Oblig 2 Statistikk

1 Kapittel 7

1.1 a) Kan du bruke $P(X <= a)$for å finne $P(X > a)$? Hvordan?

Ja man kan finne det ut ved å ta 1-P(X=<) fordi da får vi den resterende sannsynligheten. Vi vet sannsynligheten for mindre enn a så det som er igjen blir større enn a.

1.2 b) Hvorfor er $P(X < c) = P(X <= c)$ når X er kontinuerlig?

Fordi når X er kontinuerlig så kan aldri sannsynligheten være eksakt på et punkt. Dermed er sannsynligheten mindre enn c lik som når vi tar med akkurat punktet c.

1.3 c) Hvorfor kan vi ikke regne med $P(X < c) = P(X <= c)$ når X er diskret? (Hvorfor vil de for det meste være forskjellige?)

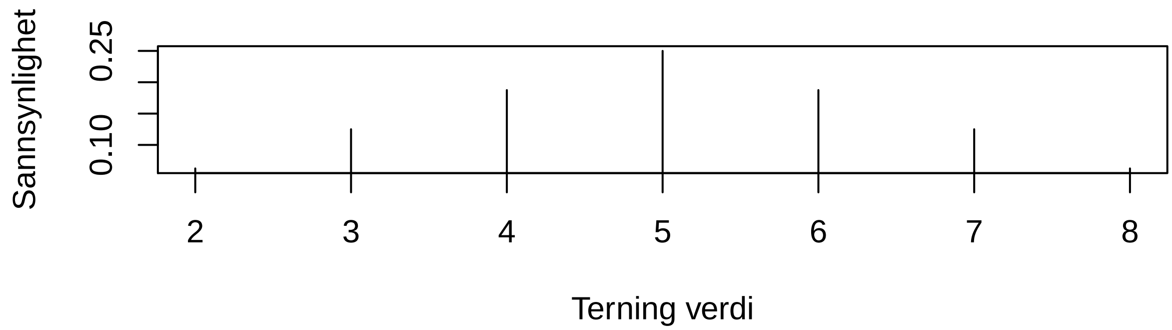
Fordi når X er diskret så kan vi ha en sannsynlighet i punktet c som ikke er lik den totale sannsynligheten mindre enn c.

1.4 d) For hånd: 1e (en: 2e) SE UNDER

1.5 e) For hånd: 2b (en: 3b) SE UNDER

1.6 f) Gjør i R: 1e (en: 2e)

```
values = c(2, 3, 4, 5, 6, 7, 8)
probs = c(1/16, 2/16, 3/16, 4/16, 3/16, 2/16, 1/16)

plot(seq(2, 8, 1), probs, type="h", xlab="Terning verdi", ylab="Sannsynlighet")
```




```
eX = 0
for (i in 1:length(values)) {
  eX = eX + values[i] * probs[i]
}

eX2 = 0
for (i in 1:length(values)) {
  eX2 = eX2 + values[i]^2 * values[i]
}

varX = eX2 - eX^2
sigma = sqrt(varX)
tauX = 1/(sigma^2)
```

E[X] = 5

var(X) = 2.5

sigma = 1.581139

tau = 0.4

1.7 g) Gjør i R: 2b (en: 3b)

```
values = c(1, 2, 3, 4)
probs = c(1/10, 2/10, 3/10, 4/10)

muX = 0
for (i in values) {
  muX = muX + values[i] * probs[i]
}

eX2 = 0
```

```r
for (i in values) {
  eX2 = eX2 + values[i]^2 * probs[i]
}

varX = eX2 - muX^2
sigma = sqrt(varX)
tau = 1 / (sigma^2)

pXeM = probs[1] + probs[4]
```

µx = 3

var(X) = 1

sigma = 1

tau = 1

P(X elementin M) = 0.5

2 Kapittel 9

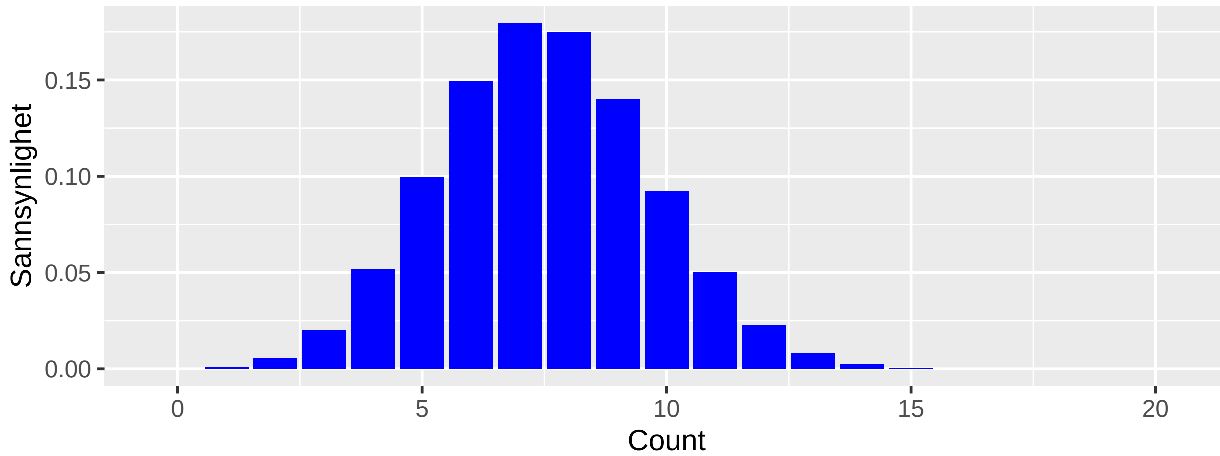
2.1 a) X bin(20,0.375). Lag tabell over sannsynligheter for x= 0,....,20, og plott både pdf og CDF for denne sannsynlighetsfordelingen.

2.1.1 Her er tabell og plot for PDF:

```
library(ggplot2)

x = c(0:20)

y = dbinom(x, 20, 0.375)

df = data.frame(x, y)

df
```

```
##     x            y
## 1   0 8.271806e-05
## 2   1 9.926167e-04
## 3   2 5.657915e-03
## 4   3 2.036850e-02
## 5   4 5.193966e-02
## 6   5 9.972415e-02
## 7   6 1.495862e-01
## 8   7 1.795035e-01
## 9   8 1.750159e-01
## 10  9 1.400127e-01
## 11 10 9.240839e-02
## 12 11 5.040458e-02
## 13 12 2.268206e-02
## 14 13 8.374914e-03
## 15 14 2.512474e-03
## 16 15 6.029938e-04
## 17 16 1.130613e-04
## 18 17 1.596160e-05
## 19 18 1.596160e-06
## 20 19 1.008101e-07
## 21 20 3.024303e-09
```

```
ggplot(data=df, aes(x=x, y=y)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```

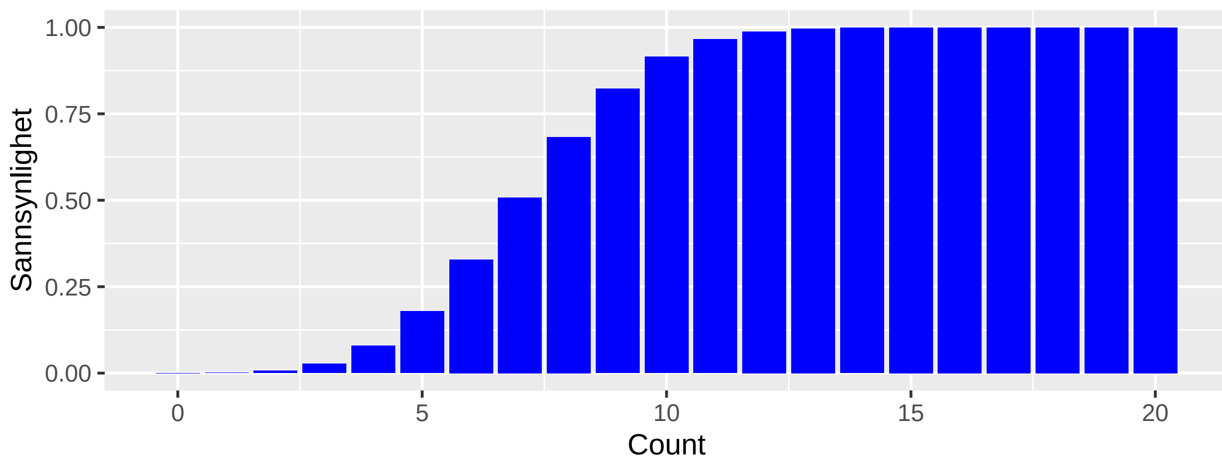


2.1.2 Her er tabell og plot for CDF:

```
y_cdf = pbinom(x, 20, 0.375)

df_cdf = data.frame(x, y_cdf)

df_cdf
```

```
##     x        y_cdf
## 1   0 8.271806e-05
## 2   1 1.075335e-03
## 3   2 6.733250e-03
## 4   3 2.710175e-02
## 5   4 7.904141e-02
## 6   5 1.787656e-01
## 7   6 3.283518e-01
## 8   7 5.078553e-01
## 9   8 6.828712e-01
## 10  9 8.228839e-01
## 11 10 9.152923e-01
## 12 11 9.656968e-01
## 13 12 9.883789e-01
## 14 13 9.967538e-01
## 15 14 9.992663e-01
## 16 15 9.998693e-01
## 17 16 9.999823e-01
## 18 17 9.999983e-01
## 19 18 9.999999e-01
## 20 19 1.000000e+00
## 21 20 1.000000e+00
```

```
ggplot(data=df_cdf, aes(x=x, y=y_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



2.1.3 E[X]

Vi summerer opp hver count ganget med sannsynligheten for å finne E[X]

```
sum(x*y)
```

