



Bruk av Matlab i statistikk

Notat for TMA4240/TMA4245 Statistikk*

Institutt for matematiske fag,
NTNU

15. august 2012

Dette notatet er ment som en kort innføring i bruk av Matlab i forbindelse med introduksjonskurset i statistikk. I Matlab representeres lister med tall som matriser eller vektorer. En $m \times n$ -matrise har m rader og n kolonner. En $1 \times n$ -matrise er det samme som en radvektor med n elementer, og en $m \times 1$ -matrise er det samme som en kolonnevektor med m elementer. Skal man f.eks. trekke hundre tilfeldige tall, er det mer praktisk å trekke en tilfeldig vektor med hundre elementer, enn å trekke ett og ett tall hundre ganger. Generelt er vektor- og matriseoperasjoner å foretrekke fremfor for-løkker, spesielt hvis datamengden er stor.

1 Sannsynlighetsfordelinger

1.1 *Probability Distribution Function Tool* (disttool)

Verktøyet disttool lar deg studere ulike kumulative sannsynlighetsfordelingsfunksjoner (cdf) og sannsynlighetstetthetsfunksjoner (pdf) og hvordan plasseringen, størrelsen og formen deres avhenger av parametrene til fordelingen. For å åpne disttool, skriv `disttool` i kommandolinja, eller velg disttool i menyen under *Start* $\rightarrow$ *Toolboxes* $\rightarrow$ *Statistics*. I vinduet som kommer opp kan du velge hvilken fordeling du vil studere, og om du vil se sannsynlighetstetthetsfunksjonen (PDF) eller den kumulative fordelingsfunksjonen (CDF). Verdier kan leses av fra grafen; flytt den røde stiplede linja ved å klikke i figurvinduet, eller skriv inn ønsket verdi i en av boksene. Bruk den nederste delen av vinduet til å sette verdien til de ulike parametrene. Navnet på disse, samt hvor mange det er, avhenger av hvilken fordeling som er valgt (se seksjon 3). Plottet oppdateres fortløpende når parametrene endres gradvis.

1.2 Tetthetsfunksjon og kumulativ fordelingsfunksjon; pdf og cdf

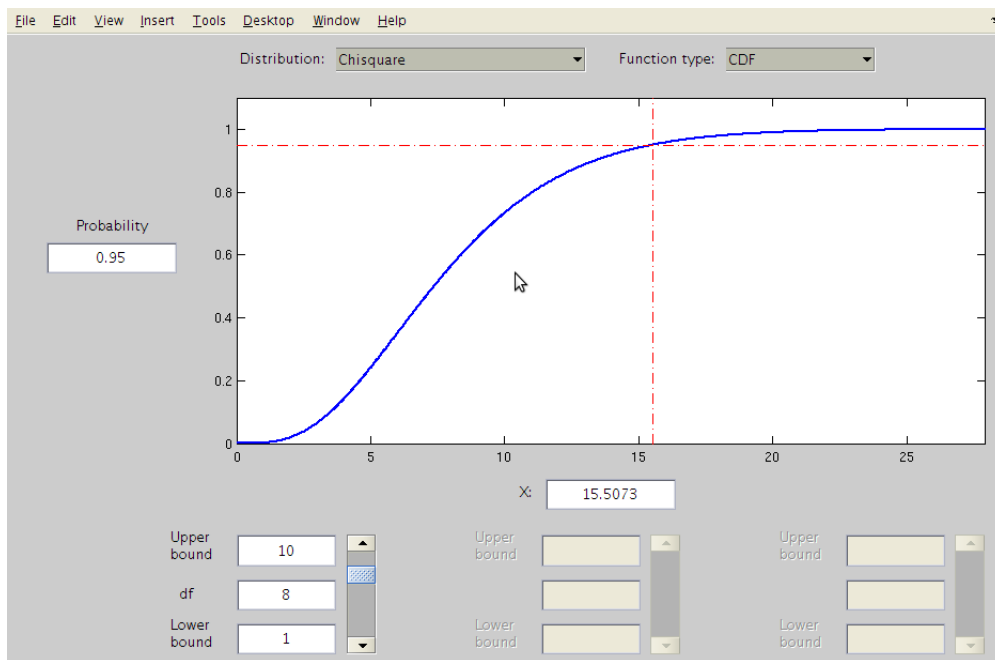
Funksjonene `pdf` og `cdf` brukes for å evaluere sannsynlighetstetthetsfunksjoner og kumulative fordelingsfunksjoner. F.eks. vil

```
X = linspace(-3,3,1000);  
Y = pdf('normal',X,0,1);
```

evaluere pdf-en til normalfordelingen med forventningsverdi 0 og standardavvik 1 i 1000 punkter mellom -3 og 3 , og lagre resultatet i vektoren `Y`.

`cdf` kalles på samme måte.

*Notatet er skrevet av Jacob Skauvold i samarbeid med Ingelin Steinsland. Dersom du finner feil eller har forslag til forbedringer, ta kontakt med Ingelin Steinsland, ingelins@math.ntnu.no.



