

Dalam buku catatan ini, kami menunjukkan plot statistik utama, uji, dan distribusi dalam Euler.

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistika. Jadi, Anda mungkin memerlukan beberapa pengetahuan dasar untuk memahami rincian tersebut.

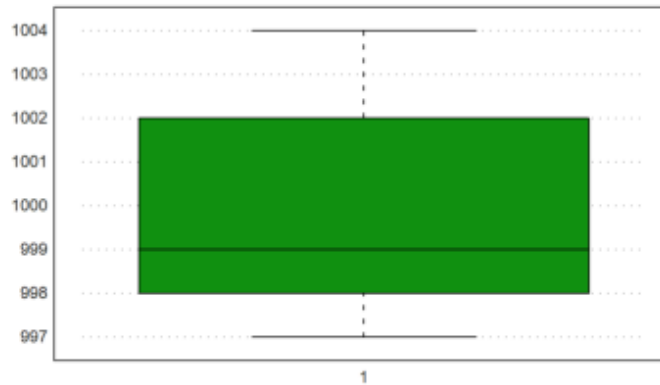
Anggaplah pengukuran berikut. Kami ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
>M=[1000,1004,998,997,1002,1001,998,1004,998,997]; ...  
>median(M), mean(M), dev(M),
```

```
999  
999.9  
2.72641400622
```

Kita dapat membuat plot diagram kotak dan garis (box-and-whiskers) untuk data ini. Dalam kasus kita, tidak ada data yang berada di luar jangkauan (outliers).

```
>aspect(1.75); boxplot(M):
```



Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai yang diukur berasal dari distribusi normal.

Semua fungsi distribusi dalam Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPD).

$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-m}{d}\right)^2} dt.$$

Kami mencetak hasilnya dalam bentuk persen dengan akurasi dua digit desimal menggunakan fungsi print.

```
>print((1-normaldis(1005,mean(M),dev(M)))*100,2,unit=" %")
```

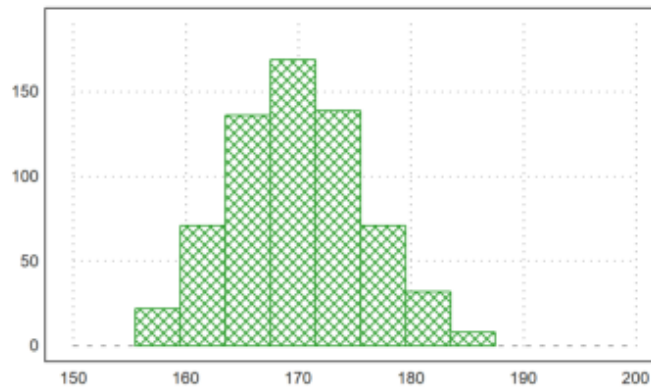
3.07 %

Untuk contoh berikutnya, kami mengasumsikan jumlah pria dalam rentang ukuran yang diberikan sebagai berikut.

```
>r=155.5:4:187.5; v=[22,71,136,169,139,71,32,8];
```

Berikut adalah plot dari distribusinya.

```
>plot2d(r,v,a=150,b=200,c=0,d=190,bar=1,style="\/"):
```



Kita dapat memasukkan data mentah seperti ini ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita seharusnya memiliki tiga kolom: Awal rentang, akhir rentang, jumlah pria dalam rentang tersebut.

Tabel dapat dicetak dengan judul. Kami menggunakan vektor string untuk menetapkan judul kolom.

```
>T:=r[1:8]' | r[2:9]' | v'; writetable(T,labc=["BB","BA","Frek"])
```

BB	BA	Frek
155.5	159.5	22
159.5	163.5	71
163.5	167.5	136
167.5	171.5	169
171.5	175.5	139
175.5	179.5	71
179.5	183.5	32
183.5	187.5	8

Jika kita membutuhkan nilai rata-rata dan statistik lainnya dari ukuran tersebut, kita perlu menghitung titik tengah dari rentang tersebut. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk ini.

Simbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" digunakan untuk menentukan judul kolom.

```
>(T[,1]+T[,2])/2 // nilai tengah dari tiap interval
```

```
157.5  
161.5  
165.5  
169.5  
173.5  
177.5  
181.5  
185.5
```

Namun, lebih mudah untuk menjumlahkan rentang tersebut dengan vektor $[1/2, 1/2]$.

```
>M=fold(r,[0.5,0.5])
```

```
[157.5, 161.5, 165.5, 169.5, 173.5, 177.5, 181.5, 185.5]
```

Sekarang kita dapat menghitung nilai rata-rata dan deviasi dari sampel dengan frekuensi yang diberikan.

```
>{m,d}=meandev(M,v); m, d,
```

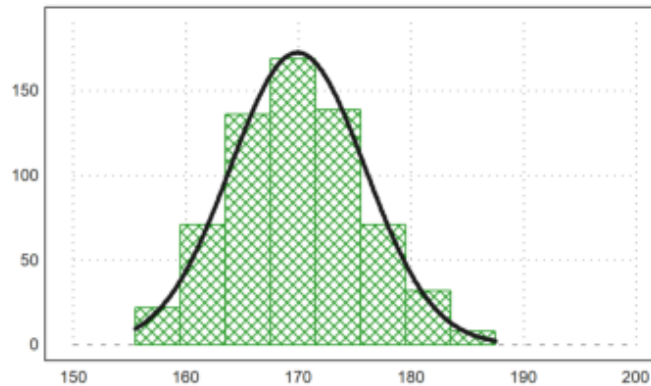
```
169.901234568  
5.98912964449
```

Mari tambahkan distribusi normal dari nilai-nilai tersebut ke plot batang di atas. Rumus untuk distribusi normal dengan rata-rata m dan deviasi standar d adalah:

$$y = \frac{1}{d\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2d^2}}.$$

Karena nilai-nilainya berada di antara 0 dan 1, untuk memplotnya pada diagram batang, nilai tersebut harus dikalikan dengan 4 kali jumlah total data.

```
>plot2d("qnormal(x,m,d)*sum(v)*4", ...  
> xmin=min(r),xmax=max(r),thickness=3,add=1):
```



Contoh Soal

1) Tentukan median, mean, deviasi, dan distribusi kumulatif normal (CDF) dari data tersebut.

$N = [199, 200, 201, 202, 203, 204, 205, 205, 206, 207, 208, 208, 209]$

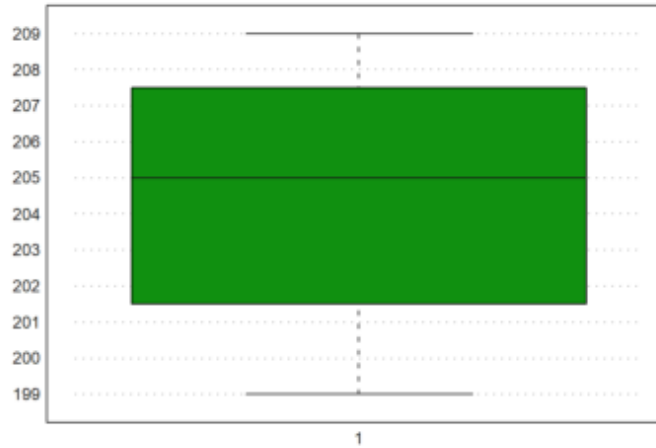
```
>N=[199, 200, 201,202,203,204,205,205,206,207,208,208,209]
```

```
[199, 200, 201, 202, 203, 204, 205, 205, 206, 207, 208, 208,  
209]
```

```
>median(N), mean(N), dev(N),
```

```
205  
204.384615385  
3.22847904176
```

```
>aspect(1.5); boxplot(N):
```

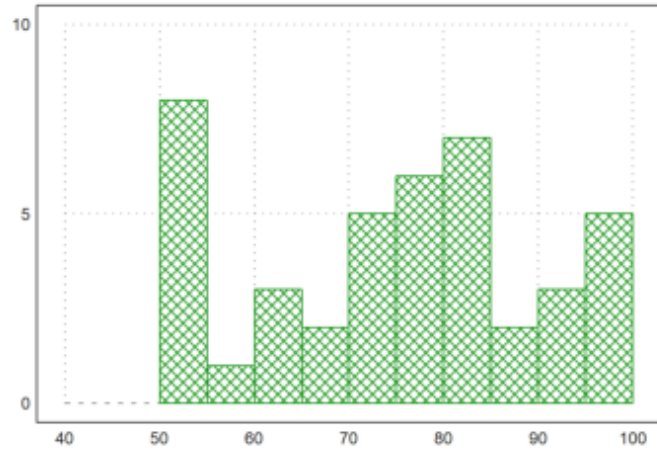


```
>print((1-normaldis(199,mean(N),dev(N)))*100,2,unit=" %")
```

95.23 %

2) Buatlah plot dari jumlah siswa dalam rentang yang diberikan sebagai berikut.

```
>r=50:5:100; v=[8,1,3,2,5,6,7,2,3,5];  
>plot2d(r,v,a=40,b=100,c=0,d=10,bar=1,style="\/"):
```

Di direktori buku catatan ini, Anda akan menemukan sebuah file berisi tabel. Data tersebut mewakili hasil dari sebuah survei. Berikut adalah empat baris pertama dari file tersebut. Data ini berasal dari sebuah buku online Jerman "Einführung in die Statistik mit R" oleh A. Handl.

```
>printfile("table.dat",4); //function printfile (filename, lines)
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
  open(filename,"r");
```

Tabel tersebut berisi 7 kolom angka atau token (string). Kami ingin membaca tabel dari file tersebut. Pertama, kami akan menggunakan terjemahan kami sendiri untuk token-token tersebut.

Untuk melakukan ini, kami mendefinisikan set token-token tersebut. Fungsi `strtokens()` mendapatkan vektor string token dari sebuah string yang diberikan.

```
>mf:=["m","f"]; yn:=["y","n"]; ev:=strtokens("g vg m b vb");
```

Sekarang kita akan membaca tabel dengan terjemahan ini.

Argumen tok2, tok4, dan sebagainya adalah terjemahan dari kolom-kolom tabel. Argumen-argumen ini tidak ada dalam daftar parameter readtable(), sehingga Anda perlu menyediakannya dengan ":=".

```
>{MT,hd}=readtable("table.dat",tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;
```

```
>load over statistics;
```

Untuk mencetak, kita perlu menentukan set token yang sama. Kita hanya akan mencetak empat baris pertama.

```
>writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,tok7:=yn);
```

```
MT is not a variable!
Error in:
writetable(MT[1:10],labc=hd,wc=5,tok2:=mf,tok4:=yn,tok5:=ev,to ...
^
```

Titik "." mewakili nilai yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan sebelumnya, kita hanya perlu menentukan kolom-kolom yang berisi token, bukan angka..

```
>ctok=[2,4,5,7]; {MT,hd,tok}=readtable("table.dat",ctok=ctok);
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;
```

Fungsi readtable() sekarang mengembalikan sebuah set token.

```
>tok
```

```
Variable tok not found!
Error in:
tok ...
  ^
```

Tabel tersebut berisi entri dari file dengan token diterjemahkan menjadi angka.

String khusus NA="." diartikan sebagai "Not Available" (Tidak Tersedia), dan diubah menjadi NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA dan NAval.

```
>MT[1]
```

```
MT is not a variable!  
Error in:  
MT[1] ...  
      ^
```

Berikut adalah konten tabel dengan angka yang tidak diterjemahkan.

```
>writetable(MT,wc=5)
```

```
Variable or function MT not found.  
Error in:  
writetable(MT,wc=5) ...  
      ^
```

Untuk kenyamanan, Anda dapat menyimpan output dari readtable() ke dalam sebuah daftar (list).

```
>Table={{readtable("table.dat",ctok=ctok)}};
```

```
Could not open the file
table.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
  if filename!=none then open(filename,"r"); endif;
```

Dengan menggunakan kolom-kolom token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel tersebut. Kita dapat menentukan ctok, tok, dll., atau menggunakan daftar (list) Table.

```
>writetable(Table,ctok=ctok,wc=5);
```

```
Variable or function Table not found.
Error in:
writetable(Table,ctok=ctok,wc=5); ...
      ^
```

Fungsi `tablecol()` mengembalikan nilai-nilai dari kolom-kolom tabel, melewati baris-baris dengan nilai NAN (".." dalam file), serta indeks-indeks kolom yang berisi nilai-nilai tersebut.

```
>{c,i}=tablecol(MT,[5,6]);
```

```
Variable or function MT not found.  
Error in:  
{c,i}=tablecol(MT,[5,6]); ...  
                ^
```

Kita dapat menggunakan ini untuk mengekstrak kolom-kolom dari tabel untuk membuat tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],lab=hd[j],ctok=[2],tok=tok)
```

```
Variable or function i not found.  
Error in:  
j=[1,5,6]; writetable(MT[i,j],lab=hd[j],ctok=[2],tok=tok) ...  
                        ^
```

Tentu saja, dalam hal ini, kita perlu mengekstrak tabel itu sendiri dari daftar (list) Table.

```
>MT=Table[1];
```

```
Table is not a variable!  
Error in:  
MT=Table[1]; ...  
      ^
```

Tentu saja, kita juga dapat menggunakannya untuk menentukan nilai rata-rata dari suatu kolom atau nilai statistik lainnya.

```
>mean(tablecol(MT,6))
```

```
Variable or function MT not found.  
Error in:  
mean(tablecol(MT,6)) ...  
      ^
```


Fungsi `getstatistics()` mengembalikan elemen-elemen dalam bentuk vektor, beserta jumlah kemunculan-nya. Kita menggunakannya untuk nilai "m" dan "f" dalam kolom kedua tabel kita.

```
>{xu,count}=getstatistics(tablecol(MT,2)); xu, count,
```

```
Variable or function MT not found.  
Error in:  
{xu,count}=getstatistics(tablecol(MT,2)); xu, count, ...  
^
```

Kita dapat mencetak hasilnya dalam bentuk tabel baru.

```
>writetable(count',labr=tok[xu])
```

```
Variable count not found!  
Error in:  
writetable(count',labr=tok[xu]) ...  
^
```

Fungsi `selecttable()` mengembalikan tabel baru dengan nilai-nilai dalam satu kolom yang dipilih dari vektor indeks. Pertama, kita mencari indeks dua nilai dalam tabel token kita.

