



Oblig 2b

Gruppe 98

av

Christopher Sanden

i

MA-223

Statistikk

Fakultet for teknologi og realfag

Universitetet i Agder

Mars 2026

Innhold

1	Innledning	1
2	Oppgave 40 fra læreboka	2
2.1	40a) Sannsynlighet for å bli tatt ved 1 passering	2
2.2	40b) Sannsynlighet for å bli tatt 2 ganger ved 5 passeringer	2
2.3	40c) Hvor mange passeringer må til for å bli tatt 2 ganger?	3
2.4	40d) Maks antall passeringer før sannsynlighet bikker 50%	4
2.5	40e) Tatt mens Hans leser i baksetet	5
2.6	40f) Sannsynlighet for nøyaktig 15 biler med <i>Wunderbaum Heute</i>	7
3	Oppgave 2 – Bernoulli-prosess	8
3.1	2a) Forskjellen mellom <code>rbinom(1,10,2/3)</code> og <code>rbinom(10,1,2/3)</code>	8
3.2	2b) Simulering av 50 serier med 10 kast	9
3.3	2c) Beskrivelse av hva koden gjør	9
3.4	2d) Fem nye kjøringar med 50 simuleringer	10
3.5	2e) Fem kjøringar med 100000 simuleringer	10
3.6	2f) Viktigste forskjell mellom 50 og 100000 simuleringer	11
3.7	2g) Funksjonsgrafen til $\text{Bin}(10, \frac{2}{3})$	11
3.8	2h) Hva forteller funksjonsgrafen oss? Hva er sammenhengen med histogrammene over?	12
4	Oppgave 3: Simulering av Poisson-fordeling	14
4.1	3a) Simulering av Poisson-fordeling med $\lambda = 2$	14
4.2	3b) Funksjonsgrafen til $\text{Poisson}(2)$	14
4.3	3c) Sammenligning mellom histogrammer og funksjonsgraf	15
4.4	3d) Simulering med 100000 runder	16
4.5	3e) Viktigste forskjell mellom 200 og 100000 simuleringer	16
5	Oppgave 4: Kontinuerlige sannsynlighetsfordelinger	17
5.1	4a) $X \sim \phi(37, 5)$, finn $P(X \geq 33)$	17
5.2	4b) Oppgave 10.7.3.c	18
5.3	4c) Oppgave 10.7.11	18
5.4	4d) Oppgave 10.7.28	20
5.5	4e) Oppgave 10.7.37	20
5.6	4f) Oppgave 10.7.51	22
6	Oppgave 5: Normalfordeling	24
6.1	5a) Graf av standard normalfordeling	24
6.2	5b) Ett tilfeldig trekk fra $\phi(0, 1)$	25
6.3	5c) 50 tilfeldige trekk og histogram med normalfordelingen tegnet inn	25
6.4	5d) Histogrammer for 500 og 50000 trekk	26
6.5	5e) Forklaring av den oppgitte R-koden for $N = 100$	28
6.6	5f) Gjentakelse med $N = 1000, 10000$ og 100000	29
7	Oppgave 6: Betafordeling	32
7.1	6a) Graf av $\beta(3, 7)$	32

7.2	6b) Grafer for alle kombinasjoner av $a, b \in \{1, 2, 5\}$	33
7.3	6c) Hvordan parameterne påvirker formen	34

Figurer

1	Resulterende funksjonsgraf fra å kjøre koden (10)	12
2	Graf for teoretisk sannsynlighetsfordeling i oppgave 3	15
3	PDF og CDF for $X \sim \phi(37, 5)$, med markering av området $P(X \geq 33)$ i oppgave 4a	17
4	PDF og CDF for $X \sim \phi(2, 1)$, med markering av kvantilen a slik at $P(X \leq a) = 0.05$ i oppgave 4b	18
5	PDF og CDF for $T \sim \exp_{4.4}$, med markering av intervallet $P(T \in (0.15, 0.28))$ i oppgave 4c	19
6	PDF og CDF for Student-fordelingen $t(4)$, med markering av kvantilen $t_{4,0.1}$ i oppgave 4d	20
7	PDF og CDF for $X \sim \beta(3, 2)$, med markering av intervallet $P(X \in (0.4, 0.65))$ i oppgave 4e	21
8	PDF og CDF for $T \sim \text{weib}(0.5, 3)$, med markering av intervallet $P(T \in (2, 4))$ i oppgave 4f	23
9	Graf av standard normalfordeling $N(0, 1)$ i oppgave 5a	24
10	Histogram av 50 trekk fra standard normalfordeling med teoretisk kurve tegnet inn i oppgave 5c	26
11	Histogram av 500 trekk fra standard normalfordeling med teoretisk kurve tegnet inn i oppgave 5d	27
12	Histogram av 50000 trekk fra standard normalfordeling med teoretisk kurve tegnet inn i oppgave 5d	28
13	Histogram for den standardiserte binomiske fordelingen ved $N = 100$ i oppgave 5e	29
14	Histogram for den standardiserte binomiske fordelingen ved $N = 1000$ i oppgave 5f	30
15	Histogram for den standardiserte binomiske fordelingen ved $N = 10000$ i oppgave 5f	30
16	Histogram for den standardiserte binomiske fordelingen ved $N = 100000$ i oppgave 5f	31
17	Graf til $\beta(3, 7)$	32
18	Grafer for alle kombinasjoner av parameterverdiene	34
19	De fem resulterende histogrammene fra å kjøre koden fra oppgave 2b)	
20	De fem resulterende histogrammene fra å kjøre koden fra oppgave 2e)	
21	De fem resulterende histogrammene fra å kjøre koden fra oppgave 3a)	

List of Listings

1	R-kode for beregning av sannsynligheten i oppgave 40b	3
2	R-kode for beregning av antall passeringer og standardavvik i oppgave 40c	4
3	R-kode for kontroll av grenseverdier i oppgave 40d	5
4	R-kode for løsning av oppgave 40e	6
5	R-kode for beregning av sannsynligheten i oppgave 40f	7
6	R-kode som viser forskjellen mellom de to binomiske simuleringene i oppgave 2a . .	8
7	R-kode for simulering av 50 serier med 10 Bernoulli-forsøk i oppgave 2b	9
8	R-kode for fem nye histogrammer med 50 simuleringer i oppgave 2d	10
9	R-kode for fem histogrammer med 100000 simuleringer i oppgave 2e	10
10	R-kode for funksjonsgrafen til den binomiske fordelingen i oppgave 2g	11
11	R-kode for fem simuleringer av Poisson-fordelingen med 200 runder	14
12	R-kode for funksjonsgrafen til Poisson(2)	14
13	R-kode for simulering av Poisson(2) med 100000 runder	16
14	R-kode for beregning av $P(X \geq 33)$ i oppgave 4a	17
15	R-kode for beregning av kvantilen i oppgave 4b	18
16	R-kode for beregningene i oppgave 4c	19
17	R-kode for beregning av $t_{4,0.1}$ i oppgave 4d	20
18	R-kode for beregningene i oppgave 4e	21
19	R-kode for beregningene i oppgave 4f	22
20	R-kode for grafen til standard normalfordeling i oppgave 5a	24
21	R-kode for ett tilfeldig trekk fra standard normalfordeling i oppgave 5b	25
22	R-kode for histogram av 50 trekk med normalfordelingen tegnet inn i oppgave 5c .	25
23	R-kode for histogrammer av 500 og 50000 trekk med normalfordelingen tegnet inn i oppgave 5d	27
24	Oppgitt R-kode i oppgave 5e	28
25	R-kode for grafen til $\beta(3, 7)$ i oppgave 6a	32
26	R-kode for de ni betafordelingene i oppgave 6b	33

1 Innledning

Denne rapporten besvarer oblig 2b i MA-223 og tar for seg navngitte sannsynlighetsfordelinger, både diskrete og kontinuerlige. Besvarelsen er bygd opp etter oppgaveteksten, med én valgt bokoppgave fra den diskrete delen, simulering av Bernoulli- og Poisson-fordeling, samt oppgaver om kontinuerlige fordelinger, normalfordeling og betafordeling.

I den diskrete delen brukes både analytiske utregninger og kommandoer i R. I den kontinuerlige delen brukes R til å tegne fordelingsfunksjoner, kumulative fordelingsfunksjoner og histogrammer, slik at de teoretiske fordelingene kan sammenlignes med simulerte data. Målet med rapporten er både å regne ut sannsynligheter og å illustrere hvordan de ulike fordelingene oppfører seg grafisk.

2 Oppgave 40 fra læreboka

2.1 40a) Sannsynlighet for å bli tatt ved 1 passering

I oppgave 40a blir hver fjerde bil kontrollert, så sannsynligheten for kontroll er

$$p = \frac{1}{4}.$$

Dersom Hans og Fritz sniker én gang, er sannsynligheten for at de blir knepet derfor

$$P(\text{knepet}) = \frac{1}{4} = 0.25.$$

Sannsynligheten for å bli tatt er altså 25%. Dette kan også uttrykkes i R som sannsynligheten for nøyaktig én kontroll i ett forsøk:

```
P_knepet = dbinom(1, 1, 1/4)
P_knepet
```

Kommandoen gir verdien 0.25, som stemmer med utregningen over.

2.2 40b) Sannsynlighet for å bli tatt 2 ganger ved 5 passeringer

Hver gang Hans og Fritz sniker gjennom bomringen, er sannsynligheten for å bli knepet

$$p = \frac{1}{4}.$$

Når de sniker 5 ganger, og vi ønsker sannsynligheten for at de blir knepet nøyaktig 2 ganger, kan dette modelleres med en binomisk fordeling:

$$X \sim \text{Bin}(5, \frac{1}{4}).$$

Vi bruker formelen for binomisk sannsynlighet:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}.$$

Her får vi

$$P(X = 2) = \binom{5}{2} \left(\frac{1}{4}\right)^2 \left(\frac{3}{4}\right)^3.$$

Regner vi dette ut, får vi

$$P(X = 2) = 10 \cdot \frac{1}{16} \cdot \frac{27}{64} = \frac{135}{512} \approx 0.2637.$$

Sannsynligheten for at Hans og Fritz blir knepet nøyaktig 2 ganger dersom de sniker 5 ganger, er altså omtrent 26.37%.

Dette kan også beregnes i R med kommandoen under:

```

1 P_2av5 = dbinom(2, 5, 1/4)
2 P_2av5

```

Listing 1: R-kode for beregning av sannsynligheten i oppgave 40b

Denne kommandoen gir verdien 0.2636719, som stemmer med utregningen over.

2.3 40c) Hvor mange passeringer må til for å bli tatt 2 ganger?

Antall ganger Hans og Fritz blir knepet kan modelleres som en binomisk stokastisk variabel

$$X \sim \text{Bin}(n, \frac{1}{4}),$$

der n er antall ganger de sniker, og sannsynligheten for å bli knepet i ett forsøk er

$$p = \frac{1}{4}.$$

For en binomisk fordeling er forventningsverdien gitt ved

$$E(X) = np.$$

Vi ønsker at forventet antall knipinger skal være 2, altså

$$np = 2.$$

Setter vi inn $p = \frac{1}{4}$, får vi

$$n \cdot \frac{1}{4} = 2$$

og dermed

$$n = 8.$$

Hans og Fritz må altså snike 8 ganger for at forventet antall knipinger skal være 2.

Standardavviket til en binomisk fordeling er

$$\sigma = \sqrt{np(1-p)}.$$

Setter vi inn $n = 8$ og $p = \frac{1}{4}$, får vi

$$\sigma = \sqrt{8 \cdot \frac{1}{4} \cdot \frac{3}{4}} = \sqrt{\frac{3}{2}} \approx 1.225.$$

Standardavviket er altså omtrent 1.225.

Dette kan kontrolleres i R med kommandoene under:

```

1 P_knepet = 1/4
2
3 ganger_tatt_totalt = 2
4 antall_passeringer = ganger_tatt_totalt / P_knepet
5 standardavvik = sqrt(antall_passeringer * P_knepet * (1 - P_knepet))
6
7 antall_passeringer
8 standardavvik

```

Listing 2: R-kode for beregning av antall passeringer og standardavvik i oppgave 40c

2.4 40d) Maks antall passeringer før sannsynlighet bikker 50%

La X være antall ganger Hans og Fritz blir knepet når de sniker n ganger. Siden hver snikekjøring kan regnes som et uavhengig forsøk med sannsynlighet

$$p = \frac{1}{4}$$

for å bli knepet, har vi

$$X \sim \text{Bin}(n, \frac{1}{4}).$$

Vi ønsker å finne det største antall ganger de kan snike før sannsynligheten for å bli knepet overstiger 50%. Dette er sannsynligheten for å bli knepet minst én gang:

$$P(X \geq 1).$$

Det er enklere å bruke komplementregelen:

$$P(X \geq 1) = 1 - P(X = 0).$$

For en binomisk fordeling er

$$P(X = 0) = \left(\frac{3}{4}\right)^n,$$

så vi får

$$P(X \geq 1) = 1 - \left(\frac{3}{4}\right)^n.$$

Vi søker største verdi av n slik at denne sannsynligheten ikke overstiger 0.5:

$$1 - \left(\frac{3}{4}\right)^n \leq 0.5.$$

Vi tester de nærmeste heltallsverdiene:

$$P(X \geq 1 \mid n = 2) = 1 - \left(\frac{3}{4}\right)^2 = 1 - \frac{9}{16} = \frac{7}{16} = 0.4375$$

og

$$P(X \geq 1 \mid n = 3) = 1 - \left(\frac{3}{4}\right)^3 = 1 - \frac{27}{64} = \frac{37}{64} \approx 0.5781.$$

Sannsynligheten overstiger altså 50% ved $n = 3$, så største antall ganger de kan snike uten at sannsynligheten overstiger 50%, er

$$n = 2.$$

Dette kan kontrolleres i R med kommandoene under:

```
1 1 - dbinom(0, 2, 1/4)
2 1 - dbinom(0, 3, 1/4)
```

Listing 3: R-kode for kontroll av grenseverdier i oppgave 40d

2.5 40e) Tatt mens Hans leser i baksetet

I denne oppgaven ser vi på 12 snikekjøringer totalt. Av disse ble Hans og Fritz knepet 4 ganger. I 7 av de 12 tilfellene satt en medhjelper i baksetet og leste. Vi ønsker sannsynligheten for at denne personen satt og leste i nøyaktig 3 av de 4 gangene de ble knepet.