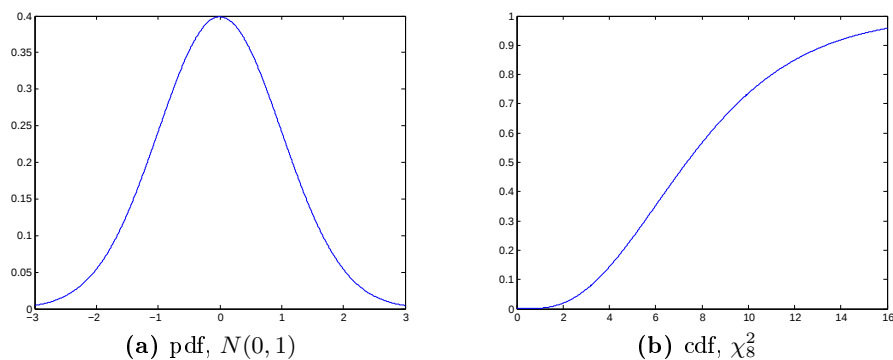
Figur 1 – cdf-en til χ^2 -fordelingen med 8 frihetsgrader

```
U = linspace(0,16,1000);
V = cdf('chisquare',U,8);
```

evaluerer cdf-en til χ^2 -fordelingen med åtte frihetsgrader i 1000 punkter mellom 0 og 16, og lagrer resultatet i vektoren V . For å plote fordelingsfunksjonene i hver sin figur kan vi skrive

```
figure(1)
plot(X,Y)
figure(2)
plot(U,V)
```

som gir plottene i fig. 2.



Figur 2

Skriv `help pdf` eller `help cdf` for å se en liste over navn på fordelinger som kan brukes i stedet for 'normal' og 'chisquare' i eksemplene over. Antall parametre og rekkefølgen

de har i funksjonskallet avhenger av fordeling (se oversikt nedenfor).

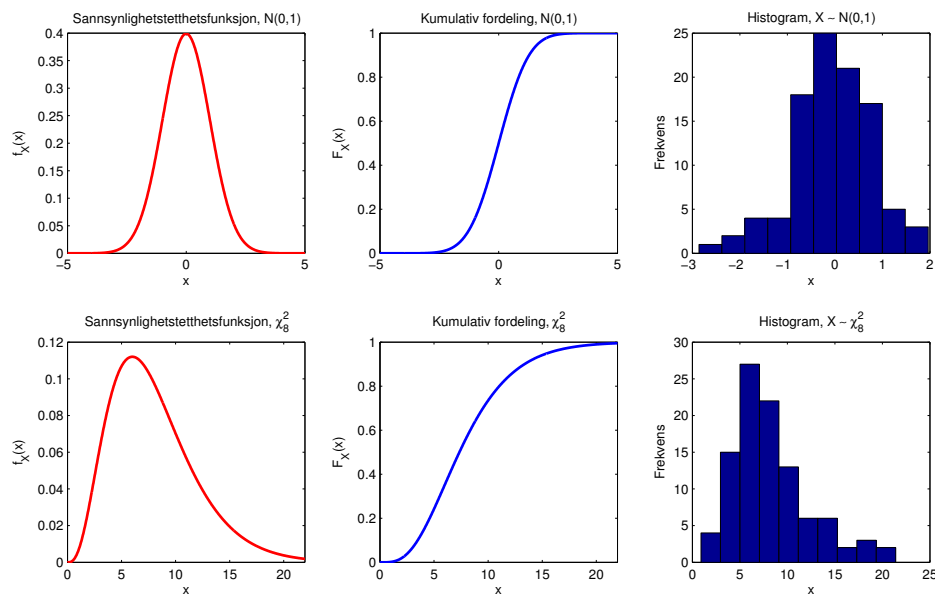
Eksempel: Plottene i fig. 3 er laget med kommandoene under. Se fila `sannsynlighetsfordelinger.m` for det fullstendige scriptet.

```
normal_pdf = pdf('normal',x1,0,1);
normal_cdf = cdf('normal',x1,0,1);
normal_random = random('normal',0,1,100,1);

chisquare_pdf = pdf('chisquare',x2,8);
chisquare_cdf = cdf('chisquare',x2,8);
chisquare_random = random('chisquare',8,100,1);

plot(x1,normal_pdf,'r-','LineWidth',2)
plot(x1,normal_cdf,'b-','LineWidth',2)
hist(normal_random)

plot(x2,chisquare_pdf,'r-','LineWidth',2)
plot(x2,chisquare_cdf,'b-','LineWidth',2)
hist(chisquare_random)
```



Figur 3 – Plott av (fra venstre) sannsynlighetstetthetsfunksjonen, den kumulative fordelingsfunksjonen, og et histogram av et sett realisasjoner trukket fra en standard normalfordeling (øverst) og fra en kji-kvadratfordeling med åtte frihetsgrader (nederst).

1.3 Invers kumulativ fordelingsfunksjon; icdf

Hvis en tilfeldig variabel X har en fordeling med sannsynlighetstetthetsfunksjon $f_X(x)$ og kumulativ fordelingsfunksjon $F_X(x)$, så kalles inversfunksjonen til $F_X(x)$ den inverse kumulative fordelingsfunksjonen. Den betegnes $F_X^{-1}(p)$ og tar verdier av p på intervallet

(0,1). Tolkningen av den kumulative fordelingen er $F_X(x) = P(X \leq x)$, mens tolkningen av den inverse kumulative fordelingen er at $F_X^{-1}(p)$ gir den verdien x som er slik at $P(X \leq x) = p$, dvs. slik at X er mindre enn eller lik x med sannsynlighet p .

`icdf` brukes for å evaluere inverse kumulative fordelingsfunksjoner. F.eks. kan en skrive `icdf('exponential',0.5,3)` for å finne medianen til eksponentialfordelingen med parameter/forventningsverdi $\mu = 3$. `icdf` kalles på samme måte som `pdf` og `cdf`.

Legg ellers merke til at vi på fig. 1 også kan lese av verdien av den inverse kumulative fordelingsfunksjonen F_X^{-1} til χ^2 -fordelingen med åtte frihetsgrader; $F_X^{-1}(0.95) = 15.5073$.

