

STK-FYS1110 - Oblig 1

Bastian Eggum Huuse

februar 2025

Oppgave 1

a)

Vi setter opp gunstige utfall over mulige utfall. Vi har 11^4 mulige utfall, siden hver person kan velge mellom 11 etasjer. De gunstige utfallene, er de der hver person velger en ny etasje, så den første personen kan velge mellom alle etasjene, den andre mellom alle minus 1, også videre.

Vi har da at

$$P(4 \text{ personer velger forskjellige etasjer}) = \frac{11 \cdot 10 \cdot 9 \cdot 8}{11^4} = 0.5409$$

Det er ca 54% sjanse for at alle går av i hver sin etasje.

b)

Dersom minst to går av i samme etasje, har vi at *ikke* alle går av i forskjellige etasjer. Vi kan da bruke resultatet fra oppgave a:

$$\begin{aligned} P(\text{minst 2 av 4 går av i samme etasje}) \\ &= 1 - P(4 \text{ personer velger forskjellige etasjer}) \\ &= 1 - 0.54 \\ &= 0.4690 \end{aligned}$$

Det er en ca 46% sjanse for at minst to personer ender opp i samme etasje.

c)

Vi ser etter et uordnet utvalg uten tilbakelegging. Vi har da fra kombinatorikken at

$$\binom{4}{3} = \frac{4!}{3!(4-3)!} = 4$$

er antallet ulike grupper.

d)

Vi setter opp et uttrykk for sannsynligheten for hele systemet:

$$P((3 \cap (4 \cup 5)) \cup (1 \cup 2))$$

Så løser vi dette uttrykket opp, og setter inn sannsynlighetene:

$$\begin{aligned} &P((3 \cap (4 \cup 5)) \cup (1 \cup 2)) \\ &= P(3 \cap (4 \cup 5)) + P(1 \cup 2) - P((3 \cap (4 \cup 5)) \cap (1 \cup 2)) \\ &= P(3)(P(4) + P(5) - P(4 \cap 5)) + (P(1) + P(2) - P(1 \cap 2)) \\ &\quad - (P(3)(P(4) + P(5) - P(4 \cap 5))(P(1) + P(2) - P(1 \cap 2))) \\ &= P(3)(P(4) + P(5) - P(4)P(5)) + (P(1) + P(2) - P(1)P(2)) \\ &\quad - (P(3)(P(4) + P(5) - P(4)P(5))(P(1) + P(2) - P(1)P(2))) \\ &= 0.95 \cdot (0.95 + 0.95 - 0.95^2) + (0.95 + 0.95 - 0.95^2) \\ &\quad - (0.95 \cdot (0.95 + 0.95 - 0.95^2) \cdot (0.95 + 0.95 - 0.95^2)) \\ &= 0.99987 \end{aligned}$$

Det er en nesten 100% sjanse for at alarmen fungerer

Oppgave 2

a)

Vi setter opp to begivenheter: A : Teksten er AI-generert, og B : Programmet mener teksten er AI-generert.

Vi ønsker å finne $P(A|B)$. Vi bruker Bayes setning:

$$\begin{aligned} P(A|B) &= \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A')P(A')} \\ &= \frac{0.9 \cdot 0.05}{0.9 \cdot 0.05 + 0.07 \cdot (1 - 0.05)} \\ &= 0.4035 \end{aligned}$$

Vi har en ca 40% sannsynlighet for at en tekst er AI-generert, gitt at maskinen mener den er det.

b)

Vi danner en funksjon, med like tall som i oppgave a), hvor vi erstatter 0.07 med en variabel:

$$f(x) = \frac{0.9 \cdot 0.05}{0.9 \cdot 0.05 + x \cdot (1 - 0.05)}$$

Dette gir $x = 0.00526$ når $f(x) = 0.9$. Vi må altså ha en lavere sannsynlighet en 0.526% for å få over 90% sannsynlighet i a).

Oppgave 3

a)

All Kode relatert til oppgavene er lagt ved i bunnen av dokumentet.

