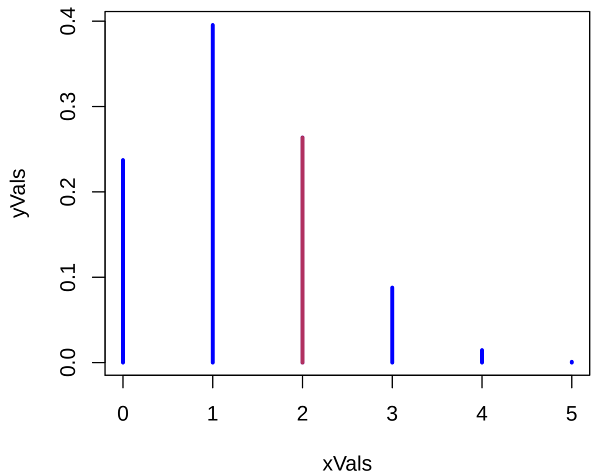
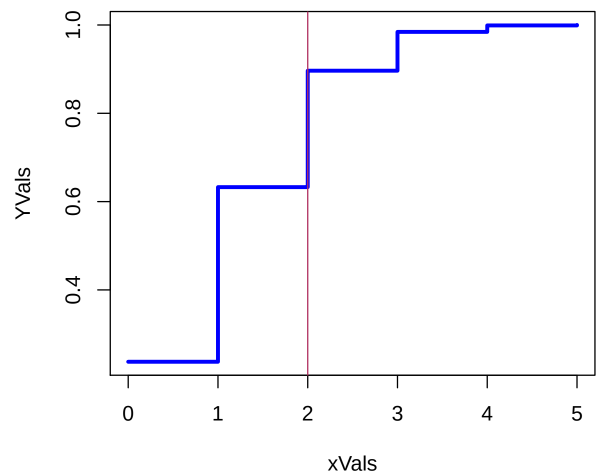
Oblig 2b-fasit

1. (25%) Kapittel 9, oppgave **40:**

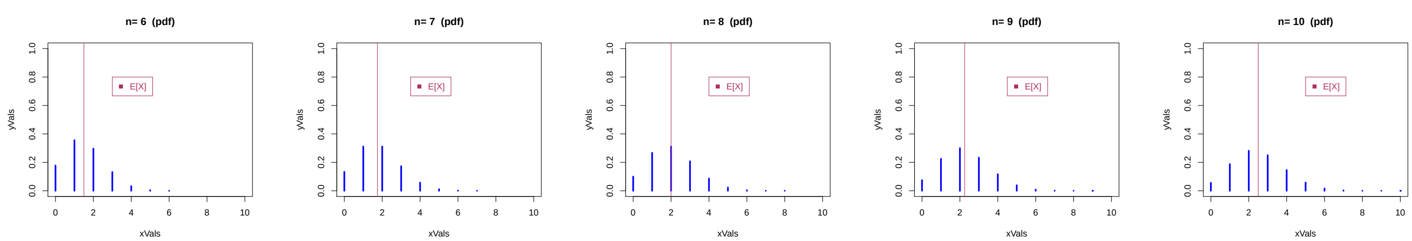
   (a) $p = 1/4 = 0.25$

   (b) $\text{bin}_{(5,0.25)}(2)$=dbinom(2,5,0.25)=0.2636719





   (c) $E[X] = np = n \cdot 0.25$. Skal $E[X] = 2$, må derfor $n = 8$.

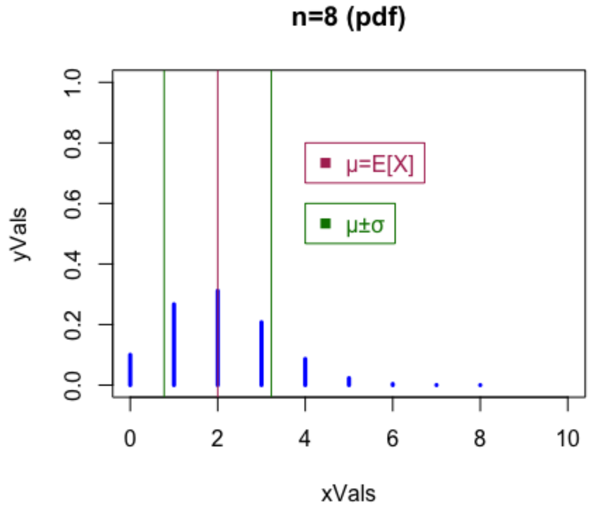


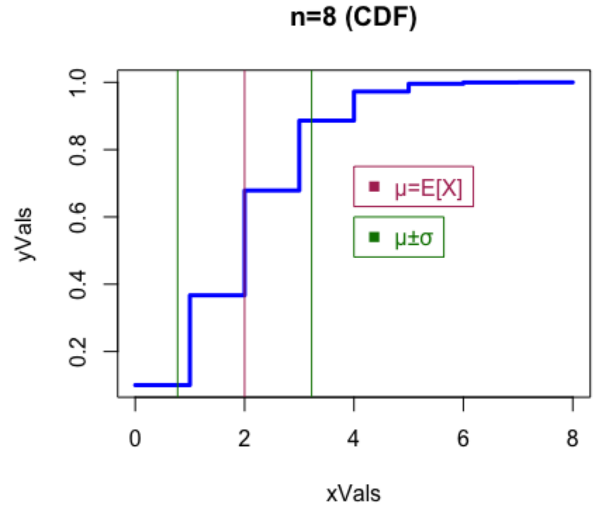
   R og **MatLab** har ikke eksplisitte kommandoer for $E[X]$, men i **Wolfram** finnes dette, og de som svarer i **Wolfram** må da skrive Mean[BinomialDistribution[n, 0.25]]. Mer presis må de skrive den fulle kommandoen