Catatan tambahan:

Fungsi `indexof(v, x)` berarti bahwa mencari `x` dalam vektor `v`.

```
>v:=indexof(tok,["g","vg"])
```

```
Variable or function tok not found.  
Error in:  
v:=indexof(tok,["g","vg"]) ...  
          ^
```

Sekarang kita dapat memilih baris-baris tabel yang memiliki salah satu dari nilai-nilai dalam vektor `v` di kolom kelima mereka.

```
>MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5);
```

```
Variable or function MT not found.  
Error in:  
MT1:=MT[selectrows(MT,5,v)]; i:=sortedrows(MT1,5); ...  
          ^
```

Sekarang kita dapat mencetak tabel dengan nilai-nilai yang diekstrak dan diurutkan dalam kolom kelima.

```
>writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7);
```

```
Variable or function i not found.  
Error in:  
writetable(MT1[i],labc=hd,ctok=ctok,tok=tok,wc=7); ...  
^
```

Untuk statistik selanjutnya, kita ingin mengaitkan dua kolom dari tabel. Jadi, kita akan mengekstrak kolom 2 dan 4, lalu mengurutkan tabelnya.

```
>i=sortedrows(MT,[2,4]); ...  
> writetable(tablecol(MT[i],[2,4])',ctok=[1,2],tok=tok)//Tabel diurutkan secara leksikografis.
```

```
Variable or function MT not found.  
Error in:  
i=sortedrows(MT,[2,4]); writetable(tablecol(MT[i],[2,4])',c ...  
^
```

Dengan menggunakan fungsi `getstatistics()`, kita dapat menghubungkan jumlah kemunculan dalam dua kolom tabel satu sama lain.

```
>MT24=tablecol(MT,[2,4]); ...  
>{xu1,xu2,count}=getstatistics(MT24[1],MT24[2]); ...  
>writetable(count,labr=tok[xu1],labc=tok[xu2])
```

```
Variable or function MT not found.  
Error in:  
MT24=tablecol(MT,[2,4]); {xu1,xu2,count}=getstatistics(MT24[1] ...  
^
```

Sebuah tabel dapat ditulis ke dalam sebuah file.

```
>filename="test.dat"; ...  
>writetable(count,labr=tok[xu1],labc=tok[xu2],file=filename);
```

```
Variable or function count not found.  
Error in:  
filename="test.dat"; writetable(count,labr=tok[xu1],labc=tok[x ...  
^
```

Kemudian kita dapat membaca tabel dari file tersebut.

```
>{MT2,hd,tok2,hdr}=readtable(filename,>clabs,>rlabs); ...  
>writetable(MT2,labr=hdr,labc=hd)
```

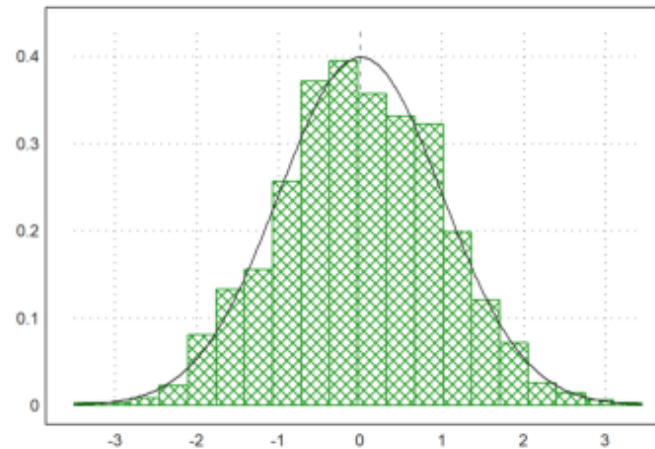
```
Could not open the file  
test.dat  
for reading!  
Try "trace errors" to inspect local variables after errors.  
readtable:  
    if filename!=none then open(filename,"r"); endif;
```

Dan menghapus file tersebut.

```
>fileremove(filename);
```

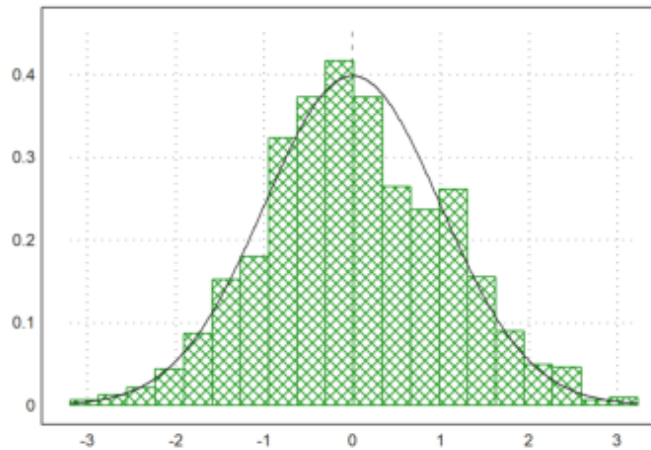
Dengan plot2d, ada metode yang sangat mudah untuk memplot distribusi data eksperimental.

```
>p=normal(1,1000); //1000 sampel acak yang terdistribusi normal p  
>plot2d(p,distribution=20,style="/"); // plot sampel acak p  
>plot2d("qnormal(x,0,1)",add=1): // tambahkan plot distribusi normal standar
```



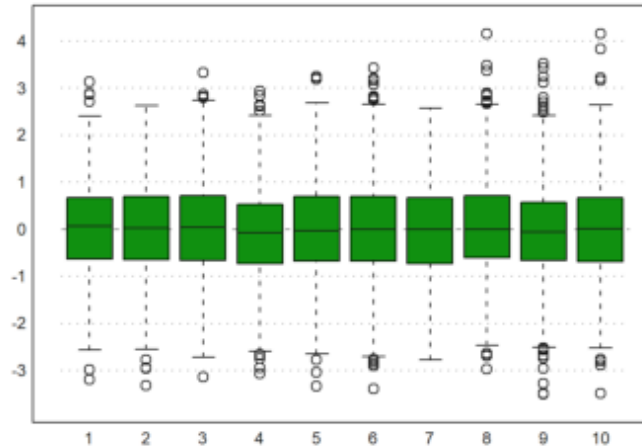
Harap perhatikan perbedaan antara diagram batang (sampel) dan kurva normal (distribusi sebenarnya). Silakan masukkan kembali tiga perintah tersebut untuk melihat hasil pengambilan sampel yang lain.

```
>p=normal(1,1000);  
>plot2d(p,distribution=20,style="\");  
>plot2d("qnormal(x,0,1)",add=1):
```



Berikut adalah perbandingan dari 10 simulasi 1000 nilai yang terdistribusi secara normal menggunakan diagram kotak (box plot). Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta nilai-nilai yang berada di luar jangkauan (outliers).

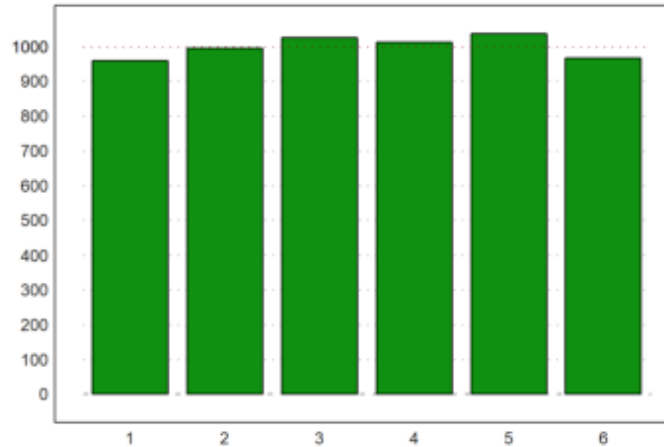
```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki fungsi `intrandom`. Mari kita simulasi lemparan dadu dan plot distribusinya.

Kita akan menggunakan fungsi `getmultiplicities(v, x)` yang menghitung seberapa sering elemen-elemen dari `v` muncul dalam `x`. Kemudian, kita akan memplot hasilnya menggunakan fungsi `columnplot()`.

```
>k=intrandom(1,6000,6); ...
>columnplot(getmultiplicities(1:6,k)); ...
>ygrid(1000,color=red):
```

Catatan tambahan :

`inrandom(n, m, k)`: Matriks dari variabel acak

`getmultiplicities` : Menghitung seberapa sering elemen-elemen dari `x` muncul dalam `y`.

Meskipun `inrandom(n, m, k)` mengembalikan bilangan bulat yang terdistribusi seragam dari 1 hingga `k`, kita juga dapat menggunakan distribusi bilangan bulat lainnya dengan menggunakan `randpint()`.

Pada contoh berikut, probabilitas untuk 1, 2, 3 adalah 0,4, 0,1, 0,5 berturut-turut.

```
>randpint(1,1000,[0.4,0.1,0.5]); getmultiplicities(1:3,%)
```

```
[394, 107, 499]
```

Euler dapat menghasilkan nilai acak dari berbagai distribusi. Silakan lihat dokumentasi (reference) untuk informasi lebih lanjut.

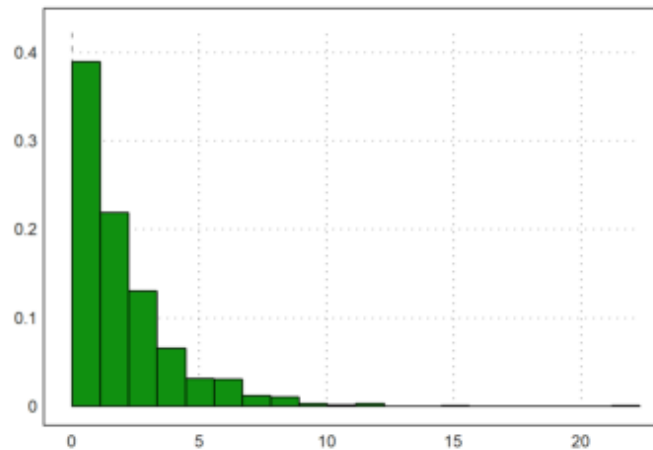
Sebagai contoh, kita akan mencoba distribusi eksponensial. Sebuah variabel acak kontinu X dikatakan memiliki distribusi eksponensial jika PDF (Probability Density Function) nya diberikan oleh:

$$f_X(x) = \lambda e^{-\lambda x}, \quad x > 0, \quad \lambda > 0,$$

dengan parameter

$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

```
>plot2d(randexponential(1,1000,2),>distribution):
```

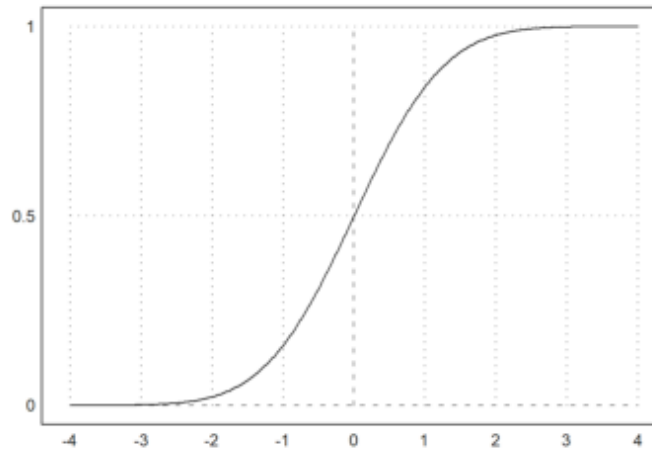


Catatan tambahan :

randexponential : matrix acak dari distribusi eksponensial

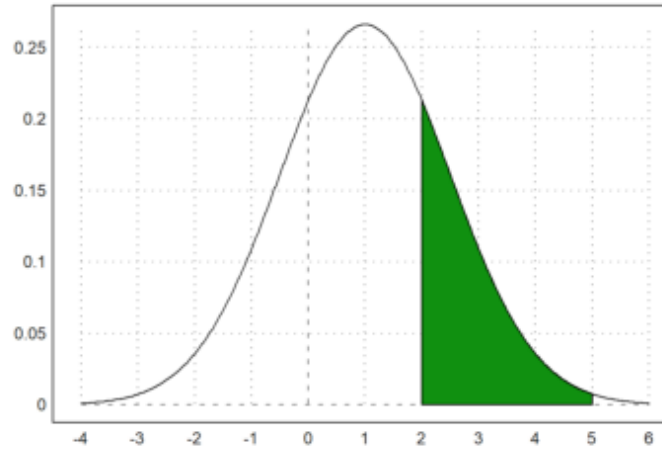
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversenya.

```
>plot2d("normaldis",-4,4):
```



Berikut adalah salah satu cara untuk memplot kuantil.

```
>plot2d("qnormal(x,1,1.5)",-4,6); ...  
>plot2d("qnormal(x,1,1.5)",a=2,b=5,>add,>filled):
```



$$\text{normaldis}(x,m,d) = \int_{-\infty}^x \frac{1}{d\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{t-m}{d}\right)^2} dt.$$

Probabilitas berada di area hijau adalah sebagai berikut.

```
>normaldis(5,1,1.5)-normaldis(2,1,1.5)
```

0.248662156979

Ini dapat dihitung secara numerik dengan integral berikut.

$$\int_2^5 \frac{1}{1.5\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-1}{1.5}\right)^2} dx.$$

```
>gauss("qnormal(x,1,1.5)",2,5)
```

```
0.248662156979
```

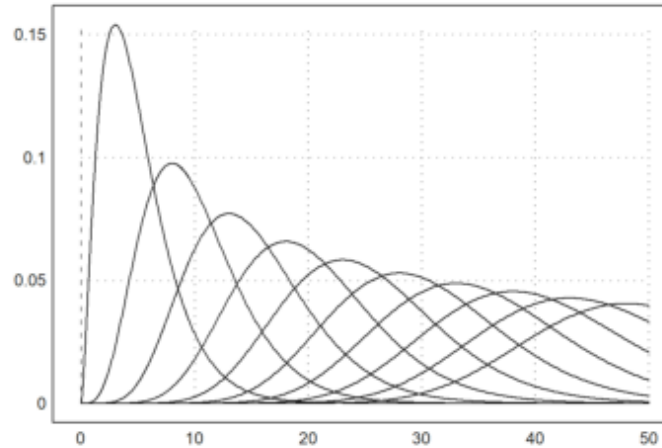
Mari bandingkan distribusi binomial dengan distribusi normal dengan rata-rata dan deviasi standar yang sama. Fungsi `invbindis()` menyelesaikan interpolasi linear antara nilai-nilai bulat.

```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219  
526.007419394
```

Fungsi `qdis()` adalah fungsi densitas distribusi chi-square. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian, kita dapat dengan mudah membuat plot untuk semua distribusi chi-square dengan derajat 5 hingga 30 dengan cara berikut.

```
>plot2d("qchidis(x,(5:5:50)')",0,50):
```



Euler memiliki fungsi yang akurat untuk mengevaluasi distribusi. Mari kita periksa `chidis()` dengan sebuah integral.

Penamaan fungsi tersebut mencoba konsisten. Misalnya,

- distribusi chi-square adalah `chidis()`,
- fungsi inversnya adalah `invchidis()`,
- fungsi densitasnya adalah `qchidis()`.

Komplemen dari distribusi (ekor atas) disebut `chicdis()`.

```
>chidis(1.5,2), integrate("qchidis(x,2)",0,1.5)
```

```
0.527633447259
```

```
0.527633447259
```

Distribusi Diskret

Untuk mendefinisikan distribusi diskret Anda sendiri, Anda dapat menggunakan metode berikut. Pertama, kita atur fungsi distribusi.

```
>wd = 0|((1:6)+[-0.01,0.01,0,0,0,0])/6
```

```
[0, 0.165, 0.335, 0.5, 0.666667, 0.833333, 1]
```

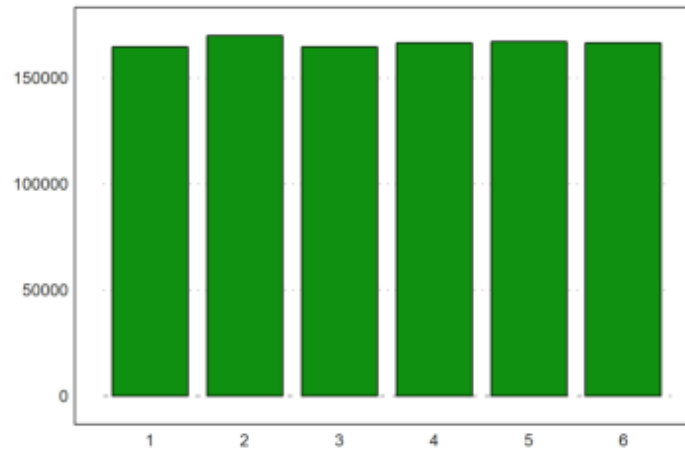
Artinya, dengan probabilitas $wd[i+1] - wd[i]$, kita menghasilkan nilai acak i .

Ini hampir merupakan distribusi seragam. Mari kita tentukan pembangkit angka acak untuk ini. Fungsi `find(v, x)` mencari nilai x dalam vektor v . Fungsi ini juga berfungsi untuk vektor x .

```
>function wrongdice (n,m) := find(wd,random(n,m))
```

Kesalahan tersebut sangat halus sehingga kita hanya bisa melihatnya dengan sangat banyak iterasi.

