Obligatorisk oppgave 1, STK1100, Vår 2021

Cory Alexander Balaton

Oppgave 1

a) Begivenheten av at hver av de 5 personene går av heisen i hver sin etasje tilsvarer et ordnet utvalg av 5 etasjer av 11.

$$\begin{aligned} N(a) &= 11 \cdot 10 \cdot 9 \cdot 8 \cdot 7 \\ &= 55440 \end{aligned} \tag{1}$$

For å finne sannsynligheten for at de 5 personene går av heisen i hver sin etasje, så må vi ta $N(A)$ og dele det med antall gunstige måter for 5 personer å velge en etasje å gå av på.

$$\begin{aligned} N &= 11^5 \\ &= 161051 \end{aligned} \tag{2}$$

$$\begin{aligned} P(A) &= \frac{N(A)}{N} \\ &= \frac{55440}{161051} \\ &\approx 0.3442 \end{aligned} \tag{3}$$

b) Begivenheten minst 2 personer går av i samme etasje er komplementærbegivenheten til at hver person går av i hver sin etasje, og kan derfor uttrykkes slik:

$$\begin{aligned} P(A)' &= 1 - P(A) \\ &\approx 1 - 0.3442 \\ &\approx 0.6558 \end{aligned} \tag{4}$$

c) For å finne ut av hvor mange grupper på 3 man kan lage av 5 personer, så bruker man binomialkoeffisienten (Jeg antar at rekkefølgen av personene ikke spiller rolle).

$$\begin{aligned}\binom{5}{3} &= \frac{5!}{3!(5-3)!} \\ &= \frac{5 \cdot 4}{2} \\ &= 10\end{aligned} \tag{5}$$

d) For å finne sannsynligheten til at alarmsystemet fungerer, så må vi først finne ut sannsynligheten for at subsystemene fungerer. Kommer til å bruke at $P(S)$ = sannsynligheten for at hele systemet fungerer, $P_1(S)$ = sannsynligheten for at komponent 1 fungerer, og $P_{12}(S)$ = sannsynligheten for at komponent 1 og 2 fungerer osv.

$$\begin{aligned}P_{12}(S) = P_{34}(S) &= 1 - (P_1(S)' \cdot P_2(S)') \\ &= 1 - (0.1 \cdot 0.1) \\ &= 1 - 0.01 \\ &= 0.99\end{aligned} \tag{6}$$

$$\begin{aligned}P_{345}(S) &= P_{34}(S) \cdot P_5(S) \\ &= 0.99 \cdot 0.9 \\ &= 0.891\end{aligned} \tag{7}$$

$$\begin{aligned}P(S) &= 1 - (P_{345}(S)' \cdot P_{12}(S)') \\ &= 1 - (0.109 \cdot 0.01) \\ &= 1 - 0.00109 \\ &= 0.99891\end{aligned} \tag{8}$$

Oppgave 2

Begynner med å sette navn på informasjonen som er gitt.
Sannsynligheten for at gutten glemmer å mate gullfiskenfisken: $P(A') = 0.25$
Sannsynligheten for at gullfisken overlever gitt at gutten har matet den: $P(B|A) = 0.9$
Sannsynligheten for at gullfisken overlever gitt at gutten har ikke matet den: $P(B|A') = 0.5$
Sannsynligheten for at gutten ikke har matet gullfisken gitt at den er død: $P(A'|B')$

$$P(A'|B') = \frac{p(A' \cap B')}{P(B')} \tag{9}$$

For å finne $P(B')$, så kan man bruke regelen $P(B) = P(B|A_1) \cdot P(A_1) + P(B|A_2) \cdot P(A_2) + \cdots + P(B|A_n) \cdot P(A_n)$, der alle A_n er disjunkte. Siden vi har $P(B|A)$ og $P(B|A')$, så må vi først finne $P(B)$ og deretter ta komplimentet.

$$\begin{aligned} P(B') &= 1 - (P(B|A) \cdot P(A) + P(B|A') \cdot P(A')) \\ &= 1 - (0.9 \cdot 0.75 + 0.5 \cdot 0.25) \\ &= 1 - (\frac{27}{40} + \frac{1}{8}) \\ &= 1 - \frac{4}{5} \\ &= \frac{1}{5} \end{aligned} \tag{10}$$

Nå vil man finne $P(A' \cap B')$, og man kan gjøre det ved å ta komplimentet til $P(A \cup B)$.
Vi vet også at $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
Vi har $P(A)$ og $P(A)$, og vi trenger nå bare å finne $P(A \cap B)$, og vi kan gjøre det ved å bruke produktregelen.

$$\begin{aligned} P(A \cap B) &= P(A) \cdot P(B|A) \\ &= \frac{3}{4} \cdot \frac{9}{10} \\ &= \frac{27}{40} \end{aligned} \tag{11}$$

$$
\begin{aligned}
P(A \cup B) &= P(A) + P(B) - P(A \cap B) \\
&= \frac{3}{4} + \frac{4}{5} - \frac{27}{40} \\
&= \frac{30}{40} + \frac{32}{40} - \frac{27}{40} \\
&= \frac{35}{40} \\
&= \frac{7}{8}
\end{aligned} \tag{12}
$$

$$
\begin{aligned}
P(A' \cap B') &= 1 - P(A \cup B) \\
&= 1 - \frac{7}{8} \\
&= \frac{1}{8}
\end{aligned} \tag{13}
$$

Nå har vi funnet det vi trenger for å løse $P(A'|B')$.

$$
\begin{aligned}
P(A'|B') &= \frac{p(A' \cap B')}{P(B')} \\
&= \frac{\frac{1}{8}}{\frac{1}{5}} \\
&= \frac{5}{8}
\end{aligned} \tag{14}
$$

Sannsynligheten for at gutten har glemt er da $\frac{5}{8}$

PS.