Vi skriver program i python, som gir oss følgende output:

```
1 '''
2
3 Forventet levealder for menn ved foedsel er 80.73
4
5 Forventet levealder for menn, gitt at man har levd 25 aar, er 56.27
6 Forventet levealder for menn, gitt at man har levd 50 aar, er 32.26
7 Forventet levealder for menn, gitt at man har levd 85 aar, er 5.44
8
9 '''
```

b)

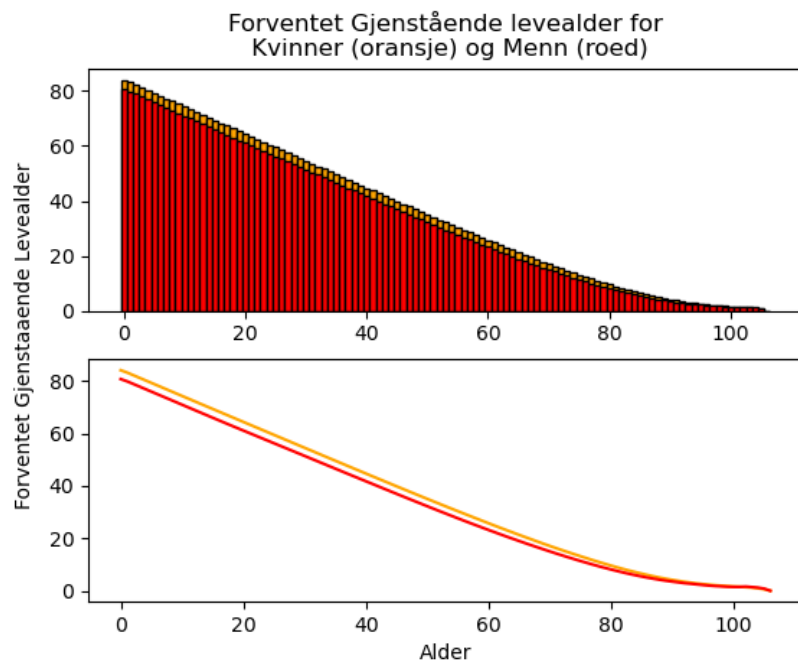
Vi skriver program i python, som gir oss følgende output:

```
1 '''
2
3 Forventet levealder for menn ved foedsel er 80.73
4
5 Forventet levealder for menn, gitt at man har levd 25 aar, er 56.27
6 Forventet levealder for menn, gitt at man har levd 50 aar, er 32.26
7 Forventet levealder for menn, gitt at man har levd 85 aar, er 5.44
8
9
10 Forventet levealder for kvinner ved foedsel er 84.11
11
12 Forventet levealder for kvinner, gitt at man har levd 25 aar, er 59.47
13 Forventet levealder for kvinner, gitt at man har levd 50 aar, er 35.06
14 Forventet levealder for kvinner, gitt at man har levd 85 aar, er 6.54
15
16 '''
```

Vi kan se at kvinner har høyere forventet levealder ved alle aldre.

c)

Vi skriver program i python, som gir følgende plot:



Oppgave 4

a)

Vi har at hver markør er en av 4 ulike nukliotider. Dette er da et ordnet utvalg uten tilbakelegging, og vi har da fra kombinatorikken at den totale mengden kombinasjoner er 4^q , der q er antall markører.

En vilkårlig person vil ha en av disse kombinasjonene. Vi setter gunstige over mulige og får at det er en $\frac{1}{4^q}$ sjanse for at en vilkårlig person har DNA-profil S .

Vi skriver program i python, som gir følgende utskrift:

```
1  '''
2
3  Sannsynligheten for at en person har en spesifikk DNA-profil S, naar DNA-profilen
   har 5 nukleotider, er 9.77e-04
4  Dersom vi har 5500000 personer, kan vi forvente at 5.37e+03 personer har profilen
   S
5
6  Sannsynligheten for at en person har en spesifikk DNA-profil S, naar DNA-profilen
   har 10 nukleotider, er 9.54e-07
7  Dersom vi har 5500000 personer, kan vi forvente at 5.25e+00 personer har profilen
   S
8
9  Sannsynligheten for at en person har en spesifikk DNA-profil S, naar DNA-profilen
   har 20 nukleotider, er 9.09e-13
10 Dersom vi har 5500000 personer, kan vi forvente at 5.00e-06 personer har profilen
    S
11
12  '''
```

Sjansen for at en tilfeldig person har profil S er altså høyere når q er lav (som passer med formelen vår), og antall personer som har profil S synker når q går opp, som samsvarer med at sjansen går ned.

b)

Gitt at den mistenkte personen er bidragsyter, har vi at den mistenkte alltid vil ha DNA-profil S , siden S er bidragsyterens profil. Dette gir $P(B|A) = 1$.