```
>columnsplot(getmultiplicities(1:6,wrongdice(1,1000000))):
```



Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai-nilai 1 hingga K dalam vektor v. Kita menerima hasilnya jika untuk semua frekuensi

$$\left| f_i - \frac{1}{K} \right| < \frac{\delta}{\sqrt{n}}.$$

```
>function checkrandom (v, delta=1) ...
```



```
K=max(v); n=cols(v);  
fr=getfrequencies(v,1:K);  
return max(fr/n-1/K)<delta/sqrt(n);  
endfunction
```

Memang, fungsi tersebut menolak distribusi seragam.

```
>checkrandom(wrongdice(1,1000000))
```

0

Dan itu menerima pembangkit angka acak bawaan.

```
>checkrandom(intrandom(1,1000000,6))
```

1

Kita dapat menghitung distribusi binomial. Pertama, ada fungsi `binomialsun()`, yang mengembalikan probabilitas i atau kurang hasil dalam n percobaan.

```
>bindis(410,1000,0.4)//Distribusi Binomial Kumulatif
```

0.751401349654

Fungsi invers Beta digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p . Tingkat kepercayaan default adalah α .

Makna dari interval ini adalah bahwa jika p berada di luar interval tersebut, hasil yang diamati 410 dari 1000 adalah hal yang jarang terjadi.

```
>clopperpearson(410,1000)
```

```
[0.37932, 0.441212]
```

Perintah-perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Namun, untuk nilai n yang besar, penjumlahan langsung tidak akurat dan lambat.

```
>p=0.4; i=0:410; n=1000; sum(bin(n,i)*p^i*(1-p)^(n-i))
```

```
0.751401349655
```

Sekadar informasi, `invbinsum()` menghitung invers dari `binomsum()`.

```
>invbindis(0.75,1000,0.4)
```

```
409.932733047
```

Dalam permainan Bridge, kita asumsikan ada 5 kartu istimewa (dari total 52 kartu) dalam dua tangan (26 kartu). Mari kita hitung probabilitas dari distribusi yang lebih buruk daripada 3:2 (misalnya 0:5, 1:4, 4:1, atau 5:0).

```
>2*hypergeomsum(1,5,13,26)
```

```
0.321739130435
```

Ada juga simulasi distribusi multinomial.

```
>randmultinomial(10,1000,[0.4,0.1,0.5])
```

418	92	490
445	90	465
403	114	483
405	95	500
384	107	509
414	93	493
419	90	491
394	101	505
382	103	515
403	128	469

Plotting Data

Untuk memplot data, kita mencoba hasil pemilihan umum Jerman sejak tahun 1990, diukur dalam kursi.

```
>BW := [ ...  
>1990,662,319,239,79,8,17; ...  
>1994,672,294,252,47,49,30; ...  
>1998,669,245,298,43,47,36; ...  
>2002,603,248,251,47,55,2; ...  
>2005,614,226,222,61,51,54; ...  
>2009,622,239,146,93,68,76; ...  
>2013,631,311,193,0,63,64];
```

Untuk partai-partai politik, kita menggunakan string nama-nama partai.

```
>P:=["CDU/CSU", "SPD", "FDP", "Gr", "Li"];
```

Mari mencetak persentasenya dengan rapi.

Pertama, kita ekstrak kolom-kolom yang diperlukan. Kolom 3 hingga 7 adalah kursi-kursi masing-masing partai, dan kolom 2 adalah total jumlah kursi. Kolom adalah tahun pemilihan umum.

```
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
```

Kemudian kita cetak statistiknya dalam bentuk tabel. Kita gunakan nama-nama partai sebagai judul kolom, dan tahun-tahun sebagai judul baris. Lebar default untuk kolom-kolom adalah wc=10, namun kita lebih memilih output yang lebih padat. Kolom-kolom akan diperluas untuk label-label kolom, jika diperlukan.

```
>writetable(BT*100,wc=6,dc=0,>fixed,labc=P,labr=YT)
```

	CDU/CSU	SPD	FDP	Gr	Li
1990	48	36	12	1	3
1994	44	38	7	7	4
1998	37	45	6	7	5
2002	41	42	8	9	0
2005	37	36	10	8	9
2009	38	23	15	11	12
2013	49	31	0	10	10

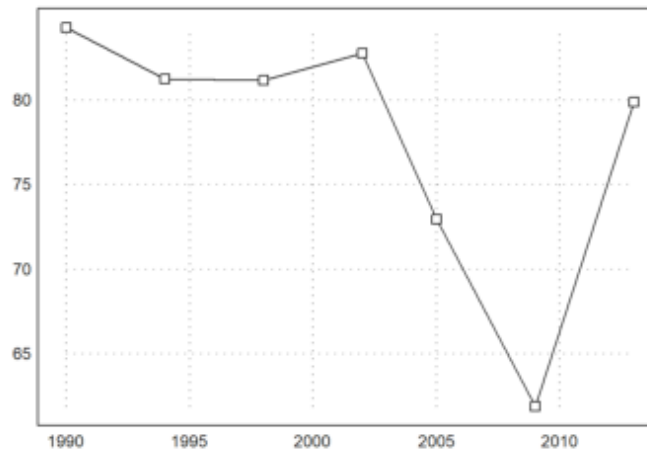
Perkalian matriks berikut mengekstrak jumlah persentase dari dua partai besar, menunjukkan bahwa partai-partai kecil telah mendapatkan perolehan kursi di parlemen hingga tahun 2009.

```
>BT1:=(BT.[1;1;0;0;0])'*100
```

```
[84.29, 81.25, 81.1659, 82.7529, 72.9642, 61.8971, 79.8732]
```

Ada juga plot statistik sederhana. Kita menggunakannya untuk menampilkan garis dan titik secara bersamaan. Alternatifnya adalah dengan memanggil plot2d dua kali dengan >add.

```
>statplot(YT,BT1,"b"):
```

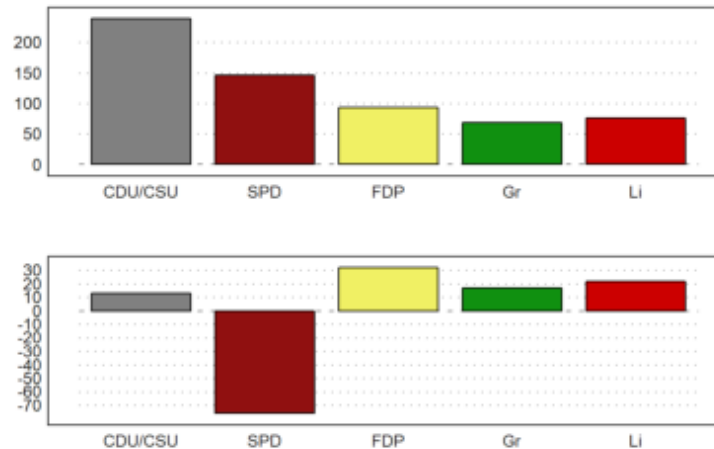


Tentukan beberapa warna untuk setiap partai.

```
>CP=[rgb(0.5,0.5,0.5),red,yellow,green,rgb(0.8,0,0)];
```

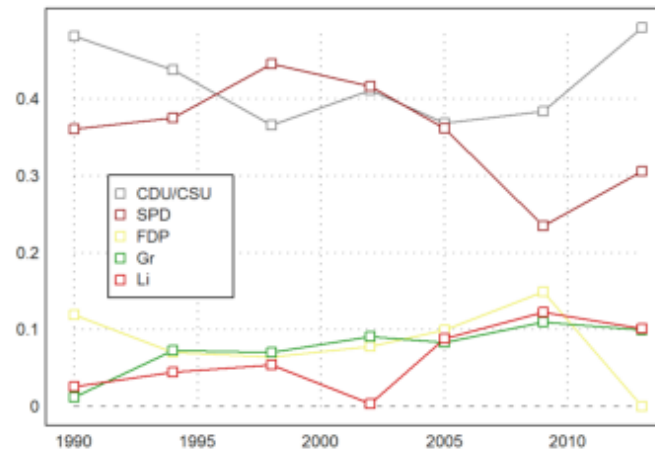
Sekarang kita dapat memplot hasil pemilihan tahun 2009 dan perubahan hasilnya dalam satu plot menggunakan fungsi figure. Kita dapat menambahkan vektor kolom ke setiap plot.

```
>figure(2,1); ...  
>figure(1); columnspot(BW[6,3:7],P,color=CP); ...  
>figure(2); columnspot(BW[6,3:7]-BW[5,3:7],P,color=CP); ...  
>figure(0):
```



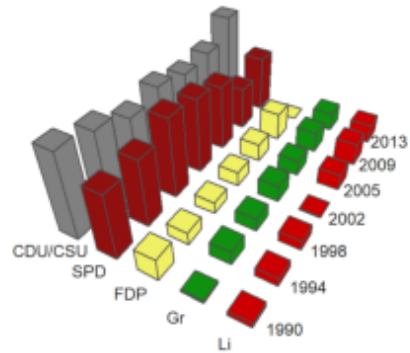
Plot data menggabungkan baris-baris data statistik dalam satu plot.

```
>J:=BW[,1]'; DP:=BW[,3:7]'; ...  
>dataplot(YT,BT',color=CP); ...  
>labelbox(P,colors=CP,styles="[]",>points,w=0.2,x=0.3,y=0.4):
```



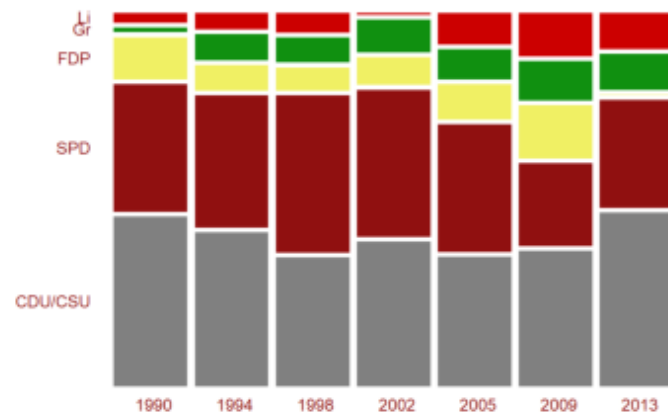
Plot kolom 3D menunjukkan baris-baris data statistik dalam bentuk kolom. Kita menyediakan label untuk baris dan kolom. Sudut (angle) adalah sudut pandang tampilan.

```
>columnplot3d(BT,scols=P,srows=YT, ...  
> angle=30°,ccols=CP):
```

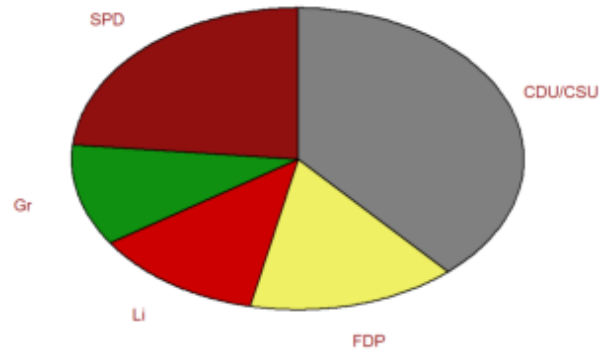
Representasi lainnya adalah plot mozaik. Perlu diingat bahwa kolom-kolom plot ini merepresentasikan kolom-kolom dari matriks di sini. Karena panjang label CDU/CSU, kita menggunakan jendela yang lebih kecil dari biasanya.

```
>shrinkwindow(>smaller); ...
>mosaicplot(BT',srows=YT,scols=P,color=CP,style="#"); ...
>shrinkwindow():
```



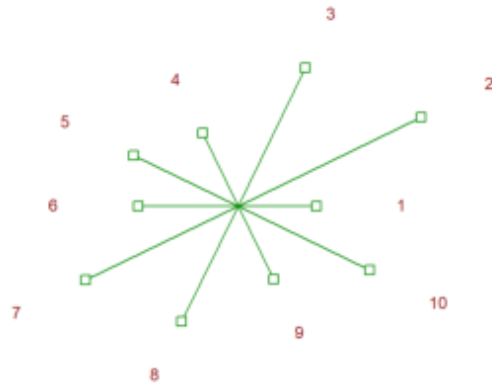
Kita juga dapat membuat diagram lingkaran (pie chart). Karena hitam dan kuning membentuk koalisi, kita akan menyusun ulang elemen-elemen tersebut.

```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



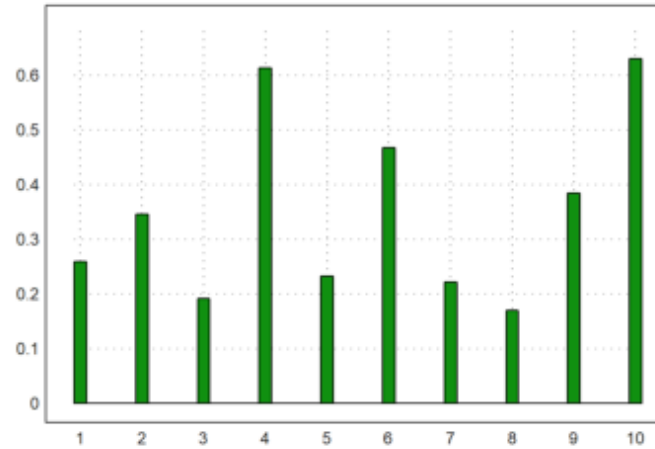
Di sini adalah jenis plot yang lain.

```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```



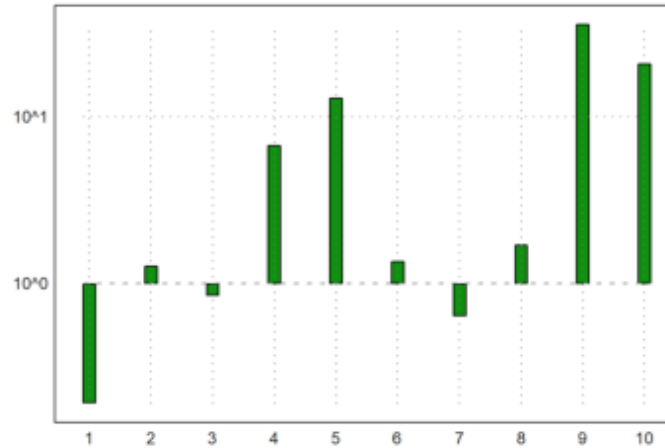
Beberapa plot dalam plot2d cocok untuk data statis. Berikut adalah plot impuls dari data acak yang terdistribusi secara merata dalam $[0,1]$.

```
>plot2d(makeimpulse(1:10,random(1,10)),>bar):
```



But for exponentially distributed data, we may need a logarithmic plot.

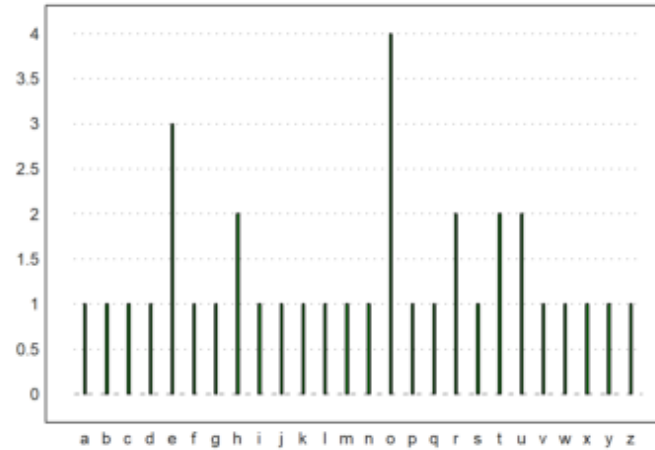
```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi `'columnplot()'` lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, kita dapat mengatur label sesuai keinginan kita, seperti yang telah kami tunjukkan dalam tutorial ini sebelumnya.