   Solve[Mean[BinomialDistribution[n, 0.25]] == 2, n]

   Standardavviket er $\sqrt{np(1-p)} = \sqrt{8 \cdot 0.25 \cdot (1 - 0.25)} = 1.22474$. I Wolfram:

   StandardDeviation[BinomialDistribution[8, 0.25]]

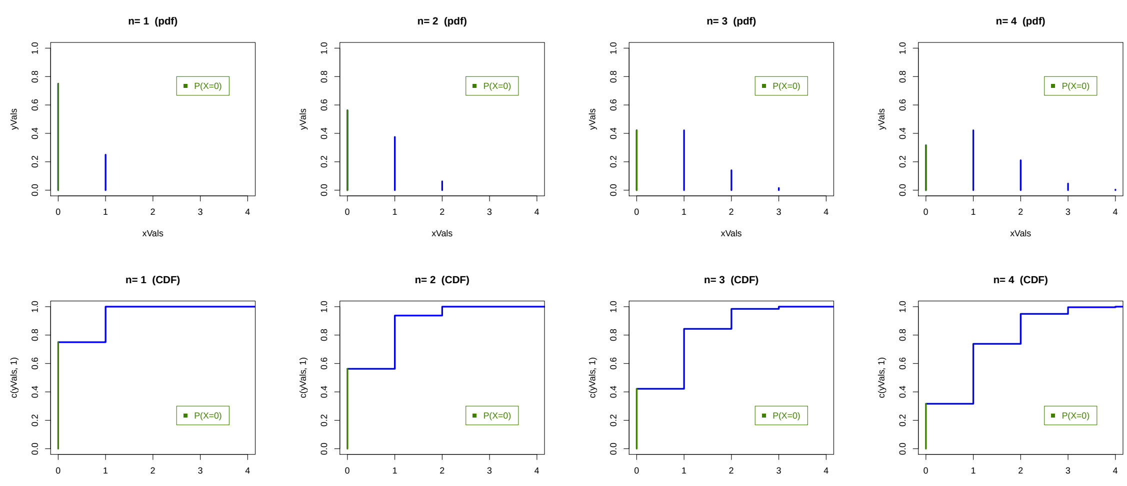




(d) Dette er n slik at $P(X > 0) \leq 0.5$, og derfor altså $P(X \leq 0) = P(X = 0) < 0.5$. Det spiller ingen rolle om du bruker pdf eller CDF, siden $\text{bin}_{(n,p)}(0) = P(X = 0) = P(X \leq 0) = \text{BIN}_{(n,p)}(0)$. Merk: Dette ville ikke virket for 1, siden $P(X = 1) < P(X \leq 1)$.

$$\begin{aligned}
P(X = 0) = \text{bin}_{(n,0.25)}(0) = (1 - 0.25)^n &< 0.5 \\
0.75^n &< 0.5 \\
\ln(0.75) \cdot n &< \ln(0.5) \\
n &> \frac{\ln(0.5)}{\ln(0.75)} = 2.40942
\end{aligned}$$

Så for 3 eller flere snikinger overstiger sjansen for på bli tatt 50%.

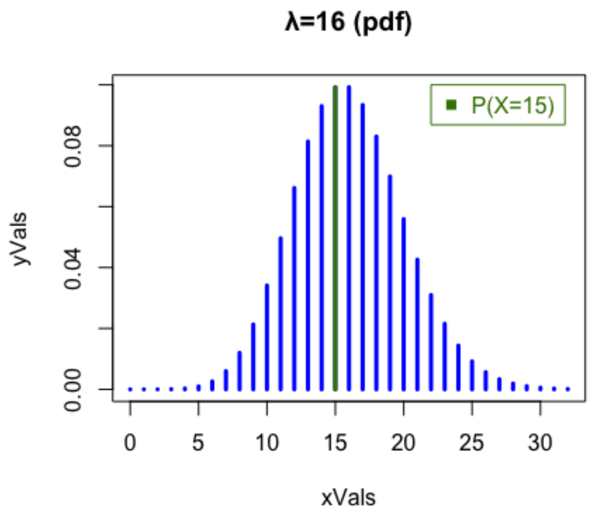


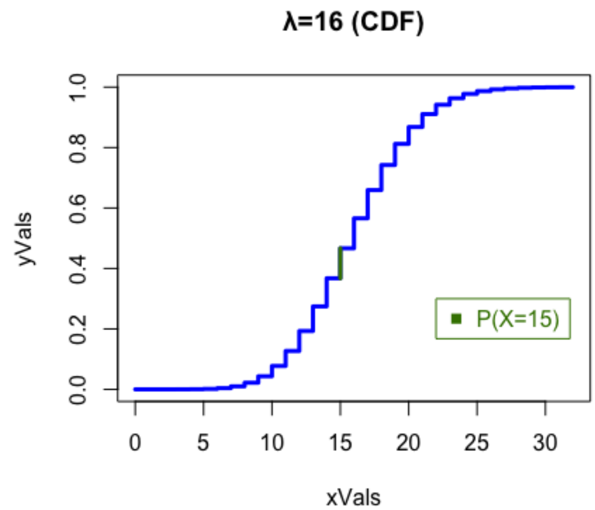
(e) Denne er hypergeometrisk, siden det er "trekking uten tilbakelegging".