Gitt at den mistenkte ikke er bidragsyter, er sjansen for at den mistenkte har DNA-profil S , lik den vi fant i a) for en vilkårlig person (siden dette nå er en vilkårlig person). Vi får $P(B|A') = \frac{1}{4^q}$, der q er antall markører.

c)

Vi bruker Bayes' setning:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A')P(A')}$$

Vi har at $P(A) = \frac{1}{N}$ og $P(A') = 1 - \frac{1}{N}$. Vi setter inn og får

$$\begin{aligned} P(A|B) &= \frac{1 \cdot \frac{1}{N}}{1 \cdot \frac{1}{N} + \frac{1}{4^q} \cdot \left(1 - \frac{1}{N}\right)} \\ &= \frac{\frac{1}{N}}{\frac{1}{N} + \frac{1}{4^q} \cdot \frac{1}{4^q N}} \\ &= \frac{4^q}{4^q + N - 1} \\ &= \frac{1}{1 + \frac{N-1}{4^q}} \end{aligned}$$

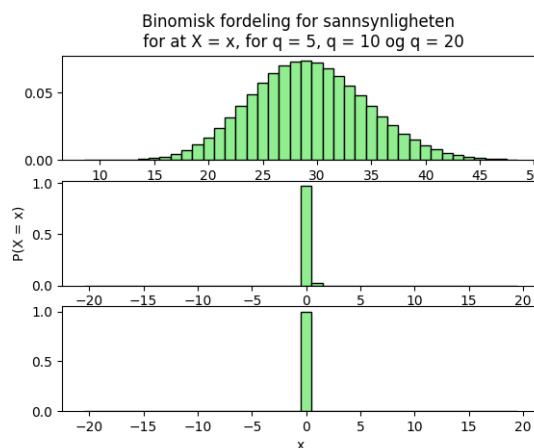
d)

Kriteriene for en binomisk fordeling er at alle eksperimentene er uavhengige, har lik sannsynlighet hver gang, og der resultatene kan beskrives med en av to verdier (som regel Success eller Failure). Alle personene i databasen har uavhengige DNA-profiler, alle har lik sannsynlighet for å matche S, og enten matcher de S, eller ikke. Vi kan altså bruke en binomisk fordeling for å modellere X.

For $X = 1$ gitt C' har vi:

$$\begin{aligned} P(X = 1|C') &= \binom{n}{x} p^x (1-p)^{n-x} \\ &= \binom{n}{1} \left(\frac{1}{4^q}\right)^1 \left(1 - \frac{1}{4^q}\right)^{n-1} \\ &= \frac{n}{4^q} \left(1 - \frac{1}{4^q}\right)^{n-1} \end{aligned}$$

For å plotte X, bruker vi python. Dette gir følgende plot:



e)

Vi har n sjanser til å trekke den riktige personen fra en gruppe på N . Vi setter gunstige over mulige, og får at $P(C) = \frac{n}{N}$.

For å finne $P(X = 1|C)$, må vi først endre litt på uttrykket. Siden vi vet at vi har bidragsyter i databasen, blir sjansen for at det kun er en person som har DNA-profil S lik sjansen for at ingen andre har DNA-profil S. Vi får altså

$$P(X = 1|C) = P(X' = 0|C')$$

Der X' er hvor mange av de resterende individene som har DNA-profil S (Vi må da også senke n med en i neste uttrykk). Vi kan da sette inn i binomialfordelingen, og finne et uttrykk.

$$\begin{aligned} P(X = 1|C) &= P(X' = 0|C') \\ &= \binom{n-1}{0} \left(\frac{1}{4^q}\right)^0 \left(1 - \frac{1}{4^q}\right)^{n-1-0} \\ &= \left(1 - \frac{1}{4^q}\right)^{n-1} \end{aligned}$$

f)