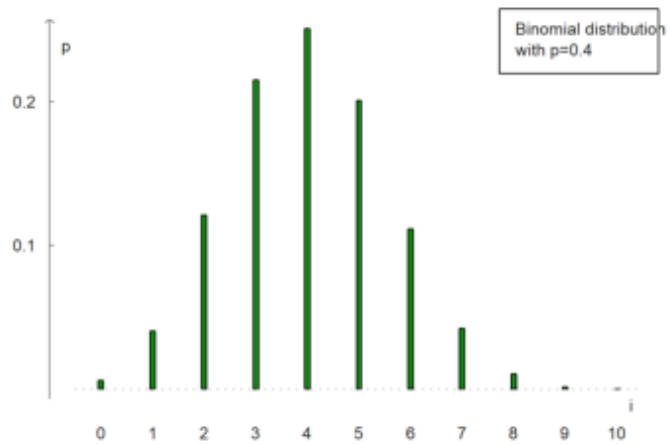
Berikut adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat plot statistiknya.

```
>v=strtochar("the quick brown fox jumps over the lazy dog"); ...  
>w=ascii("a"):ascii("z"); x=getmultiplicities(w,v); ...  
>cw=[]; for k=w; cw=cw|char(k); end; ...  
>columnplot(x,lab=cw,width=0.05):
```



Juga mungkin untuk secara manual mengatur sumbu-sumbu.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
>columnsplot(x,lab=i,width=0.05,<frame,<grid); ...
>yaxis(0,0:0.1:1,style="->",>left); xaxis(0,style="."); ...
>label("p",0,0.25), label("i",11,0); ...
>textbox(["Binomial distribution","with p=0.4"]):
```



Berikut adalah cara untuk membuat plot frekuensi angka dalam sebuah vektor.
Kita membuat vektor dari angka-angka acak integer dari 1 hingga 6.

```
>v:=inrandom(1,10,10)
```

```
[4, 5, 2, 6, 1, 10, 8, 4, 1, 2]
```

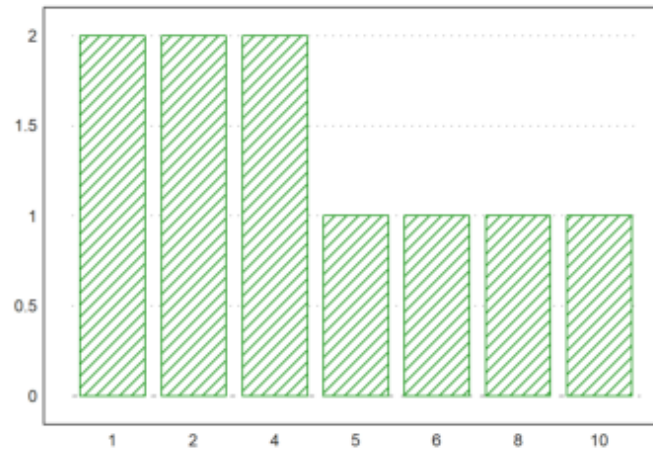

Kemudian ekstrak angka-angka unik dalam vektor v.

```
>vu:=unique(v)
```

```
[1, 2, 4, 5, 6, 8, 10]
```

Dan plot frekuensi tersebut dalam sebuah plot kolom.

```
>columnplot(getmultiplicities(vu,v),lab=vu,style="/"):
```



Kami ingin mendemonstrasikan fungsi-fungsi untuk distribusi empiris dari nilai-nilai.

```
>x=normal(1,20);
```

Fungsi 'empdist(x, vs)' memerlukan sebuah larik nilai yang telah diurutkan. Oleh karena itu, kita perlu mengurutkan x sebelum kita dapat menggunakannya.

```
>xs=sort(x);
```

Kemudian kita membuat plot distribusi empiris dan beberapa batang kepadatan dalam satu plot. Kali ini, daripada menggunakan plot batang untuk distribusi, kita akan menggunakan plot gigi gergaji (sawtooth plot).

```
>figure(2,1); ...  
>figure(1); plot2d("empdist",-4,4;xs); ...  
>figure(2); plot2d(histo(x,v=-4:0.2:4,<bar)); ...  
>figure(0):
```

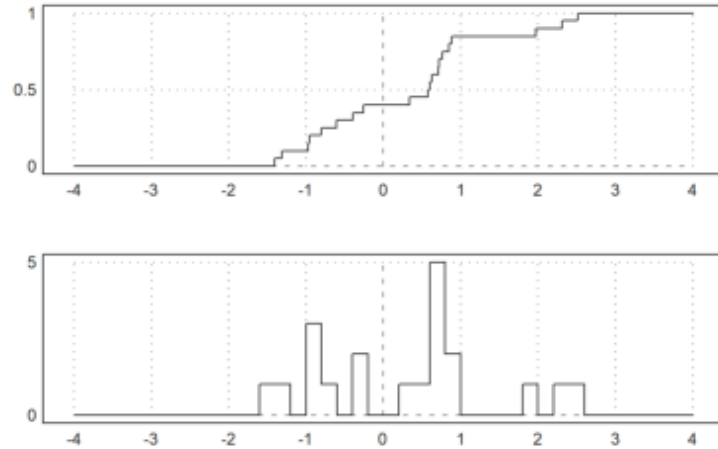
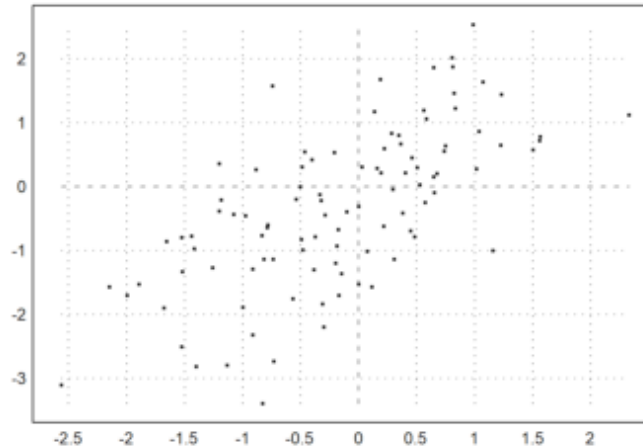


Diagram hamburan (scatter plot) mudah dilakukan dalam Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa X dan $X+Y$ jelas berkorelasi positif.

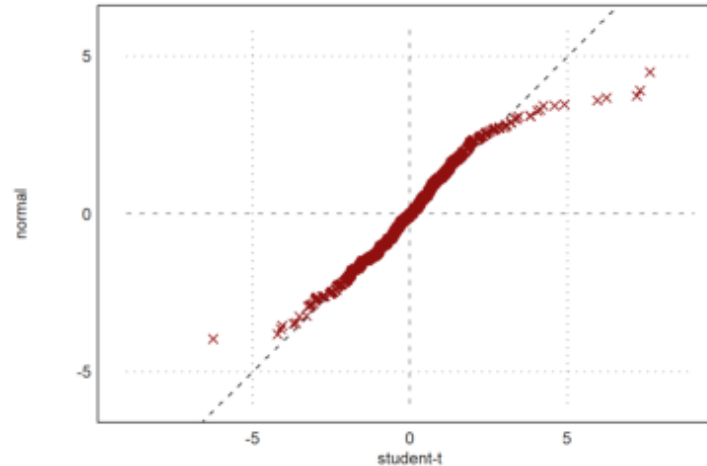
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



Seringkali, kita ingin membandingkan dua sampel dengan distribusi yang berbeda. Hal ini dapat dilakukan dengan menggunakan plot kuantil-kuantil (quantile-quantile plot).

Untuk sebuah uji, kita mencoba distribusi t-student dan distribusi eksponensial.

```
>x=randt(1,1000,5); y=randnormal(1,1000,mean(x),dev(x)); ...  
>plot2d("x",r=6,style="--",yl="normal",xl="student-t",>vertical); ...  
>plot2d(sort(x),sort(y),>points,color=red,style="x",>add):
```



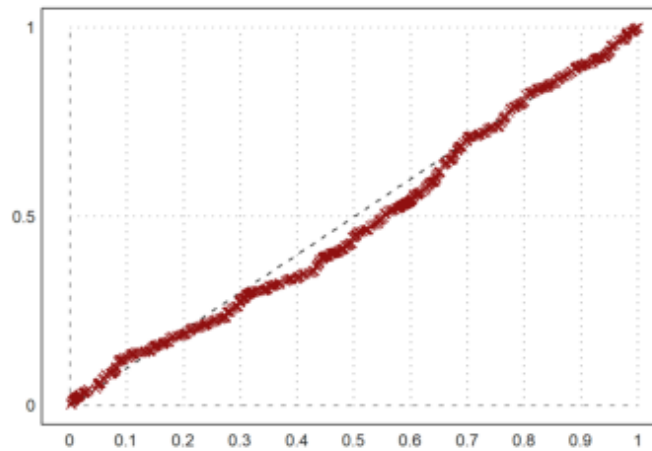
Grafik tersebut dengan jelas menunjukkan bahwa nilai yang terdistribusi secara normal cenderung lebih kecil di ujung-ujung ekstrem.

Jika kita memiliki dua distribusi dengan ukuran yang berbeda, kita dapat memperbesar yang lebih kecil atau mengecilkan yang lebih besar. Fungsi berikut baik digunakan untuk keduanya. Ini mengambil nilai median dengan persentase antara 0 dan 1.

```
>function medianexpand (x,n) := median(x,p=linspace(0,1,n-1));
```

Mari kita bandingkan dua distribusi yang sama.

```
>x=random(1000); y=random(400); ...  
>plot2d("x",0,1,style="--"); ...  
>plot2d(sort(medianexpand(x,400)),sort(y),>points,color=red,style="x",>add):
```



Regresi dan Korelasi

Regresi linear dapat dilakukan dengan menggunakan fungsi `polyfit()` atau berbagai fungsi fit lainnya. Untuk memulai, kita dapat menemukan garis regresi untuk data univariat dengan `polyfit(x, y, 1)`.

```
>x=1:10; y=[2,3,1,5,6,3,7,8,9,8]; writetable(x'|y',labc=["x","y"])
```

x	y
1	2
2	3
3	1
4	5
5	6
6	3
7	7
8	8
9	9
10	8

Kita ingin membandingkan hasil regresi tanpa bobot (non-weighted) dan dengan bobot (weighted). Pertama-tama, mari lihat koefisien regresi linearnya.

```
>p=polyfit(x,y,1)
```

```
[0.733333, 0.812121]
```

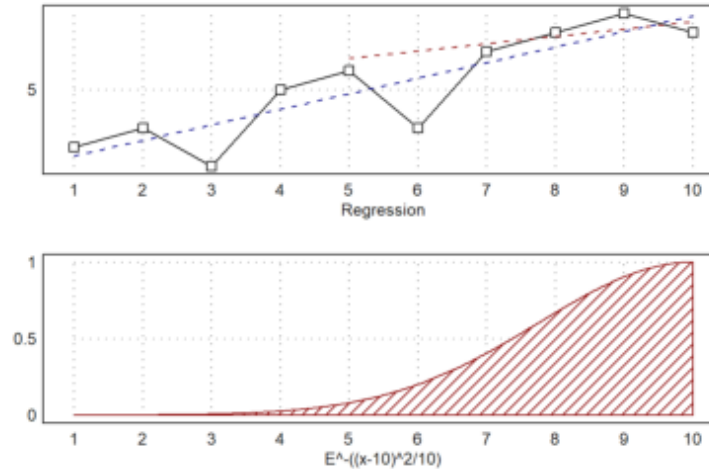
Sekarang mari lihat koefisien dengan bobot yang menekankan nilai-nilai terakhir.

```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[4.71566,  0.38319]
```

Kita gabungkan semuanya dalam satu plot untuk titik-titik data, garis regresi, dan untuk bobot yang digunakan.

```
>figure(2,1); ...  
>figure(1); statplot(x,y,"b",xl="Regression"); ...  
> plot2d("evalpoly(x,p)",>add,color=blue,style="--"); ...  
> plot2d("evalpoly(x,pw)",5,10,>add,color=red,style="--"); ...  
>figure(2); plot2d(w,1,10,>filled,style="/",fillcolor=red,xl=w); ...  
>figure(0):
```

Untuk contoh lain, kita membaca hasil survei tentang mahasiswa, usia mereka, usia orang tua mereka, dan jumlah saudara kandung dari sebuah file.

Tabel ini berisi "m" dan "f" dalam kolom kedua. Kita menggunakan variabel tok2 untuk mengatur terjemahan yang sesuai daripada membiarkan readtable() mengumpulkan terjemahan.

```
>{MS,hd}:=readtable("table1.dat",tok2=["m","f"]); ...
>writetable(MS,labc=hd,tok2=["m","f"]);
```

```
Could not open the file
table1.dat
for reading!
Try "trace errors" to inspect local variables after errors.
readtable:
    if filename!=none then open(filename,"r"); endif;
```

Bagaimana usia-usia ini bergantung satu sama lain? Kesimpulan awal dapat diperoleh dari scatterplot pasangan (pairwise scatterplot).

```
>scatterplots(tablecol(MS,3:5),hd[3:5]):
```

```
Variable or function MS not found.  
Error in:  
scatterplots(tablecol(MS,3:5),hd[3:5]): ...  
                        ^
```

Jelas bahwa usia ayah dan ibu saling bergantung. Mari kita tentukan dan plot garis regresinya.

```
>cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1)
```

```
MS is not a variable!  
Error in:  
cs:=MS[,4:5]'; ps:=polyfit(cs[1],cs[2],1) ...  
                        ^
```

Ini jelas bukan model yang tepat. Garis regresinya seharusnya adalah $s = 17 + 0.74t$, di mana t adalah usia ibu dan s adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak sebanyak itu.

Sebaliknya, kami mencurigai bahwa fungsi seperti $s = a + t$. Kemudian, a adalah rata-rata dari $s-t$. Itu adalah perbedaan usia rata-rata antara ayah dan ibu.

```
>da:=mean(cs[2]-cs[1])
```

```
cs is not a variable!  
Error in:  
da:=mean(cs[2]-cs[1]) ...  
           ^
```

Mari kita plot ini dalam satu scatter plot.

```
>plot2d(cs[1],cs[2],>points); ...  
>plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...  
>plot2d("x+da",color=blue,>add):
```

```
cs is not a variable!  
Error in:  
plot2d(cs[1],cs[2],>points); plot2d("evalpoly(x,ps)",color=re ...  
           ^
```

Berikut adalah box plot dari dua usia. Ini hanya menunjukkan bahwa usia-usia tersebut berbeda.

```
>boxplot(cs,["mothers","fathers"]):
```

```
Variable or function cs not found.  
Error in:  
boxplot(cs,["mothers","fathers"]): ...  
      ^
```

Menarik bahwa perbedaan median tidak sebesar perbedaan mean.

```
>median(cs[2])-median(cs[1])
```

```
cs is not a variable!  
Error in:  
median(cs[2])-median(cs[1]) ...  
      ^
```

Koefisien korelasi menunjukkan adanya korelasi positif.

```
>correl(cs[1],cs[2])
```

```
cs is not a variable!  
Error in:  
correl(cs[1],cs[2]) ...  
      ^
```

Korelasi peringkat adalah ukuran untuk urutan yang sama dalam kedua vektor. Juga sangat positif.

```
>rankcorrel(cs[1],cs[2])
```

```
cs is not a variable!  
Error in:  
rankcorrel(cs[1],cs[2]) ...  
      ^
```

Membuat Fungsi Baru

Tentu saja, bahasa EMT dapat digunakan untuk membuat fungsi-fungsi baru. Misalnya, kita dapat mendefinisikan fungsi keserjanaan (skewness).

$$\text{sk}(x) = \frac{\sqrt{n} \sum_i (x_i - m)^3}{(\sum_i (x_i - m)^2)^{3/2}}$$

di mana m adalah rata-rata dari x .