2 Trekke tilfeldige tall

2.1 Funksjoner for trekking av tilfeldige tall

Skal man trekke tilfeldige tall kan man enten bruke funksjonen `random` og spesifisere hvilken fordeling det skal trekkes fra, eller man kan bruke en spesialisert funksjon for den ønskede fordelingen, f.eks. `normrnd` eller `unidirnd`.

Eksempel: Både `normrnd(0,1,m,n)` og `random('normal',0,1,m,n)` genererer en $m \times n$ -matrise med tall fra normalfordelingen med forventningsverdi 0 og standardavvik 1.

Skriv `help random` for en liste med navn på ulike fordelinger, og `help stats` for en oversikt over funksjonene i Matlab Statistics Toolbox. Funksjonene som genererer tilfeldige tall står oppført under *Random Number Generators*.

Eksempel: For å generere en vektor med 10 tall fra Weibullfordelingen i eksempelet ovenfor kan en enten bruke `wblrnd` eller `random` med argumentet `'Weibull'`. Velger det første, og lager en kolonnevektor, dvs. en 20×1 -matrise.

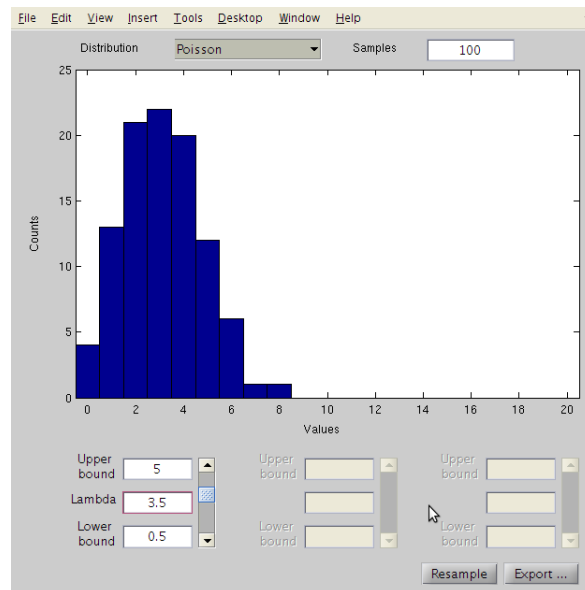
```
>> wblrnd(1/0.05,3,20,1)
```

```
ans =
```

```
20.3271
23.5211
22.2773
15.7076
18.1554
20.2962
11.4028
16.2431
16.8531
8.8429
21.5577
13.0550
13.1265
19.7735
16.5433
27.4269
28.5853
17.1758
12.5913
```

2.2 Random Number Generation Tool (randtool)

Et annet alternativ for å trekke tilfeldige tall er randtool. Skriv `randtool` eller velg randtool i menyen under *Start* → *Toolboxes* → *Statistics*. Velg fordeling, hvor mange tall som skal trekkes, og sett verdien til eventuelle parametre. Dataene vises i et histogram. Trykk på *Resample* for å trekke om igjen, og trykk på *Export...* for å lagre dataene i en vektor.



Figur 4 – Randtool, sampling fra Poissonfordelingen med parameter $\lambda = 3.5$

3 Oversikt over parametriseringer

Læreboka (W. M. M. Y.) og Matlab bruker forskjellige parametriseringer for noen av sannsynlighetsfordelingene. Denne oversikten dekker de mest brukte fordelingene. Generelt brukes de samme argumentene i `pdf`, `cdf`, `random` og andre Matlabfunksjoner som bruker sannsynlighetsfordelinger.

3.1 Binomialfordeling

Samme parametrisering i W.M.M.Y. og Matlab

Sannsynlighetsfordelingen for antall suksesser fra n uavhengige forsøk er

$$b(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, 2, \dots, n.$$

Eksempel: Et gitt forsøk resulterer i en suksess med sannsynlighet $p = 0.4$ og i en ikke-suksess/fiasco med sannsynlighet $1-p = 0.6$. Dersom det utføres tolv slike forsøk uavhengig av hverandre, hva er sannsynligheten for at akkurat åtte av dem resulterer i suksesser? Hva er sannsynligheten for at åtte eller færre resulterer i suksesser?

```
>> n = 12; p = 0.4;
>> pdf('binomial',8,n,p)
```

```
ans =
```

```
0.0420
```

```
>> cdf('binomial',8,n,p)
```

```
ans =
```

```
0.9847
```

Sannsynligheten for å få akkurat åtte suksesser er 0.0420, og sannsynligheten for å få åtte eller færre er 0.9847. Som en kontroll kan vi også legge sammen sannsynlighetsmassene for $x = 0, \dots, 8$, slik

$$B(8; n, p) = \sum_{x=0}^8 b(x; n, p).$$

I Matlab kan dette gjøres på følgende måte.

```
>> pdf('binomial',[0:8]',n,p)
```

```
ans =
```

```
0.0022
```

```
0.0174
```

```
0.0639
```

```
0.1419
```

```
0.2128
```

```
0.2270
```

```
0.1766
```

```
0.1009
```

```
0.0420
```

```
>> sum(ans)
```

```
ans =
```

```
0.9847
```

3.2 Multinomisk fordeling

Forskjellig parametrisering i W.M.M.Y. og Matlab

Hvis et forsøk har k mulige utfall $E_1, E_2, \dots, E_k$ som inntreffer med sannsynlighet $p_1, p_2, \dots, p_k$, så er sannsynlighetsfordelingen til $X_1, X_2, \dots, X_k$, antall forekomster av $E_1, E_2, \dots, E_k$, for n uavhengige forsøk

$$f(x_1, x_2, \dots, x_k; p_1, p_2, \dots, p_k, n) = \frac{n!}{x_1! x_2! \dots x_k!} p_1^{x_1} p_2^{x_2} \dots p_k^{x_k}$$

med

$$\sum_{i=1}^k x_i = n \quad \text{og} \quad \sum_{i=1}^k p_i = 1.$$

I Matlab brukes funksjonen

$$f(x|n, p) = \frac{n!}{x_1! x_2! \dots x_k!} p_1^{x_1} p_2^{x_2} \dots p_k^{x_k}.$$

Altså samme form som i boka, men argumentene $x = (x_1, x_2, \dots, x_k)^T$ og $p = (p_1, p_2, \dots, p_k)^T$ er vektorer, og n beregnes automatisk, slik at funksjonen bare tar to argumenter.

