

Introduksjon til bruk av Matlab i TMA4240 Statistikk – Høst 2016

Mandag 29. august og torsdag 1. september

Del A: Bruk av Matlab - hva er de ulike vinduene?

Når du åpner Matlab vises (sannsynligvis) følgende vinduer:

- Tekstredigeringsvinduet (Editor)
- Kommandovinduet (Command window)
- Oversikt over definerte variable (Workspace)
- Oversikt over innholdet i arbeidskatalogen (Current Folder)

Velg arbeidskatalog (working directory) ved hjelp av “adresselinja” over editor-vinduet. Opprett et nytt Matlab skript ved å høyreklikke i Current folder-vinduet, og velge *New File* ► *Script*.

Del B: Prøve ut Matlab-kommandoer - en forvokst kalkulator?

Skriv inn kommandoene under i skriptet, og utfør dem. Dette kan gjøres ved å markere kommandoene som skal utføres, høyreklikke på den markerte teksten, og velge *Evaluate Selection*. Alternativt kan man kjøre hele skriptet ved å trykke på *Run*-knappen øverst i Matlab-vinduet.

```
2 + 3
2*6
3*10^4 - 3*5^2
sqrt(9)
log(3)
help log
help log10
log10(500)
exp(34)
gamma(3)
factorial(2)
nchoosek(10,4)
1:4
[1,2,3,4]
sum(1:5)
heights = [192,185,174,195,173]
shoes = [46,43,40,45,40]
heights/shoes
heights./shoes
```

Del C: Import og eksport - data inn, figurer ut

Legg tekstfila `trondheim.txt` i arbeidskatalogen. Last så inn dataene med kommandoen

```
trondheim = load('trondheim.txt');
```

hvor semikolonet gjør at innholdet i `trondheim`-variabelen ikke skrives til skjermen. Alternativt kan man høyreklikke på tekstfila i Current Folder-vinduet og velge *Import Data...* for å åpne dialogvinduet for å importere data fra en fil. Velg så *Numeric Matrix* som format og trykk på den grønne haken for å importere dataene. Dataene finnes nå i matrisen `trondheim`, som har 365 rader og 7 kolonner. Temperaturobservasjonene er i kolonne 6. Isoler temperaturobservasjonene i en egen vektor med kommandoen

```
temp = trondheim(:,6);
```

og plott dem med

```
plot(temp)
```

Marker figuraksene med

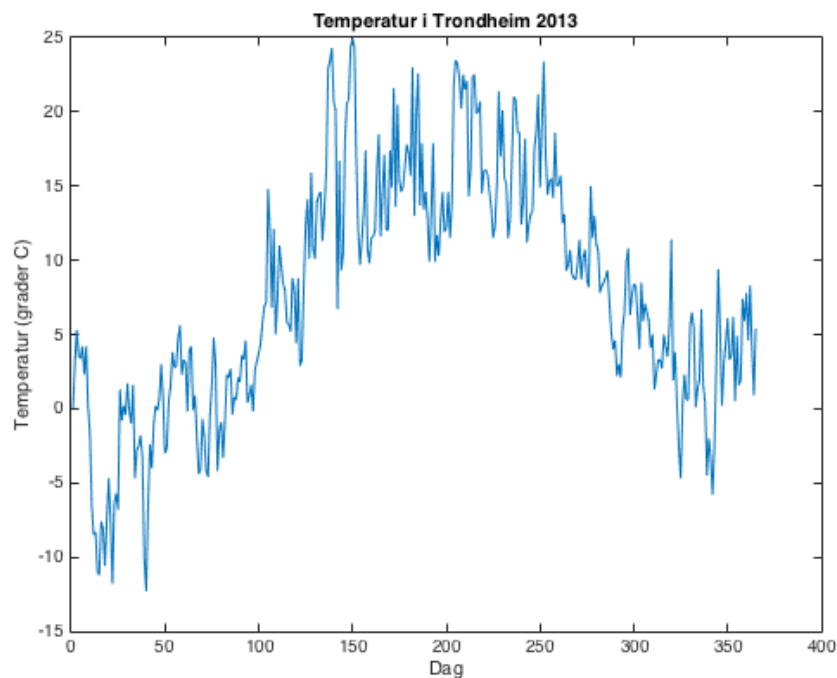
```
xlabel('Dag')
```

```
ylabel('Temperatur (grader C)')
```

og legg til en figurttittel med

```
title('Temperatur i Trondheim 2013')
```

For å lagre figuren, trykk på *File ► Save As...* i figurvinduet. Velg deretter ønsket filformat (f.eks. *.png*), gi fila et navn (f.eks. *trondheimtemp*) og trykk *Save*. Figuren bør se omtrent slik ut:



Del D: Innebygde funksjoner

Funksjonen `normrnd` kan brukes til å generere uavhengige realisasjoner fra normalfordelingen. Test ut funksjonen ved å skrive inn og utføre følgende kommandoer.

```
normrnd(0,1)
normrnd(0,1,1)
normrnd(0,1,2)
normrnd(0,1,2,1)
normrnd(0,1,1,2)
help normrnd
```

Generer tusen realisasjoner fra normalfordelingen, lagre dem i en vektor, og plott et histogram av dem.

```
x = normrnd(0,1,1000,1);
histogram(x);
xlabel('x')
ylabel('Frekvens')
```

Plott histogrammet og kurven til sannsynlighetstettheten i samme figur.

```
histogram(x,'normalization','pdf')
hold on
xx = linspace(-3,3,100);
yy = normpdf(xx,0,1);
plot(xx,yy,'LineWidth',2)
xlabel('x')
ylabel('Frekvens')
```

De ekstra argumentene i `histogram`-funksjonen gjør at søylene i histogrammet skales, slik at histogrammet blir sammenlignbart med sannsynlighetstettheten. Linjetykkelse 2 i `plot`-kommandoen gir en tydelig kurve, som er lett å se selv med histogrammet som bakgrunn. Funksjonen `linspace(-3,3,100)` brukes for å lage en vektor med 100 verdier jevnt fordelt mellom endepunktene -3 og 3 .

Del E: Egendefinerte funksjoner

Opprett en ny Matlab-fil og gi den navnet `minfunksjon.m`. Skriv så inn følgende kode, og lagre fila.

```
function y = minfunksjon(x)
y1 = x;
y2 = x.^2;
y = max(y1,y2);
end
```

Denne filen definerer nå funksjonen `minfunksjon` som tar et tall x som input. Funksjonen returnerer x hvis $x \geq x^2$ og den returnerer x^2 dersom $x^2 > x$. Prøv å kalle funksjonen med forskjellige x -verdier.

```
minfunksjon(1)
minfunksjon(2)
minfunksjon(-1)
minfunksjon(-2)
```

Plott grafen til funksjonen for x -verdier mellom -1 og 2 .

```
xx = linspace(-1,2,100);  
yy = minfunksjon(xx);  
plot(xx,yy)  
xlabel('x')  
ylabel('y')
```

Del F: Matriser og vektorer

Generer en matrise med normalfordelte tall:

```
A = normrnd(0,1,10,6)
```

Sett de negative elementene av A lik 0 .

```
A(A<0) = 0;
```

Lag en ny matrise B av de midterste elementene av A , og lag enda en matrise C av det nederste, høyre hjørnet av A .

```
B = A(4:7,3:4)
```

```
C = A(8:end,5:end)
```

Regn ut matriseproduktet av B og den transponerte til C .

```
B*C'
```

Skriv inn høydene og skostørrelsene fra del B i hver sin kolonnevektor.

```
heights = [192,185,174,195,173]'  
shoes = [46,43,40,45,40]'
```

Bruk minste kvadraters metode for å estimere proporsjonalitetskonstanten. Det vil si, sett opp likningen $\mathbf{a}^T \mathbf{a} x = \mathbf{a}^T \mathbf{b}$ hvor \mathbf{a} er `heights` og \mathbf{b} er `shoes`, og løs for x .

```
x = (heights'*heights)\(heights'*shoes);
```

Plott `shoes` som en funksjon av `heights`. Plott også `x*heights` som en funksjon av `heights` i samme figur.

```
plot(heights,shoes,'o')  
hold on  
plot(heights,x*heights,'-')  
xlabel('heights')  
ylabel('shoes')
```

Lag to kolonnevektorer med 12 normalfordelte elementer i hver.

```
u = normrnd(0,1,12,1);  
v = normrnd(0,1,12,1);
```

Regn ut skalarproduktet av de to vektorene.

```
u'*v;
```

Regn ut det elementvise produktet av de to vektorene.

```
u.*v
```