(f) "Strictly speaking" er dette også binomisk, men som vi lærte på forelesningen, så blir Poisson en greiere modell når p er liten og n veldig stor, aller da når $p \downarrow 0$ og $n \to \infty$. Det gis 100% rett for binomisk regning for de som får til det, men også 100% rett for Poisson.

Her er $p = \frac{1}{1024} \approx 0$, og $n = 16384 \approx \infty$. Det gir en parameter $\lambda = np = \frac{1}{1024} \cdot 16384 = 16$. Da blir

$$P(X = 15) = \text{pois}_{16}(15) = \text{dpois(15,16)} = 0.09921753$$





2. (25%) Kapittel 9, oppgave **39:**

   (a) Her er Bernoulli enklest. Fanget (1 av 28) $p = \frac{4}{28}$. Du kan også nummerere deltagerne $1, \ldots, 28$, med våre fire som $1, \ldots, 4$, og regne ut $P(X \in \{1, \ldots, 4\})$.

   (b) Hypergeometrisk, siden det dessverre er "uten tilbakelegging". Har man først blitt tatt, så slippes man ikke løs.

   (c) Strictly, er dette også hypergeometrisk, men her er det formålstjenlig med binomisk. Binomisk kan regnes fra studentenes perspektiv, hvor hver student har $p = \frac{300}{10000} = 0.03$ for å bli tatt, og antall forsøk da er $n = 4$. Det er også mulig å regne binomisk fra piratjgernes perspektiv, med $p = \frac{4}{10000} = 0.0004$ for at den de har tatt IP-adressen til er en av studentene, og $n = 300$ forsøk. Som økonomisering av regning er dog siste variant mindre formålstjenlig, og også en dårligere tilnærming, om du ser på tallene under.

   $$\begin{aligned} P(X=2) &= \text{bin}_{(4,0.03)}(2) = \text{dbinom(2,4,0.03)} = 0.00508086 && \text{(studentperspektiv)} \\ P(X=2) &= \text{bin}_{(300,0.0004)}(2) = \text{dbinom(2,300,0.0004)} = 0.006369483 && \text{(piratjegerperspektiv)} \end{aligned}$$

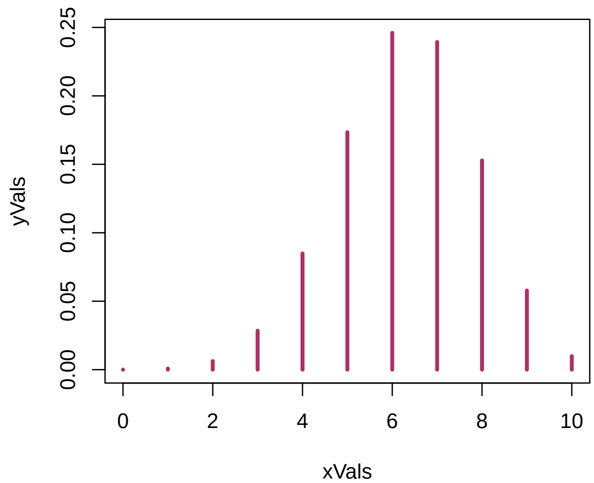
   For de som er nysgjerrige på eksakt utregning med hypergeometrisk:

   $$\begin{aligned} P(X=2) &= \text{hyp}_{(4,300,10000-300)}(2) = \text{dhyper(2,300,9700,4)} = 0.005066441 && \text{(studentperspektiv)} \\ P(X=2) &= \text{hyp}_{(300,4,10000-4)}(2) = \text{dhyper(2,4,9996,300)} = 0.005066441 && \text{(piratjegerperspektiv)} \end{aligned}$$

   (d) Her er meningen å bruke Poisson som tilnærming. Da er $\lambda = np = \frac{16384}{256} = 64$. Vi har fra Poisson-fordelingen at $E[X] = \lambda = 16$. Og $P(X = 15) = \text{pois}_{64}(15) = \text{dpois(15,64)} = 1.518 \cdot 10^{-13}$.

3. (10%) Du skal simulere en Bernoulli-prosess med $p = 0.63$. Valgfritt verktøy, men alt du gjør skal fremvises på PDF-en du leverer inn.