I etterkant innså jeg at $P(B'|A')$ er komplimentet til $P(B|A')$, og dermed kunne jeg brukt Bayes teorem til å finne $P(A'|B')$, som er mye enklere enn det jeg først gjorde.

$$
\begin{aligned}
P(A'|B') &= \frac{P(B'|A') \cdot P(A')}{P(B')} \\
&= \frac{\frac{1}{2} \cdot \frac{1}{4}}{\frac{1}{5}} \\
&= \frac{\frac{1}{8}}{\frac{1}{5}} \\
&= \frac{5}{8}
\end{aligned} \tag{15}
$$

Oppgave 3

a) Vi vil finne $P(X \leq x)$, og for å finne denne kumulative sannsynligheten, så er det enklere å finne overlevelsessannsynligheten først, og deretter ta komplimentet til den sannsynligheten.
Siden mannen er 35 år gammel og vi vil regne ut kumulative sannsynligheten for tiden han har igjen. Da kan overlevelsessannsynligheten skrives som $S(x) = P(X > x) = P(X > 35) \cdot P(X > 36|X \geq 36) \cdot \cdots \cdot P(X > x|X \geq x)$
q_x beskriver sannsynligheten for at en x år gammel person kommer til å dø innen et år, og kan uttrykkes som $P(X = x|X \geq x)$, og siden vi har data for q_x, så kan vi skrive om overlevelsessannsynligheten:

$$\begin{aligned} P(X > x) &= P(X > 35) \cdot P(X > 36|X \geq 36) \cdot \cdots \cdot P(X > x|X \geq x) \\ &= (1 - P(X = 35)) \cdot (1 - P(X = 35|X \geq 35)) \cdot \cdots \cdot (1 - P(X = x|X \geq x)) \\ &= (1 - q_{35}) \cdot (1 - q_{36} \cdot \cdots \cdot (1 - q_x)) \\ &= \prod_{y=0}^{x} (1 - q_{35+y}) \end{aligned} \tag{16}$$

Vi ser nå at $P(X > x)$ og $P(X \leq x)$ er komplimentære hendelser til hverandre, og får derfor:

$$\begin{aligned} P(X \leq x) &= 1 - P(X > x) \\ &= 1 - \prod_{y=0}^{x} (1 - q_{35+y}) \end{aligned} \tag{17}$$

b) En kumulativ sannsynlighet kan beskrives ved likningen $P(X \leq x) = P(0) + P(1) + \cdots + P(x-1) + P(x)$, og ved å forandre på uttrykket, så får man:

$$\begin{aligned} P(X \leq x) &= p(0) + p(1) + \cdots + p(x-1) + p(x) \\ p(x) &= P(X \leq x) - (p(0) + p(1) + \cdots + p(x-1)) \\ &= P(X \leq x) - P(X \leq x - 1) \end{aligned} \tag{18}$$

c)

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

def Fx_px():
    # Read and store the columns of the file
    file = pd.read_csv("~/Documents/UiO/STK1100/Oblig1/Python/dodssannsynlighet-felles.txt", sep="\t")

    # Set each column as a vector
    age = file["ald"].values
    death = file["dod"].values

    # Defininf q byconverting death from per mille to decimal
    q = death/1000

    # Using the formula in a to calculate the cdf
    Fx = 1-np.cumprod(1-q[35:107])

    # VAriable tmp that offset the values of q by one
    tmp = np.zeros(72)
    tmp[1:72] = Fx[:71]

    # variable that calculates the pmf on every point
    px = Fx - tmp

    # Creating a bar plot and saving it in the image directory to use in the latex document
    x = np.arange(0, 72)
    plt.bar(x, px, 1, edgecolor="black")
    plt.xlabel("Gjenstaaende levetid")
    plt.ylabel("Sannsynlighet")
    plt.savefig("/home/cory/Documents/UiO/STK1100/Oblig1/images/1c.png")