```
>function skew (x:vector) ...  
  
    m=mean(x);  
    return sqrt(cols(x))*sum((x-m)^3)/(sum((x-m)^2))^(3/2);  
endfunction
```

Seperti yang Anda lihat, kita dengan mudah dapat menggunakan bahasa matriks untuk mendapatkan implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

```
>data=normal(20); skew(normal(10))
```

```
-0.0794571577159
```

Berikut adalah fungsi lain yang disebut koefisien skewness Pearson.

```
>function skew1 (x) := 3*(mean(x)-median(x))/dev(x)  
>skew1(data)
```

-1.04418407161

Simulasi Monte Carlo

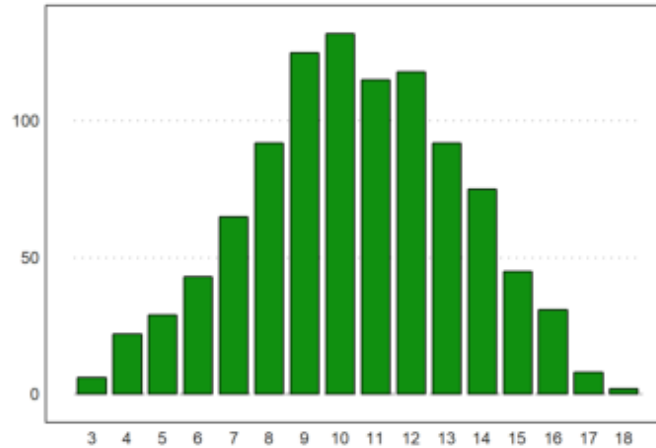
Euler dapat digunakan untuk mensimulasikan peristiwa acak. Kami telah melihat contoh-contoh sederhana di atas. Berikut adalah contoh lain yang mensimulasikan 1000 kali lemparan tiga dadu, dan menghitung distribusi jumlahnya.

```
>ds:=sum(intrandom(1000,3,6))'; fs=getmultiplicities(3:18,ds)
```

```
[6, 22, 29, 43, 65, 92, 125, 132, 115, 118, 92, 75, 45,  
31, 8, 2]
```

Sekarang kita dapat membuat plot hasil simulasi ini.

```
>columnspplot(fs,lab=3:18):
```

Untuk menentukan distribusi yang diharapkan tidaklah mudah. Kita menggunakan rekursi tingkat lanjut untuk ini.

Fungsi berikut menghitung jumlah cara di mana angka k dapat direpresentasikan sebagai jumlah dari n angka dalam rentang 1 hingga m . Ini bekerja secara rekursif dengan cara yang jelas.

```
>function map countways (k; n, m) ...

    if n==1 then return k>=1 && k<=m
    else
        sum=0;
        loop 1 to m; sum=sum+countways(k-#,n-1,m); end;
        return sum;
    end;
endfunction
```

Berikut adalah hasilnya untuk tiga lemparan dadu.

```
>countways(5:25,5,5)
```

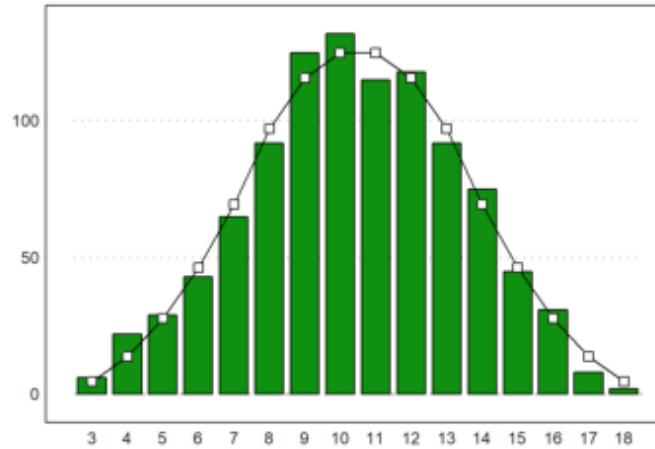
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,  
255, 185, 121, 70, 35, 15, 5, 1]
```

```
>cw=countways(3:18,3,6)
```

```
[1, 3, 6, 10, 15, 21, 25, 27, 27, 25, 21, 15, 10, 6, 3,  
1]
```

Kita tambahkan nilai-nilai yang diharapkan ke dalam plot.

```
>plot2d(cw/6^3*1000,>add); plot2d(cw/6^3*1000,>points,>add):
```



Untuk simulasi lainnya, deviasi nilai rata-rata dari n variabel acak yang terdistribusi normal antara 0 dan 1 adalah $1/\sqrt{n}$.

```
>longformat; 1/sqrt(10)
```

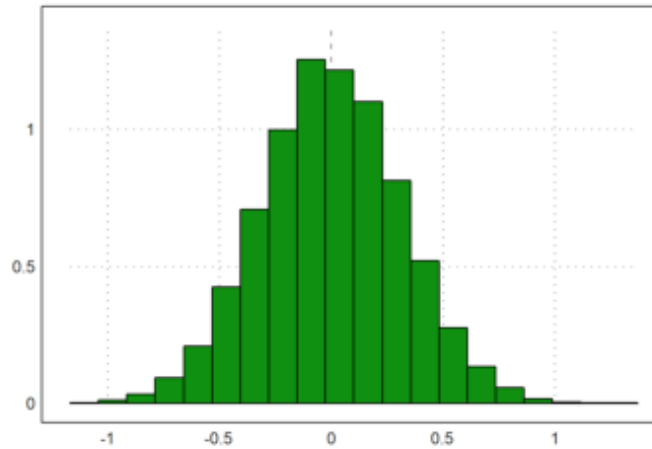
0.316227766017

Mari kita periksa ini dengan simulasi. Kami menghasilkan 10.000 kali 10 vektor acak.

```
>M=normal(10000,10); dev(mean(M)')
```

0.314944511075

```
>plot2d(mean(M)',>distribution):
```



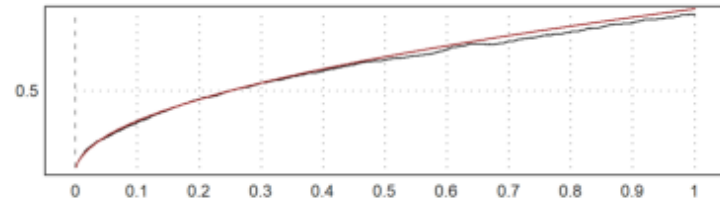
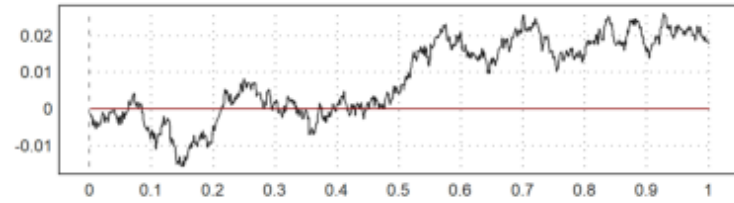
Median dari 10 angka acak yang terdistribusi normal antara 0 dan 1 memiliki deviasi yang lebih besar.

```
>dev(median(M)')
```

0.370541708221

Karena kita dapat dengan mudah menghasilkan perjalanan acak (random walks), kita dapat mensimulasikan proses Wiener. Kita ambil 1000 langkah dari 1000 proses ini. Kemudian kita membuat plot dari simpangan baku (standard deviation) dan rata-rata langkah ke-n dari proses-proses ini bersama dengan nilai-nilai yang diharapkan dalam warna merah.

```
>n=1000; m=1000; M=cumsum(normal(n,m)/sqrt(m)); ...  
>t=(1:n)/n; figure(2,1); ...  
>figure(1); plot2d(t,mean(M')'); plot2d(t,0,color=red,>add); ...  
>figure(2); plot2d(t,dev(M')'); plot2d(t,sqrt(t),color=red,>add); ...  
>figure(0):
```



Uji statistik merupakan alat penting dalam statistik. Dalam Euler, banyak uji statistik telah diimplementasikan. Semua uji ini mengembalikan kesalahan yang kita terima jika kita menolak hipotesis nol.

Sebagai contoh, kita menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kita mendapatkan nilai-nilai berikut, yang kita masukkan ke dalam uji chi-kuadrat.

```
>chitest([90,103,114,101,103,89],dup(100,6)')
```

```
0.498830517952
```

Uji chi-kuadrat juga memiliki mode yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya seharusnya hampir sama. Parameter `>p` menginterpretasikan vektor `y` sebagai vektor probabilitas.

```
>chitest([90,103,114,101,103,89],dup(1/6,6)',>p,>montecarlo)
```

```
0.507
```

Kesalahan ini jauh terlalu besar. Jadi kita tidak bisa menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Tapi kita tidak bisa menolak hipotesis kita.

Selanjutnya, kita menghasilkan 1000 lemparan dadu menggunakan pembangkit bilangan acak, dan melakukan uji yang sama.

```
>n=1000; t=random([1,n*6]); chitest(count(t*6,6),dup(n,6)')
```

0.247506458901

Mari kita uji untuk nilai rata-rata 100 dengan uji t.

```
>s=200+normal([1,100])*10; ...  
>ttest(mean(s),dev(s),100,200)
```

0.261148883454

Fungsi ttest() memerlukan nilai rata-rata, deviasi, jumlah data, dan nilai rata-rata yang akan diuji.

Sekarang mari kita periksa dua pengukuran untuk nilai rata-rata yang sama. Kita menolak hipotesis bahwa mereka memiliki nilai rata-rata yang sama jika hasilnya $< 0,05$.

```
>tcomparedata(normal(1,10),normal(1,10))
```

0.116475902968

Jika kita menambahkan bias pada salah satu distribusi, kita akan mendapatkan lebih banyak penolakan. Ulangi simulasi ini beberapa kali untuk melihat efeknya.

```
>tcomparedata(normal(1,10),normal(1,10)+2)
```

5.770990267e-05

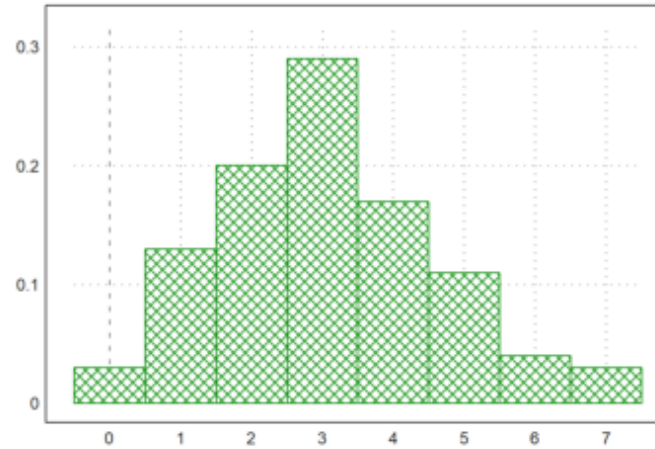
Pada contoh berikut, kita menghasilkan 20 lemparan dadu acak 100 kali dan menghitung jumlah angka satu (1) dalamnya. Rata-rata seharusnya adalah $20/6 = 3,3$ angka satu.

```
>R=random(100,20); R=sum(R*6<=1)'; mean(R)
```

3.08

Sekarang kita membandingkan jumlah angka satu dengan distribusi binomial. Pertama-tama, kita membuat plot distribusi angka satu.

```
>plot2d(R,distribution=max(R)+1,even=1,style="\/"): 
```



```
>t=count(R,21);
```

Kemudian kita menghitung nilai-nilai yang diharapkan.

```
>n=0:20; b=bin(20,n)*(1/6)^n*(5/6)^(20-n)*100;
```

Kita harus mengumpulkan beberapa angka untuk mendapatkan kategori-kategori yang cukup besar.

```
>t1=sum(t[1:2])|t[3:7]|sum(t[8:21]); ...
>b1=sum(b[1:2])|b[3:7]|sum(b[8:21]);
```

Uji chi-kuadrat menolak hipotesis bahwa distribusi kita adalah distribusi binomial, jika hasilnya $<0,05$.

```
>chitest(t1,b1)
```

```
0.717403213286
```

Contoh berikut berisi hasil dari dua kelompok orang (pria dan wanita, misalnya) yang memilih salah satu dari enam partai.

```
>A=[23,37,43,52,64,74;27,39,41,49,63,76]; ...
> writetable(A,wc=6,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	23	37	43	52	64	74
f	27	39	41	49	63	76

Kita ingin menguji independensi suara dari jenis kelamin. Uji tabel chi-kuadrat melakukannya. Hasilnya terlalu besar untuk menolak independensi. Jadi, dari data ini, kita tidak bisa mengatakan apakah pemilihan bergantung pada jenis kelamin atau tidak.

```
>tabletest(A)
```

```
0.990701632326
```

Berikut adalah tabel yang diharapkan jika kita mengasumsikan frekuensi yang diamati dalam pemilihan.

```
>writetable(expectedtable(A),wc=6,dc=1,labr=["m","f"],labc=1:6)
```

	1	2	3	4	5	6
m	24.9	37.9	41.9	50.3	63.3	74.7
f	25.1	38.1	42.1	50.7	63.7	75.3

Kita dapat menghitung koefisien kontingensi yang dikoreksi. Karena nilai yang sangat mendekati 0, kita dapat menyimpulkan bahwa pemilihan tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```

Beberapa Uji Lainnya

Selanjutnya, kita menggunakan analisis varians (uji F) untuk menguji tiga sampel data yang terdistribusi normal untuk nilai rata-rata yang sama. Metode ini disebut ANOVA (analisis varians). Dalam Euler, digunakan fungsi `varanalysis()`.

```
>x1=[109,111,98,119,91,118,109,99,115,109,94]; mean(x1),
```

```
106.545454545
```

```
>x2=[120,124,115,139,114,110,113,120,117]; mean(x2),
```

```
119.111111111
```

```
>x3=[120,112,115,110,105,134,105,130,121,111]; mean(x3)
```

```
116.3
```

```
>varanalysis(x1,x2,x3)
```

```
0.0138048221371
```

Artinya, kita menolak hipotesis nilai rata-rata yang sama. Kita melakukannya dengan probabilitas kesalahan sebesar 1,3%.

Ada juga uji median, yang menolak sampel data dengan distribusi rata-rata yang berbeda dengan menguji median dari sampel yang digabungkan.

```
>a=[56,66,68,49,61,53,45,58,54];  
>b=[72,81,51,73,69,78,59,67,65,71,68,71];  
>mediantest(a,b)
```

```
0.0241724220052
```

Uji lain untuk kesetaraan adalah uji peringkat. Ini jauh lebih tajam daripada uji median.

```
>ranktest(a,b)
```

```
0.00199969612469
```

Pada contoh berikut, kedua distribusi memiliki nilai rata-rata yang sama.

```
>ranktest(random(1,100),random(1,50)*3-1)
```

```
0.0908472946126
```

Mari kita coba mensimulasikan dua perawatan a dan b yang diberikan kepada orang-orang yang berbeda.

```
>a=[8.0,7.4,5.9,9.4,8.6,8.2,7.6,8.1,6.2,8.9];  
>b=[6.8,7.1,6.8,8.3,7.9,7.2,7.4,6.8,6.8,8.1];
```

Uji signum (signum test) menentukan apakah a lebih baik daripada b.

```
>signtest(a,b)
```

0.0546875

Ini terlalu tinggi tingkat kesalahan. Kita tidak bisa menolak bahwa a sama baiknya dengan b.

Uji Wilcoxon lebih tajam daripada uji ini, tetapi bergantung pada nilai kuantitatif dari perbedaan.