Dette kan modelleres med en hypergeometrisk fordeling, siden vi trekker 4 knepede tilfeller fra totalt 12 tilfeller uten tilbakelegging. Vi setter

$$N = 12, \quad K = 7, \quad n = 4,$$

der N er totalt antall snikekjøringer, K er antall ganger personen satt og leste, og n er antall ganger de ble knepet.

Lar vi X være antall av de knepede tilfellene der personen satt og leste, får vi

$$X \sim \text{Hypergeometric}(12, 7, 4).$$

Vi ønsker sannsynligheten

$$P(X = 3).$$

Ved bruk av formelen for hypergeometrisk sannsynlighet får vi

$$P(X = 3) = \frac{\binom{7}{3} \binom{5}{1}}{\binom{12}{4}}.$$

Dette gir

$$P(X = 3) = \frac{35 \cdot 5}{495} = \frac{175}{495} = \frac{35}{99} \approx 0.3535.$$

Sannsynligheten for at personen satt og leste i nøyaktig 3 av de 4 gangene de ble knepet, er altså omtrent 35.35%.

Dette kan kontrolleres i R med kommandoen under:

```
1  passeringer = 12
2  leser_ved_passering = 7
3  antall_tatt = 4
4  ønsket_tatt_og_leser = 3
5  P_leste_og_tatt = dhyper(ønsket_tatt_og_leser, leser_ved_passering,
6                        ( passeringer - leser_ved_passering ), antall_tatt)
7  P_leste_og_tatt
```

Listing 4: R-kode for løsning av oppgave 40e

2.6 40f) Sannsynlighet for nøyaktig 15 biler med *Wunderbaum Heute*

I denne oppgaven vet vi at i snitt har hver 1024. bil som passerer bomringen bladet *Wunderbaum Heute* liggende i baksetet. Dette betyr at sannsynligheten for at en tilfeldig bil har bladet, er

$$p = \frac{1}{1024}.$$

I løpet av uken ble 16384 bilister knepet for sniking. Vi lar X være antall av disse bilene som hadde *Wunderbaum Heute* i baksetet. Siden antall biler er stort og sannsynligheten er liten, kan dette modelleres med en Poisson-fordeling.

Parameteren blir

$$\lambda = np = 16384 \cdot \frac{1}{1024} = 16.$$

Dermed har vi

$$X \sim \text{Poisson}(16).$$

Vi ønsker sannsynligheten for at nøyaktig 15 av bilene hadde bladet i baksetet:

$$P(X = 15).$$

Ved bruk av formelen for Poisson-fordelingen får vi

$$P(X = 15) = \frac{e^{-16} 16^{15}}{15!}.$$

Dette gir

$$P(X = 15) \approx 0.0992.$$

Sannsynligheten for at nøyaktig 15 av de 16384 knepede bilene hadde *Wunderbaum Heute* i baksetet, er altså omtrent 9.92%.

Dette kan kontrolleres i R med kommandoene under:

```
1  antall_biler = 16384
2  P_wunderbaum = 1 / 1024
3  lambda = antall_biler * P_wunderbaum
4
5  P_nøyaktig_15 = dpois(15, lambda)
6  P_nøyaktig_15
```

Listing 5: R-kode for beregning av sannsynligheten i oppgave 40f

3 Oppgave 2 – Bernoulli-prosess

3.1 2a) Forskjellen mellom `rbinom(1,10,2/3)` og `rbinom(10,1,2/3)`

Begge kommandoene beskriver Bernoulli-forsøk med sannsynlighet

$$p = \frac{2}{3}$$

for suksess, men de brukes på litt forskjellig måte.

Kommandoen `rbinom(1,10,2/3)` genererer ett tilfeldig tall som angir hvor mange suksesser vi får i én serie med 10 forsøk. Resultatet blir derfor ett enkelt tall mellom 0 og 10.

Kommandoen `rbinom(10,1,2/3)` genererer derimot 10 tilfeldige tall, der hvert tall tilsvarer ett enkelt Bernoulli-forsøk. Hvert av disse tallene vil derfor være enten 0 eller 1.

Forskjellen er altså at den første kommandoen gir summen av suksesser i 10 forsøk direkte, mens den andre gir resultatet av hvert enkelt forsøk separat.

Sammenhengen mellom dem er at dersom vi summerer de 10 resultatene fra `rbinom(10,1,2/3)`, får vi et tall som følger samme binomiske fordeling som resultatet fra `rbinom(1,10,2/3)`.

Dette kan illustreres i R med kommandoene under:

```
1  rbinom(1, 10, 2/3)
2  rbinom(10, 1, 2/3)
3  sum(rbinom(10, 1, 2/3))
```

Listing 6: R-kode som viser forskjellen mellom de to binomiske simuleringene i oppgave 2a

3.2 2b) Simulering av 50 serier med 10 kast

Vi ønsker å simulere 50 serier, der hver serie består av 10 Bernoulli-forsøk med sannsynlighet

$$p = \frac{2}{3}$$

for suksess. For hver serie registrerer vi hvor mange suksesser som oppstår.

Siden vi ønsker ett samlet antall suksesser for hver serie på 10 kast, bruker vi kommandoen `rbinom(1,10,2/3)`.

```
1 data = c()
2
3 for (i in 1:50) {
4   k = rbinom(1, 10, 2/3)
5   data = c(data, k)
6 }
7
8 table(data)
9 hist(data, breaks = 100, col = "purple")
```

Listing 7: R-kode for simulering av 50 serier med 10 Bernoulli-forsøk i oppgave 2b

3.3 2c) Beskrivelse av hva koden gjør

Koden oppretter først en tom vektor kalt `data`. Deretter kjøres en `for`-løkke 50 ganger. Hver gang løkken kjøres, brukes kommandoen `rbinom(1,10,2/3)` til å simulere én serie med 10 forsøk, og resultatet k blir antall suksesser i denne serien.

Verdien k legges deretter inn i vektoren `data`. Når løkken er ferdig, inneholder `data` totalt 50 verdier, der hver verdi representerer antall suksesser i én simulert serie.

Kommandoen `table(data)` lager en frekvenstabell som viser hvor mange ganger de ulike antallene suksesser opptrer. Kommandoen `hist(data, breaks = 100, col = purple)` tegner så et histogram som viser fordelingen av de simulerte resultatene.

3.4 2d) Fem nye kjøring med 50 simuleringer

Koden fra oppgave 2b ble kjørt 5 nye ganger, med fortsatt 50 serier og 10 kast per serie. Histogrammene fra disse kjøringene er lagt ved i rapporten. (19)

Selv om alle histogrammene viser toppunkt rundt 6 til 8 suksesser, varierer de en del fra kjøring til kjøring. Dette skyldes tilfeldig variasjon, og er forventet når antall simuleringer bare er 50.

```
1  for (j in 1:5) {
2    data = c()
3
4    for (i in 1:50) {
5      k = rbinom(1, 10, 2/3)
6      data = c(data, k)
7    }
8
9    hist(data, breaks = 100, col = "purple",
10         main = paste("Histogram, kjøring", j),
11         xlab = "Antall suksesser")
12 }
```

Listing 8: R-kode for fem nye histogrammer med 50 simuleringer i oppgave 2d

3.5 2e) Fem kjøring med 100000 simuleringer

Deretter ble antall simuleringer økt fra 50 til 100000, og koden ble kjørt 5 ganger på nytt. Histogrammene fra disse kjøringene er også lagt ved i rapporten. (20)

Når antall simuleringer blir så stort, blir histogrammene langt mer stabile. Fordelingen får en tydeligere og jevnere form, og resultatene ligger mye nærmere den teoretiske binomiske fordelingen enn i oppgave 2d.

```
1  for (j in 1:5) {
2    data = c()
3
4    for (i in 1:100000) {
5      k = rbinom(1, 10, 2/3)
6      data = c(data, k)
7    }
8
9    hist(data, breaks = 100, col = "purple",
10         main = paste("Histogram, 100000 simuleringer, kjøring", j),
11         xlab = "Antall suksesser")
12 }
```

Listing 9: R-kode for fem histogrammer med 100000 simuleringer i oppgave 2e

3.6 2f) Viktigste forskjell mellom 50 og 100000 simuleringer

Den viktigste forskjellen mellom histogrammene fra 50 og 100000 simuleringer er hvor stabile de er. Når loopen bare kjøres 50 ganger, blir histogrammet ujevnt og påvirkes mye av tilfeldig variasjon. Små endringer fra kjøring til kjøring gir da tydelige utslag i søylene.

Når loopen derimot kjøres 100000 ganger, blir histogrammet mye jevnere. Fordelingen får da en form som ligger svært nær den teoretiske binomiske fordelingsgraf. Dette viser at store simuleringer gir bedre samsvar med den underliggende sannsynlighetsfordelingen.

3.7 2g) Funksjonsgrafen til $\text{Bin}(10, \frac{2}{3})$

I denne oppgaven er sannsynligheten for suksess

$$p = \frac{2}{3},$$

og antall forsøk er $n = 10$. Den teoretiske fordelingen er derfor

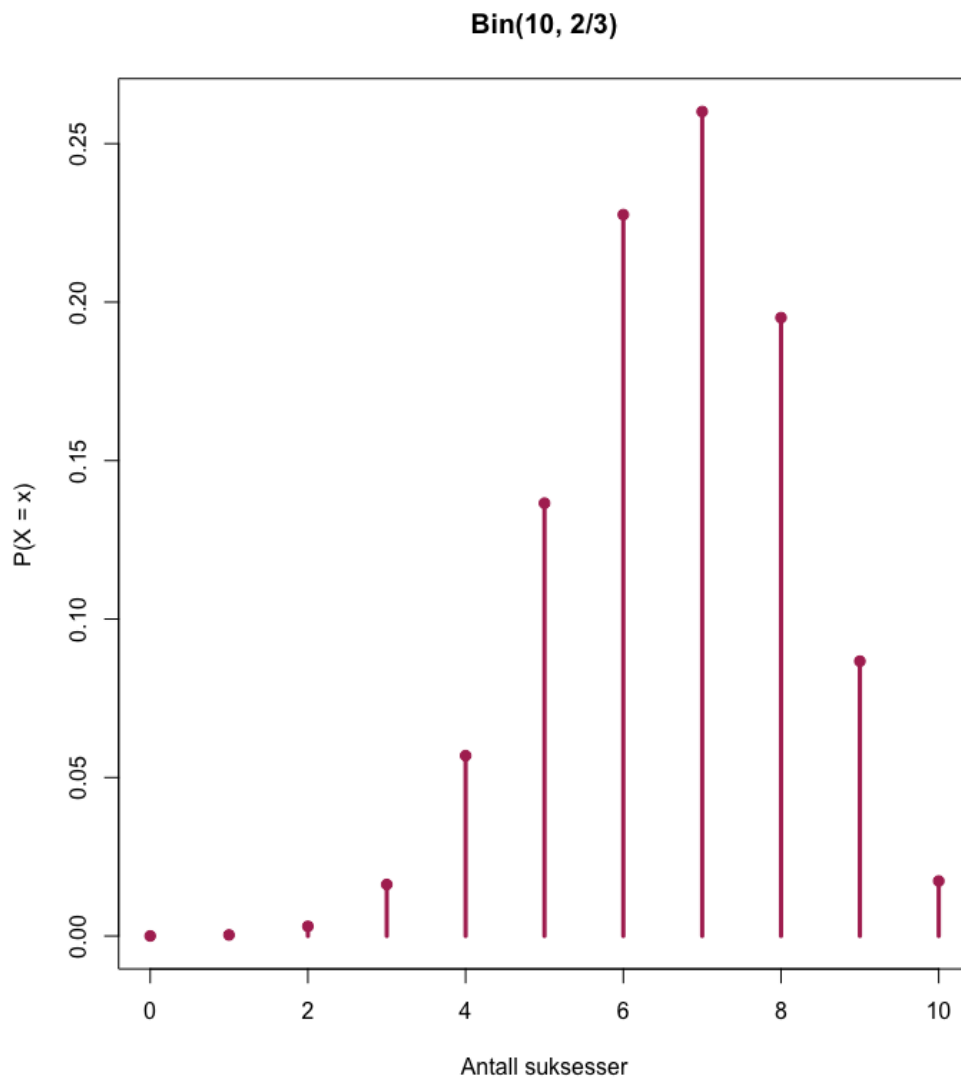
$$X \sim \text{Bin}(10, \frac{2}{3}).$$

For å tegne funksjonsgrafen til denne fordelingen kan vi beregne sannsynlighetene

$$P(X = x), \quad x = 0, 1, 2, \dots, 10.$$

```
1  x = 0:10
2  y = dbinom(x, 10, 2/3)
3
4  plot(x, y, type = "h", lwd = 3, col = "maroon",
5        main = "Bin(10, 2/3)",
6        xlab = "Antall suksesser", ylab = "P(X = x)")
7  points(x, y, pch = 19, col = "maroon")
```

Listing 10: R-kode for funksjonsgrafen til den binomiske fordelingen i oppgave 2g



Figur 1: Resulterende funksjonsgraf fra å kjøre koden (10)

3.8 2h) Hva forteller funksjonsgrafen oss? Hva er sammenhengen med histogrammene over?