E[X] = 7.5

2.1.4 Var(X)

For å finne var(X) kjører vi bare følgende R-kode

```
ex = (sum(x^2*y)-(sum(x*y))^2)
```

Var(X) = 4.6875

2.1.5 $P(2 < X < 7)$

For å finne ut sannsynligheten for X mellom 2 og 7 skriver vi følgende R-kode

```
lessthan7 = pbinom(6, length(x), 0.375)
lessthan2 = pbinom(1, length(x), 0.375)

print(lessthan7-lessthan2)
```

```
## [1] 0.2715539
```

2.2 b) X nb(3,0.3). Lag tabell over sannsynligheter for x = 0,....,20, og plott både pdf og CDF for denne sannsynlighetsfordelingen

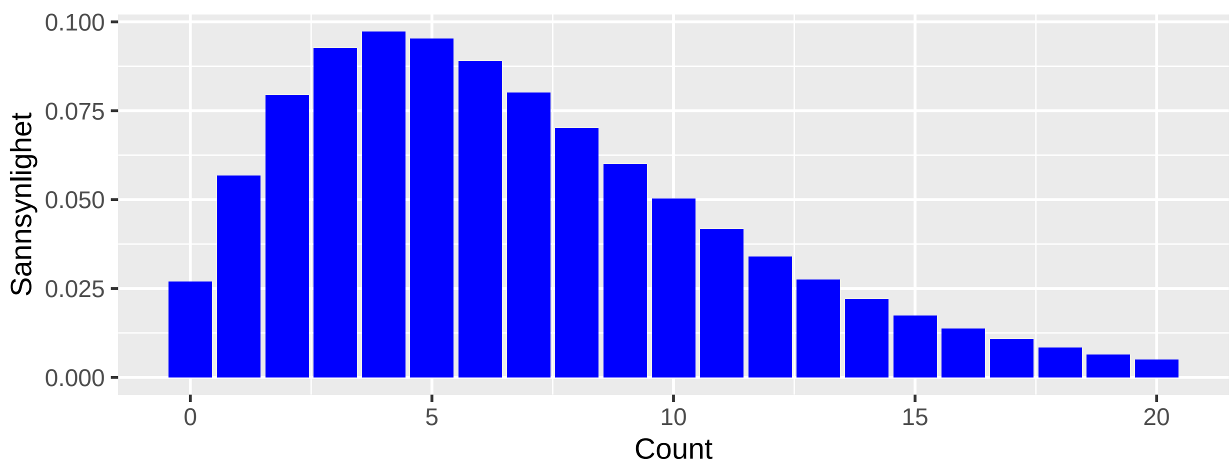
2.2.1 Her er tabell og plot for PDF:

```
x = c(0:20)

negativb_y = dnbinom(x, 3, 0.3)

nbdf_pdf = data.frame(x, y)

nbdf_pdf
```

```
##      x            y
```

```
## 1    0 8.271806e-05
## 2    1 9.926167e-04
## 3    2 5.657915e-03
## 4    3 2.036850e-02
## 5    4 5.193966e-02
## 6    5 9.972415e-02
## 7    6 1.495862e-01
## 8    7 1.795035e-01
## 9    8 1.750159e-01
## 10   9 1.400127e-01
## 11  10 9.240839e-02
## 12  11 5.040458e-02
## 13  12 2.268206e-02
## 14  13 8.374914e-03
## 15  14 2.512474e-03
## 16  15 6.029938e-04
## 17  16 1.130613e-04
## 18  17 1.596160e-05
## 19  18 1.596160e-06
## 20  19 1.008101e-07
## 21  20 3.024303e-09
```

```r
ggplot(data=nbdf_pdf, aes(x=x, y=negativb_y)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



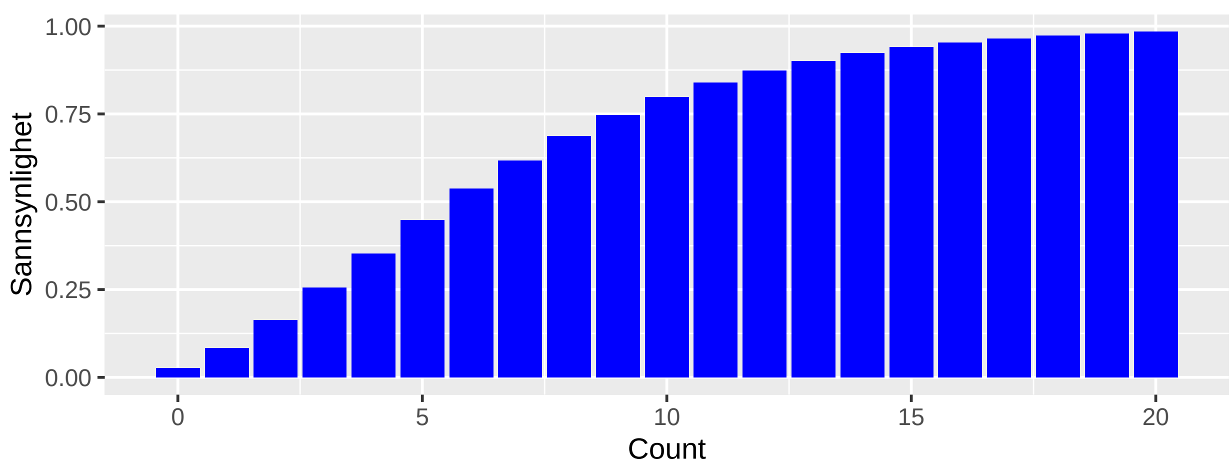
2.2.2 Her er tabell og plot for CDF:

```r
nby_cdf = pnbinom(x, 3, 0.3)

nbdf_cdf = data.frame(x, nby_cdf)

nbdf_cdf
```

```
##     x   nby_cdf
## 1   0 0.0270000
## 2   1 0.0837000
## 3   2 0.1630800
## 4   3 0.2556900
```

```
## 5   4 0.3529305
## 6   5 0.4482262
## 7   6 0.5371688
## 8   7 0.6172172
## 9   8 0.6872595
## 10  9 0.7471847
## 11 10 0.7975217
## 12 11 0.8391642
## 13 12 0.8731723
## 14 13 0.9006403
## 15 14 0.9226147
## 16 15 0.9400478
## 17 16 0.9537763
## 18 17 0.9645169
## 19 18 0.9728706
## 20 19 0.9793338
## 21 20 0.9843104
```

```
ggplot(data=nbdf_cdf, aes(x=x, y=nby_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



2.2.3 E[X]

Vi summerer opp hver count ganget med sannsynligheten for den counten for å finne E[X]

```
sum(x*negativb_y)
```

E[X] = 6.6230933

2.2.4 Var(X)

For å finne var(X) kjører vi bare følgende R-kode

```
e = (sum(x^2*negativb_y)-(sum(x*negativb_y))^2)
```

Var(X) = 19.2316685

2.2.5 $P(2 < X < 7)$

For å finne ut sannsynligheten for X mellom 2 og 7 skriver vi følgende R-kode

```r
lessthan7 = pnbinom(6, 3, 0.3)
lessthan2 = pnbinom(1, 3, 0.3)

print(lessthan7-lessthan2)
```

```
## [1] 0.4534688
```

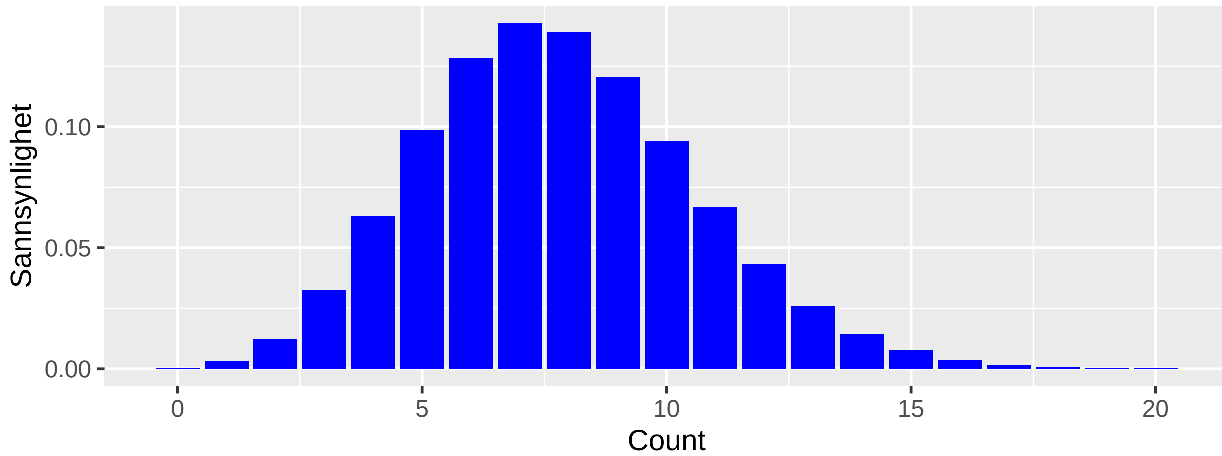
2.3 X pois7.8. Lag tabell over sannsynligheter forx= 0,....,20, og plott både pdf og CDF for denne sannsynlighetsfordelingen