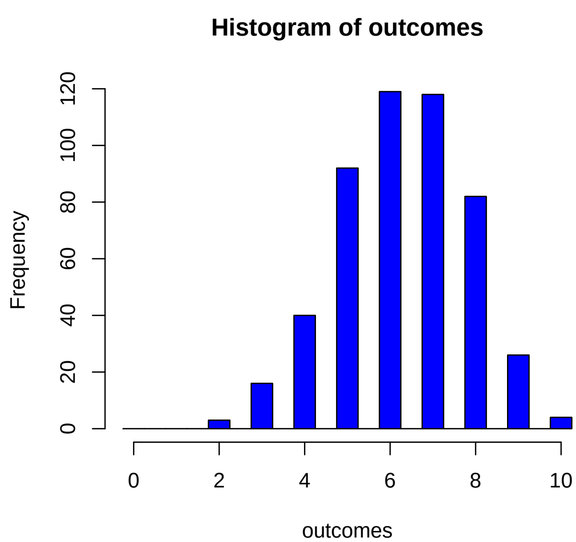
(a) Tegner funksjonsgrafen for $\text{bin}_{(10,p)}$: dbinom(x,10,0.63)

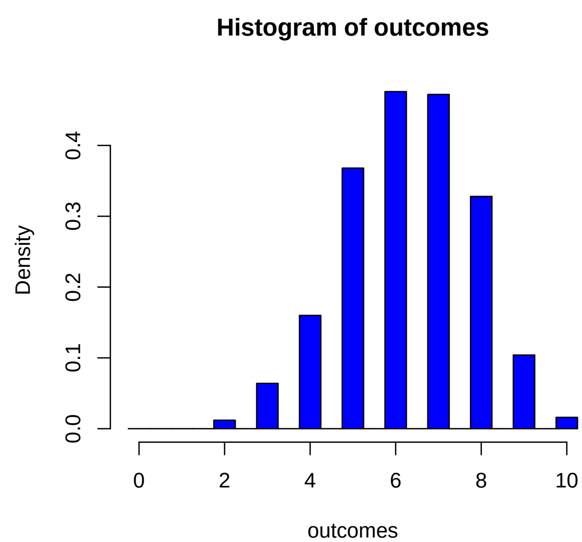


(b) En sekvens: rbinom(10,1,0.63). Mitt resultat: 0 1 1 1 0 1 1 0 1 1

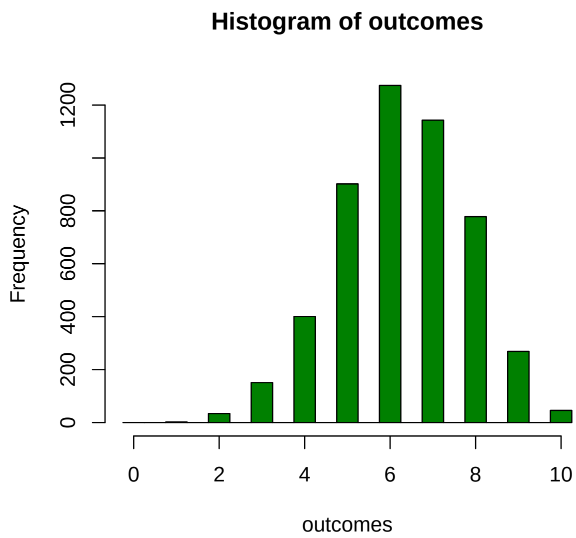
(c) Kode for 500 kjøringer:

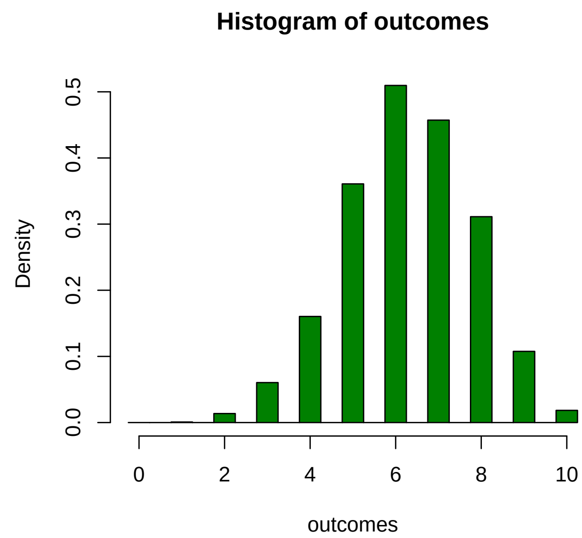
```
outcomes=c()
for (k in 1:500) {
sequence = rbinom(10,1,0.63)
count = sum(sequence)
outcomes=c(outcomes,count)
}
hist(outcomes, breaks=seq(-0.25,10.25,0.5))
hist(outcomes, breaks=seq(-0.25,10.25,0.5),probability=TRUE)
```



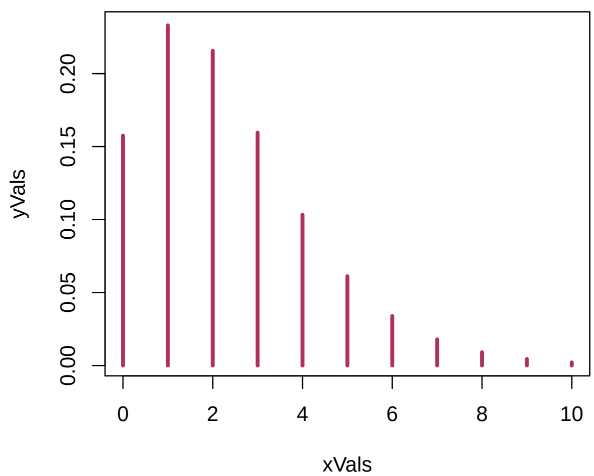


(d) Koden for 5000 kjøringer er som den over, men bytter ut for (k in 1:500) med for (k in 1:5000).