Vi ønsker å finne et uttrykk for $P(C|X = 1)$. Vi har fra produktsetningen at:

$$\begin{aligned} P(C|X = 1) &= \frac{P(C \cap X = 1)}{P(X = 1)} \\ &= \frac{P(X = 1|C)P(C)}{P(X = 1)} \end{aligned}$$

Vi bruker så setningen om total sannsynlighet:

$$\begin{aligned} &\frac{P(X = 1|C)P(C)}{P(X = 1)} \\ &= \frac{P(X = 1|C)P(C)}{P(X = 1|C)P(C) + P(X = 1|C')P(C')} \\ &= \frac{P(X = 1|C)P(C)}{P(X = 1|C)P(C) + P(X = 1|C')(1 - P(C))} \\ &= \frac{P(X = 1|C)P(C)}{P(X = 1|C)P(C) + P(X = 1|C') - P(X = 1|C')P(C)} \\ &= \frac{P(X = 1|C)}{P(X = 1|C) + \frac{P(X=1|C')}{P(C)} - P(X = 1|C')} \end{aligned}$$

Så setter vi inn for uttrykkene:

$$\begin{aligned}
& \frac{P(X=1|C)}{P(X=1|C) + \frac{P(X=1|C')}{P(C)} - P(X=1|C')} \\
&= \frac{\left(1 - \frac{1}{4^q}\right)^{n-1}}{\left(1 - \frac{1}{4^q}\right)^{n-1} + \frac{\frac{n}{4^q} \left(1 - \frac{1}{4^q}\right)^{n-1}}{\frac{n}{N}} - \left(1 - \frac{1}{4^q}\right)^{n-1}} \\
&= \frac{\left(1 - \frac{1}{4^q}\right)^{n-1}}{\left(1 - \frac{1}{4^q}\right)^{n-1} + \frac{N}{4^q} \left(1 - \frac{1}{4^q}\right)^{n-1} - \frac{n}{4^q} \left(1 - \frac{1}{4^q}\right)^{n-1}} \\
&= \frac{1}{1 + \frac{N}{4^q} - \frac{n}{4^q}} \\
&= \frac{1}{1 + \frac{N-n}{4^q}}
\end{aligned}$$

Spesielt for $n = 1$ får vi $\frac{1}{1 + \frac{N-1}{4^q}}$, som er sannsynligheten vi fant i oppgave c), og spesielt for $n = N$ får vi $\frac{1}{1 + \frac{N-N}{4^q}} = 1$, som er sannsynligheten for at bidragsyter finnes i hele befolkningen beskrevet ved N , som vi vet er 100%.

g)

Vi skriver kode i python, som gir følgende utskrift:

```

1  '''
2
3  Vi har N = 5500000, og n = 30000
4
5  Naar q = 5, har vi at P(C|X = 1) = 0.000187
6  Naar q = 10, har vi at P(C|X = 1) = 0.160860
7  Naar q = 20, har vi at P(C|X = 1) = 0.999995
8
9  '''

```

Vi kan se at når q er lav, da er $P(X=1|C)$ nær 0, fordi sannsynligheten for at en vilkårlig person har DNA-profil S er relativt høy. Når q er høy, nærmer $P(X=1|C)$ seg 1, fordi sannsynligheten for at en vilkårlig person har DNA-profil S nærmer seg 0.

Kode

Oppgave 3

a)