2.3.1 Her er tabell og plot for PDF:

```r
py_pdf = dpois(x, 7.8)

p_df_pdf = data.frame(x, py_pdf)

p_df_pdf
```

```
##     x       py_pdf
## 1   0 0.0004097350
## 2   1 0.0031959328
## 3   2 0.0124641381
## 4   3 0.0324067590
## 5   4 0.0631931800
## 6   5 0.0985813607
## 7   6 0.1281557690
## 8   7 0.1428021426
## 9   8 0.1392320890
## 10  9 0.1206678105
## 11 10 0.0941208922
## 12 11 0.0667402690
## 13 12 0.0433811748
## 14 13 0.0260287049
## 15 14 0.0145017070
## 16 15 0.0075408877
## 17 16 0.0036761827
## 18 17 0.0016867191
## 19 18 0.0007309116
## 20 19 0.0003000585
## 21 20 0.0001170228
```

```r
ggplot(data=p_df_pdf, aes(x=x, y=py_pdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```

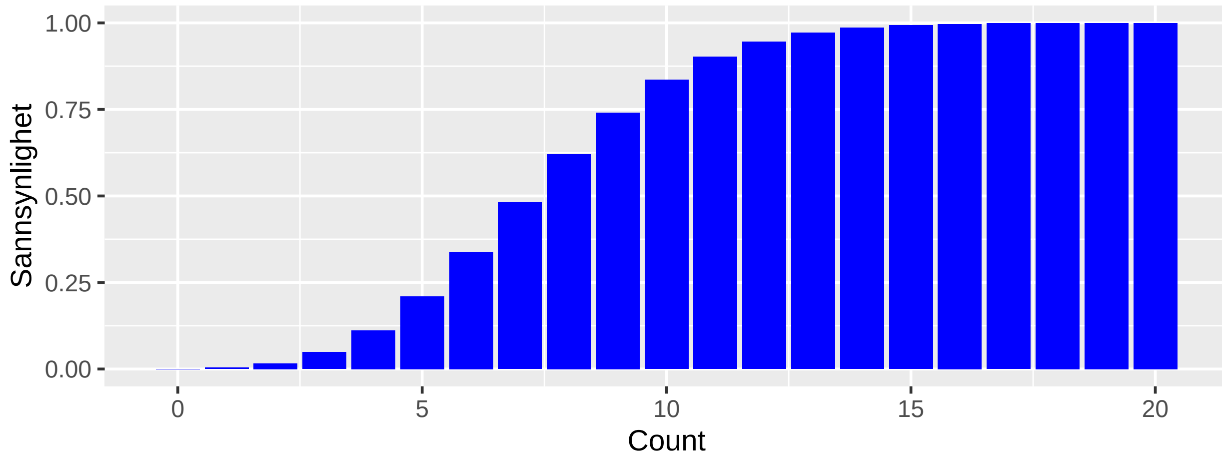


2.3.2 Her er tabell og plot for CDF:

```
pby_cdf = ppois(x, 7.8)

pdf_cdf = data.frame(x, nby_cdf)

pdf_cdf
```

```
##     x   nby_cdf
## 1   0 0.0270000
## 2   1 0.0837000
## 3   2 0.1630800
## 4   3 0.2556900
## 5   4 0.3529305
## 6   5 0.4482262
## 7   6 0.5371688
## 8   7 0.6172172
## 9   8 0.6872595
## 10  9 0.7471847
## 11 10 0.7975217
## 12 11 0.8391642
## 13 12 0.8731723
## 14 13 0.9006403
## 15 14 0.9226147
## 16 15 0.9400478
## 17 16 0.9537763
## 18 17 0.9645169
## 19 18 0.9728706
## 20 19 0.9793338
## 21 20 0.9843104
```

```
ggplot(data=pdf_cdf, aes(x=x, y=pby_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



2.3.3 E[X]

Vi summerer opp hver count ganget med sannsynligheten for å finne E[X]

```
sum(x*py_pdf)
```

E[X] = 7.7985681

2.3.4 Var(X)

For å finne var(X) kjører vi bare følgende R-kode

```
e = (sum(x^2*py_pdf)-(sum(x*py_pdf))^2)
```

Var(X) = 7.7914793

2.3.5 $P(2 < X < 7)$

For å finne ut sannsynligheten for X mellom 2 og 7 skriver vi følgende R-kode

```
lessthan7 = ppois(6, 7.8)
lessthan2 = ppois(1, 7.8)

print(lessthan7-lessthan2)
```

```
## [1] 0.3348012
```

2.4 Hypergeometrisk:X hyp(20,30,80). Lag tabell over sannsynligheter forx=0,....,20, og plott både pdf og CDF for denne sannsynlighetsfordelingen.

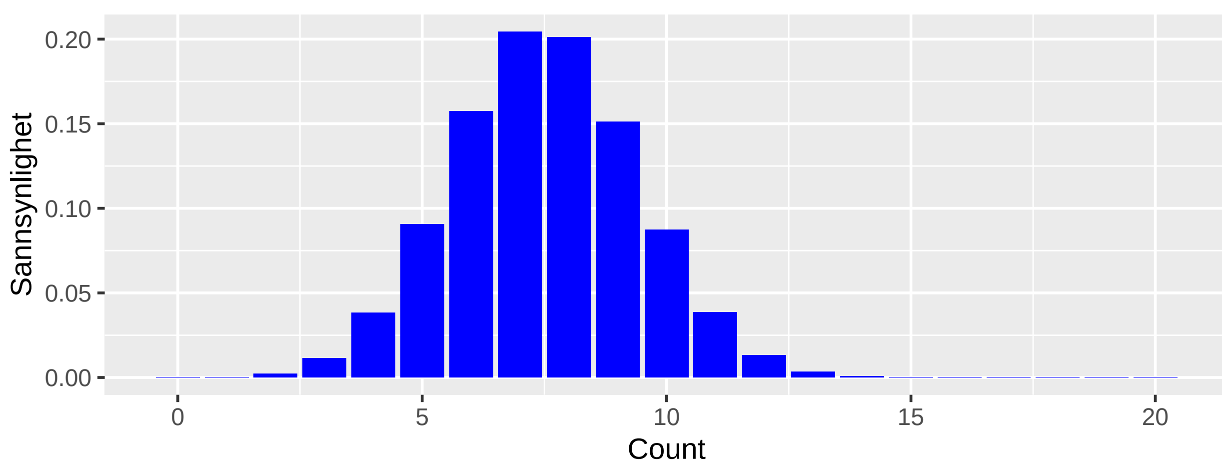
2.4.1 Her er tabell og plot for PDF:

```
hypery_pdf = dhyper(x, 30, 50, 20)

hyperydf_pdf = data.frame(x, hypery_pdf)

hyperydf_pdf
```

```
##     x   hypery_pdf
## 1   0 1.333098e-05
```

```
## 2    1 2.580189e-04
## 3    2 2.221381e-03
## 4    3 1.130885e-02
## 5    4 3.816737e-02
## 6    5 9.072929e-02
## 7    6 1.575161e-01
## 8    7 2.043452e-01
## 9    8 2.009843e-01
## 10   9 1.511677e-01
## 11  10 8.729934e-02
## 12  11 3.871367e-02
## 13  12 1.313500e-02
## 14  13 3.383613e-03
## 15  14 6.536525e-04
## 16  15 9.296391e-05
## 17  16 9.473224e-06
## 18  17 6.639556e-07
## 19  18 2.997022e-08
## 20  19 7.725943e-10
## 21  20 8.498537e-12
```

```
ggplot(data=hyperydf_pdf, aes(x=x, y=hypery_pdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



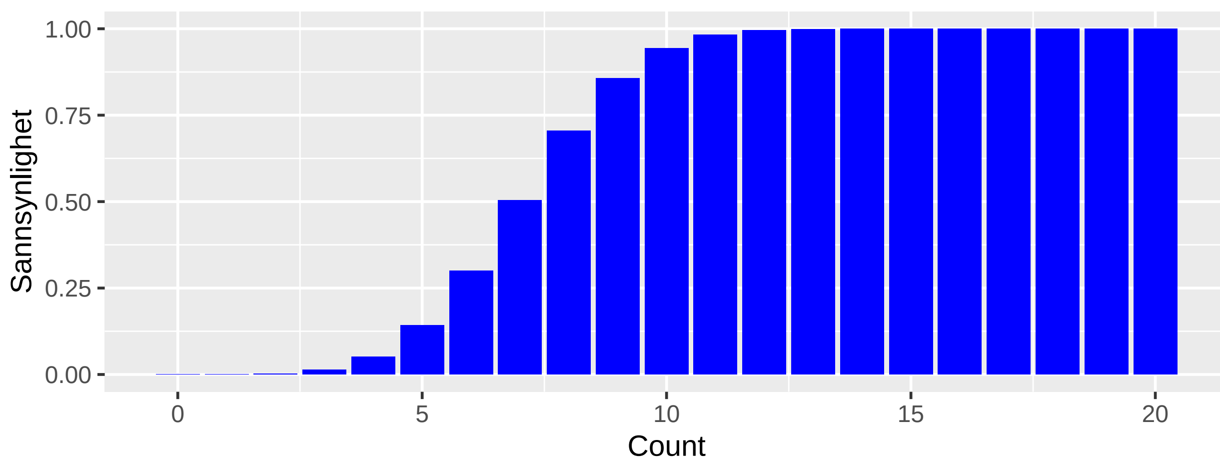
2.4.2 Her er tabell og plot for CDF:

```
hypery_cdf = phyper(x, 30,50,20)

hyperydf_cdf = data.frame(x, hypery_cdf)

hyperydf_cdf
```

```
##      x   hypery_cdf
## 1    0 1.333098e-05
## 2    1 2.713499e-04
## 3    2 2.492731e-03
## 4    3 1.380158e-02
## 5    4 5.196895e-02
```

```
## 6   5 1.426982e-01
## 7   6 3.002144e-01
## 8   7 5.045596e-01
## 9   8 7.055439e-01
## 10  9 8.567116e-01
## 11 10 9.440109e-01
## 12 11 9.827246e-01
## 13 12 9.958596e-01
## 14 13 9.992432e-01
## 15 14 9.998969e-01
## 16 15 9.999898e-01
## 17 16 9.999993e-01
## 18 17 1.000000e+00
## 19 18 1.000000e+00
## 20 19 1.000000e+00
## 21 20 1.000000e+00
```