Eksempel: Hvis et myntkast gir mynt (M) eller kron (K) med sannsynlighet henholdsvis $P(M) = 0.5$ og $P(K) = 0.5$, så vil to uavhengige kast gi to mynt med sannsynlighet $P(MM) = P(M)^2 = 0.25$, to kron med sannsynlighet $P(KK) = P(K)^2 = 0.25$ og en av hver med sannsynlighet $P(MK) + P(KM) = P(M)P(K) + P(K)P(M) = 2 \cdot 0.5^2 = 0.5$. Om vi betrakter to slike uavhengige myntkast som ett forsøk, og utfører tolv slike forsøk, hva er da sannsynligheten for at vi observerer utfallene MM , KK og $MK \cup KM$ akkurat fire ganger hver?

```
>> x = [4 4 4]'; p = [1/4 1/2 1/4]';  
>> mnpdf(x,p)
```

```
ans =
```

```
0.0330
```

3.3 Hypergeometrisk fordeling

Forskjellig parametrisering i W.M.M.Y. og Matlab

Sannsynlighetsfordelingen for antall suksesser om man trekker n medlemmer fra en populasjon på N , hvorav k gir suksess, er

$$h(x; N, n, k) = \frac{\binom{k}{x} \binom{N-k}{n-x}}{\binom{N}{n}}, \quad \max\{0, n - (N - k)\} \leq x \leq \min\{n, k\}.$$

I Matlab er rekkefølgen på parametrene N, k, n .

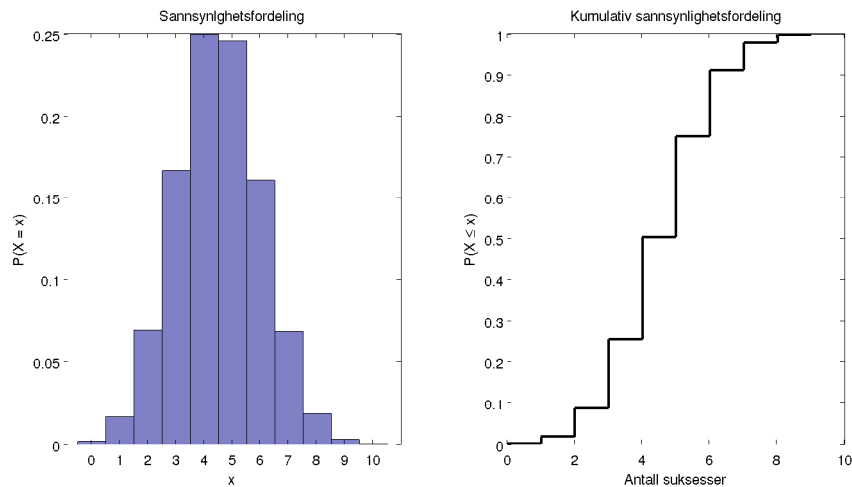
Eksempel: Fra en populasjon på 100 lyspærer hvorav 45 er defekte, gjøres et tilfeldig utvalg av ti pærer, som så testes. Hva er sannsynligheten for at nøyaktig fem av de ti pærene som testes viser seg å være defekte?

```
>> N = 100; n = 10; k = 45;  
>> pdf('hypergeometric',5,N,k,n)  
ans =
```

```
0.2455
```

Plotter også sannsynlighetsfordelingen og den kumulative fordelingen, se `hypergeometrisk.m` for det fullstendige scriptet. Plottene er vist i fig. 5

```
y1 = pdf('hypergeometric',x,N,k,n);  
y2 = cdf('hypergeometric',x,N,k,n);  
bar(x,y1,1)  
stairs(x,y2,'k-','LineWidth',2)
```



Figur 5 – Sannsynlighetsfordeling (venstre) og kumulativ fordeling (høyre) for den hypergeometriske fordelingen i lyspæreeksemepelet, $h(x; 100, 10, 45)$.

3.4 Negativ binomialfordeling

Forskjellig parametrisering i W.M.M.Y. og Matlab

Sannsynlighetsfordelingen for antall forsøk *til og med* den k te suksessen, når sannsynligheten for suksess er p , er

$$b^*(x; k, p) = \binom{x-1}{k-1} p^k (1-p)^{x-k}, \quad x = k, k+1, k+2, \dots$$

Fordelingen Matlab bruker er

$$f(\tilde{x}|k, p) = \binom{k+\tilde{x}-1}{\tilde{x}} p^k (1-p)^{\tilde{x}},$$

som er sannsynlighetsfordelingen for antall fiaskoer/ikke-suksesser før suksess nr. k inntrer. Summen av antall suksesser og antall fiaskoer er antall forsøk, så $x = \tilde{x} + k$. Ser ved innsetting at $f(x-k|k, p) = b^*(x; k, p)$.

