MATLAB for STK1100

Matematisk institutt
Univeristetet i Oslo
Januar 2014

1 Enkel generering av stokastiske variabler

MATLAB har et stort antall funksjoner for å generere tilfeldige tall. Skriv `help stats` for å se en liste av alle statistikk-relaterte funksjoner. Funksjonene som genererer tilfeldige tall står under overskriften Random Number Generators.

For eksempel, for å simulere terningkast kan en bruke `unidrnd` som trekker fra en diskret uniform fordeling. 10 terningkast kan simuleres med

```
>> x = unidrnd(6, [1, 10])

x =
     3     6     2     3     6     4     6     1     3
4
```

Det første argumentet (`6`) angir øvre grense for tallene som det skal trekkes fra – i dette tilfellet $\{1, 2, \ldots, 6\}$ og det andre argumentet `[1,10]` angir at vi ønsker 10 elementer (eller mer presist: `1` rad og `10` kolonner).

Vi kan finne de kastene som ga en sekser ved å bruke for eksempel

```
>> x == 6

ans =

     0     1     0     0     1     0     1     0     0
0
```

Her er det et ett-tall for alle kast som ga en sekser og en null alle andre steder. Vi kan finne indeksene til kastene som ga sekser ved å bruke

```
>> find(x == 6)

ans =

     2     5     7
```

og fra dette kan vi finne hvor mange kast som ble utført fra vi fikk en sekser til neste gang vi fikk en sekser, ved å bruke

```
>> diff(find(x == 6))

ans =

     3     2
```

Disse verdiene kan også sees direkte fra sekvensen av nuller og enere over. Det er to nuller (altså to kast som ikke ga sekser) mellom de to første ett-tallene, og det er en null (altså ett kast som ikke ga sekser) mellom de to siste ett-tallene.

2 Deskriptiv statistikk

MATLAB har en rekke funksjoner for deskriptiv statistikk. Her er noen av dem

`sum`	sum	`min`	minimum
`mean`	gjennomsnitt	`max`	maximum
`var`	varians	`median`	median
`std`	standardavvik	`iqr`	kvartildifferans
`cov`	kovarians	`range`	variansbredde
`corr`	korrelasjon	`prctile`	persentil (kvantil)

Se i utskriften for `help stats` under overskriften Descriptive Statistics for flere detailjer. Navnene på funksjonene for statistiske fordelinger består alle av et prefiks som angir fordelingen (`unid` for diskret uniform fordeling, `norm` for normalfordeling, `hyge` for hypergeometrisk fordeling, osv.) og et suffiks som angir hva en ønsker fra fordelingen (`rnd` for tilfeldige tall, `pdf` for sannsynlighetstetthet, `cdf` for kumulativ fordeling, `inv` for invers kumulativ fordeling, osv.).

Dette genererer 1000 standard normalfordelte variabler og beregner deres gjennomsnitt og standardavvik

```
>> x = normrnd(0, 1, [1, 1000]);
>> m = mean(x)

m =
    0.0455

>> s = std(x)

s =
    1.0313
```

3 Andre funksjoner for vektorer

MATLAB har en del nyttige funksjoner for behandling av vektorer. Her er noen av dem.

`sum`	sum	`min`	minimum
`prod`	produkt	`max`	maximum
`cumsum`	kumulativ sum	`sort`	sortering
`cumprod`	kumulativt produkt	`find`	finn indekser for sanne verdier
`length`	antall elementer	`diff`	differens av påfølgende verdier

Se `help datafun` for flere funksjoner og detaljer.