```
ggplot(data=hyperydf_cdf, aes(x=x, y=hypery_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



2.4.3 E[X]

Vi summerer opp hver count ganget med sannsynligheten for å finne E[X]

```
sum(x*hypery_pdf)
```

E[X] = 7.5

2.4.4 Var(X)

For å finne var(X) kjører vi bare følgende R-kode

```
e = (sum(x^2*hypery_pdf)-(sum(x*hypery_pdf))^2)
```

Var(X) = 3.5601266

2.4.5 P(2 < X < 7)

For å finne ut sannsynligheten for X mellom 2 og 7 skriver vi følgende R-kode

```r
lessthan7 = phyper(6,30,50,20)
lessthan2 = phyper(1,30,50,20)

print(lessthan7-lessthan2)
```

```
## [1] 0.299943
```

2.5 X Betab(3a,5a,20) for a= 2. Lag tabell over sannsynligheter for x= 0,....,20, og regn deretter ut E[X],Var(X), og P(2 < X < 7). Plott både pdf og CDF for disse sannsynlighetsfordelingene, med a= 1,a= 2,a= 4, og a= 10, og sammenlign medtilsvarende plott for bin(20,0.375)

2.5.1 Her er tabell for PDF:

```r
library(extraDistr)
```

```
## Warning: package 'extraDistr' was built under R version 3.6.3
```

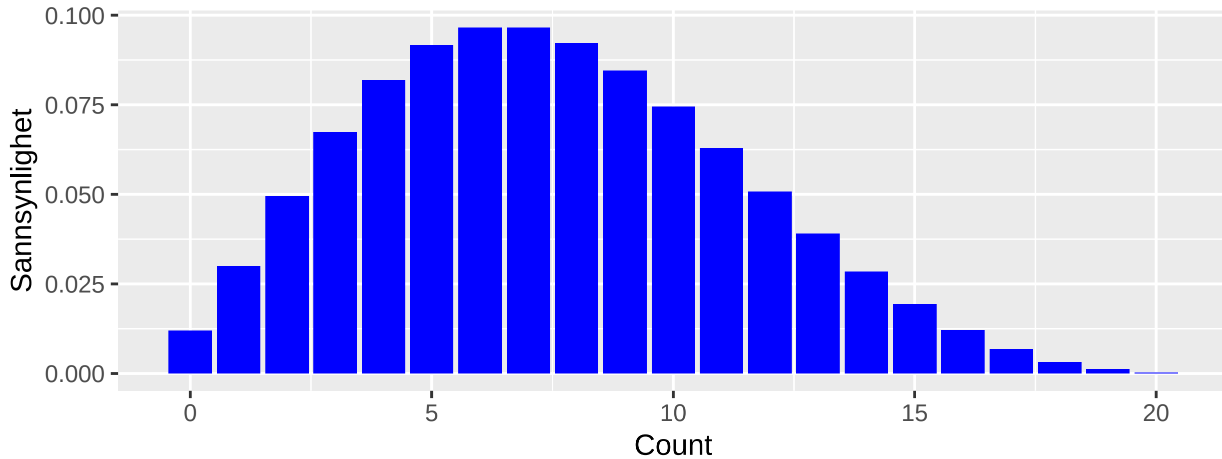
```r
a = 1
betabinomy1_pdf = dbbinom(x, 20, 3*a, 5*a)

a2 = 2
betabinomy2_pdf = dbbinom(x, 20, 3*a2, 5*a2)

a3 = 4
betabinomy3_pdf = dbbinom(x, 20, 3*a3, 5*a3)

a4 = 10
betabinomy4_pdf = dbbinom(x, 20, 3*a4, 5*a4)

betabinomy1_df_pdf = data.frame(x, betabinomy1_pdf)
betabinomy2_df_pdf = data.frame(x, betabinomy2_pdf)
betabinomy3_df_pdf = data.frame(x, betabinomy3_pdf)
betabinomy4_df_pdf = data.frame(x, betabinomy4_pdf)

betabinomy1_df_pdf
```

```
##     x betabinomy1_pdf
## 1   0    0.0119658120
## 2   1    0.0299145299
## 3   2    0.0494240059
## 4   3    0.0673963717
## 5   4    0.0818384514
## 6   5    0.0916590656
## 7   6    0.0964832269
## 8   7    0.0964832269
## 9   8    0.0922266140
## 10  9    0.0845410628
## 11 10    0.0743961353
## 12 11    0.0628019324
## 13 12    0.0507246377
## 14 13    0.0390189521
## 15 14    0.0283774197
## 16 15    0.0192966454
```

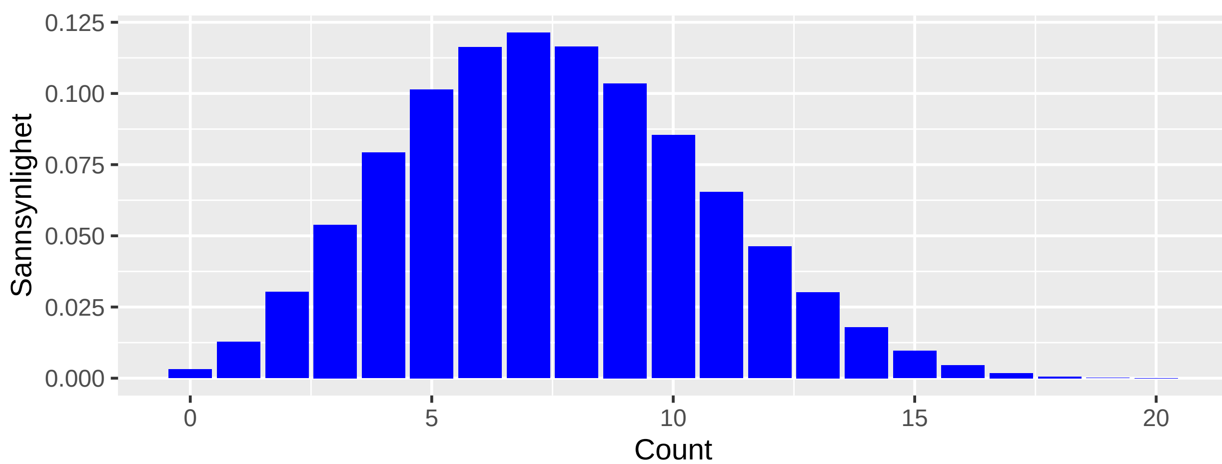
```
## 17 16     0.0120604034
## 18 17     0.0067396372
## 19 18     0.0032093510
## 20 19     0.0011823925
## 21 20     0.0002601263
```

2.5.2 Her er plot for PDF med a=1,a=2,a=4 og a=10

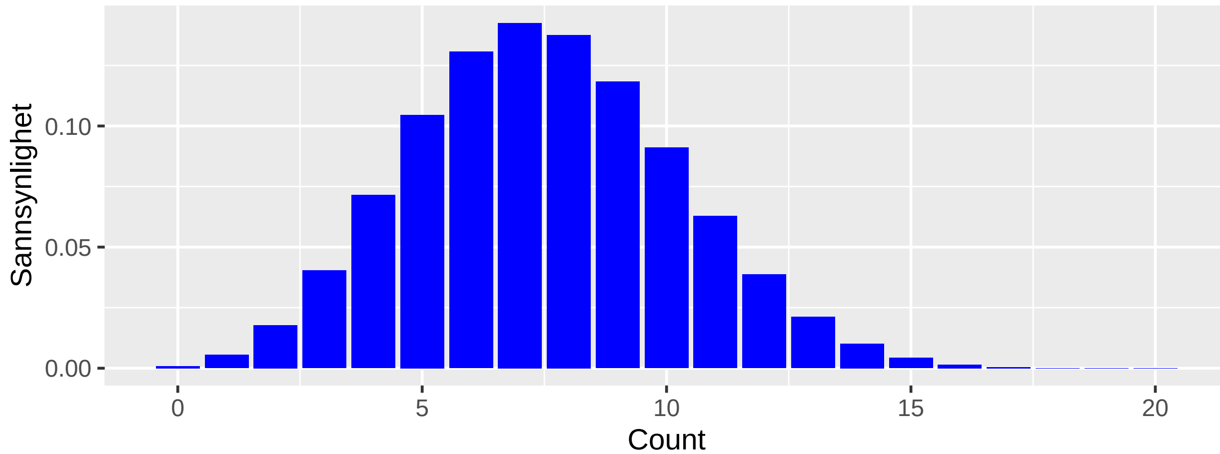
```r
ggplot(data=betabinomy1_df_pdf, aes(x=x, y=betabinomy1_pdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



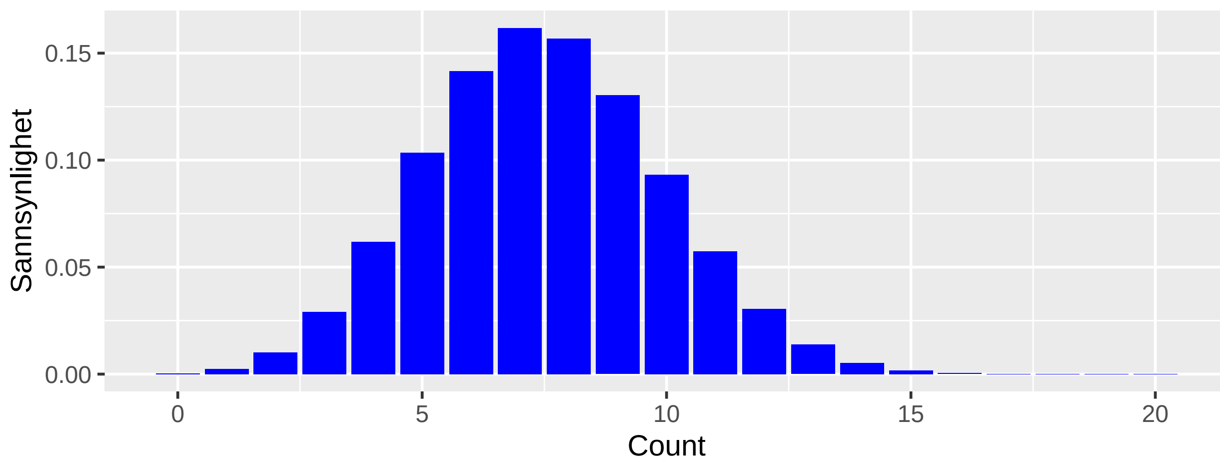