```
>wilcoxon(a,b)
```

0.0296680599405

Mari kita mencoba dua uji lainnya menggunakan rangkaian data yang dihasilkan.

```
>wilcoxon(normal(1,20),normal(1,20)-1)
```

0.00499819423799

```
>wilcoxon(normal(1,20),normal(1,20))
```

0.559353645673

Bilangan Acak

Berikut adalah uji untuk generator bilangan acak. Euler menggunakan generator yang sangat baik, jadi kita tidak perlu mengharapkan masalah.

Pertama-tama kita menghasilkan sepuluh juta bilangan acak dalam rentang $[0,1]$.

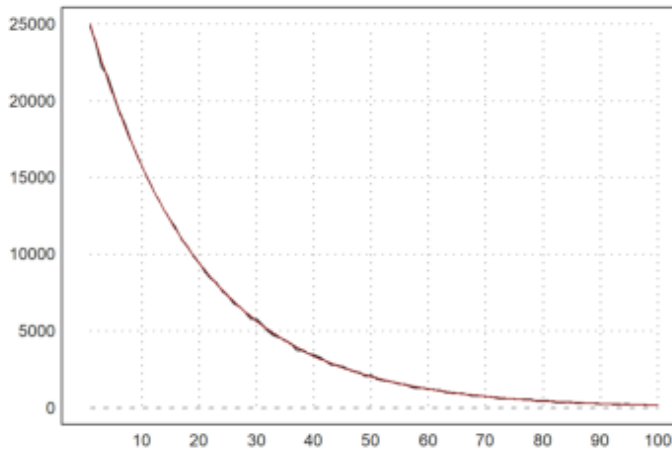
```
>n:=10000000; r:=random(1,n);
```

Selanjutnya, kita menghitung jarak antara dua bilangan yang kurang dari 0,05.

```
>a:=0.05; d:=differences(nonzeros(r<a));
```

Akhirnya, kita membuat plot jumlah kali setiap jarak terjadi, dan membandingkannya dengan nilai yang diharapkan.

```
>m=getmultiplicities(1:100,d); plot2d(m); ...  
> plot2d("n*(1-a)^(x-1)*a^2",color=red,>add):
```



Hapus data.

```
>remvalue n;
```

Pengantar untuk Pengguna Proyek R

Jelas, EMT tidak bersaing dengan R sebagai paket statistik. Namun, ada banyak prosedur statistik dan fungsi yang tersedia dalam EMT juga. Jadi, EMT dapat memenuhi kebutuhan dasar. Pada akhirnya, EMT dilengkapi dengan paket-paket numerik dan sistem aljabar komputer.

Buku catatan ini untuk Anda jika Anda akrab dengan R, tetapi perlu mengetahui perbedaan dalam sintaks antara EMT dan R. Kami akan memberikan gambaran tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami akan mengeksplorasi cara pertukaran data antara kedua sistem ini.

Harap dicatat bahwa ini adalah sebuah proyek yang masih dalam proses pengembangan. **Sintaks**

Dasar

Hal pertama yang Anda pelajari dalam R adalah membuat vektor. Di EMT, perbedaan utamanya adalah operator : dapat mengambil langkah. Selain itu, operator : memiliki prioritas yang rendah.

```
>n=10; 0:n/20:n-1
```

```
[0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5,  
7, 7.5, 8, 8.5, 9]
```

Fungsi c() tidak ada. Namun, Anda dapat menggunakan vektor untuk menggabungkan hal-hal.

Contoh berikut, seperti banyak contoh lainnya, diambil dari "Pengantar ke R" yang disertakan dalam proyek R. Jika Anda membaca PDF tersebut, Anda akan menemukan bahwa saya mengikuti jalurnya dalam tutorial ini.

```
>x=[10.4, 5.6, 3.1, 6.4, 21.7]; [x,0,x]
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 0, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Operator titik dua (colon operator) dengan langkah di EMT digantikan oleh fungsi seq() di R. Kita bisa menulis fungsi ini di EMT.

```
>function seq(a,b,c) := a:b:c; ...  
>seq(0,-0.1,-1)
```

```
[0, -0.1, -0.2, -0.3, -0.4, -0.5, -0.6, -0.7, -0.8, -0.9, -1]
```

Fungsi rep() dalam R tidak ada di EMT. Untuk input berupa vektor, fungsi ini dapat ditulis sebagai berikut.

```
>function rep(x:vector,n:index) := flatten(dup(x,n)); ...  
>rep(x,2)
```

```
[10.4, 5.6, 3.1, 6.4, 21.7, 10.4, 5.6, 3.1, 6.4, 21.7]
```

Perlu dicatat bahwa "=" atau ":=" digunakan untuk penugasan dalam EMT. Operator "->" digunakan untuk satuan dalam EMT.

```
>125km -> " miles"
```

```
77.6713990297 miles
```

Operator "<-" untuk penugasan memang membingungkan dan bukan ide yang baik dalam R. Berikut akan membandingkan a dengan -4 dalam EMT.

```
>a=2; a<-4
```

0

Di R, "a <- 4 < 3" berfungsi, tetapi "a <- 4 <- 3" tidak. Saya juga memiliki ambiguitas yang serupa di EMT, tetapi mencoba mengatasinya sedikit demi sedikit.

EMT dan R memiliki vektor tipe boolean. Namun, di EMT, angka 0 dan 1 digunakan untuk mewakili false dan true. Di R, nilai true dan false tetap dapat digunakan dalam aritmatika biasa seperti di EMT.

```
>x<5, %*x
```

```
[0, 0, 1, 0, 0]  
[0, 0, 3.1, 0, 0]
```

EMT dapat menghasilkan kesalahan atau mengembalikan NAN tergantung pada pengaturan flag "errors".

```
>errors off; 0/0, isNAN(sqrt(-1)), errors on;
```

```
NAN  
1
```

String (tali) adalah sama dalam R dan EMT. Keduanya berada dalam bahasa yang berlaku, bukan dalam Unicode.

Di R, ada paket untuk Unicode. Di EMT, sebuah string dapat menjadi string Unicode. Sebuah string Unicode dapat diterjemahkan ke dalam enkodean lokal dan sebaliknya. Selain itu, u"..." dapat mengandung entitas HTML.

```
>u"&#169; Ren&eacute; Grothmann"
```

© René Grothmann

Berikut mungkin atau mungkin tidak ditampilkan dengan benar di sistem Anda sebagai A dengan titik di atas dan tanda garis di atasnya. Ini tergantung pada font yang Anda gunakan.

```
>chartoutf([480])
```

Penggabungan string dilakukan dengan "+" atau "|". Ini dapat mencakup angka, yang akan mencetak dalam format yang berlaku.

```
>"pi = "+pi
```

pi = 3.14159265359

Sebagian besar waktu, ini akan berfungsi seperti dalam R.

Namun, EMT akan menginterpretasikan indeks negatif dari belakang vektor, sementara R menginterpretasikan `x[n]` sebagai `x` tanpa elemen ke-`n`.

```
>x, x[1:3], x[-2]
```

```
[10.4,  5.6,  3.1,  6.4, 21.7]  
[10.4,  5.6,  3.1]  
6.4
```

Perilaku R dapat dicapai dalam EMT dengan menggunakan `drop()`.

```
>drop(x,2)
```

```
[10.4,  3.1,  6.4, 21.7]
```


Vektor logika tidak diperlakukan secara berbeda sebagai indeks dalam EMT, berbeda dengan R. Anda perlu mengekstraksi elemen-elemen yang bukan nol terlebih dahulu di EMT.

```
>x, x>5, x[nonzeros(x>5)]
```

```
[10.4,  5.6,  3.1,  6.4, 21.7]  
[1,  1,  0,  1,  1]  
[10.4,  5.6,  6.4, 21.7]
```

Sama seperti dalam R, vektor indeks dapat mengandung pengulangan.

```
>x[[1,2,2,1]]
```

```
[10.4,  5.6,  5.6, 10.4]
```

Tetapi penggunaan nama untuk indeks tidak mungkin di EMT. Untuk paket statistik, ini mungkin seringkali diperlukan untuk memudahkan akses ke elemen-elemen vektor.

Untuk meniru perilaku ini, kita dapat mendefinisikan fungsi seperti berikut.

```
>function sel (v,i,s) := v[indexof(s,i)]; ...  
>s=["first","second","third","fourth"]; sel(x,["first","third"],s)
```

```
Trying to overwrite protected function sel!  
Error in:  
function sel (v,i,s) := v[indexof(s,i)]; ... ...  
      ^
```

```
Trying to overwrite protected function sel!  
Error in:  
function sel (v,i,s) := v[indexof(s,i)]; ... ...  
      ^
```

```
Trying to overwrite protected function sel!  
Error in:  
function sel (v,i,s) := v[indexof(s,i)]; ... ...  
      ^
```

```
[10.4, 3.1]
```

Tipe Data

EMT memiliki lebih banyak tipe data yang telah ditentukan daripada R. Dalam R, jelas terdapat vektor yang bisa tumbuh. Anda dapat mengatur vektor numerik kosong `v` dan memberikan nilai ke elemen `v[17]`. Hal ini tidak mungkin dilakukan di EMT.

Berikut ini sedikit tidak efisien.

```
>v=[]; for i=1 to 10000; v=v|i; end;
```

EMT akan mengkonstruksi sebuah vektor dengan `v` dan `i` yang ditambahkan di atas tumpukan (stack) dan menyalin vektor tersebut kembali ke variabel global `v`.

Cara yang lebih efisien adalah dengan mendefinisikan vektor sebelumnya.

```
>v=zeros(10000); for i=1 to 10000; v[i]=i; end;
```

Untuk mengubah tipe data dalam EMT, Anda dapat menggunakan fungsi seperti `complex()`.

```
>complex(1:4)
```

```
[ 1+0i , 2+0i , 3+0i , 4+0i ]
```

Konversi ke string hanya mungkin untuk tipe data dasar. Format saat ini digunakan untuk penggabungan string sederhana. Tetapi ada fungsi seperti `print()` atau `frac()`.

Untuk vektor, Anda dapat dengan mudah menulis fungsi sendiri.

```
>function tostr (v) ...  
  
    s="[";  
    loop 1 to length(v);  
        s=s+print(v[#],2,0);  
        if #<length(v) then s=s+","; endif;  
    end;  
    return s+"]";  
endfunction
```

```
>tostr(linspace(0,1,10))
```

```
[0.00,0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1.00]
```

Untuk berkomunikasi dengan Maxima, ada fungsi `convertmxm()`, yang juga dapat digunakan untuk memformat vektor untuk keluaran.

```
>convertmxm(1:10)
```

```
[1,2,3,4,5,6,7,8,9,10]
```

Untuk LaTeX, perintah tex dapat digunakan untuk mendapatkan perintah LaTeX.

```
>tex(&[1,2,3])
```

```
\left[ 1 , 2 , 3 \right]
```

Faktor dan Tabel

Dalam pengantar ke R, ada contoh dengan faktor-faktor yang disebutkan.

Berikut adalah daftar wilayah dari 30 negara bagian.

```
>austates = ["tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", ...  
>"qld", "vic", "nsw", "vic", "qld", "qld", "sa", "tas", ...  
>"sa", "nt", "wa", "vic", "qld", "nsw", "nsw", "wa", ...  
>"sa", "act", "nsw", "vic", "vic", "act"];
```

Misalkan, kita memiliki pendapatan yang sesuai di setiap negara bagian.

```
>incomes = [60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, ...  
>61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46, ...  
>59, 46, 58, 43];
```

Sekarang, kita ingin menghitung rata-rata pendapatan di wilayah-wilayah tersebut. Sebagai program statistik, R memiliki fungsi `factor()` dan `tapply()` untuk ini.

EMT dapat melakukan ini dengan menemukan indeks wilayah di daftar unik wilayah.

```
>auterr=sort(unique(austates)); f=indexofsorted(auterr,austates)
```

```
[6, 5, 4, 2, 2, 3, 8, 8, 4, 7, 2, 7, 4, 4, 5, 6, 5, 3,  
8, 7, 4, 2, 2, 8, 5, 1, 2, 7, 7, 1]
```

Pada saat itu, kita dapat menulis fungsi loop sendiri untuk melakukan sesuatu hanya untuk satu faktor saja.

Atau kita dapat meniru fungsi `tapply()` dengan cara berikut.

```
>function map tappl (i; f$:call, cat, x) ...
```

```
u=sort(unique(cat));  
f=indexof(u,cat);  
return f$(x[nonzeros(f==indexof(u,i))]);  
endfunction
```

Ini agak tidak efisien, karena menghitung wilayah-wilayah unik untuk setiap i, tetapi ini berfungsi.

```
>tappl(auterr,"mean",austates,incomes)
```

```
[44.5, 57.3333333333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perlu dicatat bahwa ini berfungsi untuk setiap vektor wilayah.

```
>tappl(["act","nsw"],"mean",austates,incomes)
```

```
[44.5, 57.3333333333]
```

Sekarang, paket statistik EMT mendefinisikan tabel seperti dalam R. Fungsi readtable() dan writetable() dapat digunakan untuk input dan output.

Jadi kita bisa mencetak rata-rata pendapatan negara bagian di wilayah-wilayah dengan cara yang ramah.

```
>writetable(tappl(auterr,"mean",austates,incomes),labc=auterr,wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Kita juga dapat mencoba meniru perilaku R sepenuhnya.

Faktor-faktor harus jelas disimpan dalam koleksi dengan tipe dan kategori (negara bagian dan wilayah dalam contoh kita). Untuk EMT, kita tambahkan indeks yang telah dihitung sebelumnya.

```
>function makef (t) ...  
  
## Factor data  
## Returns a collection with data t, unique data, indices.  
## See: tapply  
u=sort(unique(t));  
return {{t,u,indexofsorted(u,t)}};  
endfunction
```

```
>statef=makef(austates);
```

Sekarang, elemen ketiga dari koleksi akan berisi indeks.

```
>statef[3]
```

```
[6, 5, 4, 2, 2, 3, 8, 8, 4, 7, 2, 7, 4, 4, 5, 6, 5, 3,  
8, 7, 4, 2, 2, 8, 5, 1, 2, 7, 7, 1]
```

Sekarang kita bisa meniru `tapply()` dengan cara berikut. Ini akan mengembalikan tabel sebagai koleksi data tabel dan judul kolom.

```
>function tapply (t:vector,tf,f$:call) ...
```

```
## Makes a table of data and factors
## tf : output of makef()
## See: makef
uf=tf[2]; f=tf[3]; x=zeros(length(uf));
for i=1 to length(uf);
    ind=nonzeros(f==i);
    if length(ind)==0 then x[i]=NAN;
    else x[i]=f$(t[ind]);
    endif;
end;
return {{x,uf}};
endfunction
```

Kami tidak menambahkan banyak pemeriksaan tipe di sini. Satu-satunya tindakan pencegahan adalah berkaitan dengan kategori (faktor) tanpa data. Namun, seharusnya memeriksa panjang yang benar dari `t` dan kebenaran koleksi `tf`.

Tabel ini dapat dicetak sebagai tabel dengan `writetable()`.

```
>writetable(tapply(incomes,statef,"mean"),wc=7)
```

act	nsw	nt	qld	sa	tas	vic	wa
44.5	57.33	55.5	53.6	55	60.5	56	52.25

Arrays

EMT hanya memiliki dua dimensi untuk array. Tipe data ini disebut matriks. Namun, akan mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau membuat perpustakaan C untuk ini.

Di R, array adalah vektor dengan bidang dimensi.