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 # Leser fra fil
6 doed=pd.read_csv("https://www.uio.no/studier/emner/matnat/math/STK1100/data/
   doedelighet.txt",sep="\t")
7 alder=doed["alder"].values
8 menn=doed["menn"].values
9
10 # Sannsynligheten i datasettet er gitt per 1000 individer.
11 # Finner sannsynligheten for en person
12 qx=menn/1000
13
14 Sx=np.cumprod(1-qx)
15
16 # Finner Kumulativ Fordeling.
17 Fx=1-Sx
18
19 # Finner alle punktsannsynlighetene
20 tmp=np.zeros(107)
21 tmp[1:107] = Fx[0:106]
22 px=Fx-tmp
23
24 # Finner forventning ( $E(X) = \sum(x*p(x))$ )
25 x=alder
26 E_x = np.sum(x*px)
27
28 # Finner forventet resterende alder ved gitt alder a
29 A = [25,50,85]
30
31 print(f"\nForventet levealder for menn ved foedsel er {E_x:.2f}\n")
32
33 for a in A:
34     x = alder[a:107]
35     Fx = np.cumsum(px)
36     E_x_a = np.sum((alder[a:107]-a)*px[a:107])/(1-Fx[a-1])
37
38     print(f"Forventet levealder for menn, gitt at man har levd {a} aar, er {(E_x_a
   ):.2f}")
39
40 print()
```

b)

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 # Leser fra fil
6 doed=pd.read_csv("https://www.uio.no/studier/emner/matnat/math/STK1100/data/
   doedelighet.txt",sep="\t")
7 alder=doed["alder"].values
8 menn=doed["menn"].values
```

```

9 kvinner = doed["kvinner"].values
10
11 # Sannsynligheten i datasettet er gitt per 1000 individer.
12 # Finner sannsynligheten for en person
13 qx=menn/1000
14 qy=kvinner/1000
15
16 Sx=np.cumprod(1-qx)
17 Sy=np.cumprod(1-qy)
18
19 # Finner Kumulativ Fordeling.
20 Fx=1-Sx
21 Fy=1-Sy
22
23 # Finner alle punktsannsynlighetene
24 tmp=np.zeros(107)
25 tmp[1:107] = Fx[0:106]
26 px=Fx-tmp
27 tmp[1:107] = Fy[0:106]
28 py=Fy-tmp
29
30 # Finner forventning ( $E(X) = \sum(x \cdot p(x))$ )
31 x=alder
32 E_x = np.sum(x*px)
33 y=alder
34 E_y = np.sum(y*py)
35
36 # Finner forventet resterende alder ved gitt alder a
37 A = [25,50,85]
38
39 print(f"\nForventet levealder for menn ved foedsel er {E_x:.2f}\n")
40
41 for a in A:
42     x = alder[a:107]
43     Fx = np.cumsum(px)
44     E_x_a = np.sum((alder[a:107]-a)*px[a:107])/(1-Fx[a-1])
45
46     print(f"Forventet levealder for menn, gitt at man har levd {a} aar, er {(E_x_a-
47     ):.2f}")
48
49 print()
50 print(f"\nForventet levealder for kvinner ved foedsel er {E_y:.2f}\n")
51
52 for a in A:
53     y = alder[a:107]
54     Fy = np.cumsum(py)
55     E_y_a = np.sum((alder[a:107]-a)*py[a:107])/(1-Fy[a-1])
56
57     print(f"Forventet levealder for kvinner, gitt at man har levd {a} aar, er {(
58     E_y_a):.2f}")
59
60 print()

```

c)

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import pandas as pd
4
5 # Leser fra fil

```

```

6 doed=pd.read_csv("https://www.uio.no/studier/emner/matnat/math/STK1100/data/
  doedelighet.txt",sep="\t")
7 alder=doed["alder"].values
8 menn=doed["menn"].values
9 kvinner = doed["kvinner"].values
10
11 # Sannsynligheten i datasettet er gitt per 1000 individer.
12 # Finner sannsynligheten for en person
13 qx=menn/1000
14 qy=kvinner/1000
15
16 Sx=np.cumprod(1-qx)
17 Sy=np.cumprod(1-qy)
18
19 # Finner Kumulativ Fordeling.
20 Fx=1-Sx
21 Fy=1-Sy
22
23 # Finner alle punktsannsynlighetene
24 tmp=np.zeros(107)
25 tmp[1:107] = Fx[0:106]
26 px=Fx-tmp
27 tmp[1:107] = Fy[0:106]
28 py=Fy-tmp
29
30 # Finner forventning ( $E(X) = \sum(x \cdot p(x))$ )
31 x=alder
32 E_x = np.sum(x*px)
33 y=alder
34 E_y = np.sum(y*py)
35
36 # Finner forventet resterende alder ved gitt alder a
37 A = range(1,107,1)
38 plot_m = np.zeros(107)
39 plot_k = np.zeros(107)
40 plot_m[0] = E_x
41 plot_k[0] = E_y
42
43 for a in A:
44     x = alder[a:107]
45     y = alder[a:107]
46     Fx = np.cumsum(px)
47     Fy = np.cumsum(py)
48     E_x_a = np.sum((alder[a:107]-a)*px[a:107])/(1-Fx[a-1])
49     E_y_a = np.sum((alder[a:107]-a)*py[a:107])/(1-Fy[a-1])
50
51     plot_m[a] = E_x_a
52     plot_k[a] = E_y_a
53
54 width=1
55 edge = "black"
56 color_k = "orange"
57 color_m = "red"
58
59 fig = plt.figure()
60 ax = fig.add_subplot(111) # The big subplot
61 ax1 = fig.add_subplot(211)
62 ax2 = fig.add_subplot(212)
63
64 ax.spines['top'].set_color('none')
65 ax.spines['bottom'].set_color('none')
66 ax.spines['left'].set_color('none')

```