```r
ggplot(data=betabinomy2_df_pdf, aes(x=x, y=betabinomy2_pdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```




```r
ggplot(data=betabinomy3_df_pdf, aes(x=x, y=betabinomy3_pdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```

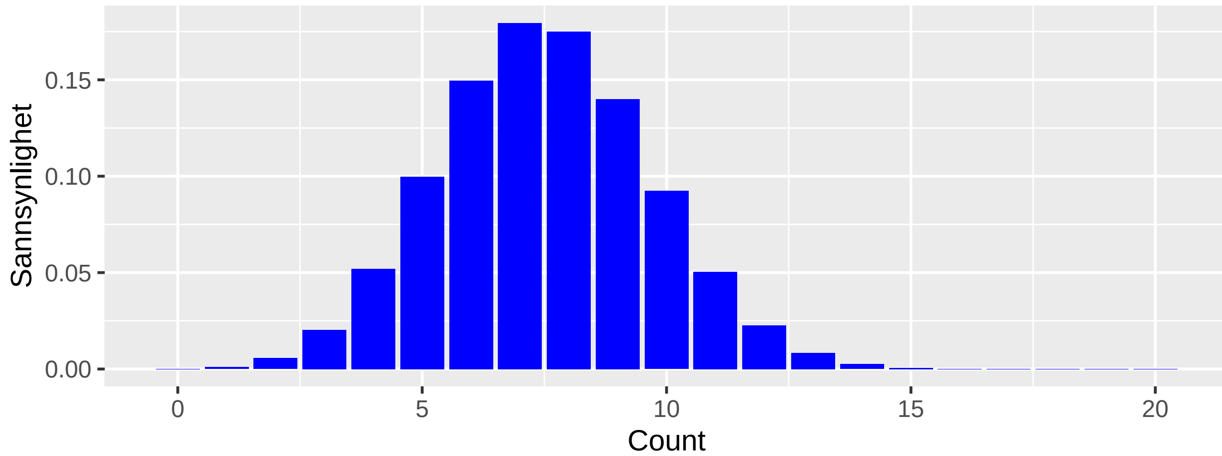



```
ggplot(data=betabinomy4_df_pdf, aes(x=x, y=betabinomy4_pdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



Her kommer plot for bin(20, 0.375)

```
ggplot(data=df, aes(x=x, y=y)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



Det vi kan se er at når a variabelen, som vi ganger med øker, så nærmer vi oss bin(20, 0.375) fordelingen.

2.5.3 Her er tabell for CDF:

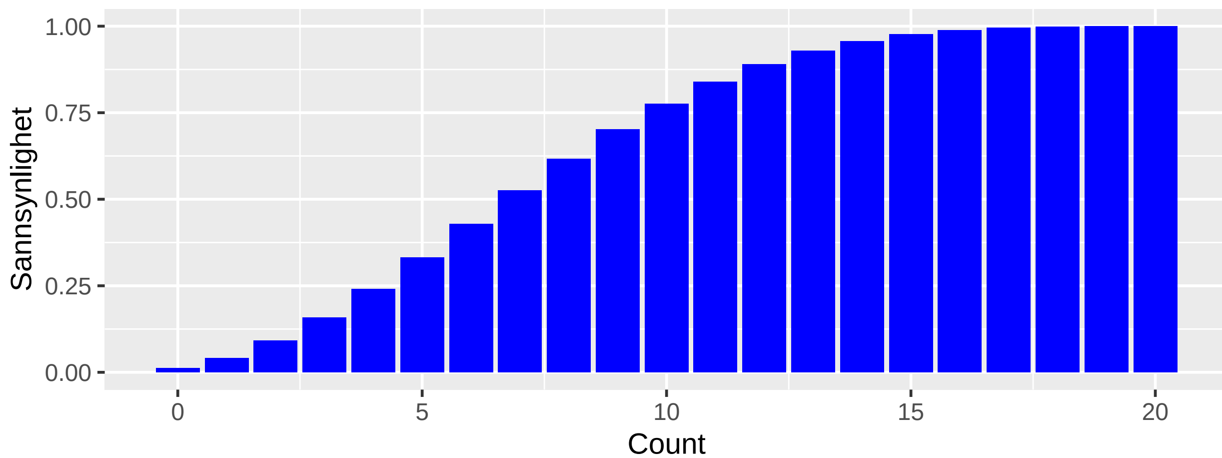
```
a1 = 1
betabinomy1_cdf = pbbinom(x, 20, 3*a1, 5*a1)

a2 = 2
betabinomy2_cdf = pbbinom(x, 20, 3*a2, 5*a2)

a3 = 4
betabinomy3_cdf = pbbinom(x, 20, 3*a3, 5*a3)

a4 = 10
betabinomy4_cdf = pbbinom(x, 20, 3*a4, 5*a4)

betabinomy1_df_cdf = data.frame(x, betabinomy1_cdf)
betabinomy2_df_cdf = data.frame(x, betabinomy2_cdf)
betabinomy3_df_cdf = data.frame(x, betabinomy3_cdf)
betabinomy4_df_cdf = data.frame(x, betabinomy4_cdf)

betabinomy1_df_cdf
```

```
##     x betabinomy1_cdf
## 1   0      0.01196581
## 2   1      0.04188034
## 3   2      0.09130435
## 4   3      0.15870072
## 5   4      0.24053917
## 6   5      0.33219824
## 7   6      0.42868146
## 8   7      0.52516469
## 9   8      0.61739130
## 10  9      0.70193237
## 11 10      0.77632850
## 12 11      0.83913043
## 13 12      0.88985507
## 14 13      0.92887402
## 15 14      0.95725144
```

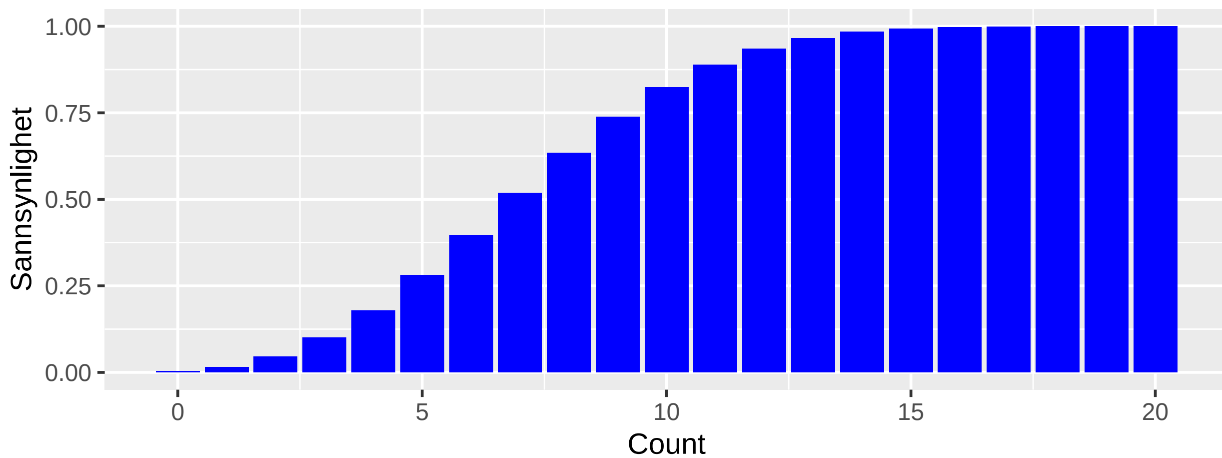
```
## 16 15       0.97654809
## 17 16       0.98860849
## 18 17       0.99534813
## 19 18       0.99855748
## 20 19       0.99973987
## 21 20       1.00000000
```

2.5.4 Her er plot for CDF med a=1,a=2,a=4 og a=10

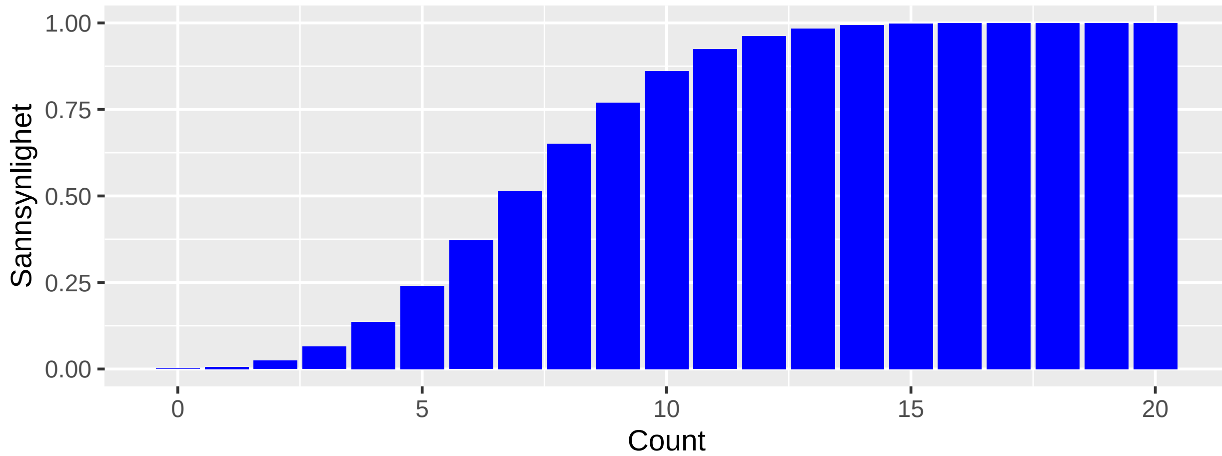
```r
ggplot(data=betabinomy1_df_cdf, aes(x=x, y=betabinomy1_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