4 Plotting

MATLAB har et stort utvalg av funksjoner for plotting. De mest vanlige er

`plot`	plotte linjer, kurver og punkter
`hist`	histogram (også `histc`)
`bar`	søylediagram
`mesh`	3-dimensjonalt gitterplott
`surf`	3-dimensjonalt flateplott

Her er et eksempel som plotter tettheten for en standard normalfordeling som en rød heltrukken linje

```
>> x = -4 : 0.1 : 4;                    % lag vektor med x-verdier
>> y = normpdf(x);                      % lag vektor med y-verdier
>> plot(x, y, 'r-');                    % plott med r{\o}d linje
>> xlabel('x'); ylabel('tetthet');      % sett merkelapper p{\aa} akser
>> title('Standard normalfordling');    % sett tittel p{\aa} figuren
>> box on                               % sett ramme rundt figuren
>> grid on                              % legg til stiplete linjer
```

Og her er et eksempel som plotter sannsynligheten for å få x rette i lotto dersom en satser 8 rekker. Dette kan sammenlignes med at vi har en urne med 34 baller hvorav 7 er røde og resten hvite. Hvis vi trekker ut 8 baller, hvor stor er sannsynligheten for at x av disse er røde?

```
>> m = 34;                              % antall baller i alt
>> n = 7;                               % antall r{\o}de baller
>> k = 8;                               % antall som trekkes ut
>> x = 0 : 7;                           % antall r{\o}de blant de som ble trukk
>> y = hygepdf(x, m, k, n);             % beregn sannsynligheter
>> bar(x, y);                           % lage s{\o}ylediagram
>> xlabel('k'); ylabel('sannsynlighet');
>> title('Sannsynlighet for x rette');
```

Kommentar-tegnet i MATLAB er `%`.

Andre funksjoner relatert til figurer er

`figure`	åpne et figurvindu
`clf`	lukk gjeldende figurvindu
`close`	lukk angitt figurvindu
`print`	skriv ut eller lagre figuren

For å kontrollere plottingen finnes andre nyttige funksjoner

`axis`	juster aksene	`title`	lag en tittel
`hold`	overskriv plott, ikke erstatt	`xlabel`	sett merkelapp på x-aksen
`grid`	legg linjer i aksesystemet	`ylabel`	sett merkelapp på y-aksen
`box`	lag en boks rundt aksesystemet	`zlabel`	sett merkelapp på z-aksen

For å sette tekst i en figur brukes for eksempel `text` og `gtext`.

Se hjelpeteksten for de respektive funksjonene for mer informasjon om hvordan de brukes, for eksempel `help plot`

5 Lesing og skriving av data til fil

Den mest brukte funksjonen for å lese data fra fil er `load`. Den kan lese vanlige tekstfiler med tall og filer i MATLABs eget format (filer som slutter på `.mat`). Hvis vi har en fil som heter `data.txt` og som inneholder for eksempel

```
40.6   93.5   91.7   41.0   89.4
 5.8   35.3   81.3    1.0   13.9
20.3   19.9   60.4   27.2   19.9
```

så kan denne leses inn i MATLAB og lagres i variabelen `X`, ved å bruke

```
>> X = load('data.txt');
```

For å skrive data til fil kan en bruke `save`. Funksjonen `save` lagrer vanligvis data i MATLABs eget format, men en kan lagre i tekstformat ved å kalle `save` med `-ascii` til slutt. For eksempel, dersom en vil lagre dataene i variabelen `X` til tekstfilen `data.txt`, så kan en bruke

```
>> save data.txt X -ascii
```

Se `help save` for flere detaljer.

6 Programfiler

Det kan ofte være greit å putte kommandoer i en fil dersom de skal kjøres om og om igjen med bare små endringer mellom hver gang. Vi kan for eksempel lage en fil som heter `prog.m` som inneholder

```
n = 100;                          % antall tilfeldige tall
x = normrnd(0, 1, [1, n]);        % generer tilfeldige tall
hist(x)                           % plott histogram
```

Denne filen må ligge slik at MATLAB finner den. Det enkleste er å legge filen på det samme filområdet som MATLAB ble startet fra. I Unix vil MATLAB alltid finne filene som ligger på filområdet `~/matlab/`, så det kan være et greit sted å legge slike filer.

For å utføre funksjonene i filen `prog.m` skriver en

```
>> prog
```

og så vil MATLAB utføre innholdet av `prog.m`. For å se innholdet av en fil i MATLAB bruk funksjonen `type`, for eksempel `type prog.m`.

En kan også legge filer på andre filområder, men da må filområdene legges til MATLABs søkesti. Dette kan for eksempel gjøres ved å bruke funksjonen `addpath` (se avsnittet nedenfor om oppstartfil).