(e) Tegner funksjonsgrafen for $\text{nb}_{(4,p)}$: dnbinom(x,4,0.63)



(f) Generator som kjører til du har 4 positive utfall.

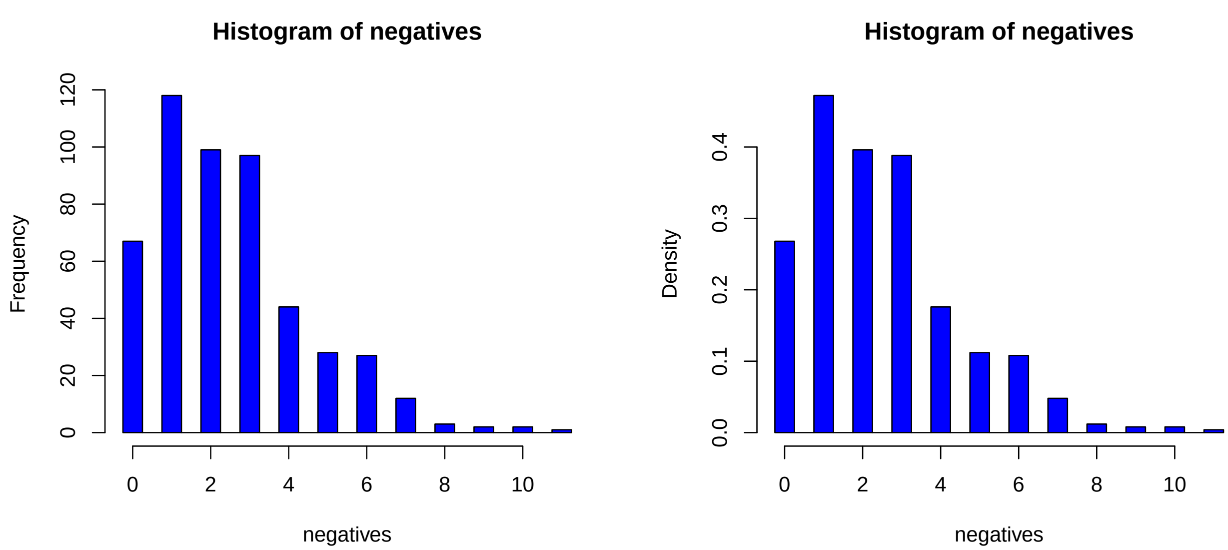
```
k=0
samples=c()
while (k<4) {
  saample = rbinom(1,1,0.63)
  if (saample==1) {k=k+1}
  samples=c(samples,saample)
}
length(samples) - 4
samples
```

Denne kan gjøres mer effektiv, men jeg bruker akkurat denne her fordi den også visert sekvensen. 10 utfall herfra:

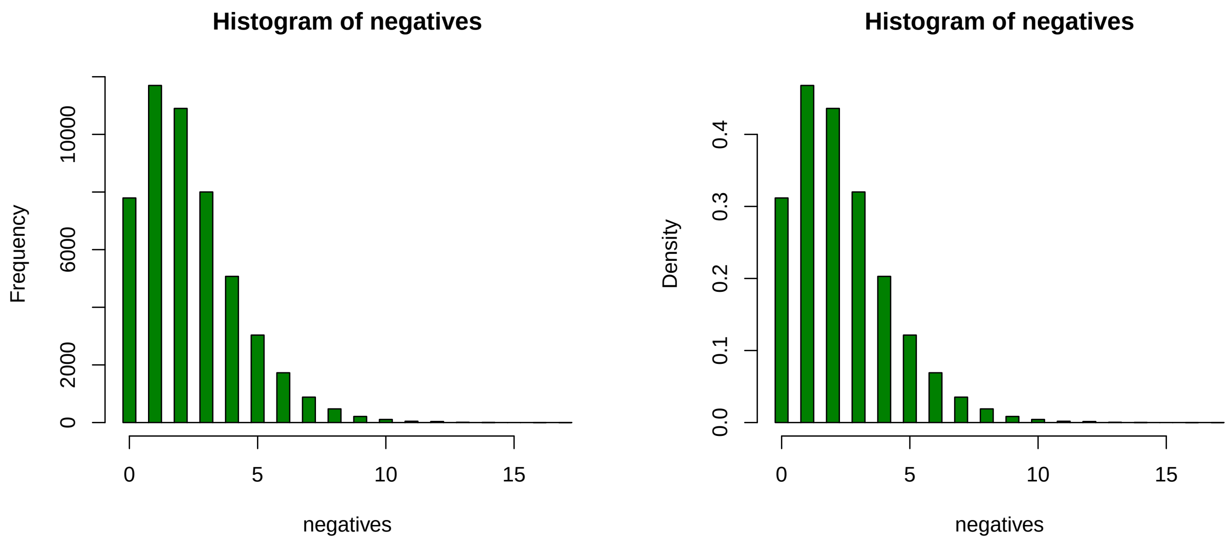
```
1 0 0 0 1 1 1 - 3
0 0 1 1 0 0 1 1 -  4
1 1 1 0 1 -  1
0 0 0 0 1 0 0 1 1 0 1 -  7
1 0 0 1 1 1 -  2
0 0 1 0 0 1 1 1 -  4
1 0 0 1 1 0 0 0 1 -  5
1 0 1 1 0 1 -  2
0 1 1 1 0 1 -  2
0 0 0 0 1 0 0 1 0 1 1 -  7
```