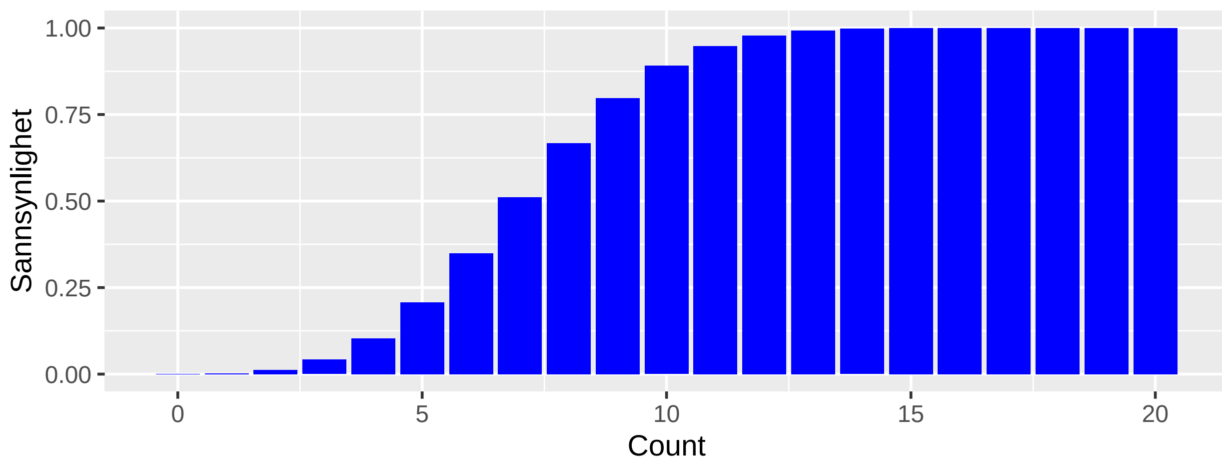

```r
ggplot(data=betabinomy2_df_cdf, aes(x=x, y=betabinomy2_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```




```r
ggplot(data=betabinomy3_df_cdf, aes(x=x, y=betabinomy3_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```




```r
ggplot(data=betabinomy4_df_cdf, aes(x=x, y=betabinomy4_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



2.5.5 E[X]

Vi summerer opp hver count ganget med sannsynligheten for å finne E[X]

```r
sum(x*betabinomy_pdf)
```

$E[X] = 7.5$

2.5.6 Var(X)

For å finne var(X) kjører vi bare følgende R-kode

```r
e = (sum(x^2*betabinomy2_pdf)-(sum(x*betabinomy2_pdf))^2)
```

$Var(X) = 9.9264706$

2.5.7 $P(2 < X < 7)$

For å finne ut sannsynligheten for X mellom 2 og 7 skriver vi følgende R-kode

```
lessthan7 = pbbinom(6, 20, 3*a2, 5*a2)
lessthan2 = pbbinom(1, 20, 3*a2, 5*a2)

print(lessthan7-lessthan2)
```

```
## [1] 0.3811168
```

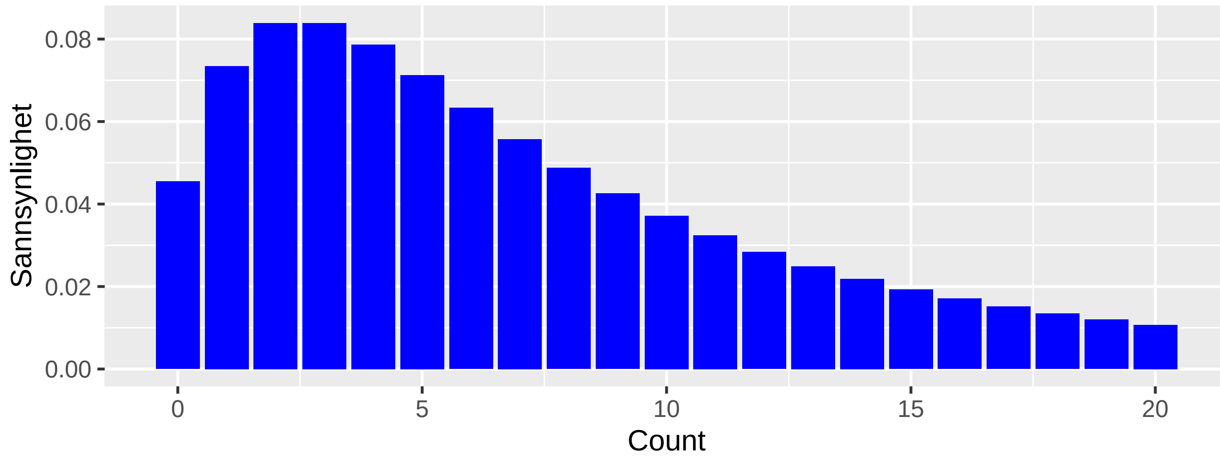
2.6 Beta negativ binomisk,Bnb(a,b,k):X Bnb(3a,7a,3)for a= 1. Lag tabell over sannsynligheter for x = 0,....,20, og plott både pdf og CDF for denne sannsynlighetsfordelingen. Regn deretter ut E[X],Var(X), og P(2 < X < 7)

2.6.1 Her er tabell og plot for PDF:

```
a_1 = 1
dbnbinom_y_pdf = dbnbinom(x, 3, 3*a_1, 7*a_1)

dbnbinom_ydf_pdf = data.frame(x, dbnbinom_y_pdf)

dbnbinom_ydf_pdf
```

```
##     x dbnbinom_y_pdf
## 1   0     0.04545455
## 2   1     0.07342657
## 3   2     0.08391608
## 4   3     0.08391608
## 5   4     0.07867133
## 6   5     0.07126697
## 7   6     0.06334842
## 8   7     0.05572755
## 9   8     0.04876161
## 10  9     0.04256966
## 11 10     0.03715170
## 12 11     0.03245267
## 13 12     0.02839609
## 14 13     0.02490119
## 15 14     0.02189115
## 16 15     0.01929665
## 17 16     0.01705686
## 18 17     0.01511936
## 19 18     0.01343943
## 20 19     0.01197912
## 21 20     0.01070634
```

```
ggplot(data=dbnbinom_ydf_pdf, aes(x=x, y=dbnbinom_y_pdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```

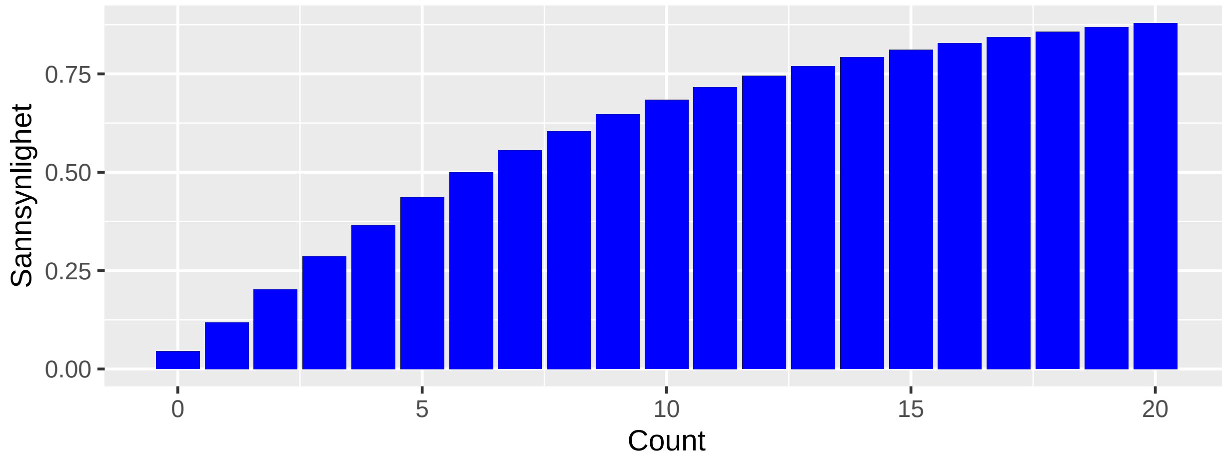


2.6.2 Her er tabell og plot for CDF:

```
dbnbinom_y_cdf = pbnbinom(x, 3, 3*a_1, 7*a_1)

dbnbinom_ydf_cdf = data.frame(x, dbnbinom_y_cdf)

dbnbinom_ydf_cdf
```

```
##     x dbnbinom_y_cdf
## 1   0     0.04545455
## 2   1     0.11888112
## 3   2     0.20279720
## 4   3     0.28671329
## 5   4     0.36538462
## 6   5     0.43665158
## 7   6     0.50000000
## 8   7     0.55572755
## 9   8     0.60448916
## 10  9     0.64705882
## 11 10     0.68421053
## 12 11     0.71666320
## 13 12     0.74505929
## 14 13     0.76996047
## 15 14     0.79185163
## 16 15     0.81114827
## 17 16     0.82820513
## 18 17     0.84332449
## 19 18     0.85676393
## 20 19     0.86874305
## 21 20     0.87944939
```

```
ggplot(data=dbnbinom_ydf_cdf, aes(x=x, y=dbnbinom_y_cdf)) +
  geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



2.6.3 E[X]

Vi summerer opp hver count ganget med sannsynligheten for å finne E[X]