Eksempel: Anta at vi ønsker å finne sannsynligheten for at det trengs nøyaktig 12 kast med en terning for å få tre sekser. Definer en sekser som en suksess og ethvert annet kast som en fiasko. Ønsker da å finne $b^*(12; 3, \frac{1}{6})$.

```
>> x = 12; k = 3; p = 1/6;
>> pdf('nbin', x-k, k, p)
```

```
ans =
```

```
0.0493
```

Hva er sannsynligheten for at det trengs 12 eller flere kast for å få tre sekser?

```
>> 1 - cdf('nbin', (x-1)-k, k, p)
```

```
ans =
```

```
0.7268
```

Finner sannsynligheten for at det trengs $x - 1 = 11$ eller færre kast, og tar sannsynligheten for den komplementære hendelsen.

3.5 Geometrisk fordeling

Forskjellig parametrisering i W.M.M.Y. og Matlab

Sannsynlighetsfordelingen for antall forsøk *til og med* første suksess inntreffer er

$$g(x; p) = p(1 - p)^{x-1}, \quad x = 1, 2, 3, \dots$$

Funksjonen som brukes i Matlab er sannsynlighetsfordelingen for antall forsøk *før* første suksess inntreffer,

$$f(\tilde{x}|p) = p(1 - p)^{\tilde{x}},$$

slik at $f(x - 1|p) = g(x; p)$.

Eksempel: Av 160 Nonstop i en skål er 40 stykker gule. Hvis Nonstop trekkes tilfeldig fra skåla enkeltvis og med tilbakelegg, hva er sannsynligheten for å trekke den første ikke-gule Nonstop-en på femte forsøk?

```
>> x = 5; p = 0.75;  
>> pdf('geo', x-1, p)
```

```
ans =
```

```
0.0029
```

3.6 Poissonfordeling

Forskjellig parametrisering i W.M.M.Y. og Matlab

Sannsynlighetsfordelingen for antall hendelser som inntreffer i et intervall med lengde t er

$$p(x; \lambda t) = \frac{e^{-\lambda t} (\lambda t)^x}{x!}, \quad x = 0, 1, 2, \dots,$$

hvor λ er raten til prosessen. Matlab bruker funksjonen

$$f(x|\tilde{\lambda}) = \frac{\tilde{\lambda}^x}{x!} e^{-\tilde{\lambda}},$$

dvs. $f(x|\lambda t) = p(x; \lambda \cdot t)$.

Eksempel: Du observerer en veistrekning hvor det i gjennomsnitt passerer én bil hvert femte minutt. Hvis trafikken modelleres som en Poissonprosess med intensitet $\lambda = \frac{1}{5}$, hva er da sannsynligheten for at det i løpet av en periode på $t = 5$ minutter passerer akkurat to biler?

$$P(X = 2) = p(2; \lambda t)$$

```
>> x = 2; lambda = 0.2; t = 5;  
>> pdf('poisson', x, lambda*t)
```

```
ans =
```

```
0.1839
```

Hva er sannsynligheten for at det i løpet av en like lang periode passerer tre eller flere biler?

$$P(X \geq 3) = 1 - P(X \leq 2) = 1 - \sum_{x=0}^2 p(x; \lambda t)$$

```
>> 1 - cdf('poisson',2,lambda*t)
```

```
ans =
```

```
0.0803
```

3.7 Kontinuerlig uniform fordeling

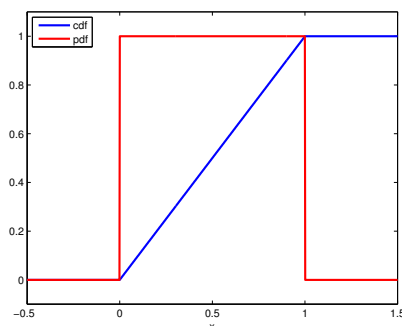
Samme parametrisering i W.M.M.Y. og Matlab

Sannsynlighetstetthetsfunksjonen til den kontinuerlige uniforme fordelingen på intervallet $[a, b] \subset \mathbb{R}$ er

$$f(x; a, b) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{ellers} \end{cases}.$$

Eksempel: Plotter den kumulative fordelingsfunksjonen og tetthetsfunksjonen til den kontinuerlige uniforme fordelingen på $[0, 1]$. Se fig. 6.

```
>> x = linspace(-1/2,3/2,1000);
>> y = cdf('uniform',x,0,1);
>> z = pdf('uniform',x,0,1);
>> figure(1), hold on
>> plot(x,y,'b-','LineWidth',2);
>> plot(x,z,'r-','LineWidth',2);
>> legend('cdf','pdf','Location','NorthWest');
>> xlabel('x');
>> ylim([-0.1 1.1])
```



Figur 6 – Den kontinuerlige uniforme sannsynlighetsfordelingen på $[0, 1]$.

3.8 Normalfordeling

Samme parametrisering i W.M.M.Y. og Matlab

Tetthetsfunksjonen til normalfordelingen er

$$n(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}, \quad x \in \mathbb{R}.$$

Matlab leser parametrene på samme måte, dvs. bruk standardavviket σ , ikke variansen σ^2 .

Eksempel: Genererer normalfordelte data og bruker en ensidig t -test til å sammenlikne utvalgene. Hypotesene er $H_0 : \mu_X = \mu_Y$ og $H_1 : \mu_X > \mu_Y$.

```
>> X = random('normal',0,1,30,1);  
>> Y = random('normal',0.1,1,40,1);  
>> [H, P] = ttest2(X,Y,0.05,'right')
```

H =

1

P =

0.0406

H_0 forkastes på signifikansnivå 5%, P-verdien er 0.0406.