    return Fx, px
```

d) Hvis mannen dør før han fyller 67 år, så vil han ikke få noen pensjonsutbetalinger. Siden han er 35 år gammel, så vil $X < 67 - 35 \Rightarrow X < 32 \Rightarrow X \leq 31$. Hvis han blir 67 eller eldre, så vil han få pensjonsutbetalinger, og nåverdien av utbetalingene uttrykkes som:

$$\begin{aligned} h(X) &= \sum_{k=32}^{X} \frac{100000}{1.03^k} \\ &= \frac{100000}{1.03^{32}} + \frac{100000}{1.03^{32+1}} + \cdots + \frac{100000}{1.03^{32+X}} \\ &= \frac{100000}{1.03^{32}} \cdot (1 + \frac{1}{1.03} + \frac{1}{1.03^2} + \cdots + \frac{1}{1.03^X}) \\ &= \frac{100000}{1.03^{32}} \cdot \sum_{k=0}^{X-32} \frac{1}{1.03^k} \\ &= \frac{100000}{1.03^{32}} \cdot \frac{1 - (\frac{1}{1.03})^{X-31}}{1 - (\frac{1}{1.03})} \end{aligned} \tag{19}$$

e) Man regner ut den forventede nåverdien ved å ta summen av alle nåverdiene av utbetalingene ganget med punktsannsynlighetene som man fant i b ($h(x) \cdot p(x)$), og tar dem fra $x = 0$ til $x = 106 - 35 = 71$ fordi maks levetid er definert som 106 år, og får da uttrykket:

$$E[h(X)] = \sum_{x=0}^{71} h(x) \cdot p(x) \tag{20}$$

For å regne ut den forventede nåverdien, så ekspanderer man på programmet i 3c:

```
# Import the function created in three_c.py
import numpy as np
from three_c import Fx_px

# Execute the function and store Fx and px as variables
Fx, px = Fx_px()

# Create funhction that calculate
h = lambda X: (100000/1.03**32) * (1 - (1/1.03)**(X-31))/(1 -
    (1/1.03)) if X >= 32 else 0

# Create arange from 0 to including 71
arr = np.arange(72)

# Map through arr and apply the function h to every element.
   Should return 0 on the 32 first elements.
hx = np.array(list(map(h, arr)))

# Sum of the product of each element in hx and px
Ex = sum(hx*px)

print(Ex)

'''
cory@Nickelback:~/Documents/UiO/STK1100/Oblig1$ /usr/bin/
   python3 /home/cory/Documents/UiO/STK1100/Oblig1/Python/
   three_d.py
501511.92449024785
'''
```

Etter å ha kjørt programmet, så får man ut a forventningsverdien er 501511.92449024785 $\approx$ 501512

f) Når man skal regne ut den samlede nåverdien av premieinnbetalingene, så tar man summen fra mannen er 35 år gammel til og med han fyller 66, eller om til han dør hvis han dør før han rekker å bli 66, som kan uttrykkes som $min(X, 31)$. Siden vi har i hele oppgaven tatt i utgangspunktet at forsikringen starter fra mannen er fylt 35, så summerer vi uttrykket fra 0 til 31. Vi kan forenkle uttrykket ved å ta K utenfor summeringen, og deretter omgjøre summeringen om til formelen til en endelig geometrisk rekke.

$$\begin{aligned} K \cdot g(X) &= \sum_{k=0}^{min(X,31)} \frac{K}{1.03^k} \\ &= K \cdot \sum_{k=0}^{min(X,31)} \frac{1}{1.03^k} \\ &= K \cdot \frac{1 - (\frac{1}{1.03}^{min(X,31)+1})}{1 - (\frac{1}{1.03})} \end{aligned} \tag{21}$$

g) Akkurat som i e, så får vi den forventede nåverdien ved å summere alle nåverdier av premieinnbetalingene ganget med punktsannsynlighetene, og deretter ganger det med K og får den forventede nåverdien per krone ganget med k. Summeringen er da fra 0 til $106 - 35 = 71$ og man får uttrykket:

$$K \cdot E[g(X)] = K \cdot \sum_{x=0}^{71} g(x) \cdot p(x) \tag{22}$$

For å regne dette ut, så ekspanderer vi enda en gang på programmet i 3c:

```
# Import the function created in three_c.py
import numpy as np
from three_c import Fx_px

# Execute the function and store Fx and px as variables
Fx, px = Fx_px()

# Create funhction that calculate
g = lambda X: (1 - (1/1.03)**(np.minimum(X, 31)+1))/(1 -
    (1/1.03))

# Create arange from 0 to including 71
arr = np.arange(72)

# Map through arr and apply the function h to every element.
    Should return 0 on the 40 last elements
gx = np.array(list(map(g, arr)))

# Sum of the product of each element in hx and px
Ex = sum(gx*px)

print(Ex)

'''
cory@Nickelback:~/Documents/UiO/STK1100/Oblig1$ /usr/bin/
    python3 /home/cory/Documents/UiO/STK1100/Oblig1/Python/
    three_g.py
20.603314835251663
'''
```

Etter å ha kjørt programmet, så ser man ut a forventningsverdien er $K \cdot 20.603314835251663 \approx K \cdot 20.6033$

h)

$$\begin{aligned} K \cdot E[g(X)] &= E[h(X)] \\ K &= \frac{E[h(X)]}{E[g(X)]} \\ &= \frac{501512}{20.6033} \\ &= 24341.3434 \end{aligned} \tag{23}$$

Den årlige premien blir på 24341.3434 kr.