Funksjonsgrafen til $\text{Bin}(10, \frac{2}{3})$ viser den teoretiske sannsynlighetsfordelingen for antall suksesser i 10 forsøk, der sannsynligheten for suksess i hvert forsøk er $\frac{2}{3}$. For hver mulig verdi $x = 0, 1, \dots, 10$ viser grafen hvor sannsynlig det er å få akkurat x suksesser.

Histogrammene fra simuleringene viser derimot observerte resultater fra mange tilfeldige kjøring av samme prosess. Når antall simuleringer er lite, vil histogrammene variere en del fra kjøring til

kjøring på grunn av tilfeldig variasjon. Når antall simuleringer blir stort, vil histogrammet få en form som ligner mer og mer på funksjonsgrafen til den teoretiske binomiske fordelingen.

Sammenhengen er derfor at funksjonsgrafen viser den underliggende, teoretiske fordelingen, mens histogrammene viser empiriske tilnærminger til denne fordelingen. Jo flere simuleringer vi gjør, desto bedre samsvarer histogrammet med funksjonsgrafen.

4 Oppgave 3: Simulering av Poisson-fordeling

4.1 3a) Simulering av Poisson-fordeling med $\lambda = 2$

I denne oppgaven skal koden fra oppgave 2 endres slik at den simulerer en Poisson-fordeling med parameter

$$\lambda = 2.$$

Det betyr at vi i stedet for binomisk simulering bruker kommandoen `rpois(1,2)` for å generere ett tilfeldig trekk fra `Poisson(2)`.

Koden ble kjørt 5 ganger, der hver kjøring bestod av 200 simuleringer. Histogrammene fra disse kjøringene er lagt ved i rapporten. 21

```
1   for (j in 1:5) {
2     data = c()
3
4     for (i in 1:200) {
5       k = rpois(1, 2)
6       data = c(data, k)
7     }
8
9     hist(data,
10      breaks = seq(-0.5, max(data) + 0.5, 1),
11      col = "purple",
12      main = paste("Poisson(2), kjøring", j),
13      xlab = "Antall hendelser")
14 }
```

Listing 11: R-kode for fem simuleringer av Poisson-fordelingen med 200 runder

4.2 3b) Funksjonsgrafen til `Poisson(2)`

Den teoretiske sannsynlighetsfordelingen er her

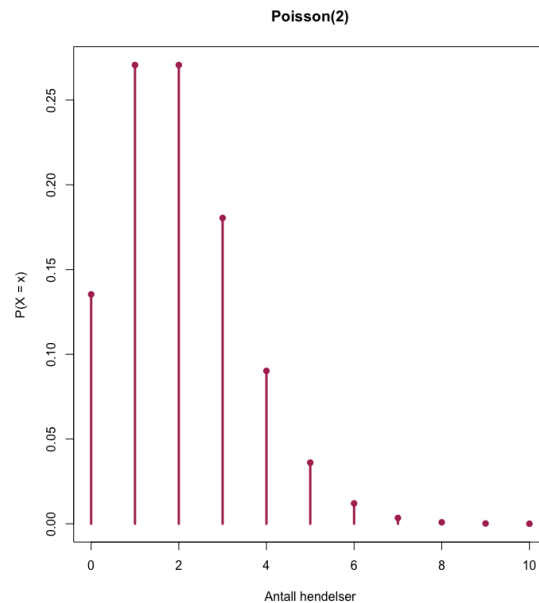
$$X \sim \text{Poisson}(2).$$

For å tegne funksjonsgrafen beregner vi sannsynlighetene

$$P(X = x), \quad x = 0, 1, 2, \dots, 10.$$

```
1   x = 0:10
2   y = dpois(x, 2)
3
4   plot(x, y, type = "h", lwd = 3, col = "maroon",
5        main = "Poisson(2)",
6        xlab = "Antall hendelser", ylab = "P(X = x)")
7   points(x, y, pch = 19, col = "maroon")
```

Listing 12: R-kode for funksjonsgrafen til `Poisson(2)`



Figur 2: Graf for teoretisk sannsynlighetsfordeling i oppgave 3

4.3 3c) Sammenligning mellom histogrammer og funksjonsgraf

Histogrammene fra simuleringene viser observerte frekvenser fra tilfeldige trekk, mens funksjonsgrafen viser de eksakte sannsynlighetene i den teoretiske Poisson-fordelingen.

I histogrammene med 200 simuleringer ser vi at de fleste observasjonene ligger rundt 1 og 2, og i noe mindre grad rundt 3. Dette stemmer godt med funksjonsgrafen til $\text{Poisson}(2)$, der sannsynligheten er størst for $x = 1$ og $x = 2$, og deretter avtar for større verdier.

Histogrammene er høyreskjeve og har en tynn hale mot større verdier som 5, 6, 7 og 8. Samtidig varierer søylehøydene noe fra kjøring til kjøring. Dette skyldes tilfeldig variasjon og er forventet når antall simuleringer bare er 200.

4.4 3d) Simulering med 100000 runder

Deretter ble loopen kjørt med 100000 simuleringer i stedet for 200. For denne delen er ett histogram tilstrekkelig.

```
1 data = c()
2
3 for (i in 1:100000) {
4   k = rpois(1, 2)
5   data = c(data, k)
6 }
7
8 hist(data,
9       breaks = seq(-0.5, max(data) + 0.5, 1),
10      col = "purple",
11      main = "Poisson(2), 100000 simuleringer",
12      xlab = "Antall hendelser")
```

Listing 13: R-kode for simulering av Poisson(2) med 100000 runder

4.5 3e) Viktigste forskjell mellom 200 og 100000 simuleringer

Den viktigste forskjellen mellom histogrammene med 200 og 100000 simuleringer er hvor tydelig den underliggende fordelingen kommer fram. Ved 200 simuleringer ser vi riktig hovedform, men histogrammene varierer fortsatt noe fra kjøring til kjøring, og søylene blir derfor litt ujevne.

Når antall simuleringer økes til 100000, blir histogrammet mye glattere og mer stabilt. Da ser vi svært tydelig at de største frekvensene ligger ved 1 og 2, og at frekvensene avtar gradvis for større verdier. Histogrammet samsvarer derfor svært godt med funksjonsgrafen til Poisson(2).

Dette illustrerer at store simuleringer gir et mer presist bilde av den teoretiske sannsynlighetsfordelingen enn små simuleringer gjør.

5 Oppgave 4: Kontinuerlige sannsynlighetsfordelinger

5.1 4a) $X \sim \phi(37, 5)$, finn $P(X \geq 33)$

I denne oppgaven er X normalfordelt med forventningsverdi

$$\mu = 37$$

og standardavvik

$$\sigma = 5.$$

Vi skriver dette som

$$X \sim \phi(37, 5).$$

Vi skal finne sannsynligheten

$$P(X \geq 33).$$

Dette tilsvarer arealet under tetthetsfunksjonen til høyre for $x = 33$.

Ved hjelp av fordelingsfunksjonen får vi

$$P(X \geq 33) = 1 - P(X \leq 33) = 1 - F(33).$$

I R beregnes dette med

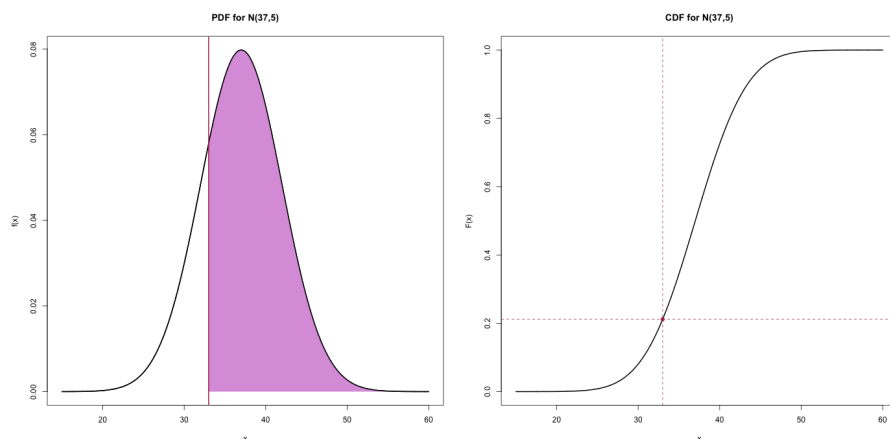
$$P(X \geq 33) = 1 - \text{pnorm}(33, 37, 5) \approx 0.7881.$$

Sannsynligheten for at X er større enn eller lik 33, er altså omtrent 0.7881, som tilsvarer 78.81%.

```
1  pnorm(33, mean = 37, sd = 5, lower.tail = FALSE)
```

Listing 14: R-kode for beregning av $P(X \geq 33)$ i oppgave 4a

Figurene under viser både tetthetsfunksjonen med markert område til høyre for $x = 33$, og fordelingsfunksjonen med markert verdi $F(33)$.



Figur 3: PDF og CDF for $X \sim \phi(37, 5)$, med markering av området $P(X \geq 33)$ i oppgave 4a

5.2 4b) Oppgave 10.7.3.c

Her er

$$X \sim \phi(2, 1),$$

og vi skal finne en verdi a slik at

$$P(X \leq a) = 0.05.$$

Dette betyr at vi skal finne 5%-kvantilen til normalfordelingen $\phi(2, 1)$. I R kan dette beregnes med den inverse fordelingsfunksjonen `qnorm`.

Vi får

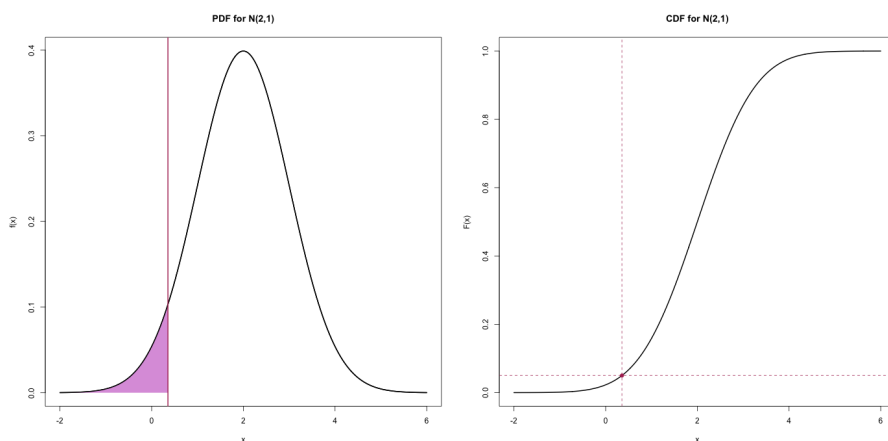
$$a = \text{qnorm}(0.05, 2, 1) \approx 0.3551.$$

Verdien av a som gir $P(X \leq a) = 0.05$, er altså omtrent

$$a \approx 0.3551.$$

```
1  qnorm(0.05, mean = 2, sd = 1)
```

Listing 15: R-kode for beregning av kvantilen i oppgave 4b



Figur 4: PDF og CDF for $X \sim \phi(2, 1)$, med markering av kvantilen a slik at $P(X \leq a) = 0.05$ i oppgave 4b

5.3 4c) Oppgave 10.7.11

Her er

$$T \sim \exp_{4.4}.$$

For en eksponentialfordeling med rateparameter $\lambda = 4.4$ gjelder at

$$\mu_T = \frac{1}{\lambda}, \quad \sigma_T = \frac{1}{\lambda}.$$

Dermed får vi

$$\mu_T = \frac{1}{4.4} \approx 0.2273$$

og

$$\sigma_T = \frac{1}{4.4} \approx 0.2273.$$

Vi skal også finne sannsynligheten

$$P(T \in (0.15, 0.28)).$$

Dette beregnes som

$$P(0.15 < T < 0.28) = F(0.28) - F(0.15).$$

I R får vi

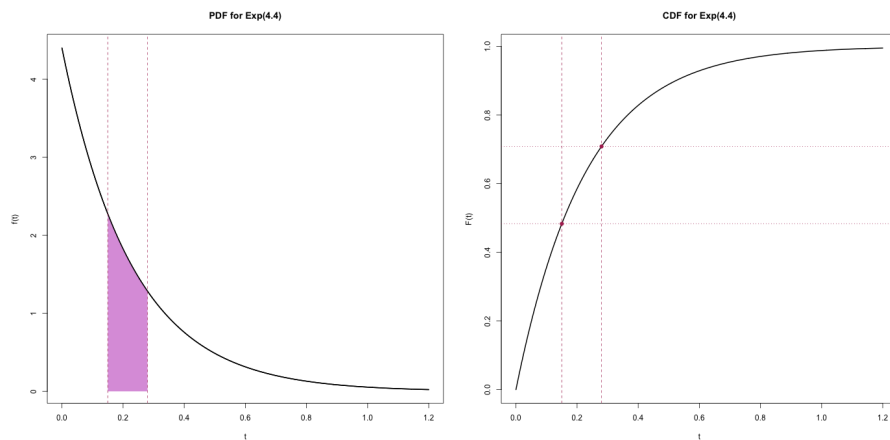
$$P(T \in (0.15, 0.28)) \approx 0.2251.$$

Altså er

$$\mu_T \approx 0.2273, \quad \sigma_T \approx 0.2273, \quad P(T \in (0.15, 0.28)) \approx 0.2251.$$

```
1 lambda = 4.4
2 1 / lambda
3 pexp(0.28, rate = lambda) - pexp(0.15, rate = lambda)
```

Listing 16: R-kode for beregningene i oppgave 4c



Figur 5: PDF og CDF for $T \sim \exp_{4.4}$, med markering av intervallet $P(T \in (0.15, 0.28))$ i oppgave 4c

5.4 4d) Oppgave 10.7.28

I denne oppgaven skal vi finne

$$t_{4,0.1},$$

altså 10%-kvantilen til Student-fordelingen med 4 frihetsgrader.