```
sum(x*dbnbinom_y_pdf)
```

E[X] = 5.9098999

2.6.4 Var(X)

For å finne var(X) kjører vi bare følgende R-kode

```
e = (sum(x^2*dbnbinom_y_pdf)-(sum(x*dbnbinom_y_pdf))^2)
```

Var(X) = 27.1270321

2.6.5 $P(2 < X < 7)$

For å finne ut sannsynligheten for X mellom 2 og 7 skriver vi følgende R-kode

```
lessthan7 = pbnbinom(6, 3, 3*a_1, 7*a_1)
lessthan2 = pbnbinom(1, 3, 3*a_1, 7*a_1)

print(lessthan7-lessthan2)
```

```
## [1] 0.3811189
```

2.7 Plott Borel-Tanner-fordelingen med parameter u = 0.2.

2.7.1 Her er plot for Borel-Tanner:

```
library(VGAM)
```

```
## Warning: package 'VGAM' was built under R version 3.6.3
```

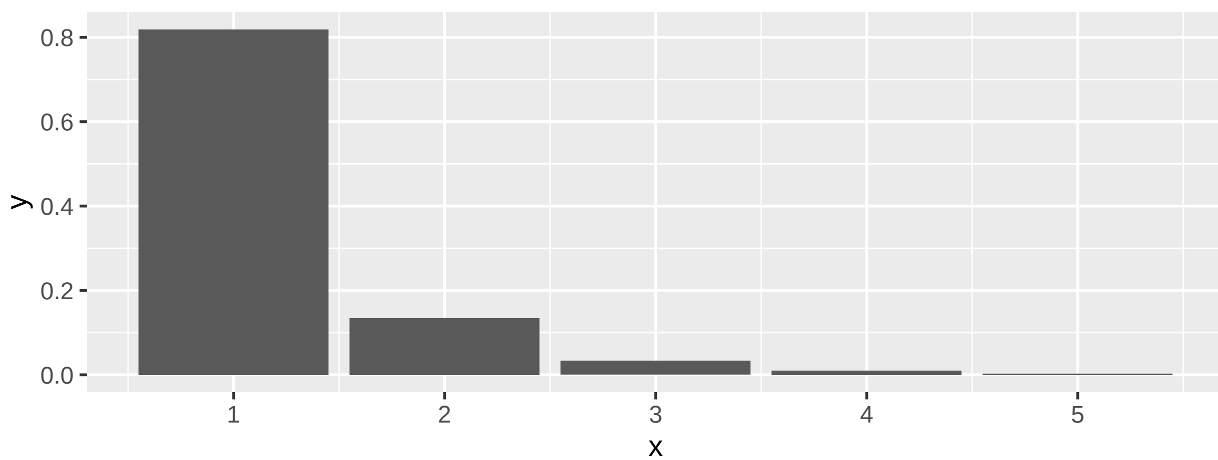
```
## Loading required package: stats4
```

```
## Loading required package: splines
```

```
##
## Attaching package: 'VGAM'
```

```
## The following objects are masked from 'package:extraDistr':
##
##     dfrechet, dgev, dgompertz, dgpd, dgumbel, dhuber, dkumar, dlaplace,
##     dlomax, dpareto, drayleigh, dskellam, dslash, pfrechet, pgev,
##     pgompertz, pgpd, pgumbel, phuber, pkumar, plaplace, plomax,
##     ppareto, prayleigh, pslash, qfrechet, qgev, qgompertz, qgpd,
##     qgumbel, qhuber, qkumar, qlaplace, qlomax, qpareto, qrayleigh,
##     rfrechet, rgev, rgompertz, rgpd, rgumbel, rhuber, rkumar, rlaplace,
##     rlomax, rpareto, rrayleigh, rskellam, rslash
```

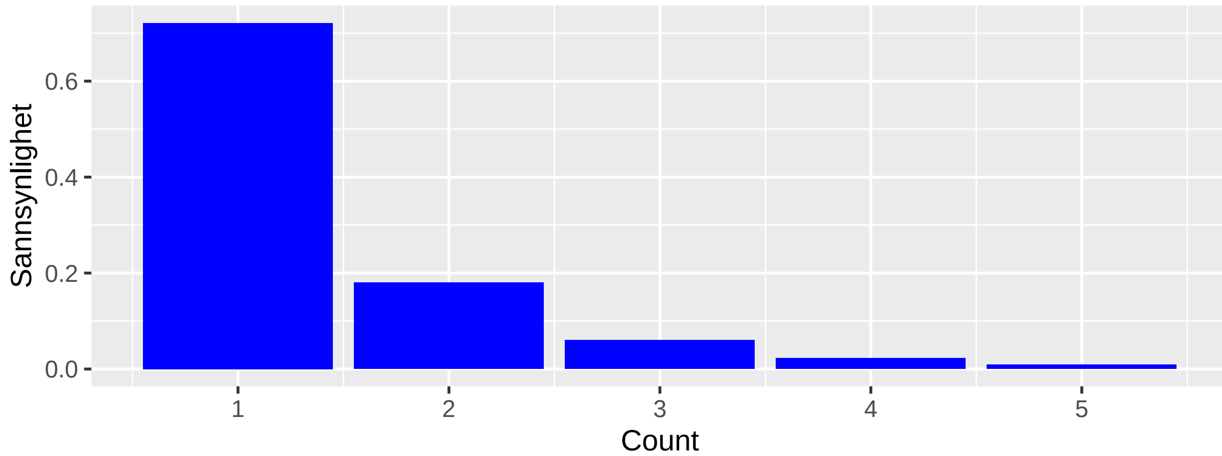
```r
u=0.2

x = c(1:5)
y = dbort(x, a=u)

distr = data.frame(x, y)

ggplot(distr,aes(x,y))+ geom_bar(stat = "identity")
```



2.8 Plott Logaritmisk fordeling med parameter p = 0.5

2.8.1 Her er plot for Logaritmisk fordeling:

```r
library(VGAM)

x = c(1:5)

y = dlog(x, 0.5)

distri = data.frame(x,y)

ggplot(distri, aes(x=x,y=y)) + geom_bar(stat="identity", fill="blue") +
  labs(x = "Count", y = "Sannsynlighet")
```



2.9 Plott Skellam-fordelingen med parametere lambda1 = 3 og lambda2 = 5

2.9.1 Her er plot for Skellam-fordelingen:

```
library(skellam)
```

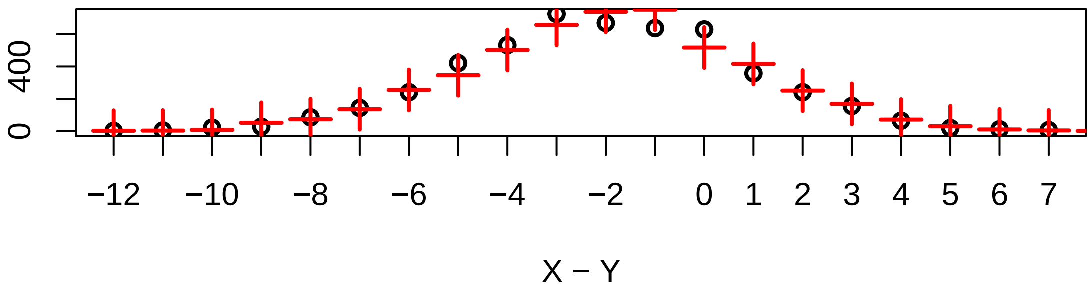
```
## Warning: package 'skellam' was built under R version 3.6.3
```

```
##
## Attaching package: 'skellam'
```

```
## The following objects are masked from 'package:VGAM':
##
##     dskellam, rskellam
```

```
## The following objects are masked from 'package:extraDistr':
##
##     dskellam, rskellam
```

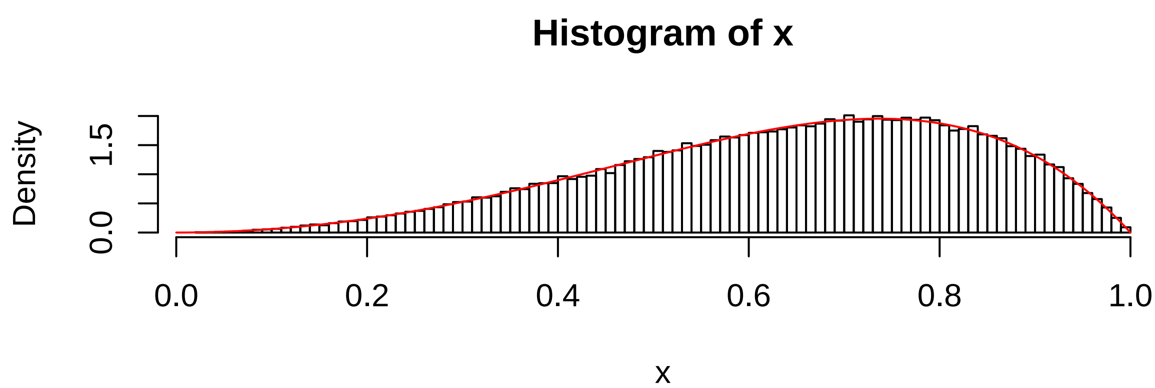
```
N = 5000
lambda1 = 3
lambda2 = 5

