subplots(2, 2, constrained_layout=True)
# histogram över de Mst x[0,:]-värdena (Poisson)
sns.histplot(x[0,:], binwidth=1, binrange=binrange, ax=axs[0,0])
axs[0,0].set_title('Exponential variabler')
# histogram över de Mst x-medelvärdena i samma skala som för x
sns.histplot(xmedel, binwidth=0.5, binrange=binrange, ax=axs[0,1])
axs[0,1].set_title('Medelvärde av ' + str(n))
# histogram över de Mst x-medelvärdena i fri skala
sns.histplot(xmedel, ax=axs[1,0])
axs[0,1].set_title('Medelvärde av ' + str(n))
# kvantilplot av medelvärdena för att undersöka normalfördelning
fig = stats.probplot(xmedel, dist="norm", fit=True, plot=axs[1,1])
```