Dette betyr at vi ønsker verdien t slik at

$$P(T \leq t) = 0.1$$

når

$$T \sim t(4).$$

I R beregnes dette med den inverse fordelingsfunksjonen `qt`:

$$t_{4,0.1} = \text{qt}(0.1, 4) \approx -1.5332.$$

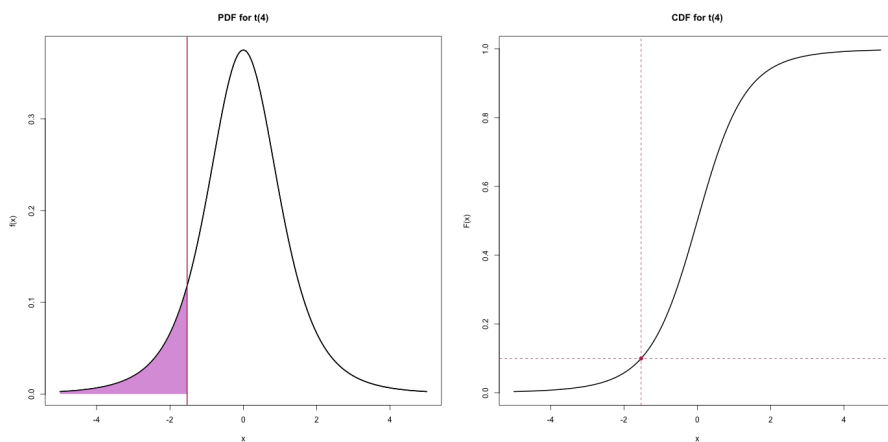
Verdien blir altså

$$t_{4,0.1} \approx -1.5332.$$

1

```
qt(0.1, df = 4)
```

Listing 17: R-kode for beregning av $t_{4,0.1}$ i oppgave 4d



Figur 6: PDF og CDF for Student-fordelingen $t(4)$, med markering av kvantilen $t_{4,0.1}$ i oppgave 4d

5.5 4e) Oppgave 10.7.37

Her er

$$X \sim \beta(3, 2).$$

For en betafordeling med parametere $a = 3$ og $b = 2$ gjelder at

$$\mu_X = \frac{a}{a+b} = \frac{3}{5} = 0.6$$

og

$$\sigma_X = \sqrt{\frac{ab}{(a+b)^2(a+b+1)}} = \sqrt{\frac{3 \cdot 2}{5^2 \cdot 6}} = 0.2.$$

Vi skal også finne sannsynligheten

$$P(X \in (0.4, 0.65)).$$

Denne beregnes som

$$P(0.4 < X < 0.65) = F(0.65) - F(0.4).$$

I R får vi

$$P(X \in (0.4, 0.65)) \approx 0.3838.$$

Dermed er

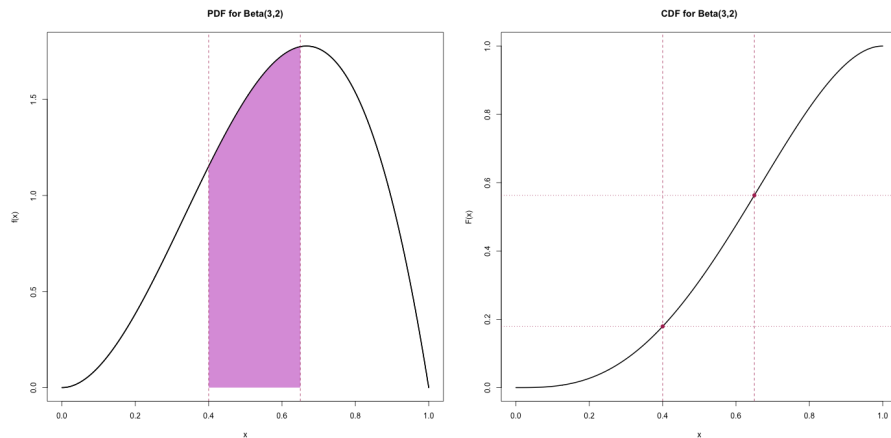
$$\mu_X = 0.6, \quad \sigma_X = 0.2, \quad P(X \in (0.4, 0.65)) \approx 0.3838.$$

```

1  a = 3
2  b = 2
3  a / (a + b)
4  sqrt((a * b) / (((a + b)^2) * (a + b + 1)))
5  pbeta(0.65, a, b) - pbeta(0.4, a, b)

```

Listing 18: R-kode for beregningene i oppgave 4e



Figur 7: PDF og CDF for $X \sim \beta(3, 2)$, med markering av intervallet $P(X \in (0.4, 0.65))$ i oppgave 4e

5.6 4f) Oppgave 10.7.51

Her er

$$T \sim \text{weib}(0.5, 3).$$

For Weibull-fordelingen med parametere $k = 0.5$ og $\lambda = 3$ gjelder at

$$\mu_T = \lambda \Gamma\left(1 + \frac{1}{k}\right)$$

og

$$\sigma_T = \sqrt{\lambda^2 \left(\Gamma\left(1 + \frac{2}{k}\right) - \Gamma\left(1 + \frac{1}{k}\right)^2 \right)}.$$

Når vi setter inn verdiene, får vi

$$\mu_T = 6$$

og

$$\sigma_T \approx 13.4164.$$

Vi skal også finne sannsynligheten

$$P(T \in (2, 4)).$$

Denne beregnes som

$$P(2 < T < 4) = F(4) - F(2).$$

I R får vi

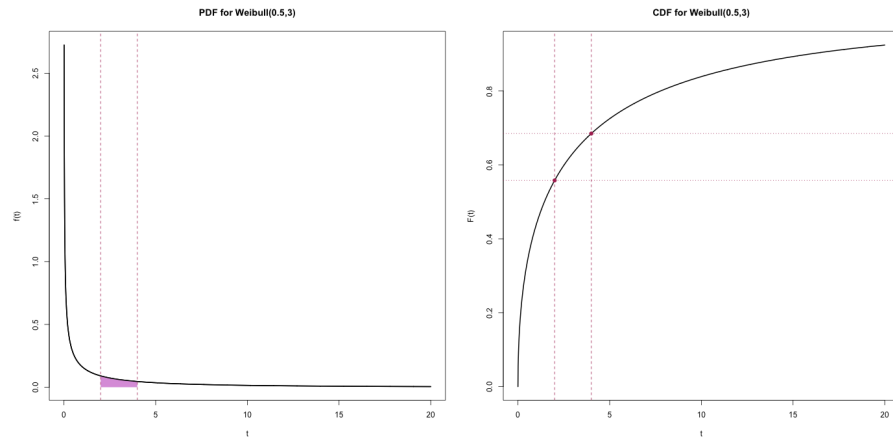
$$P(T \in (2, 4)) \approx 0.1268.$$

Dermed er

$$\mu_T = 6, \quad \sigma_T \approx 13.4164, \quad P(T \in (2, 4)) \approx 0.1268.$$

```
1 k = 0.5
2 lambda = 3
3 lambda * gamma(1 + 1/k)
4 sqrt(lambda^2 * (gamma(1 + 2/k) - gamma(1 + 1/k)^2))
5 pweibull(4, shape = k, scale = lambda) - pweibull(2, shape = k, scale = lambda)
```

Listing 19: R-kode for beregningene i oppgave 4f



Figur 8: PDF og CDF for $T \sim \text{weib}(0.5, 3)$, med markering av intervallet $P(T \in (2, 4))$ i oppgave 4f

6 Oppgave 5: Normalfordeling

6.1 5a) Graf av standard normalfordeling

Standard normalfordeling er normalfordelingen med forventningsverdi

$$\mu = 0$$

og standardavvik

$$\sigma = 1.$$

Denne fordelingen skrives som

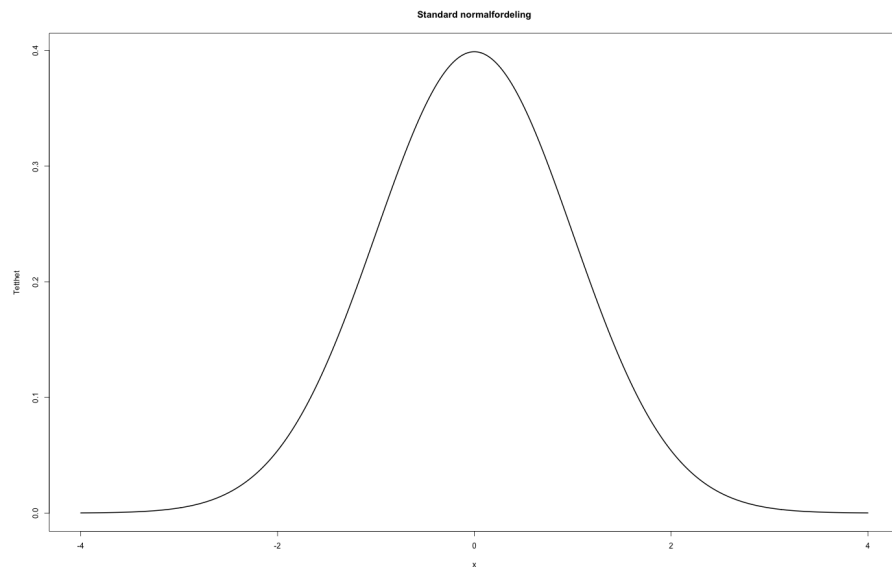
$$X \sim \phi(0, 1).$$

Grafen til standard normalfordelingen kan tegnes i R ved hjelp av tetthetsfunksjonen `dnorm`.

```
1 x = seq(-4, 4, length = 1000)
2 y = dnorm(x, mean = 0, sd = 1)
3
4 plot(x, y, type = "l", lwd = 2,
5      main = "Standard normalfordeling",
6      xlab = "x", ylab = "Tetthet")
```

Listing 20: R-kode for grafen til standard normalfordeling i oppgave 5a

Figuren under viser grafen til standard normalfordeling.



Figur 9: Graf av standard normalfordeling $N(0, 1)$ i oppgave 5a

6.2 5b) Ett tilfeldig trekk fra $\phi(0, 1)$

Et tilfeldig trekk fra standard normalfordelingen kan genereres i R med kommandoen `rnorm(1, 0, 1)`.

```
1 tilfeldig_trekk = rnorm(1, mean = 0, sd = 1)
2 tilfeldig_trekk
```

Listing 21: R-kode for ett tilfeldig trekk fra standard normalfordeling i oppgave 5b

Verdien vil variere fra gang til gang siden trekket er tilfeldig.

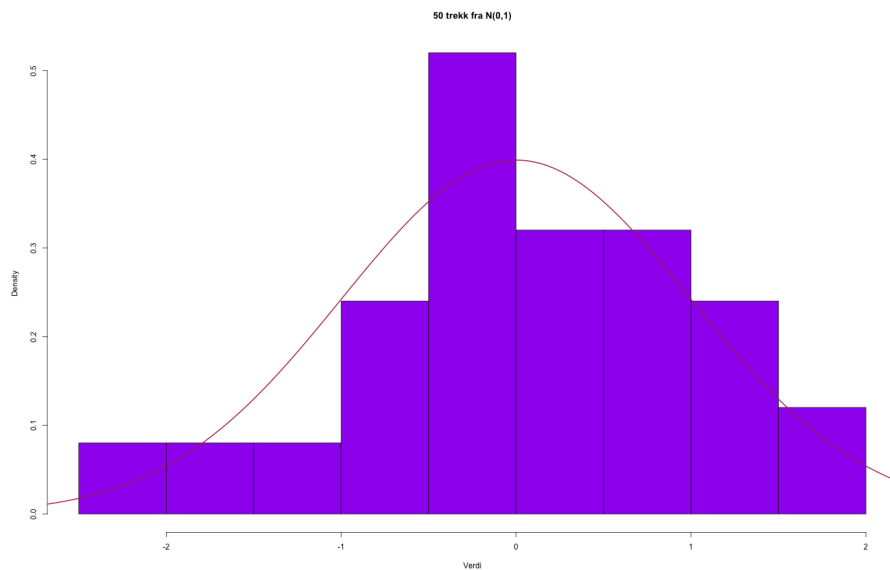
6.3 5c) 50 tilfeldige trekk og histogram med normalfordelingen tegnet inn

Deretter genereres 50 tilfeldige trekk fra $\phi(0, 1)$, og det lages et histogram av disse. Oppå histogrammet tegnes den teoretiske normalfordelingen.

```
1 data_50 = rnorm(50, mean = 0, sd = 1)
2
3 hist(data_50, probability = TRUE, breaks = 10,
4      col = "purple",
5      main = "50 trekk fra N(0,1)",
6      xlab = "Verdi")
7
8 xVals = seq(-4, 4, 0.01)
9 yVals = dnorm(xVals, mean = 0, sd = 1)
10 lines(xVals, yVals, col = "maroon", lwd = 2)
```

Listing 22: R-kode for histogram av 50 trekk med normalfordelingen tegnet inn i oppgave 5c

Histogrammet gir et grovt bilde av standard normalfordelingen. Siden utvalget bare består av 50 observasjoner, vil histogrammet være ganske ujevnt og påvirket av tilfeldig variasjon. Likevel ser vi at formen er omtrent klokkeformet og sentrert rundt 0.