(g) Kjører sekvensgeneratoren over 500 ganger:

```
negatives=c()
for (k in 1:500) {
  k=0
  count=-4
  while (k<4) {
    saample = rbinom(1,1,0.63)
    if (saample==1) {k=k+1}
    count=count+1
  }
  negatives=c(negatives,count)
}
hist(outcomes, breaks=seq(-0.25,10.25,0.5), col="blue")
hist(outcomes,probability=TRUE, breaks=seq(-0.25,10.25,0.5), col="blue")
```



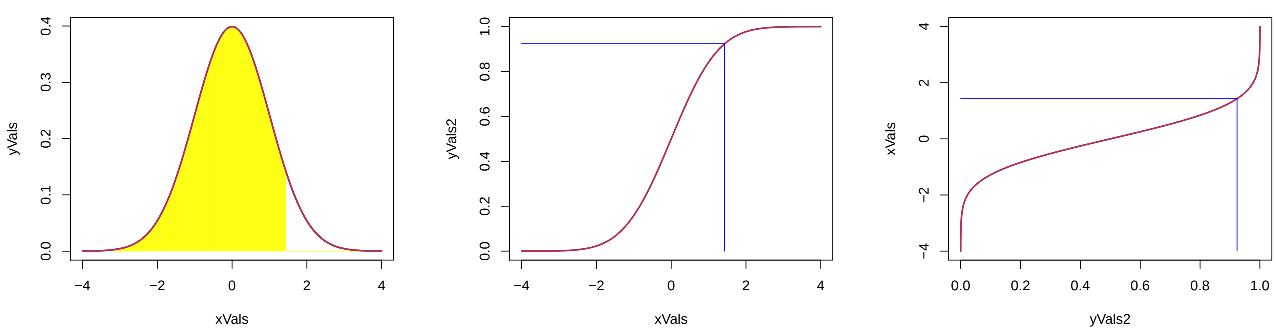
(h) Kjører sekvensgeneratoren over 50000 ganger:



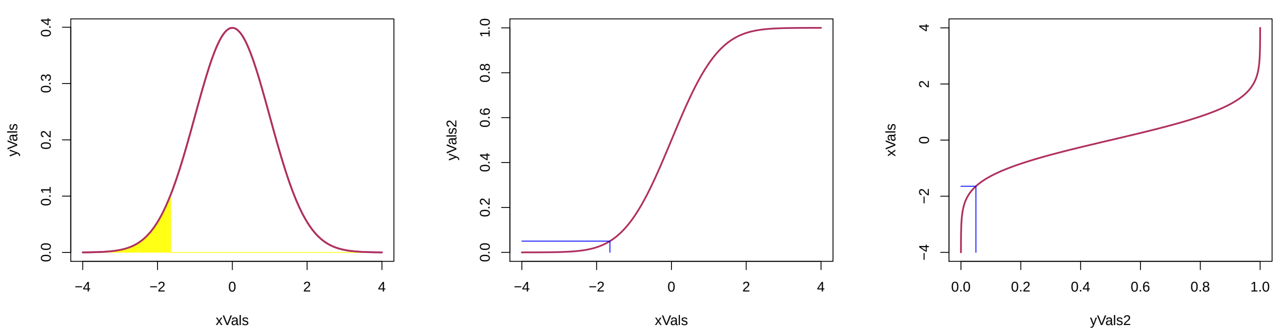
Kontinuerlige sannsynlighetsfordelinger

4. (15%) Kapittel 10, oppgave **2a og 3a:**

   2a) $P(X \leq 1.43) = \Phi_{(0,1)}(1.43)$ =pnorm(1.43,0,1)=0.9236415

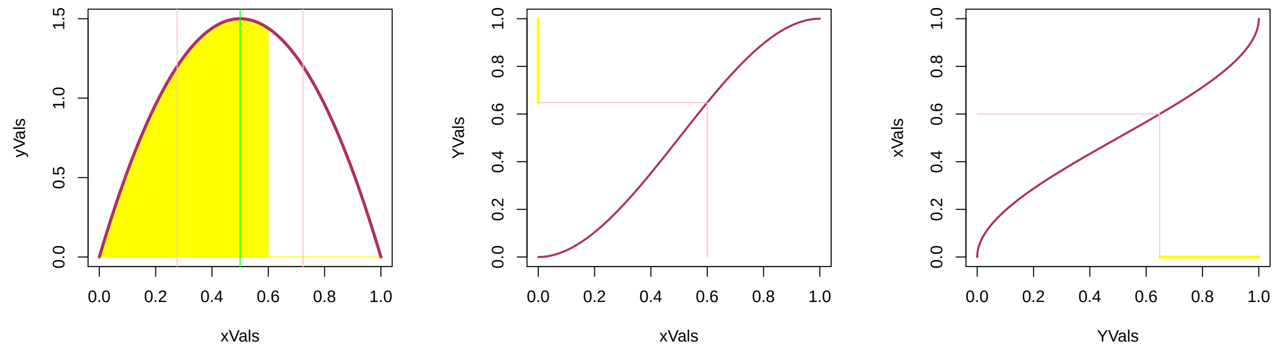


   3a) $z_{0.05} = \Phi_{(0,1)}(0.05)$ =qnorm(0.05,0,1)=-1.644854



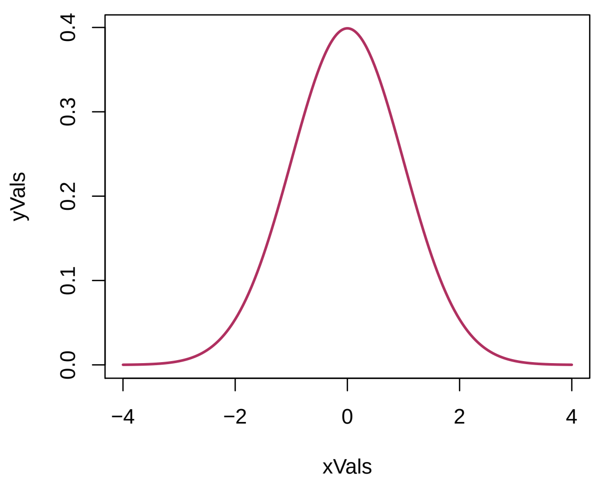
5. (10%) Kapittel 10, oppgave **35:**

- $\mu_X = \frac{2}{2+2} = 0.5$
- $\sigma_X = \sqrt{\frac{2 \cdot 2}{(2+2)^2 \cdot (2+2+1)}} = \sqrt{\frac{1}{20}} \approx 0.2236068$
- $P(X > 0.6) = 1 - P(X \leq 0.6) = 1 - \text{ß}_{(2,2)}(0.6) = 1 - pbeta(0.6, 2, 2) = 0.352$



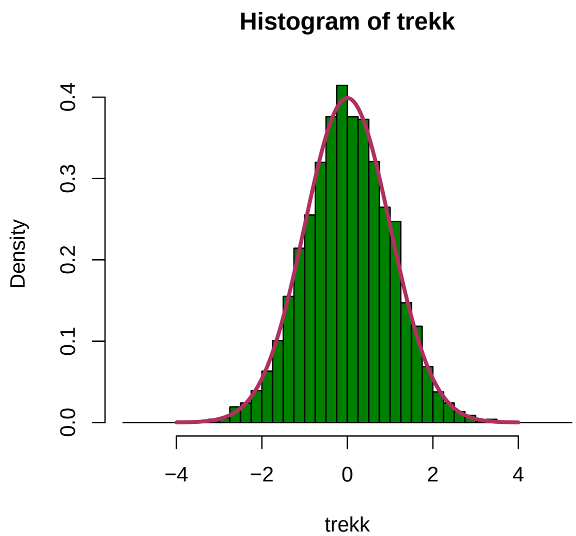
6. (10%) Normalfordeling

   (a) Tegn opp normalfordelingen $\phi_{(0,1)}$.



   (b) Ett tilfeldig trekk fra $\phi_{(0,1)}$: rnorm(1,0,1). Jeg fikk -1.493463

   (c) 5000 tilfeldige trekk fra $\phi_{(0,1)}$: rnorm(5000,0,1).



(d) Kjør følgende kode i **R**, og fortell med egne ord hva som skjer her, og hva koden gjør:

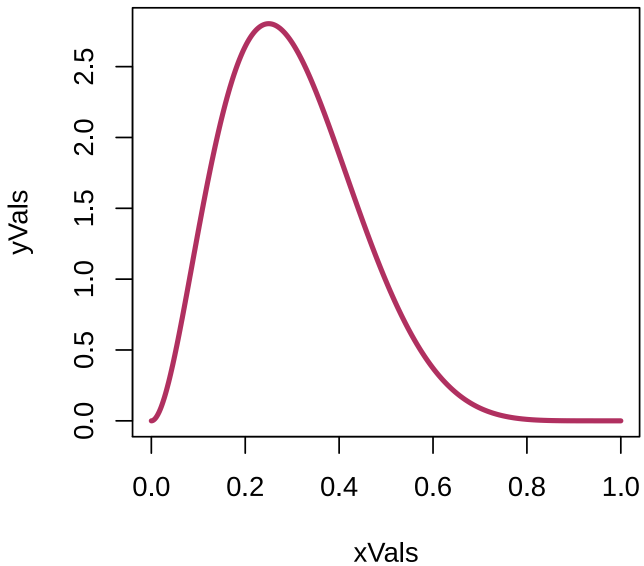
```
h=(rbinom(10000,100,0.5)-50)/5
hist(h,probability = TRUE,breaks=seq(-8.05,8.05,0.1))
xVals=seq(-4,4,0.01)
yVals=2*dnorm(xVals,0,1)
lines(xVals,yVals,col="maroon",type="l")
```

Koden kjører 10000 trekk fra en *binomisk* fordeling med $n = 100$ og $p = 0.5$, altså fra $X \sim \text{bin}_{(100,0.5)}$, og teller opp. Deretter blir antallet den forflyttet 50 enheter mot venstre så midten havner over 0, og så blir den skalert med en faktor på 5. Så blir denne tegnet opp. Etterpå tegner vi på normalfordelingen, og ser at den er nesten helt lik den transformerte binomiske fordelingen.