```

67 ax.spines['right'].set_color('none')
68 ax.tick_params(labelcolor='w', top=False, bottom=False, left=False, right=False)
69
70 ax.set_xlabel('Alder')
71 ax.set_ylabel('Forventet Gjenstaaende Levealder')
72 ax.set_title("Forventet Gjenstaaende levealder for\n Kvinne (oransje) og Menn (
    roed)")
73
74 ax1.bar(alder, plot_k, width, color=color_k, edgecolor=edge)
75 ax1.bar(alder, plot_m, width, color=color_m, edgecolor=edge)
76
77 ax2.plot(alder, plot_k, color_k)
78 ax2.plot(alder, plot_m, color_m)
79 plt.show()

```

Oppgave 4

a)

```

1 # Vi har 4 ulike valg, og q antall valg vi skal gjoere:
2
3 def n(q):
4     return 4**q
5
6 # Vi skal finne sannsynligheten for akkurat en kombinasjon
7
8 def P_x_s(n):
9     # n er antall kombinasjoner
10    return (1/n)
11
12 Q = [5,10,20]
13 N = 5500000
14
15 print()
16
17 for q in Q:
18     p = P_x_s(n(q))
19     print(f"Sannsynligheten for at en person har en spesifikk DNA-profil S, naar
    DNA-profilen har {q:2} nukleotider, er {p:.2e}")
20     # E_x = sum(x*px)
21     E = N * p
22     print(f"Dersom vi har {N} personer, kan vi forvente at {E:.2e} personer har
    profilen S\n")

```

d)

```

1 from scipy.stats import binom
2 import numpy as np
3 import matplotlib.pyplot as plt
4 # Vi har 4 ulike valg, og q antall valg vi skal gjoere:
5
6 def n(q):
7     return 4**q
8
9 # Vi skal finne sannsynligheten for akkurat en kombinasjon
10
11 def P_x_s(n):
12     # n er antall kombinasjoner
13     return (1/n)
14

```

```

15 Q = [5,10,20]
16 N = 30000
17
18 print()
19
20
21 print()
22
23 fig = plt.figure()
24 ax = fig.add_subplot(111)    # The big subplot
25
26 ax.spines['top'].set_color('none')
27 ax.spines['bottom'].set_color('none')
28 ax.spines['left'].set_color('none')
29 ax.spines['right'].set_color('none')
30 ax.tick_params(labelcolor='w', top=False, bottom=False, left=False, right=False)
31
32 ax.set_xlabel('x')
33 ax.set_ylabel('P(X = x)',labelpad = 4)
34 ax.set_title("Binomisk fordeling for sannsynligheten \n for at X = x, for q = 5, q
    = 10 og q = 20")
35
36 i = 1
37 for q in Q:
38     E = int(N*(1/4**(q)))
39     ax = plt.subplot(3,1,i)
40     X = np.arange(E - 20, E + 20)
41     ax.bar(X, binom.pmf(X, N, 1/(4**q)), width = 1, edgecolor="black", color = "
        lightgreen")
42     i += 1
43     #pass
44
45 plt.show()

```

g)

```

1 # Utrykket vi fant i oppgave f)
2 def p(N,n,q):
3     return 1/(1+(N-n)/4**q))
4
5 # Variabler fra oppgaven
6 N = 5500000
7 n = 30000
8 Q = [5,10,20]
9
10 print(f"\nVi har N = {N}, og n = {n}\n")
11 for q in Q:
12     print(f"Naar q = {q:2}, har vi at P(X = 1|C) = {p(N,n,q):.6f}")
13 print()

```