Figur 10: Histogram av 50 trekk fra standard normalfordeling med teoretisk kurve tegnet inn i oppgave 5c

6.4 5d) Histogrammer for 500 og 50000 trekk

Oppgaven ber deretter om å gjøre det samme for 500 og 50000 trekk. Når antall trekk økes, vil histogrammet i større grad ligne den teoretiske normalfordelingen.

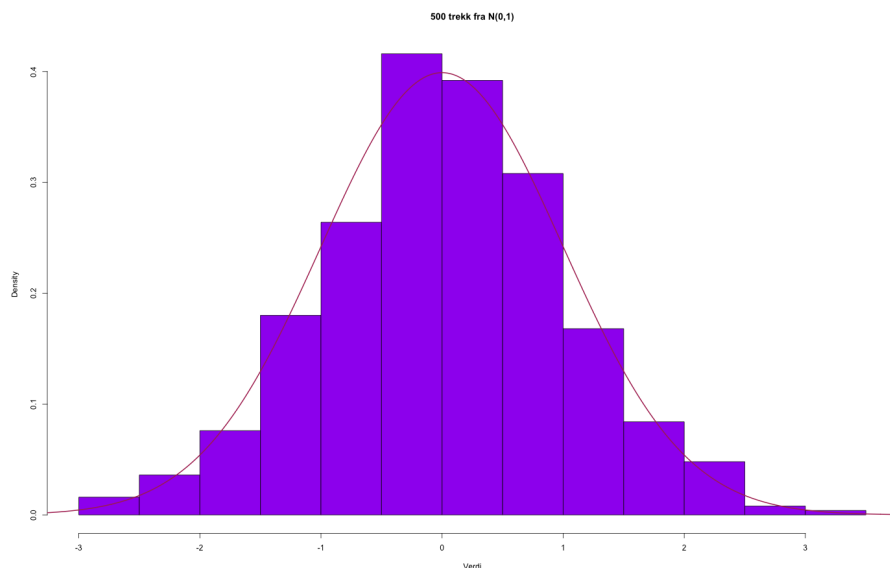
```

1 data_500 = rnorm(500, mean = 0, sd = 1)
2
3 hist(data_500, probability = TRUE, breaks = 20,
4       col = "purple",
5       main = "500 trekk fra N(0,1)",
6       xlab = "Verdi")
7
8 xVals = seq(-4, 4, 0.01)
9 yVals = dnorm(xVals, mean = 0, sd = 1)
10 lines(xVals, yVals, col = "maroon", lwd = 2)
11
12 data_50000 = rnorm(50000, mean = 0, sd = 1)
13
14 hist(data_50000, probability = TRUE, breaks = 40,
15       col = "purple",
16       main = "50000 trekk fra N(0,1)",
17       xlab = "Verdi")
18
19 lines(xVals, yVals, col = "maroon", lwd = 2)

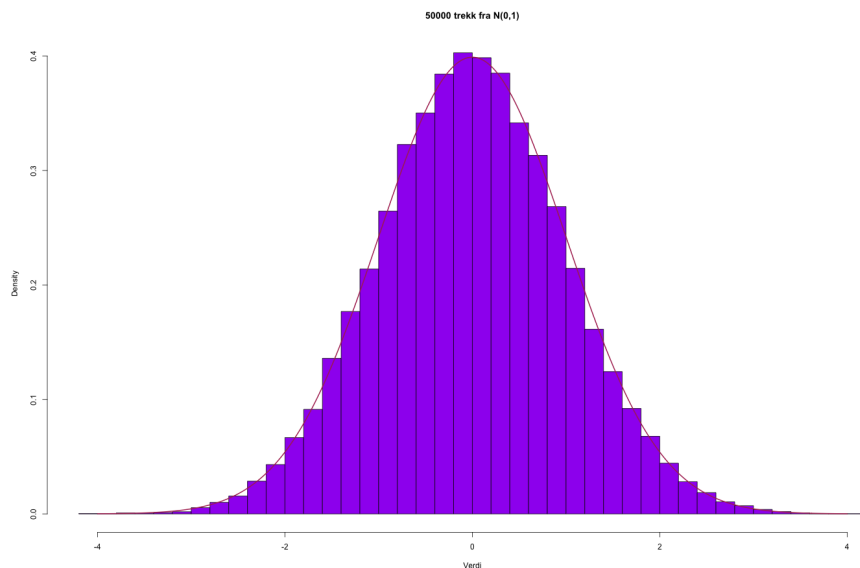
```

Listing 23: R-kode for histogrammer av 500 og 50000 trekk med normalfordelingen tegnet inn i oppgave 5d

Når vi går fra 50 til 500 og videre til 50000 trekk, blir histogrammene gradvis glattere og mer symmetriske. Samtidig ser vi at histogrammet passer stadig bedre med den teoretiske normalfordelingen som er tegnet inn.



Figur 11: Histogram av 500 trekk fra standard normalfordeling med teoretisk kurve tegnet inn i oppgave 5d



Figur 12: Histogram av 50000 trekk fra standard normalfordeling med teoretisk kurve tegnet inn i oppgave 5d

6.5 5e) Forklaring av den oppgitte R-koden for $N = 100$

I oppgaven skal følgende kode kjøres og forklares:

```

1  N = 100
2  h = (rbinom(N,100,0.5)-50)/5
3  hist(h, probability = TRUE, breaks = seq(-8.05,8.05,0.1))
4  xVals = seq(-4,4,0.01)
5  yVals = 2 * dnorm(xVals,0,1)
6  lines(xVals, yVals, col = "maroon", type = "l")

```

Listing 24: Oppgitt R-kode i oppgave 5e

Koden genererer $N = 100$ binomiske tilfeldige variabler der hver av dem er antall suksesser i 100 forsøk med sannsynlighet 0.5 for suksess. Deretter sentreres og skales verdiene ved uttrykket

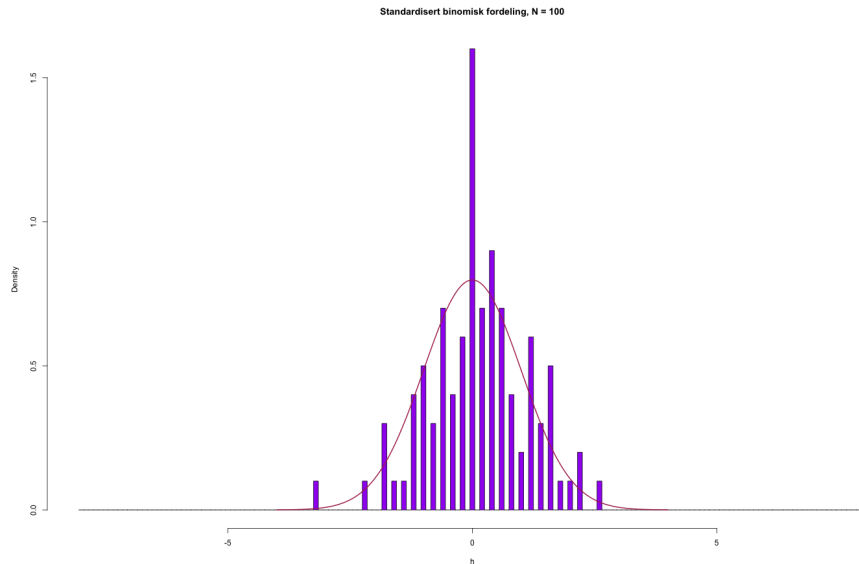
$$h = \frac{X - 50}{5}.$$

Siden en binomisk fordeling med parametere $n = 100$ og $p = 0.5$ har forventningsverdi 50 og standardavvik 5, betyr dette at verdiene i h er standardiserte versjoner av de binomiske trekningene.

Histogrammet til h sammenlignes så med kurven

$$2 \cdot \phi(0,1),$$

som tegnes inn med `lines`. Koden illustrerer dermed hvordan en binomisk fordeling, etter standardisering, får en form som ligner normalfordelingen.

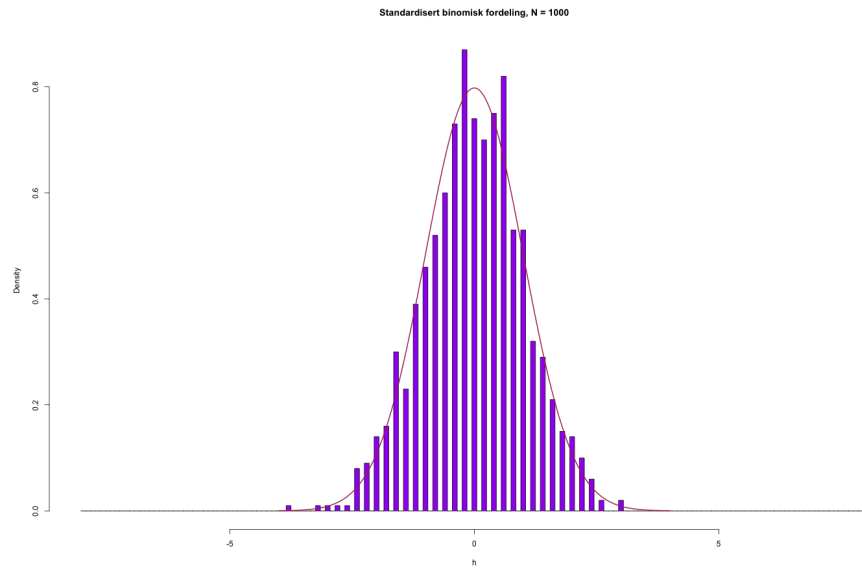


Figur 13: Histogram for den standardiserte binomiske fordelingen ved $N = 100$ i oppgave 5e

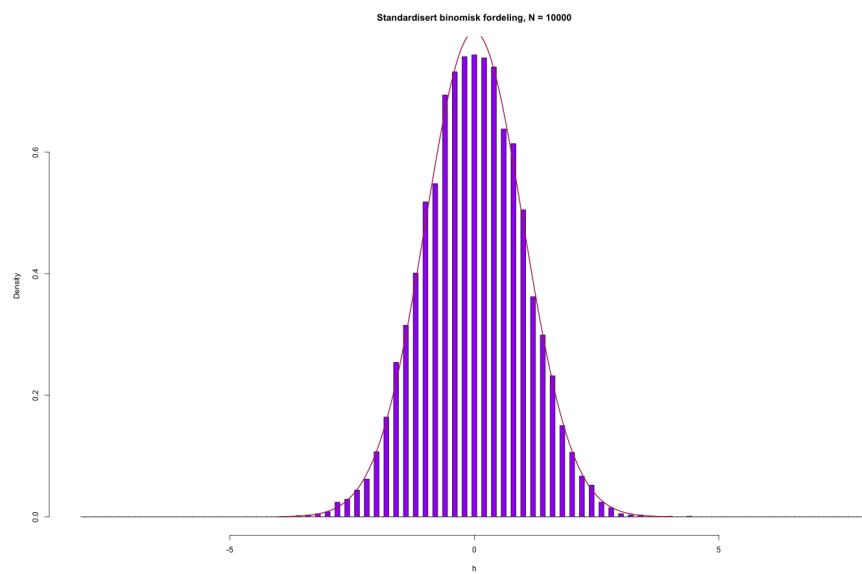
6.6 5f) Gjentakelse med $N = 1000$, 10000 og 100000

Når samme kode kjøres med større verdier av N , blir histogrammet stadig glattere og følger den inntegnede kurven tydeligere. Dette viser at når vi bruker mange simuleringer, kommer sammenhengen mellom den standardiserte binomiske fordelingen og normalfordelingen tydeligere fram.

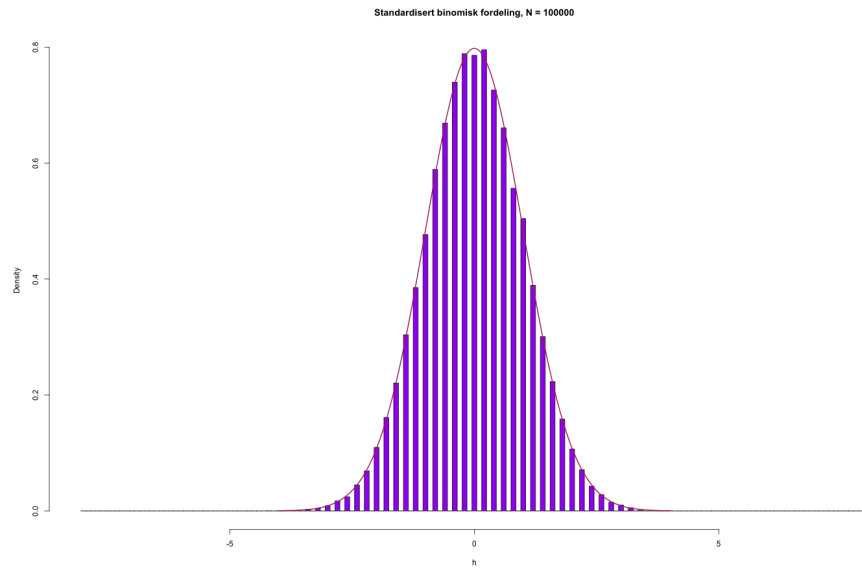
Resultatet illustrerer derfor normaltilnærming til binomisk fordeling, og viser at samsvaret med normalfordelingen blir klarere når antall simuleringer øker.