X = rpois(N, lambda1)
Y = rpois(N, lambda2)
XminusY = X - Y

Z = rskellam(N, lambda1, lambda2)

plot(table(XminusY), xlab="X - Y", ylab="", type="p", pch=1)

points(table(Z), col="red", type="p", pch=3, cex=2)
```


0
400
−12
−10
−8
−6
−4
−2
0
1
2
3
4
5
6
7
X − Y

3 Tilfedlige sannsynlighetsfordelinger

3.1 For å bruke kumar distribusjonen må vi bruke pakken "extraDistr"

```
library(extraDistr)

x <- rkumar(1e5, 3, 2)
hist(x, 100, freq = FALSE)
curve(dkumar(x, 3, 2), 0, 1, col = "red", add = TRUE)
```

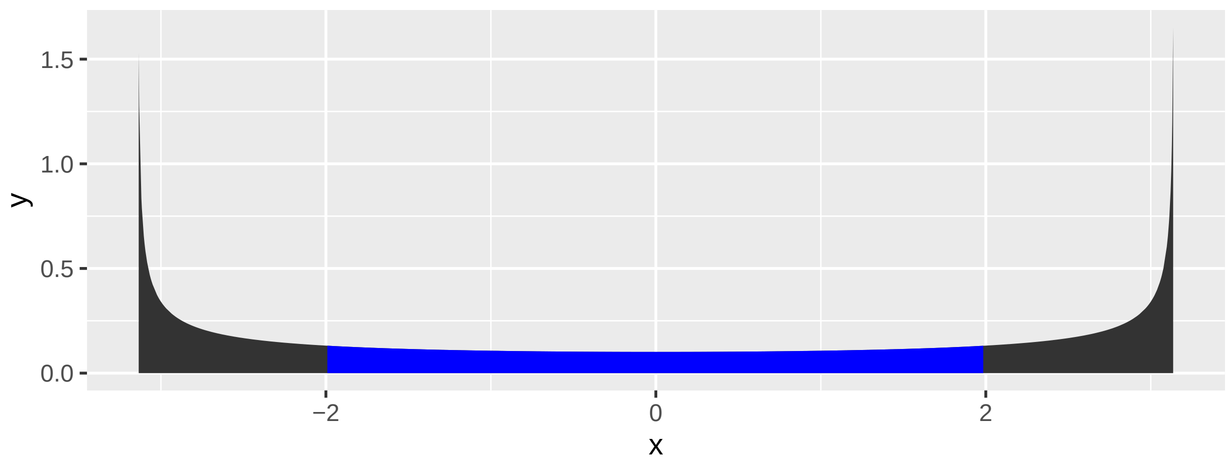


3.2 For å bruke arcsine-fordelineg må vi bruke pakken "VaRES"

```
library(ggplot2)
library(VaRES)
```

```
##
## Attaching package: 'VaRES'

## The following objects are masked from 'package:VGAM':
##
##     dbetanorm, ddagum, dexplog, dexppois, dfrechet, dgev, dgompertz,
##     dgumbel, dlaplace, dlomax, dpareto, dperks, pbetanorm, pdagum,
##     pexplog, pexppois, pfrechet, pgev, pgompertz, pgumbel, plaplace,
##     plomax, ppareto, pperks

## The following objects are masked from 'package:extraDistr':
##
##     ddweibull, dfrechet, dgev, dgompertz, dgumbel, dinvgamma, dlaplace,
##     dlomax, dpareto, pdweibull, pfrechet, pgev, pgompertz, pgumbel,
##     pinvgamma, plaplace, plomax, ppareto
```

```
x=runif(1000,min=-pi,max=pi)
y = darcsine(x, a=-pi, b=pi)
df = data.frame(x,y)
subset = df[df$x>=-2,]
subset = subset[subset$x<=2,]
ggplot(df, aes(x,y))+geom_area()+geom_area(fill="blue",data=subset)
```

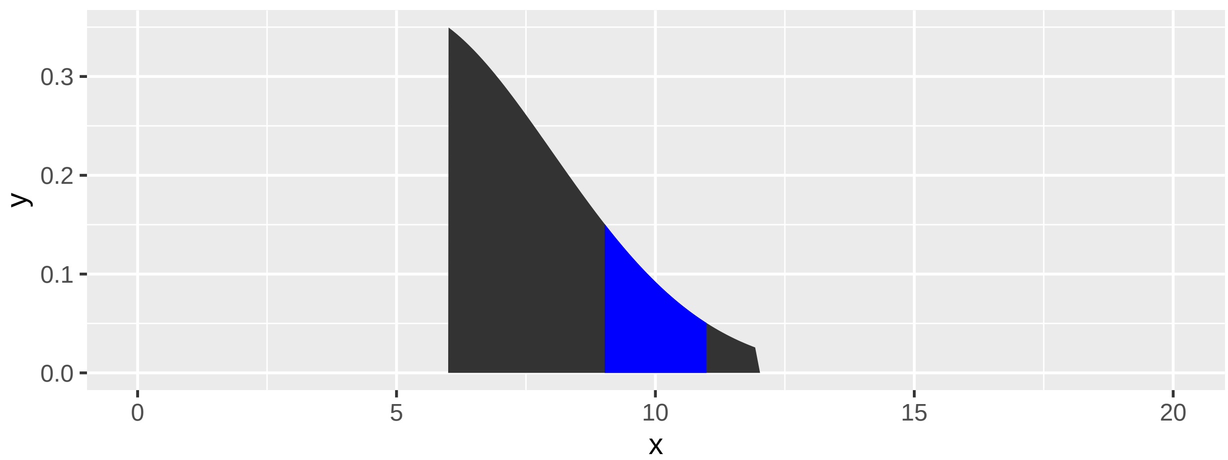


3.3 For å bruke truncnorm må vi installere pakken "truncnorm"

```
library(truncnorm)
```

```
## Warning: package 'truncnorm' was built under R version 3.6.3
```

```
x=runif(1000, min = 0, max = 20)
y = dtruncnorm(x, a=6, b=12, mean = 5, sd = 3)
df = data.frame(x, y)

subset = df[df$x>=9,]
subset = subset[subset$x<=11,]

ggplot(df, aes(x,y))+geom_area()+geom_area(fill="blue",data=subset)
```



3.4 For å bruke vonMises må vi installere pakken "circular"

```
library(circular)
```

```
## Warning: package 'circular' was built under R version 3.6.3
```

```
##
## Attaching package: 'circular'
```

```
## The following object is masked from 'package:VaRES':
##
##     dtriangular
```

```
## The following objects are masked from 'package:stats':
##
##     sd, var
```

```
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 3.6.3
```

```
p=ggdistribution(pvonmises,seq(-2,4.3,0.001),mu=1.2,kappa=2,colour =
"black")
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularqradians0counter
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularmuradians0counter
```

```
p=ggdistribution(pvonmises,seq(-2,4.3,0.001),mu=1.2,kappa=0.5,colour =
"red",p=p)
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularqradians0counter
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularmuradians0counter
```

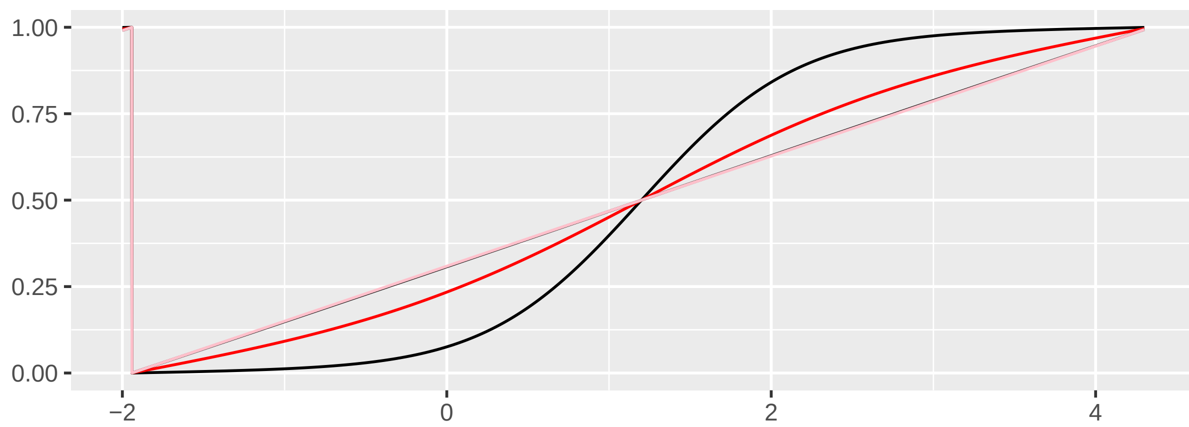
```
p=ggdistribution(pvonmises,seq(-2,4.3,0.001),mu=1.2,kappa=0.01,colour =
"black",p=p)
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the
```

```
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularqradians0counter

## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the f
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularmuradians0counter
```

```
ggdistribution(pvonmises,seq(-2,4.3,0.001),mu=1.2,kappa=0.00001,colour =
"pink",p=p)
```

```
## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the f
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularqradians0counter

## Warning in as.circular(x): an object is coerced to the class 'circular' using default value for the f
##   type: 'angles'
##   units: 'radians'
##   template: 'none'
##   modulo: 'asis'
##   zero: 0
##   rotation: 'counter'
## conversion.circularmuradians0counter
```



3.5 For å kjøre den brettede normalfordelingen må vi installere pakken "VGAM"

Område i blått er P(U > 2)

```
library(VGAM)
d = pfoldnorm(2,mean = 7, sd=4,a1=1,a2=1,lower.tail = FALSE)
d # 0.9065747
```

```
## [1] 0.9065747
```

```
p = ggdistribution(dfoldnorm,seq(-5,20,0.01),mean=7,sd=4,a1=1,a2=1)
ggdistribution(dfoldnorm,seq(d,20,0.01),mean=7,sd=4,a1=1,a2=1,fill="blue",p=p)
```