3.9 Gammafordeling

Samme parametrisering i W.M.M.Y. og Matlab

Tetthetsfunksjonen til gammafordelingen er

$$f(x; \alpha, \beta) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-x/\beta}, \quad x > 0,$$

hvor $\Gamma(\cdot)$ er gammafunksjonen.

Eksempel: Lager et plott av tetthetsfunksjonen til gammafordelingen.

```
>> a = 3; b = 2;  
>> x1_max = icdf('gamma',0.995,a,b);  
>> x1 = linspace(0,x_max,1000);  
>> gamma_pdf = pdf('gamma',x,a,b);  
>> plot(x1,gamma_pdf,'r-','LineWidth',2)
```

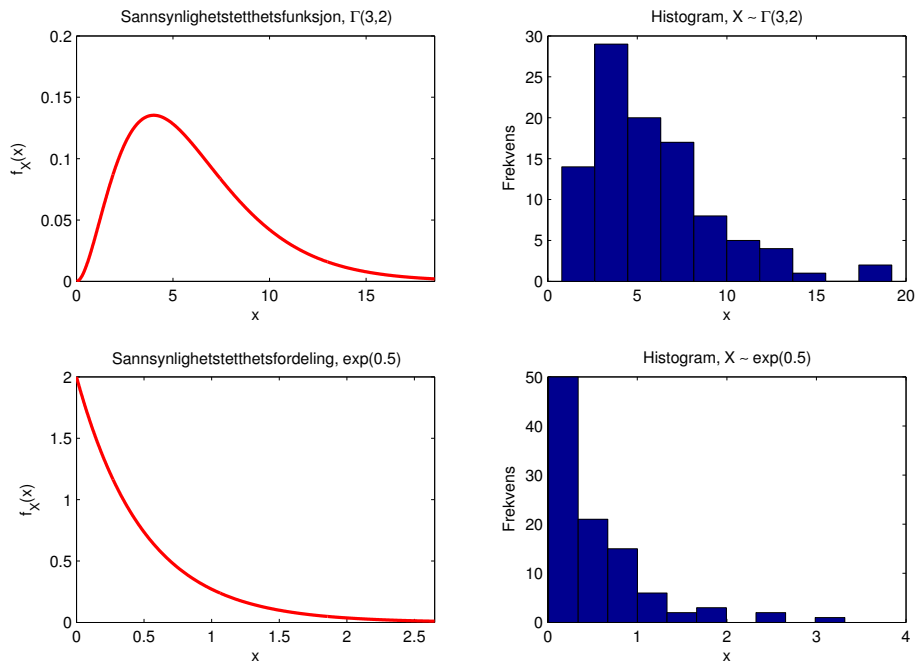
Resultatet er vist i fig. 7.

3.10 Eksponentialfordeling

Samme parametrisering i W.M.M.Y. og Matlab

Tetthetsfunksjonen til eksponentialfordelingen med parameter $\beta > 0$ er

$$f(x; \beta) = \frac{1}{\beta} e^{-x/\beta}, \quad x > 0,$$



Figur 7 – Venstre: Sannsynlighetstetthetsfunksjoner for gammafordlingen med $\alpha = 3$ og $\beta = 2$ (øverst), og for eksponentialfordelingen med $\beta = 0.5$ (nederst). Høyre: Histogram av 100 realisasjoner trukket fra de samme fordelingene. Figuren er laget med `gamma_eksp.m`.

og Matlab bruker samme parameter som boka.

Eksempel: Trekker realisasjoner fra en eksponentialfordeling med parameter $\beta = \frac{1}{2}$, og plottes dem i et histogram.

```
>> eksponential_random = random('exponential',0.5,100,1);
>> hist(eksponential_random)
```

Histogrammet er vist i fig. 7.

3.11 Kji-kvadratfordeling

Samme parametrisering i W.M.M.Y. og Matlab

Sannsynlighetstetthetsfunksjonen til χ^2 -fordelingen er

$$f(x; v) = \frac{1}{2^{v/2} \Gamma(v/2)} x^{v/2-1} e^{-x/2}, \quad x > 0,$$

hvor $\Gamma(\cdot)$ fortsatt er gammafunksjonen, og v er antall frihetsgrader.

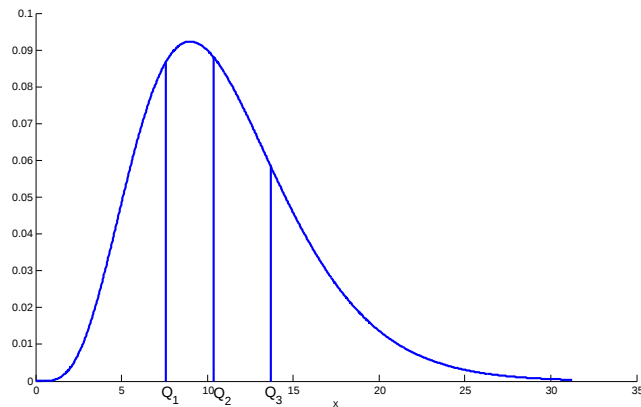
Eksempel: Finner kvartilene, Q_1 , Q_2 og Q_3 (henholdsvis 25- 50- og 75-prosent kvantilene) til en χ^2 -fordeling med $v = 11$ frihetsgrader. La $F(x)$ være den kumulative fordelingsfunksjonen til denne fordelingen. Da er

$$Q_1 = F^{-1}\left(\frac{1}{4}\right), \quad Q_2 = F^{-1}\left(\frac{1}{2}\right), \quad \text{og} \quad Q_3 = F^{-1}\left(\frac{3}{4}\right).$$

```
>> v = 11;
>> icdf('chisquare',[1/4 1/2 3/4],v)
```

```
ans =
```

```
7.5841    10.3410    13.7007
```



Figur 8 – Q_1 , Q_2 og Q_3 er henholdsvis 25, 50 og 75 prosent-kvantilene til χ^2 -fordelingen med 11 frihetsgrader. Figuren er laget med `kjikkvadratplott.m`.