Figur 14: Histogram for den standardiserte binomiske fordelingen ved $N = 1000$ i oppgave 5f



Figur 15: Histogram for den standardiserte binomiske fordelingen ved $N = 10000$ i oppgave 5f



Figur 16: Histogram for den standardiserte binomiske fordelingen ved $N = 100000$ i oppgave 5f

7 Oppgave 6: Betafordeling

7.1 6a) Graf av $\beta(3, 7)$

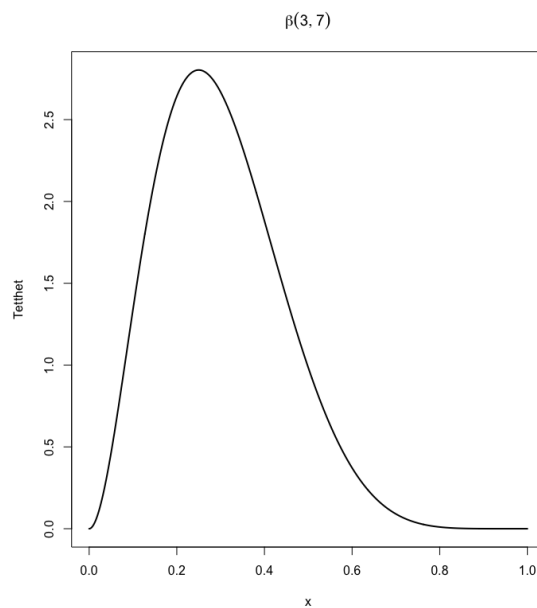
I denne oppgaven skal vi tegne grafen til betafordelingen med parametere

$$a = 3 \quad \text{og} \quad b = 7.$$

Tetthetsfunksjonen er definert på intervallet $[0, 1]$, så vi beregner verdier i dette intervallet og tegner grafen i R.

```
1 x = seq(0, 1, length = 1000)
2 y = dbeta(x, 3, 7)
3
4 plot(x, y, type = "l", lwd = 2,
5      main = expression(beta(3,7)),
6      xlab = "x", ylab = "Tetthet")
```

Listing 25: R-kode for grafen til $\beta(3, 7)$ i oppgave 6a



Figur 17: Graf til $\beta(3, 7)$

7.2 6b) Grafer for alle kombinasjoner av $a, b \in \{1, 2, 5\}$

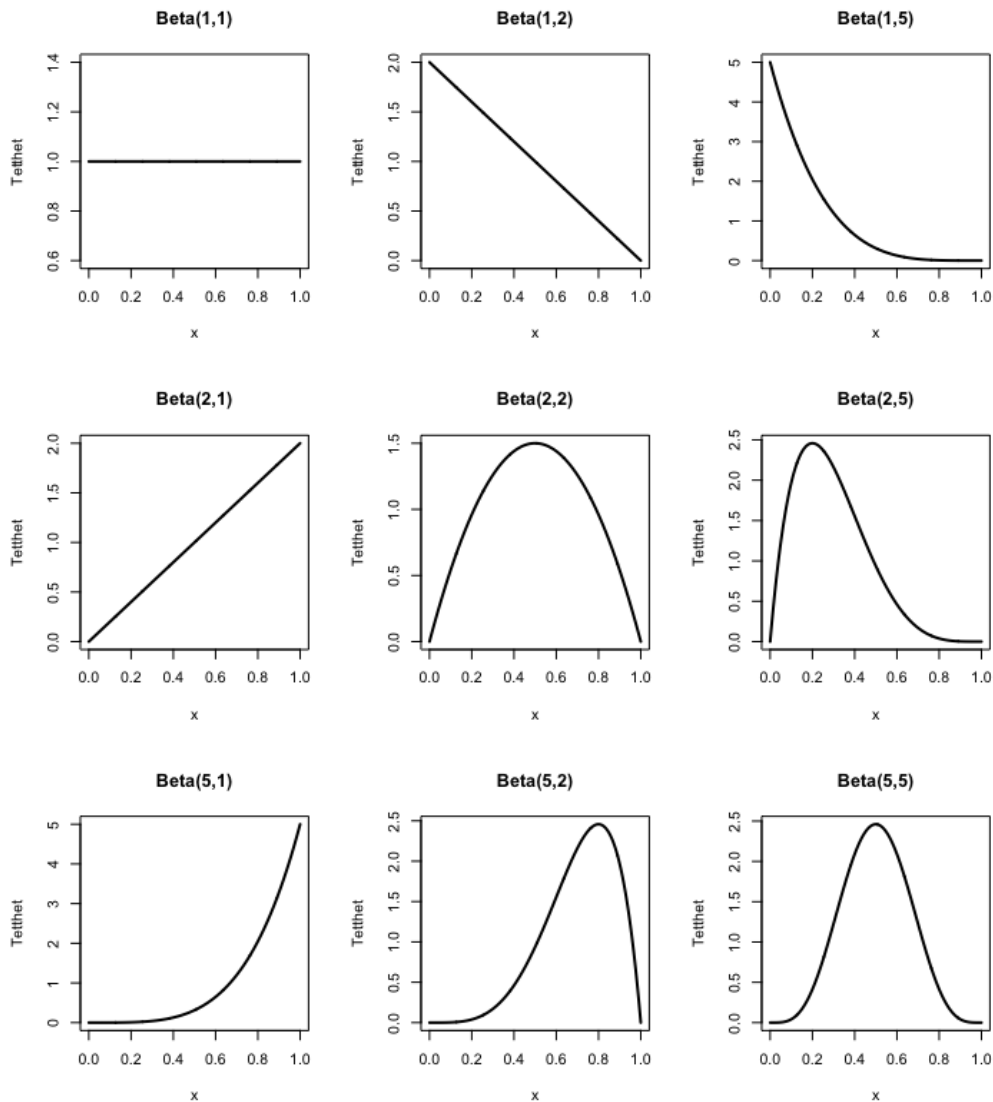
Vi tegner deretter grafene for alle kombinasjoner av parameterverdiene

$$a, b \in \{1, 2, 5\}.$$

Dette gir totalt 9 ulike betafordelinger.

```
1  par(mfrow = c(3, 3))
2
3  verdier = c(1, 2, 5)
4
5  for (a in verdier) {
6    for (b in verdier) {
7      x = seq(0, 1, length = 1000)
8      y = dbeta(x, a, b)
9
10     plot(x, y, type = "l", lwd = 2,
11          main = paste("Beta(", a, ",", b, ")", sep = ""),
12          xlab = "x", ylab = "Tetthet")
13    }
14  }
```

Listing 26: R-kode for de ni betafordelingene i oppgave 6b



Figur 18: Grafer for alle kombinasjoner av parameterverdiene

7.3 6c) Hvordan parameterne påvirker formen

Grafene viser at parameterne a og b har stor betydning for formen til betafordelingen.

Når $a = b$, blir fordelingen symmetrisk rundt $x = 0.5$. Dette ser vi for eksempel for $\beta(1,1)$, $\beta(2,2)$ og $\beta(5,5)$. Fordelingen $\beta(1,1)$ er uniform, mens $\beta(2,2)$ og $\beta(5,5)$ får en tydelig topp rundt midten. Jo større de like parameterne er, desto mer konsentrert blir fordelingen rundt $x = 0.5$.

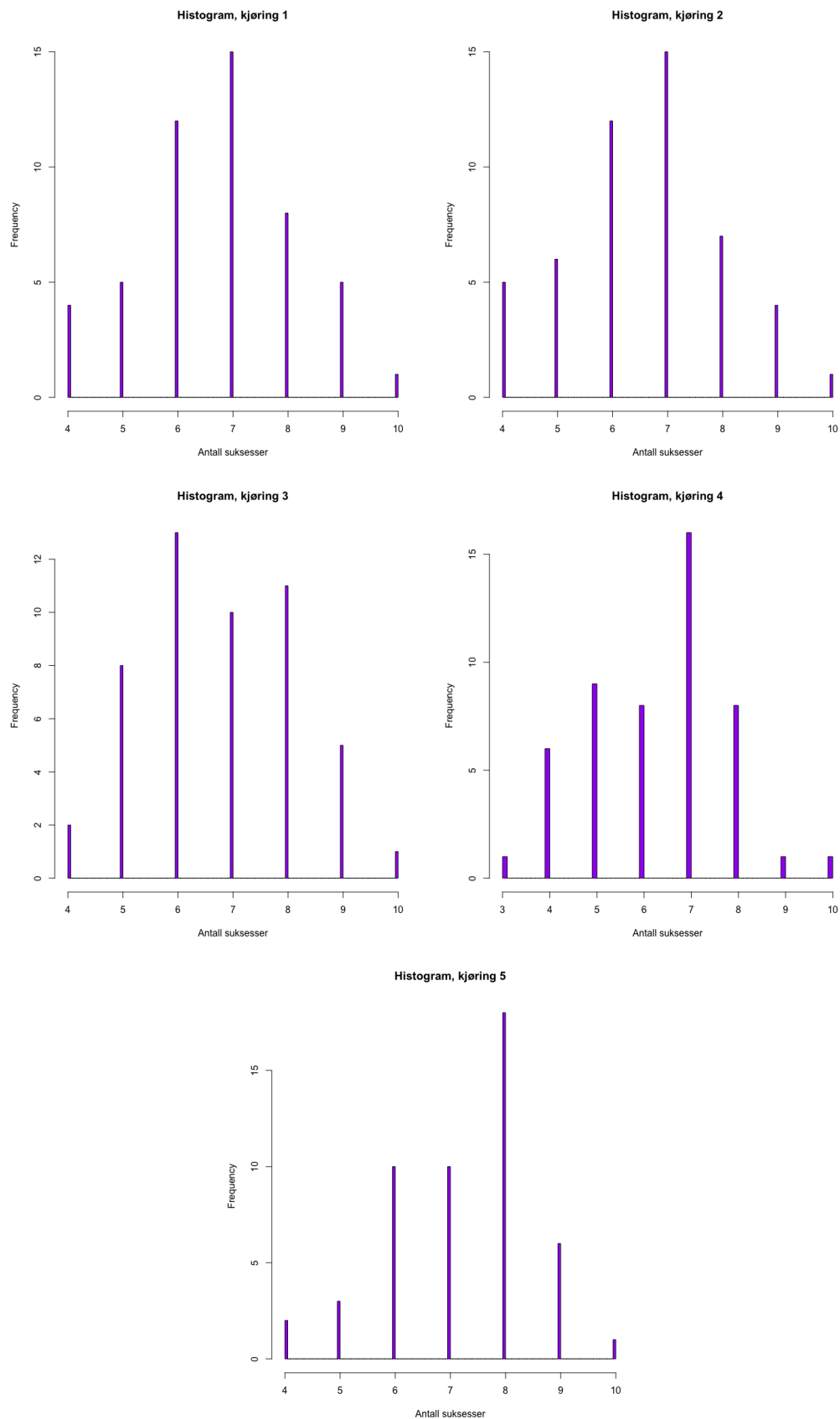
Når $a < b$, blir fordelingen trukket mot venstre, slik at små verdier av x blir mer sannsynlige.

Dette ser vi blant annet for $\beta(1, 2)$, $\beta(1, 5)$, $\beta(2, 5)$ og $\beta(3, 7)$.

Når $a > b$, blir fordelingen trukket mot høyre, slik at store verdier av x blir mer sannsynlige. Dette ser vi for eksempel for $\beta(2, 1)$, $\beta(5, 1)$ og $\beta(5, 2)$.