Di EMT, sebuah vektor adalah matriks dengan satu baris. Ini dapat diubah menjadi matriks dengan `redim()`.

```
>shortformat; X=redim(1:20,4,5)
```

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Pengambilan baris dan kolom, atau sub-matriks, mirip dengan dalam R.

```
>X[,2:3]
```

2	3
7	8
12	13
17	18

Namun, di R, Anda dapat mengatur daftar indeks khusus dari vektor ke suatu nilai. Hal yang sama hanya mungkin di EMT dengan menggunakan perulangan.

```
>function setmatrixvalue (M, i, j, v) ...
```

```
  loop 1 to max(length(i),length(j),length(v))
    M[i{#},j{#}] = v{#};
  end;
endfunction
```

Kami menunjukkan ini untuk menunjukkan bahwa matriks dilewatkan dengan referensi di EMT. Jika Anda tidak ingin mengubah matriks asli M, Anda perlu menyalinnya dalam fungsi.

```
>setmatrixvalue(X,1:3,3:-1:1,0); X,
```

1	2	0	4	5
6	0	8	9	10
0	12	13	14	15
16	17	18	19	20

Produk luar (outer product) dalam EMT hanya dapat dilakukan antara vektor. Ini dilakukan secara otomatis karena bahasa matriks. Salah satu vektor harus menjadi vektor kolom dan yang lainnya vektor baris.

```
>(1:5)*(1:5)'
```

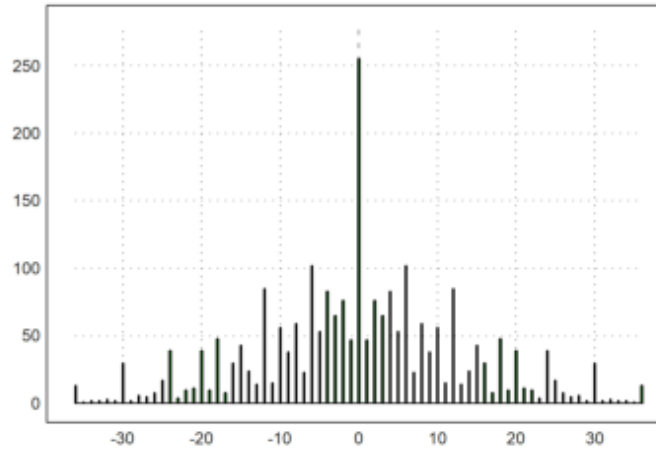
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Di dalam PDF pengantar untuk R terdapat contoh yang menghitung distribusi ab-cd untuk a, b, c, d yang dipilih dari 0 hingga n secara acak. Solusi dalam R adalah dengan menggunakan matriks 4 dimensi dan menjalankan fungsi `table()` di atasnya.

Tentu saja, ini dapat dicapai dengan menggunakan perulangan. Namun, perulangan tidak efektif di EMT atau R. Di EMT, kita bisa menulis perulangan dalam bahasa C dan itu akan menjadi solusi yang paling cepat.

Namun, kita ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat matriks ab-cd.

```
>a=0:6; b=a'; p=flatten(a*b); q=flatten(p-p'); ...  
>u=sort(unique(q)); f=getmultiplicities(u,q); ...  
>statplot(u,f,"h"):
```



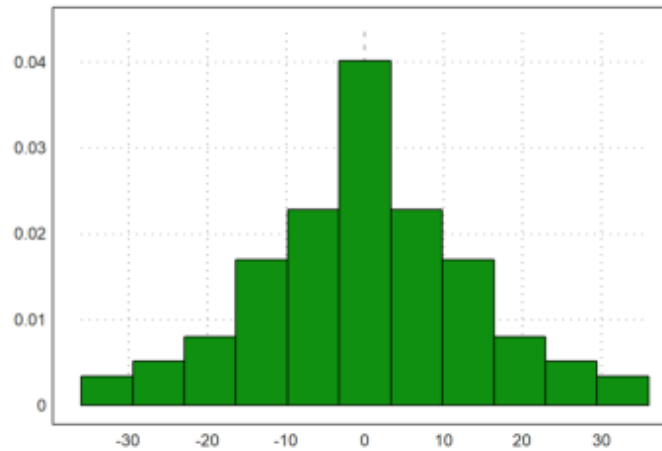
Selain jumlah yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
>getfrequencies(q,-50:10:50)
```

```
[0, 23, 132, 316, 602, 801, 333, 141, 53, 0]
```

Cara paling mudah untuk menggambarannya sebagai distribusi adalah sebagai berikut.

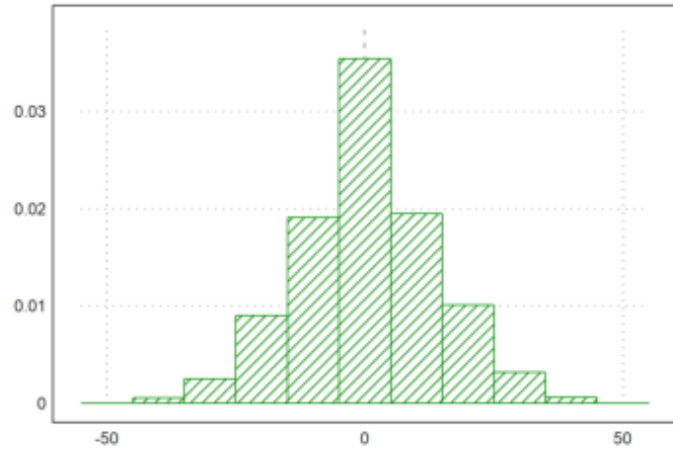
```
>plot2d(q,distribution=11):
```



Tetapi juga mungkin untuk menghitung jumlah dalam interval yang telah dipilih sebelumnya. Tentu saja, berikut ini menggunakan `getfrequencies()` secara internal.

Karena fungsi `histo()` mengembalikan frekuensi, kita perlu menyesuaikan sehingga integral di bawah grafik batangnya adalah 1.

```
>{x,y}=histo(q,v=-55:10:55); y=y/sum(y)/differences(x); ...  
>plot2d(x,y,>bar,style="/"):
```



EMT memiliki dua jenis daftar. Satu adalah daftar global yang dapat diubah, dan yang lainnya adalah tipe daftar yang tidak dapat diubah. Kami tidak membahas tentang daftar global di sini.

Tipe daftar yang tidak dapat diubah disebut koleksi dalam EMT. Ini berperilaku seperti struktur dalam C, tetapi elemennya hanya diberi nomor dan tidak dinamai.

```
>L={{ "Fred", "Flintstone", 40, [1990, 1992] }}
```

```
Fred  
Flintstone  
40  
[1990, 1992]
```

Saat ini, elemen-elemen tidak memiliki nama, meskipun nama dapat diatur untuk tujuan khusus. Mereka diakses dengan nomor.

```
>(L[4])[2]
```

```
1992
```

Input dan Output Berkas (Membaca dan Menulis Data)

Anda seringkali akan ingin mengimpor matriks data dari sumber lain ke EMT. Panduan ini memberi tahu Anda tentang banyak cara untuk mencapai ini. Fungsi sederhana adalah `writematrix()` dan `readmatrix()`.

Mari kita tunjukkan bagaimana cara membaca dan menulis vektor bilangan riil ke berkas.

```
>a=random(1,100); mean(a), dev(a),
```

```
0.54004
```

```
0.28329
```

Untuk menulis data ke berkas, kita menggunakan fungsi `writematrix()`.

Karena pengantar ini kemungkinan besar berada di direktori di mana pengguna tidak memiliki izin menulis, kita menulis data ke direktori beranda pengguna. Untuk notebook Anda sendiri, hal ini tidak perlu, karena berkas data akan ditulis ke dalam direktori yang sama.

```
>filename="test.dat";
```

Sekarang kita menulis vektor kolom \mathbf{a} ke dalam file. Hal ini akan menghasilkan satu angka di setiap baris file tersebut.

```
>writematrix(a',filename);
```

Untuk membaca data tersebut, kita menggunakan fungsi `readmatrix()`.

```
>a=readmatrix(filename)';
```

Dan menghapus file tersebut.

```
>fileremove(filename);  
>mean(a), dev(a),
```

0.54004

0.28329

Fungsi `writematrix()` atau `writetable()` dapat dikonfigurasi untuk bahasa lain.

Contohnya, jika Anda menggunakan sistem berbahasa Indonesia (titik desimal digantikan oleh koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (yang secara default dipisahkan oleh koma). File berikut "test.csv" akan muncul di folder Anda saat ini.

```
>filename="test.csv"; ...  
>writematrix(random(5,3),file=filename,separator=",");
```

Anda sekarang dapat membuka file ini langsung dengan Excel berbahasa Indonesia.

```
>fileremove(filename);
```

Terkadang kita memiliki string dengan token-token seperti contoh berikut.

```
>s1="f m m f m m m f f f m m f"; ...  
>s2="f f f m m f f";
```

Untuk mengonversi ini menjadi token, kita akan mendefinisikan sebuah vektor dari token-token tersebut.

```
>tok=["f","m"]
```

```
f  
m
```

Kemudian kita dapat menghitung berapa kali setiap token muncul dalam string tersebut, dan memasukkan hasilnya ke dalam sebuah tabel.

```
>M:=getmultiplicities(tok,strtokens(s1))_ ...  
> getmultiplicities(tok,strtokens(s2));
```

Menuliskan tabel dengan judul token.

```
>writetable(M,labc=tok,labr=1:2,wc=8)
```

	f	m
1	6	7
2	5	2

Untuk statistik, EMT dapat membaca dan menulis tabel.

```
>file="test.dat"; open(file,"w"); ...  
>writeln("A,B,C"); writematrix(random(3,3)); ...  
>close();
```

File tersebut terlihat seperti ini.

```
>printfile(file)
```

```
A,B,C  
0.3084163350124789,0.7860228753704319,0.3640480404459329  
0.3235527726365187,0.5355164285371529,0.4209486538454125  
0.9647906143251817,0.4305557169007305,0.8295841224094156
```

Fungsi `readtable()` dalam bentuk paling sederhana dapat membaca ini dan mengembalikan kumpulan nilai dan baris judul.

```
>L=readtable(file,>list);
```

Kumpulan ini dapat dicetak menggunakan `writetable()` ke buku catatan, atau ke dalam sebuah file.

```
>writetable(L,wc=10,dc=5)
```

A	B	C
0.30842	0.78602	0.36405
0.32355	0.53552	0.42095
0.96479	0.43056	0.82958

Matriks nilai adalah elemen pertama dari L. Perlu diperhatikan bahwa fungsi mean() dalam EMT menghitung nilai rata-rata dari baris-baris dalam suatu matriks.

```
>mean(L[1])
```

```
0.48616  
0.42667  
0.74164
```

File CSV

Pertama, mari tulis sebuah matriks ke dalam sebuah file. Untuk output, kita akan membuat file dalam direktori kerja saat ini.

```
>file="test.csv"; ...  
>M=random(3,3); writematrix(M,file);
```

Berikut adalah konten dari file tersebut.

```
>printfile(file)
```

```
0.4985755441612197,0.8908902130674888,0.230993153822803  
0.5388022720805338,0.03150264484701902,0.9359045715778547  
0.6011875635483036,0.1012503400474223,0.4840335655691349
```

File CSV ini dapat dibuka pada sistem berbahasa Inggris di Excel dengan mengklik dua kali. Jika Anda mendapatkan file seperti ini di sistem berbahasa Jerman, Anda perlu mengimpor data ke dalam Excel dengan memperhatikan tanda titik desimal.

Namun, tanda titik desimal adalah format default untuk EMT juga. Anda dapat membaca sebuah matriks dari sebuah file dengan menggunakan fungsi `readmatrix()`.

```
>readmatrix(file)
```

```
0.49858    0.89089    0.23099
0.5388    0.031503    0.9359
0.60119    0.10125    0.48403
```

Memungkinkan untuk menulis beberapa matriks ke dalam satu file. Perintah `open()` dapat membuka sebuah file untuk penulisan dengan parameter "w". Nilai default adalah "r" untuk membaca.

```
>open(file,"w"); writematrix(M); writematrix(M'); close();
```

Matriks-matriks tersebut dipisahkan oleh sebuah baris kosong. Untuk membaca matriks-matriks tersebut, buka file dan panggil fungsi `readmatrix()` beberapa kali.

```
>open(file); A=readmatrix(); B=readmatrix(); A==B, close();
```

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


Di Excel atau spreadsheet serupa, Anda dapat mengekspor sebuah matriks sebagai CSV (comma separated values). Di Excel 2007, gunakan "save as" dan "other formats", lalu pilih "CSV". Pastikan bahwa tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut adalah contohnya.

```
>printfile("excel-data.csv")
```

```
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
printfile:
  open(filename,"r");
```

Seperti yang dapat Anda lihat, sistem Jerman saya menggunakan titik koma sebagai pemisah dan tanda desimal koma. Anda dapat mengubah pengaturan ini dalam pengaturan sistem atau di Excel, tetapi ini tidak diperlukan untuk membaca matriks ke dalam EMT.

Cara paling mudah untuk membaca ini ke dalam Euler adalah menggunakan `readmatrix()`. Semua koma akan digantikan oleh titik dengan menggunakan parameter `>comma`. Untuk CSV berbahasa Inggris, cukup abaikan parameter ini.

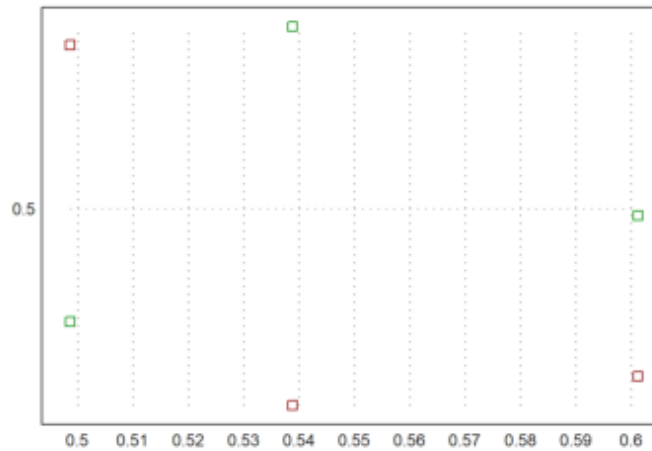
```
>M=readmatrix("excel-data.csv",>comma)
```

```
Could not open the file
excel-data.csv
for reading!
Try "trace errors" to inspect local variables after errors.
```

```
readmatrix:  
  if filename<>" " then open(filename,"r"); endif;
```

Mari memplotnya.

```
>plot2d(M'[1],M'[2:3],>points,color=[red,green] '):
```



Ada cara-cara dasar lain untuk membaca data dari sebuah file. Anda dapat membuka file dan membaca angka-angka satu per satu dari setiap baris. Fungsi `getvectorline()` akan membaca angka-angka dari sebuah baris data. Secara default, fungsi ini mengharapkan titik desimal. Namun, Anda juga dapat menggunakan koma desimal dengan memanggil `setdecimaldot(",")` sebelum Anda menggunakan fungsi ini.

Berikut adalah contoh fungsi untuk membaca data tersebut. Fungsi ini akan berhenti pada akhir file atau baris kosong.

```
>function myload (file) ...
```

```
    open(file);
    M=[];
    repeat
        until eof();
        v=getvectorline(3);
        if length(v)>0 then M=M_v; else break; endif;
    end;
    return M;
    close(file);
endfunction
```

```
>myload(file)
```

0.49858	0	0.89089	0	0.23099
0.5388	0	0.031503	0	0.9359
0.60119	0	0.10125	0	0.48403

Juga memungkinkan untuk membaca semua angka dalam file tersebut menggunakan fungsi `getvector()`.

```
>open(file); v=getvector(10000); close(); redim(v[1:9],3,3)
```

```
0.49858      0    0.89089
      0    0.23099    0.5388
      0    0.031503      0
```

Dengan demikian, sangat mudah untuk menyimpan vektor nilai, satu nilai dalam setiap baris, dan membaca kembali vektor ini.