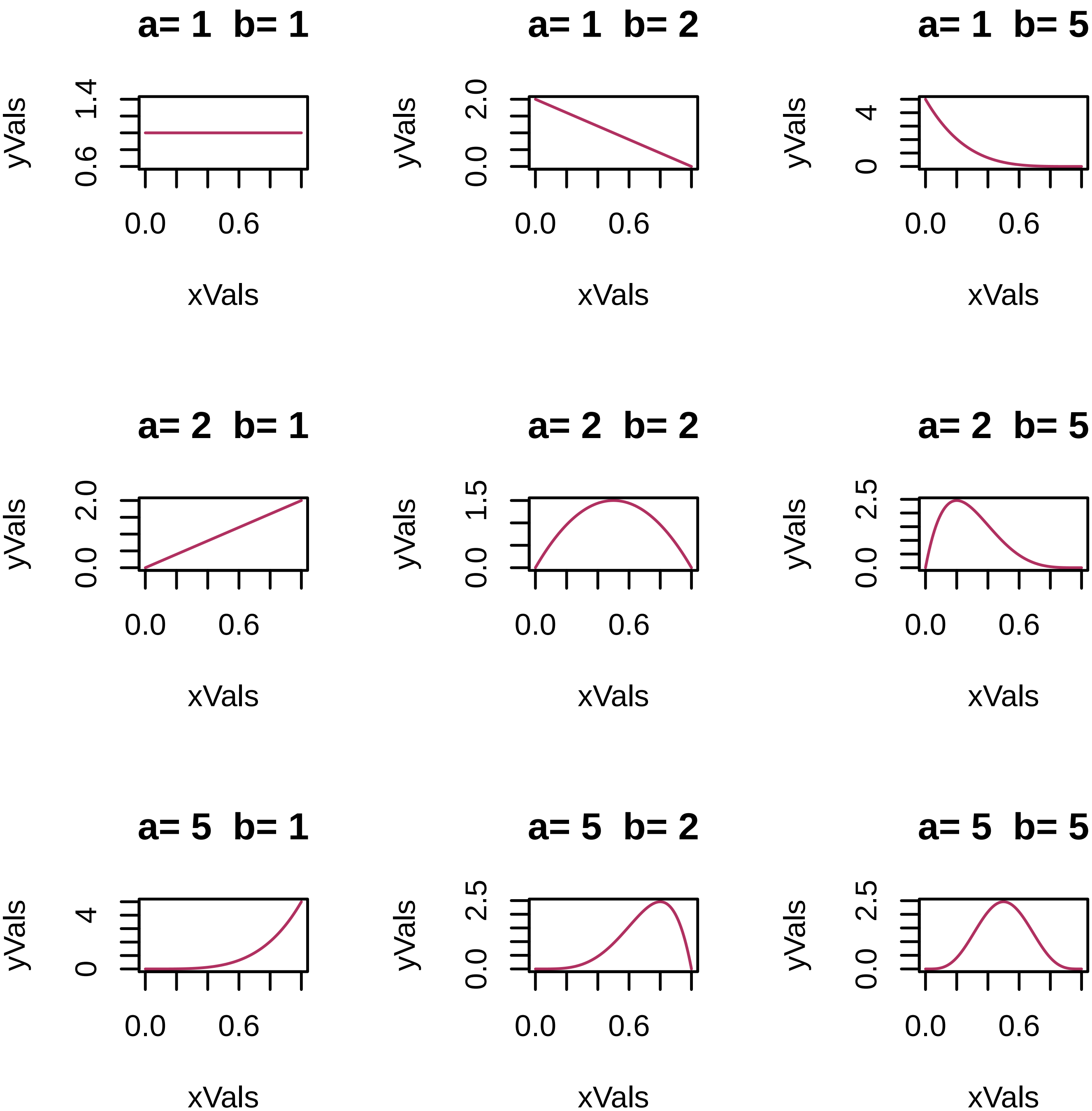
BONUS for å påpeke at $50 = E[X]$, altså forventet verdi, og enda mer bonus for å påpeke at $5 = \sigma_X$, standardavviket.

7. (5%) Betafordeling

(a) Tegn opp grafen til $\beta_{(3,7)}$



(b) Tegn opp grafene til $\beta_{(a,b)}$ for alle de 9 parene med verdier der $a, b = 1, 2, 5$.



(c) Effekten av de respektive parameterne a og b er at grafen og dermed tyngden av sannsynlighetsfordelingen forskyver seg mot høyre, mot 1, når a vokser, men mot venstre, mot 0, når b vokser. Samtidig blir den ”smalere” og mer konsentrert uansett hvilken av disse to paremeterne som vokser, så vokser begge to jevnt, så er effekten bare at grafen blir smalere og sannsynlighetsfordelingen blir konsentrert på et minde område.