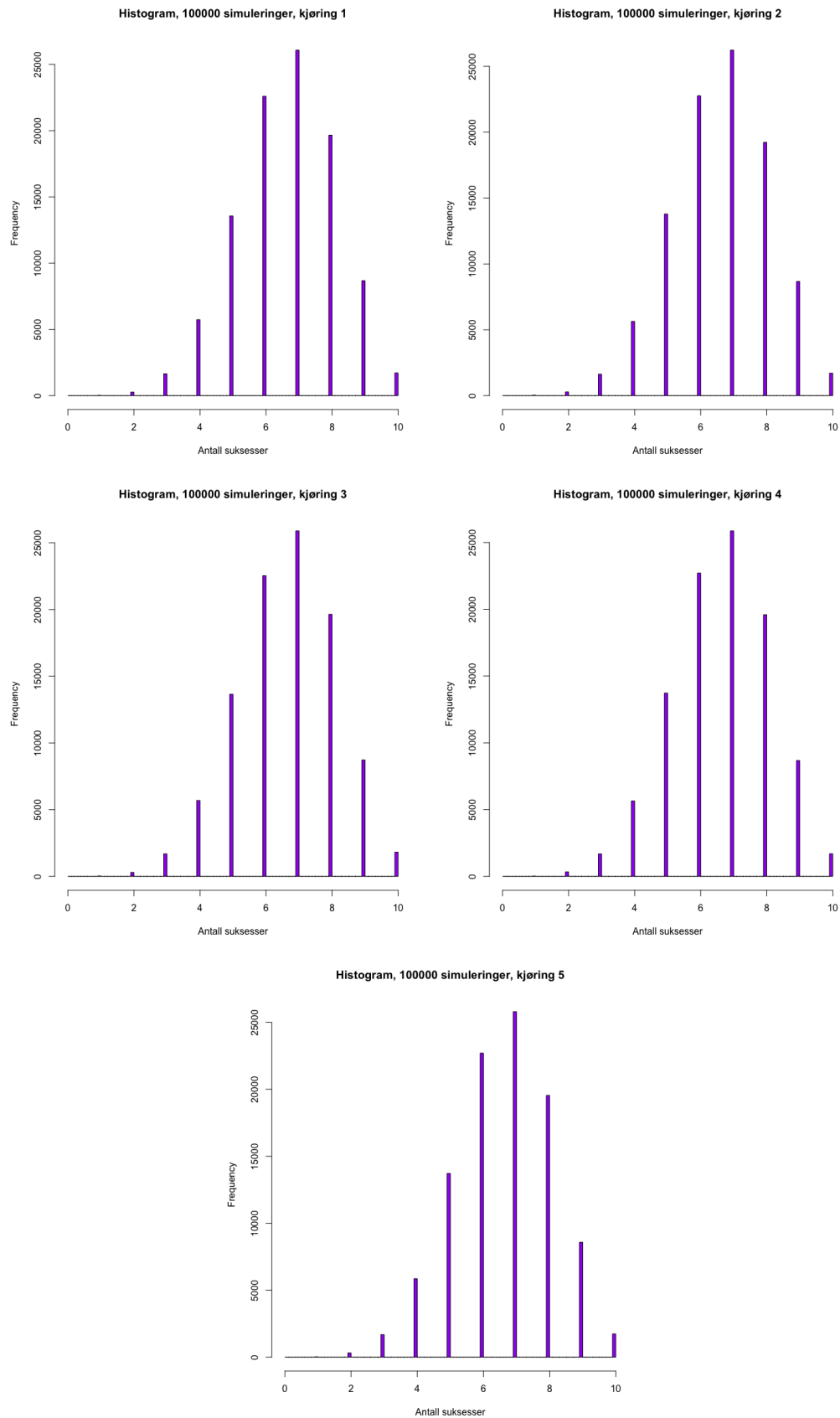
Vedlegg

5 histogrammer tilhørende oppgave 2d)



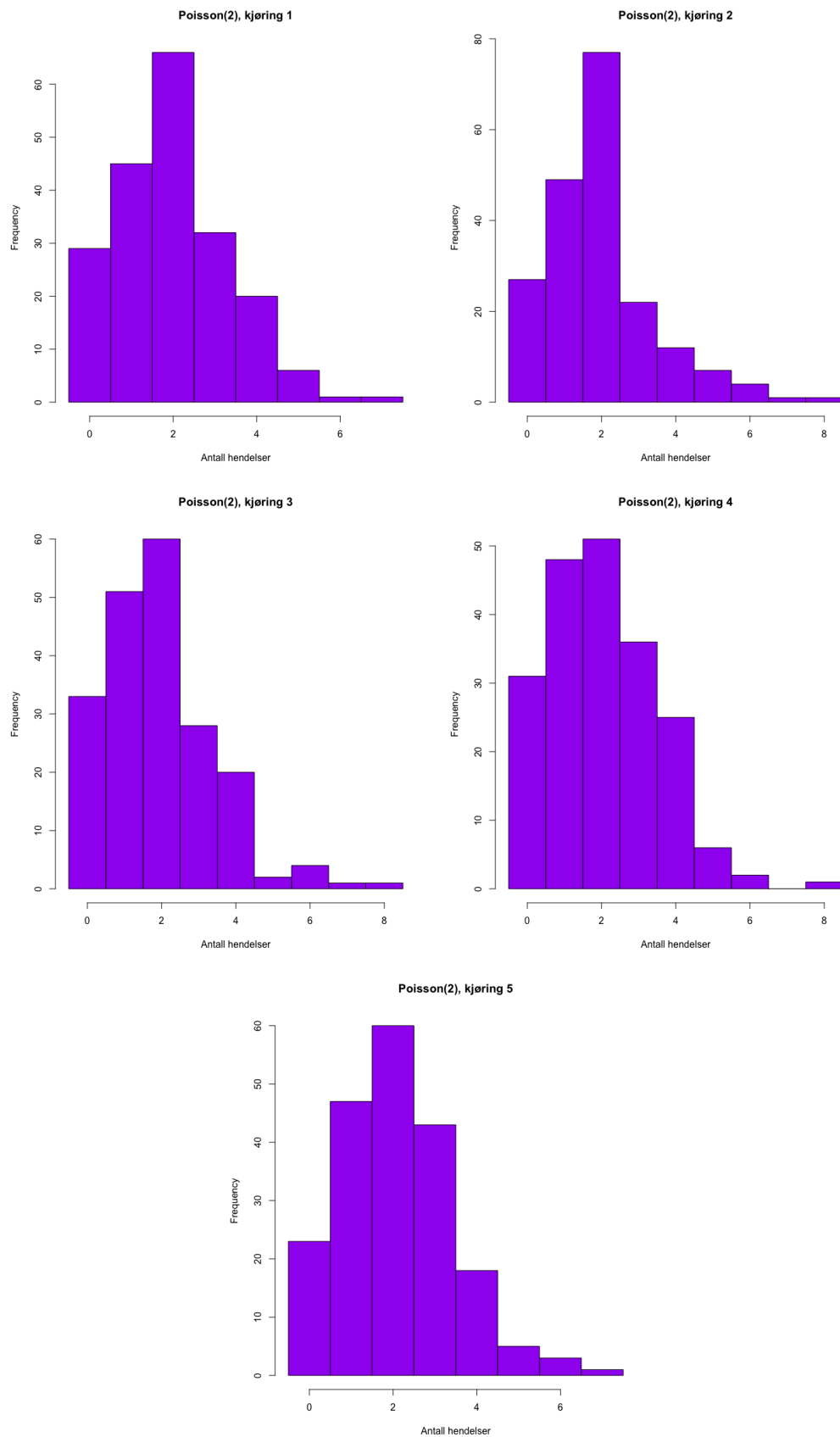
Figur 19: De fem resulterende histogrammene fra å kjøre koden fra oppgave 2b)

5 histogrammer tilhørende oppgave 2e)



Figur 20: De fem resulterende histogrammene fra å kjøre koden fra oppgave 2e)

5 histogrammer tilhørende oppgave 3a)



Figur 21: De fem resulterende histogrammene fra å kjøre koden fra oppgave 3a)

```

1  #Oppgave 40 i læreboka
2  #a)
3  P_knepet = 1/4
4  P_knepet
5
6
7  #b)
8  P_2av5 = dbinom(2, 5, 1/4)
9  P_2av5
10
11
12  #c)
13  ganger_tatt_totalt = 2
14  antall_passeringer = ganger_tatt_totalt / P_knepet
15  standardavvik = sqrt(antall_passeringer * P_knepet * (1 - P_knepet))
16
17  antall_passeringer
18  standardavvik
19
20
21  #d)
22  passeringer = 12
23  leser_ved_passering = 7
24  antall_tatt = 4
25  ønsket_tatt_og_leser = 3
26  P_leste_og_tatt = dhyper(ønsket_tatt_og_leser, leser_ved_passering,
27                          ( passeringer - leser_ved_passering ), antall_tatt)
28  P_leste_og_tatt
29
30
31  #f)
32  antall_biler = 16384
33  P_wunderbaum = 1 / 1024
34  lambda = antall_biler * P_wunderbaum
35
36  P_nøyaktig_15 = dpois(15, lambda)
37  P_nøyaktig_15
38
39
40  #2a)
41  rbinom(1, 10, 2/3)
42  rbinom(10, 1, 2/3)
43  sum(rbinom(10, 1, 2/3))
44
45
46  #b)
47  data = c()

```

```

48
49 for (i in 1:50) {
50   k = rbinom(1, 10, 2/3)
51   data = c(data, k)
52 }
53
54 table(data)
55 hist(data, breaks = 100, col = "purple")
56
57 #d)
58 for (j in 1:5) {
59   data = c()
60
61   for (i in 1:50) {
62     k = rbinom(1, 10, 2/3)
63     data = c(data, k)
64   }
65
66   hist(data, breaks = 100, col = "purple",
67        main = paste("Histogram, kjøring", j),
68        xlab = "Antall suksesser")
69 }
70
71
72 #e)
73 for (j in 1:5) {
74   data = c()
75
76   for (i in 1:100000) {
77     k = rbinom(1, 10, 2/3)
78     data = c(data, k)
79   }
80
81   hist(data, breaks = 100, col = "purple",
82        main = paste("Histogram, 100000 simuleringer, kjøring", j),
83        xlab = "Antall suksesser")
84 }
85
86
87 #f)
88 x = 0:10
89 y = dbinom(x, 10, 2/3)
90
91 plot(x, y, type = "h", lwd = 3, col = "maroon",
92      main = "Bin(10, 2/3)",
93      xlab = "Antall suksesser", ylab = "P(X = x)")
94 points(x, y, pch = 19, col = "maroon")
95

```

```

96
97
98 #3)
99 for (j in 1:5) {
100   data = c()
101
102   for (i in 1:200) {
103     k = rpois(1, 2)
104     data = c(data, k)
105   }
106
107   hist(data,
108         breaks = seq(-0.5, max(data) + 0.5, 1),
109         col = "purple",
110         main = paste("Poisson(2), kjøring", j),
111         xlab = "Antall hendelser")
112 }
113
114
115
116 x = 0:10
117 y = dpois(x, 2)
118
119 plot(x, y, type = "h", lwd = 3, col = "maroon",
120      main = "Poisson(2)",
121      xlab = "Antall hendelser", ylab = "P(X = x)")
122 points(x, y, pch = 19, col = "maroon")
123
124
125
126 data = c()
127
128 for (i in 1:100000) {
129   k = rpois(1, 2)
130   data = c(data, k)
131 }
132
133 hist(data,
134       breaks = seq(-0.5, max(data) + 0.5, 1),
135       col = "purple",
136       main = "Poisson(2), 100000 simuleringer",
137       xlab = "Antall hendelser")
138
139
140
141 par(mfrow = c(1, 1))
142
143 # 4a

```

```

144 png("/Users/chris/Maths/MA-223/Oblig/2b/oppg4a.png", width = 1400, height = 700)
145 par(mfrow = c(1, 2))
146
147 mu = 37
148 sigma = 5
149
150 # Sannsynlighet
151 P_X_ge_33 = pnorm(33, mean = mu, sd = sigma, lower.tail = FALSE)
152 P_X_ge_33
153
154 # PDF med skyggelegging
155 xVals = seq(15, 60, 0.01)
156 yVals = dnorm(xVals, mean = mu, sd = sigma)
157
158 plot(xVals, yVals, type = "l", lwd = 2,
159      main = "PDF for N(37,5)",
160      xlab = "x", ylab = "f(x)")
161
162 shade_x = seq(33, 60, 0.01)
163 shade_y = dnorm(shade_x, mean = mu, sd = sigma)
164
165 polygon(c(33, shade_x, 60),
166        c(0, shade_y, 0),
167        col = "plum", border = NA)
168
169 lines(xVals, yVals, lwd = 2)
170 abline(v = 33, col = "maroon", lwd = 2)
171
172 # CDF med markering
173 yCumVals = pnorm(xVals, mean = mu, sd = sigma)
174
175 plot(xVals, yCumVals, type = "l", lwd = 2,
176      main = "CDF for N(37,5)",
177      xlab = "x", ylab = "F(x)")
178
179 F_33 = pnorm(33, mean = mu, sd = sigma)
180
181 points(33, F_33, pch = 19, col = "maroon")
182 abline(v = 33, col = "maroon", lty = 2)
183 abline(h = F_33, col = "maroon", lty = 2)
184
185 dev.off()
186
187
188 # 4b
189 png("/Users/chris/Maths/MA-223/Oblig/2b/oppg4b.png", width = 1400, height = 700)
190 par(mfrow = c(1, 2))
191

```



```

192 mu = 2
193 sigma = 1
194
195 a = qnorm(0.05, mean = mu, sd = sigma)
196 a
197
198 # PDF
199 xVals = seq(-2, 6, 0.01)
200 yVals = dnorm(xVals, mean = mu, sd = sigma)
201
202 plot(xVals, yVals, type = "l", lwd = 2,
203      main = "PDF for N(2,1)",
204      xlab = "x", ylab = "f(x)")
205
206 shade_x = seq(-2, a, 0.01)
207 shade_y = dnorm(shade_x, mean = mu, sd = sigma)
208
209 polygon(c(-2, shade_x, a),
210        c(0, shade_y, 0),
211        col = "plum", border = NA)
212
213 lines(xVals, yVals, lwd = 2)
214 abline(v = a, col = "maroon", lwd = 2)
215
216 # CDF
217 yCumVals = pnorm(xVals, mean = mu, sd = sigma)
218
219 plot(xVals, yCumVals, type = "l", lwd = 2,
220      main = "CDF for N(2,1)",
221      xlab = "x", ylab = "F(x)")
222
223 points(a, 0.05, pch = 19, col = "maroon")
224 abline(v = a, col = "maroon", lty = 2)
225 abline(h = 0.05, col = "maroon", lty = 2)
226
227 dev.off()
228
229
230 # 4c
231 png("/Users/chris/Maths/MA-223/Oblig/2b/oppg4c.png", width = 1400, height = 700)
232 par(mfrow = c(1, 2))
233
234 lambda = 4.4
235
236 mu_T = 1 / lambda
237 sigma_T = 1 / lambda
238 P_interval = pexp(0.28, rate = lambda) - pexp(0.15, rate = lambda)
239