3.12 Weibullfordeling

Forskjellig parametrisering i W.M.M.Y. og Matlab

Tetthetsfunksjonen til Weibullfordelingen er

$$f(x; \alpha, \beta) = \alpha \beta x^{\beta-1} e^{-\alpha x^\beta}, \quad x > 0,$$

hvor $\alpha > 0$, $\beta > 0$. Funksjonen som brukes i Matlab er

$$\tilde{f}(x|a, b) = \frac{b}{a} \left(\frac{x}{a}\right)^{b-1} e^{-\frac{x^b}{a}}.$$

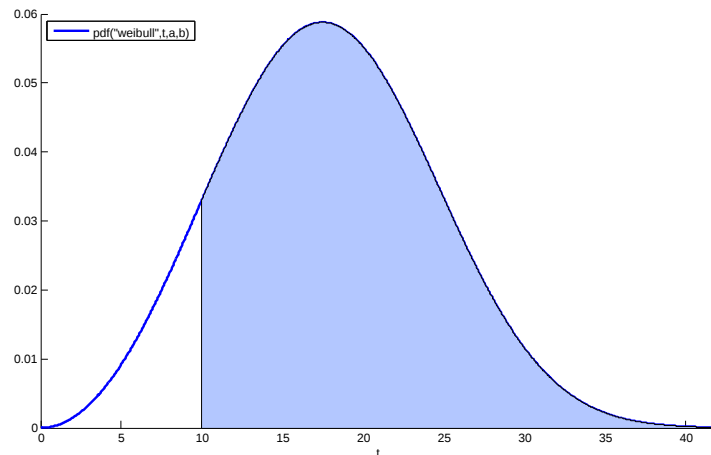
Velger en $a = \frac{1}{\alpha}$ og $b = \beta$, så er $\tilde{f}(x|a, b) = f(x; \alpha, \beta)$.

Eksempel: Hvis levetiden til en komponent følger en Weibullfordeling med parametre $\alpha = 0.05$ og $\beta = 3$, hva er sannsynligheten for at komponenten svikter ved tiden $t = 10$ eller senere?

```
>> alfa = 0.05; beta = 3;
>> a = 1/alfa; b = beta;
>> 1 - cdf('weibull',10,a,b)
```

```
ans =
```

```
0.8825
```



Figur 9 – Arealet av det skraverte området er $1 - F(10)$, der $F(t)$ er den kumulative fordelingsfunksjonen til Weibullfordelingen med $\alpha = 0.05$ og $\beta = 3$. Figuren er laget med `weibullPlott.m`.

4 Deskriptiv statistikk

Matlab har mange funksjoner for å beskrive en samling data, skriv `help stats` og se under *Descriptive Statistics* for en oversikt. Noen av de viktigste er mål for sentralitet og spredning, som `mean`, `median` og `mode` for å beregne henholdsvis midelverdien, medianen og typetallet til dataene, samt `var` og `std` som beregner empirisk varians og standardavik.

4.1 Histogram

For å visualisere hvordan dataene er fordelt er et histogram nyttig. Hvis vektoren `X` inneholder elementene $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$, så vil `hist(X,M)` dele intervallet $[x_{(1)}, x_{(n)}]$ inn i M like store underintervaller, og telle antall observasjoner som faller i hvert underintervall for så å plote resultatet i et søylediagram. `hist(X)` gir samme resultat som `hist(X,10)`. Ønsker man bare tallene uten plottet, kan eventuelt `histc` brukes i stedet. Det er også mulig å spesifisere underintervallene selv, skriv `help hist` for flere detaljer.

Eksempel: Trekker hundre tall fra standard normalfordelingen og teller hvor mange som faller i intervallene $(-\infty, -1.96)$, $[-1.96, 1.96]$ og $[1.96, \infty)$.

```
>> X = normrnd(0,1,1,100);
>> C = histc(X,[-inf -0.96 0.96 inf])
```

C =

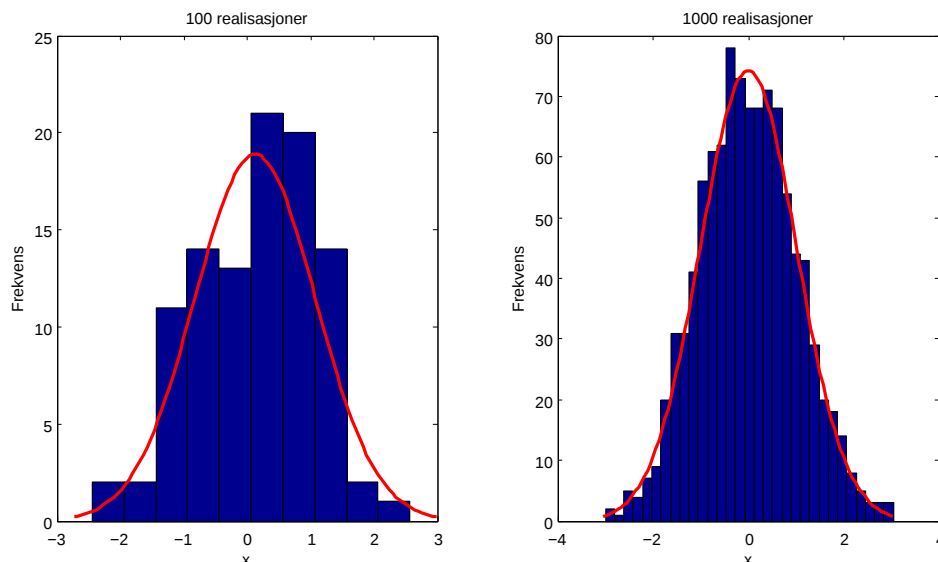
```
14    73    13     0
```

Det siste elementet i `C` er null siden ingen av obserasjonene har verdien `inf`.

Som alternativ til `hist` og `histc` kan også `histfit` brukes. En får da samme histogram som med `hist`, men i tillegg plottes en skalert sannsynlighetstetthetsfunksjon for en spesifisert fordeling, slik at en kan sammenlikne.

Eksempel: Trekker tusen realisasjoner fra en standard normalfordeling, og bruker `histfit` på de første hundre av dem, og deretter på alle. Plottene er vist i fig. 10.

```
X = random('normal',0,1,N,1);
histfit(X(1:n1),nbins1,'normal')
histfit(X(1:n2),nbins2,'normal')
```



Figur 10 – `histfit` anvendt på 100 (venstre) og 1000 (høyre) realisasjoner trukket fra en standard normalfordeling. Figuren er laget med `histfitplot.m`.

4.2 Box Plot

En annen måte å visualisere data på er box plot, som er spesielt nyttig for å sammenlikne flere sett data. `boxplot(X)` lager et box plot med en boks for hver kolonne i matrisen `X`. Boksene strekker seg fra første kvartil, eller 25-prosent kvantilen, Q_1 , til tredje kvartil, eller 75-prosent kvantilen, Q_3 , og har en tversgående linje som markerer medianen. De har også utstikkere som går ut til de mest ekstreme observasjonene som ligger nærmere boksen enn 1.5 ganger kvartildifferansen $Q_3 - Q_1$.

Eksempel: Anta at vi ønsker å sammenlikne to sett med observasjoner, lagret i vektorene `X` og `Y`. Hvis vektorene har like mange elementer, kan vi skrive `boxplot(X,Y)`. Hvis vektorene har forskjellig antall elementer, kan vi sette dem sammen til én vektor, og bruke en grupperingsvariabel.

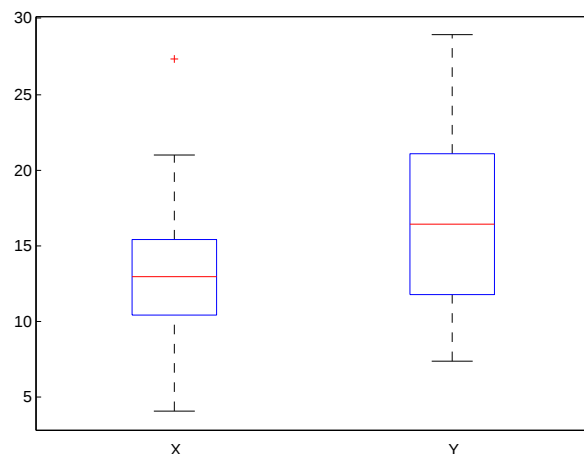
```
X = random('chisquare',v1,n,1);
Y = random('chisquare',v2,m,1);
G = [ repmat('X',n,1); repmat('Y',m,1) ];
boxplot([X; Y],G)
```

Det resulterende plottet er vist i fig. 11

4.3 Q-Q plott

Funksjonen `qqplot` har to viktige anvendelser:

1. Sjekke om et sett data er normalfordelt ved å plote kvantilene til dataene mot kvantilene til standard normalfordelingen

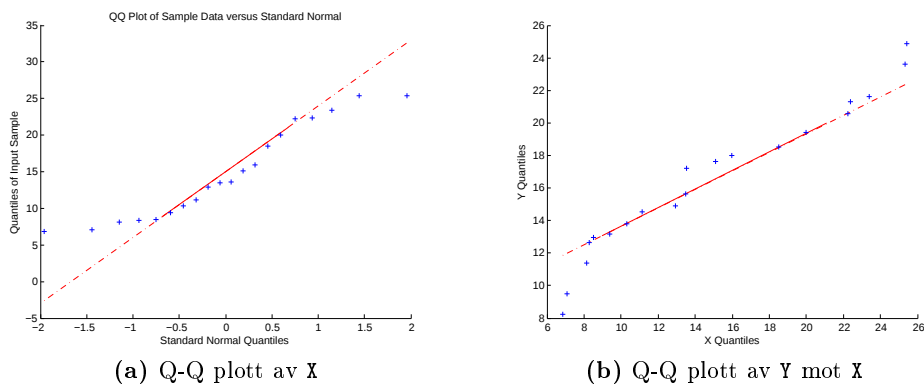


Figur 11 – Box plot av de kji-kvadratfordelte dataene i X og Y . Figuren er laget med `boksploott.m`.

2. Sjekke om to sett data er likt fordelte ved å plotte kvantilene til det ene settet mot kvantilene til det andre

I begge tilfeller er kriteriet rettlinjethet. Se forøvrig eget notat om Kvantil- og Q-Q plott.

Eksempel: Betrakt vektorene X og Y fra forrige eksempel. Vi kan skrive `qqplot(X)` for å sjekke om dataene i X er normalfordelte. Krumningen i plottet (fig. 12a), tydeligst for lave kvantiler, tyder på at dataene ikke er normalfordelte. For å vurdere om dataene i X og Y er likt fordelte kan vi bruke `qqplot(X,Y)`, som gir plottet i fig. 12b. I dette tilfellet er begge settene samplet fra kji-kvadratfordelingen, så plottet ser noenlunde rettlinjet ut.



Figur 12