```

```

240 mu_T
241 sigma_T
242 P_interval
243
244 # PDF
245 xVals = seq(0, 1.2, 0.001)
246 yVals = dexp(xVals, rate = lambda)
247
248 plot(xVals, yVals, type = "l", lwd = 2,
249      main = "PDF for Exp(4.4)",
250      xlab = "t", ylab = "f(t)")
251
252 shade_x = seq(0.15, 0.28, 0.001)
253 shade_y = dexp(shade_x, rate = lambda)
254
255 polygon(c(0.15, shade_x, 0.28),
256        c(0, shade_y, 0),
257        col = "plum", border = NA)
258
259 lines(xVals, yVals, lwd = 2)
260 abline(v = 0.15, col = "maroon", lty = 2)
261 abline(v = 0.28, col = "maroon", lty = 2)
262
263 # CDF
264 yCumVals = pexp(xVals, rate = lambda)
265
266 plot(xVals, yCumVals, type = "l", lwd = 2,
267      main = "CDF for Exp(4.4)",
268      xlab = "t", ylab = "F(t)")
269
270 F_015 = pexp(0.15, rate = lambda)
271 F_028 = pexp(0.28, rate = lambda)
272
273 points(c(0.15, 0.28), c(F_015, F_028), pch = 19, col = "maroon")
274 abline(v = 0.15, col = "maroon", lty = 2)
275 abline(v = 0.28, col = "maroon", lty = 2)
276 abline(h = F_015, col = "maroon", lty = 3)
277 abline(h = F_028, col = "maroon", lty = 3)
278
279 dev.off()
280
281
282 # 4d
283 png("/Users/chris/Maths/MA-223/Oblig/2b/oppg4d.png", width = 1400, height = 700)
284 par(mfrow = c(1, 2))
285
286 t_verdi = qt(0.1, df = 4)
287 t_verdi

```

```

288
289 # PDF
290 xVals = seq(-5, 5, 0.01)
291 yVals = dt(xVals, df = 4)
292
293 plot(xVals, yVals, type = "l", lwd = 2,
294      main = "PDF for t(4)",
295      xlab = "x", ylab = "f(x)")
296
297 shade_x = seq(-5, t_verdi, 0.01)
298 shade_y = dt(shade_x, df = 4)
299
300 polygon(c(-5, shade_x, t_verdi),
301        c(0, shade_y, 0),
302        col = "plum", border = NA)
303
304 lines(xVals, yVals, lwd = 2)
305 abline(v = t_verdi, col = "maroon", lwd = 2)
306
307 # CDF
308 yCumVals = pt(xVals, df = 4)
309
310 plot(xVals, yCumVals, type = "l", lwd = 2,
311      main = "CDF for t(4)",
312      xlab = "x", ylab = "F(x)")
313
314 points(t_verdi, 0.1, pch = 19, col = "maroon")
315 abline(v = t_verdi, col = "maroon", lty = 2)
316 abline(h = 0.1, col = "maroon", lty = 2)
317
318 dev.off()
319
320
321 # 4e
322 png("/Users/chris/Maths/MA-223/Oblig/2b/oppg4e.png", width = 1400, height = 700)
323 par(mfrow = c(1, 2))
324
325 a = 3
326 b = 2
327
328 mu_X = a / (a + b)
329 sigma_X = sqrt((a * b) / (((a + b)^2) * (a + b + 1)))
330 P_interval = pbeta(0.65, a, b) - pbeta(0.4, a, b)
331
332 mu_X
333 sigma_X
334 P_interval
335

```

```

336 # PDF
337 xVals = seq(0, 1, 0.001)
338 yVals = dbeta(xVals, a, b)
339
340 plot(xVals, yVals, type = "l", lwd = 2,
341      main = "PDF for Beta(3,2)",
342      xlab = "x", ylab = "f(x)")
343
344 shade_x = seq(0.4, 0.65, 0.001)
345 shade_y = dbeta(shade_x, a, b)
346
347 polygon(c(0.4, shade_x, 0.65),
348        c(0, shade_y, 0),
349        col = "plum", border = NA)
350
351 lines(xVals, yVals, lwd = 2)
352 abline(v = 0.4, col = "maroon", lty = 2)
353 abline(v = 0.65, col = "maroon", lty = 2)
354
355 # CDF
356 yCumVals = pbeta(xVals, a, b)
357
358 plot(xVals, yCumVals, type = "l", lwd = 2,
359      main = "CDF for Beta(3,2)",
360      xlab = "x", ylab = "F(x)")
361
362 F_04 = pbeta(0.4, a, b)
363 F_065 = pbeta(0.65, a, b)
364
365 points(c(0.4, 0.65), c(F_04, F_065), pch = 19, col = "maroon")
366 abline(v = 0.4, col = "maroon", lty = 2)
367 abline(v = 0.65, col = "maroon", lty = 2)
368 abline(h = F_04, col = "maroon", lty = 3)
369 abline(h = F_065, col = "maroon", lty = 3)
370
371 dev.off()
372
373
374 # 4f
375 png("/Users/chris/Maths/MA-223/Oblig/2b/oppg4f.png", width = 1400, height = 700)
376 par(mfrow = c(1, 2))
377
378 k = 0.5
379 lambda = 3
380
381 mu_T = lambda * gamma(1 + 1/k)
382 sigma_T = sqrt(lambda^2 * (gamma(1 + 2/k) - gamma(1 + 1/k)^2))
383 P_interval = pweibull(4, shape = k, scale = lambda) - pweibull(2, shape = k, scale = lambda)

```

```

384
385 mu_T
386 sigma_T
387 P_interval
388
389 # PDF
390 xVals = seq(0, 20, 0.01)
391 yVals = dweibull(xVals, shape = k, scale = lambda)
392
393 plot(xVals, yVals, type = "l", lwd = 2,
394      main = "PDF for Weibull(0.5,3)",
395      xlab = "t", ylab = "f(t)")
396
397 shade_x = seq(2, 4, 0.01)
398 shade_y = dweibull(shade_x, shape = k, scale = lambda)
399
400 polygon(c(2, shade_x, 4),
401        c(0, shade_y, 0),
402        col = "plum", border = NA)
403
404 lines(xVals, yVals, lwd = 2)
405 abline(v = 2, col = "maroon", lty = 2)
406 abline(v = 4, col = "maroon", lty = 2)
407
408 # CDF
409 yCumVals = pweibull(xVals, shape = k, scale = lambda)
410
411 plot(xVals, yCumVals, type = "l", lwd = 2,
412      main = "CDF for Weibull(0.5,3)",
413      xlab = "t", ylab = "F(t)")
414
415 F_2 = pweibull(2, shape = k, scale = lambda)
416 F_4 = pweibull(4, shape = k, scale = lambda)
417
418 points(c(2, 4), c(F_2, F_4), pch = 19, col = "maroon")
419 abline(v = 2, col = "maroon", lty = 2)
420 abline(v = 4, col = "maroon", lty = 2)
421 abline(h = F_2, col = "maroon", lty = 3)
422 abline(h = F_4, col = "maroon", lty = 3)
423
424 dev.off()
425
426 # Reset layout afterwards
427 par(mfrow = c(1, 1))
428
429
430
431 # =====

```

```

432 # Oppgave 5 - lagrer alle figurer i valgt mappe
433 # =====
434
435 setwd("/Users/chris/Maths/MA-223/Oblig/2b")
436 set.seed(223)
437
438 # Reset plotting layout
439 par(mfrow = c(1, 1))
440
441 # -----
442 # 5a) Standard normalfordeling
443 # -----
444 png("oppg5a_standard_normal.png", width = 1400, height = 900)
445
446 x = seq(-4, 4, length = 1000)
447 y = dnorm(x, mean = 0, sd = 1)
448
449 plot(x, y, type = "l", lwd = 2,
450      main = "Standard normalfordeling",
451      xlab = "x", ylab = "Tetthet")
452
453 dev.off()
454
455 # -----
456 # 5b) Ett tilfeldig trekk
457 # -----
458 tilfeldig_trekk = rnorm(1, mean = 0, sd = 1)
459 tilfeldig_trekk
460
461 # -----
462 # 5c) 50 trekk + teoretisk kurve
463 # -----
464 png("oppg5c_50_trekk.png", width = 1400, height = 900)
465
466 data_50 = rnorm(50, mean = 0, sd = 1)
467
468 hist(data_50, probability = TRUE, breaks = 10,
469      col = "purple",
470      main = "50 trekk fra N(0,1)",
471      xlab = "Verdi")
472
473 xVals = seq(-4, 4, 0.01)
474 yVals = dnorm(xVals, mean = 0, sd = 1)
475 lines(xVals, yVals, col = "maroon", lwd = 2)
476
477 dev.off()
478
479 # -----

```

```

480 # 5d) 500 trekk + teoretisk kurve
481 # -----
482 png("oppg5d_500_trekk.png", width = 1400, height = 900)
483
484 data_500 = rnorm(500, mean = 0, sd = 1)
485
486 hist(data_500, probability = TRUE, breaks = 20,
487       col = "purple",
488       main = "500 trekk fra N(0,1)",
489       xlab = "Verdi")
490
491 xVals = seq(-4, 4, 0.01)
492 yVals = dnorm(xVals, mean = 0, sd = 1)
493 lines(xVals, yVals, col = "maroon", lwd = 2)
494
495 dev.off()
496
497 # -----
498 # 5d) 50000 trekk + teoretisk kurve
499 # -----
500 png("oppg5d_50000_trekk.png", width = 1400, height = 900)
501
502 data_50000 = rnorm(50000, mean = 0, sd = 1)
503
504 hist(data_50000, probability = TRUE, breaks = 40,
505       col = "purple",
506       main = "50000 trekk fra N(0,1)",
507       xlab = "Verdi")
508
509 xVals = seq(-4, 4, 0.01)
510 yVals = dnorm(xVals, mean = 0, sd = 1)
511 lines(xVals, yVals, col = "maroon", lwd = 2)
512
513 dev.off()
514
515 # -----
516 # 5e) Oppgitt kode med N = 100
517 # -----
518 png("oppg5e_N100.png", width = 1400, height = 900)
519
520 N = 100
521 h = (rbinom(N, 100, 0.5) - 50) / 5
522 hist(h, probability = TRUE, breaks = seq(-8.05, 8.05, 0.1),
523       col = "purple",
524       main = "Standardisert binomisk fordeling, N = 100",
525       xlab = "h")
526 xVals = seq(-4, 4, 0.01)
527 yVals = 2 * dnorm(xVals, 0, 1)

```

```

528 lines(xVals, yVals, col = "maroon", type = "l", lwd = 2)
529
530 dev.off()
531
532 # -----
533 # 5f) N = 1000
534 # -----
535 png("oppg5f_N1000.png", width = 1400, height = 900)
536
537 N = 1000
538 h = (rbinom(N, 100, 0.5) - 50) / 5
539 hist(h, probability = TRUE, breaks = seq(-8.05, 8.05, 0.1),
540      col = "purple",
541      main = "Standardisert binomisk fordeling, N = 1000",
542      xlab = "h")
543 xVals = seq(-4, 4, 0.01)
544 yVals = 2 * dnorm(xVals, 0, 1)
545 lines(xVals, yVals, col = "maroon", type = "l", lwd = 2)
546
547 dev.off()
548
549 # -----
550 # 5f) N = 10000
551 # -----
552 png("oppg5f_N10000.png", width = 1400, height = 900)
553
554 N = 10000
555 h = (rbinom(N, 100, 0.5) - 50) / 5
556 hist(h, probability = TRUE, breaks = seq(-8.05, 8.05, 0.1),
557      col = "purple",
558      main = "Standardisert binomisk fordeling, N = 10000",
559      xlab = "h")
560 xVals = seq(-4, 4, 0.01)
561 yVals = 2 * dnorm(xVals, 0, 1)
562 lines(xVals, yVals, col = "maroon", type = "l", lwd = 2)
563
564 dev.off()
565
566 # -----
567 # 5f) N = 100000
568 # -----
569 png("oppg5f_N100000.png", width = 1400, height = 900)
570
571 N = 100000
572 h = (rbinom(N, 100, 0.5) - 50) / 5
573 hist(h, probability = TRUE, breaks = seq(-8.05, 8.05, 0.1),
574      col = "purple",
575      main = "Standardisert binomisk fordeling, N = 100000",

```



```

576     xlab = "h")
577 xVals = seq(-4, 4, 0.01)
578 yVals = 2 * dnorm(xVals, 0, 1)
579 lines(xVals, yVals, col = "maroon", type = "l", lwd = 2)
580
581 dev.off()
582
583 # Reset plotting layout afterwards
584 par(mfrow = c(1, 1))
585
586
587 #6a)
588 x = seq(0, 1, length = 1000)
589 y = dbeta(x, 3, 7)
590
591 plot(x, y, type = "l", lwd = 2,
592      main = expression(beta(3,7)),
593      xlab = "x", ylab = "Tetthet")
594
595 #b)
596 par(mfrow = c(3, 3))
597
598 verdier = c(1, 2, 5)
599
600 for (a in verdier) {
601   for (b in verdier) {
602     x = seq(0, 1, length = 1000)
603     y = dbeta(x, a, b)
604
605     plot(x, y, type = "l", lwd = 2,
606          main = paste("Beta(", a, ",", b, ")", sep = ""),
607          xlab = "x", ylab = "Tetthet")
608   }
609 }

```