```
>v=random(1000); mean(v)
```

```
0.49726
```

```
>writematrix(v',file); mean(readmatrix(file)')
```

```
0.49726
```

Penggunaan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kita akan menulis sebuah tabel dengan judul baris dan kolom ke dalam sebuah file.

```
>file="test.tab"; M=random(3,3); ...  
>open(file,"w"); ...  
>writetable(M,separator=",",labc=["one","two","three"]); ...  
>close(); ...  
>printfile(file)
```

one	two	three
0.58,	0.69,	0.89
0.88,	0.43,	0.71
0.18,	1,	0.35

Ini dapat diimpor ke dalam Excel.

Untuk membaca file tersebut di EMT, kita menggunakan fungsi `readtable()`.

```
>{M,headings}=readtable(file,>clabs); ...  
>writetable(M,labc=headings)
```

one	two	three
0.58	0.69	0.89
0.88	0.43	0.71
0.18	1	0.35

Menganalisis Sebuah Baris

Anda bahkan dapat mengevaluasi setiap baris secara manual. Misalkan, kita memiliki sebuah baris dengan format berikut.

```
>line="2020-11-03,Tue,1'114.05"
```

```
2020-11-03,Tue,1'114.05
```

Pertama, kita dapat melakukan tokenisasi terhadap baris tersebut.

```
>vt=strtokens(line)
```

```
2020-11-03
```

```
Tue
```

```
1'114.05
```

Kemudian, kita dapat mengevaluasi setiap elemen dari baris tersebut menggunakan evaluasi yang sesuai.

```
>day(vt[1]), ...  
>indexof(["mon","tue","wed","thu","fri","sat","sun"],tolower(vt[2])), ...  
>strrepl(vt[3],"',"")()
```

```
7.3816e+05  
2  
1114
```

Dengan menggunakan regular expressions, mungkin untuk mengekstrak hampir semua informasi dari sebuah baris data.

Misalkan kita memiliki baris berikut dalam sebuah dokumen HTML.

```
>line("<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>"
```

```
<tr><td>1145.45</td><td>5.6</td><td>-4.5</td><tr>
```

Untuk mengekstrak ini, kita menggunakan ekspresi reguler, yang mencari:

- tanda kurung penutup > ,
- string apa pun yang tidak mengandung tanda kurung dengan sub-pencocokan "(...)",
- tanda kurung buka dan tanda kurung tutup dengan solusi terpendek,
- lagi string apa pun yang tidak mengandung tanda kurung,
- dan tanda kurung buka < .

Ekspresi reguler agak sulit untuk dipelajari tetapi sangat kuat.

```
>{pos,s,vt}=strxfind(line,">([<>]+)<.+?>([<>]+)<");
```

The result is the position of the match, the matched string, and a vector of strings for sub-matches.

```
>for k=1:length(vt); vt[k](), end;
```

1145.5

5.6

Berikut adalah contoh fungsi yang membaca semua item numerik antara <td> dan </td>.

```
>function readtd (line) ...
```

```
v=[]; cp=0;
repeat
    {pos,s,vt}=strxfind(line,"<td.*?>(.*?)</td>",cp);
    until pos==0;
    if length(vt)>0 then v=v|vt[1]; endif;
    cp=pos+strlen(s);
end;
return v;
endfunction
```

```
>readtd(line+"<td>non-numerical</td>")
```

```
1145.45
5.6
-4.5
non-numerical
```

Membaca dari Web

Sebuah situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris per baris.

Pada contoh ini, kita membaca versi terbaru dari situs EMT. Kita menggunakan ekspresi reguler untuk mencari "Versi ..." dalam judul.

```
>function readversion () ...
```

```
    urlopen("http://www.euler-math-toolbox.de/Programs/Changes.html");  
    repeat  
        until urleof();  
        s=urlgetline();  
        k=strfind(s,"Version ",1);  
        if k>0 then substring(s,k,strfind(s,"<",k)-1), break; endif;  
    end;  
    urlclose();  
endfunction
```

```
>readversion
```

Version 2022-05-18

Input dan Output Variabel

Anda dapat menulis sebuah variabel dalam bentuk definisi Euler ke sebuah file atau ke baris perintah.

```
>writevar(pi,"mypi");
```

```
mypi = 3.141592653589793;
```

Untuk uji coba, kita akan menghasilkan sebuah file Euler di direktori kerja EMT.

```
>file="test.e"; ...  
>writevar(random(2,2),"M",file); ...  
>printfile(file,3)
```

```
M = [ ..  
      0.1715864118118049, 0.7902075714281032;  
      0.3235267463604796, 0.08344844686478282];
```

Sekarang kita dapat memuat file tersebut. Ini akan mendefinisikan matriks M.

```
>load(file); show M,
```

```
M =  
  0.17159    0.79021  
  0.32353    0.083448
```

Sebagai informasi tambahan, jika fungsi `writevar()` digunakan pada sebuah variabel, itu akan mencetak definisi variabel dengan nama variabel tersebut.

```
>writevar(M); writevar(inch$)
```

```
M = [ ..  
  0.1715864118118049, 0.7902075714281032;  
  0.3235267463604796, 0.08344844686478282];  
inch$ = 0.0254;
```

Kita juga dapat membuka sebuah file baru atau menambahkan ke dalam file yang sudah ada. Pada contoh ini, kita menambahkan ke dalam file yang telah dihasilkan sebelumnya.

```
>open(file,"a"); ...  
>writevar(random(2,2),"M1"); ...  
>writevar(random(3,1),"M2"); ...  
>close();  
>load(file); show M1; show M2;
```

```
M1 =  
  0.64914  0.13714  
  0.112   0.43951  
M2 =  
  0.77904  
  0.52075  
  0.88929
```

Untuk menghapus file-file, gunakan fungsi `fileremove()`.

```
>fileremove(file);
```

Sebuah vektor baris dalam sebuah file tidak memerlukan koma jika setiap angka berada di baris baru. Mari kita hasilkan file seperti itu, menulis setiap baris satu per satu dengan menggunakan `writeln()`.

```
>open(file,"w"); writeln("M = ["); ...  
>for i=1 to 5; writeln(""+random()); end; ...  
>writeln("];"); close(); ...  
>printfile(file)
```

```
M = [  
0.441463853011  
0.602559586157  
0.8008250194  
0.624852131639  
0.53481766277  
];
```

```
>load(file); M
```

```
[0.44146, 0.60256, 0.80083, 0.62485, 0.53482]
```

Contoh Soal

Contoh Soal 1

Banyaknya perawat di 6 klinik adalah 3,5,6,4,5, dan 6. Dengan memandang data itu sebagai data populasi, hitunglah nilai rata-rata banyaknya perawat di 6 klinik tersebut!

Penyelesaian:

```
>x=[3,5,6,4,5,6]; mean(x),
```

4.8333

Jadi, nilai rata-rata banyaknya pegawai di 6 klinik tersebut adalah 4.83

Contoh Soal 2

1. Data berikut menunjukkan tinggi badan dari 20 siswa SMA 2 Surabaya.

Siswa yang tinggi badannya dalam rentang 145-150 sebanyak 1 orang, dalam rentang 151-155 sebanyak 2 orang, dalam rentang 156-160 sebanyak 4 orang, dalam rentang 161-165 sebanyak 3 orang, dalam rentang 166-170 sebanyak 3 orang, dalam rentang 171-175 sebanyak 4 orang, dan dalam rentang 176-180 sebanyak 3 orang.

Tentukan rata-rata tinggi badan dari 20 siswa tersebut!

Penyelesaian:

Menentukan tepi bawah kelas yang terkecil

$$>146-0.5$$

$$145.5$$

Menentukan panjang kelas

$$>(150-146)+1$$

$$5$$

Menentukan tepi atas kelas yang terbesar

```
>180+0.5
```

180.5

```
>r=145.5:5:180.5; v=[1,2,4,3,3,4,3];  
>T=r[1:7]' | r[2:8]' | v'; writetable(T,labc=["TB","TA","Frek"])
```

TB	TA	Frek
145.5	150.5	1
150.5	155.5	2
155.5	160.5	4
160.5	165.5	3
165.5	170.5	3
170.5	175.5	4
175.5	180.5	3

Menentukan titik tengah

```
>(T[,1]+T[,2])/2 //titik tengah dari tiap interval
```

148
153
158

```
163  
168  
173  
178
```

```
>t=fold(r,[0.5,0.5])
```

```
[148, 153, 158, 163, 168, 173, 178]
```

```
>mean(t,v)
```

```
165.25
```

Jadi, nilai rata-rata tinggi badan dari 20 siswa tersebut adalah 165.25.

Contoh 3

Misalnya kita akan menghitung nilai rata-rata(mean) yang terdapat dalam file "test.dat"

```
>filename="test.dat"; ...  
>V=random(3,3); writematrix(V,filename);  
>printfile(filename),
```

```
0.1858294139595237,0.2359542823657924,0.2704760003790539  
0.82636390358253,0.156173368112527,0.9576388301324568  
0.04846986155954107,0.8722870876753714,0.5779228155144673
```

```
>readmatrix(filename)
```

```
0.18583      0  0.23595      0  0.27048  
0.82636      0  0.15617      0  0.95764  
0.04847      0  0.87229      0  0.57792
```

```
>mean(V),
```

```
0.23075  
0.64673  
0.49956
```

Contoh 4

Disajikan data urut yaitu 45,48,49,50,52,52,52,53,53,54,54,54,54,54,56,56, 56,56,57,57,58,58,58,58,58,58,58,59,59,60,60,60, 62,62,62,63,63,64,64,65,67,68,69,70,70,71,73,74.

Buatlah distribusi frekuensi berdasarkan data diatas!

Penyelesaian:

- Menentukan range
range= nilai maks-nilai min
= 74-45
= 29
- Menentukan banyak kelas dengan aturan struges.
= $1+3,3 \log n$, n banyaknya data
= $1+3,3 \log 48$
= 6,64
= 7
- Menentukan panjang kelas

$$p = \frac{range}{banyakkelas}$$

$$p = \frac{29}{7}$$

$$p = 4.14 = 5$$

Berdasarkan pertimbangan beberapa unsur dalam data urut diatas yaitu nilai minimum 45, nilai maksimum 74, banyak kelas yaitu 7, dan panjang kelas yaitu 5 maka dapat dibuat tabel distribusi frekuensi dengan batas bawah kelas pertama yaitu 43 dan batas atas kelas ketujuh yaitu 77. Sehingga dapat ditentukan tepi bawah kelas pertama yaitu $43-0.5=42.5$ dan tepi atas kelas ketujuh yaitu $77+0.5=77.5$.

```
>r=42.5:5:77.5; v=[1,6,13,15,6,5,2];
>T:=r[1:7]' | r[2:8]' | v'; writetable(T,labc=["TB","TA","Frek"])
```

TB	TA	Frek
42.5	47.5	1
47.5	52.5	6
52.5	57.5	13
57.5	62.5	15
62.5	67.5	6
67.5	72.5	5
72.5	77.5	2

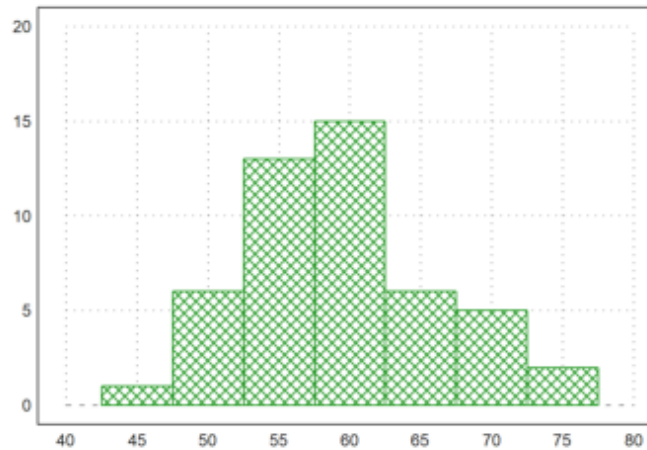
Mencari titik tengah

```
>(T[,1]+T[,2])/2 // titik tengah tiap interval
```

```
45
50
55
60
65
70
75
```

Sajian dalam bentuk histogram

```
>plot2d(r,v,a=40,b=80,c=0,d=20,bar=1,style="/"):
```



Contoh 5

Berikut daftar ukuran sepatu anak kelas 5 SDN 1 Makassar.

39,35,35,36,37,38,42,40,38,41,37,35,38,40,41,40.

Tentukan median dari data tersebut!

Penyelesaian :

```
>data=[39,35,35,36,37,38,42,40,38,41,37,35,38,40,41,40];  
>urut=sort(data)
```

```
[35, 35, 35, 36, 37, 37, 38, 38, 38, 39, 40, 40, 40, 41,  
41, 42]
```

```
>median(data)
```

38

Jadi, median dari data tersebut adalah 38.

Berikut adalah data hasil dari pengukuran berat badan 50 siswa SDN 4 Banten. Siswa yang mempunyai berat badan dalam rentang 21-26 kg sebanyak 6 orang, yang mempunyai berat badan dalam rentang 27-32 kg sebanyak 9 orang, yang mempunyai berat badan dalam rentang 33-38 kg sebanyak 14 orang, yang mempunyai berat badan dalam rentang 39-44 kg sebanyak 12 orang, yang mempunyai berat badan dalam rentang 45-50 kg sebanyak 7 orang, dan yang mempunyai berat badan 51-56 kg sebanyak 2 orang. Tentukan median dari data hasil pengukuran berat badan 50 siswa di SD tersebut!

Penyelesaian:

Menentukan tepi bawah kelas yang terkecil

$$>21-0.5$$

$$20.5$$

Menentukan panjang kelas

$$>(26-21)+1$$

$$6$$

Menentukan tepi atas kelas yang terbesar

```
>56+0.5
```

56.5

```
>r=20.5:6:56.5; v=[6,9,14,12,7,2];  
>T:=r[1:6]' | r[2:7]' | v'; writetable(T,labc=["TB","TA","frek"])
```

TB	TA	frek
20.5	26.5	6
26.5	32.5	9
32.5	38.5	14
38.5	44.5	12
44.5	50.5	7
50.5	56.5	2

Berdasarkan data, median berada pada urutan ke 25, maka median berada pada kelas 32.5-38.5.

```
>Tb=32.5, p=6, n=50, Fks=15, fm=14
```

32.5
6
50
15
14

$$>Tb+p*(1/2*n-Fks)/fm$$

36.786

Jadi, median dari data hasil pengukuran berat badan 50 siswa SDN 4 Banten adalah 36.785.

Contoh 7

Berikut adalah data hasil dari pengukuran berat badan 30 siswa SDN 5 Jember. Siswa yang mempunyai berat badan dalam rentang 21-25 kg sebanyak 1 orang, yang mempunyai berat badan dalam rentang 26-30 kg sebanyak 8 orang, yang mempunyai berat badan dalam rentang 31-35 kg sebanyak 10 orang, yang mempunyai berat badan dalam rentang 36-40 kg sebanyak 5 orang, yang mempunyai berat badan dalam rentang 41-45 kg sebanyak 4 orang, dan yang mempunyai berat badan 46-50 kg sebanyak 2 orang. Tentukan modus dari data hasil pengukuran berat badan 30 siswa di SD tersebut!

Penyelesaian:

Menentukan tepi bawah kelas yang terkecil

$$>21-0.5$$

20.5

Menentukan panjang kelas

```
>(25-21)+1
```

5

Menentukan tepi atas yang terbesar

```
>50+0.5
```

50.5

```
>r=20.5:5:50.5; v=[1,8,10,5,4,2];  
>T:=r[1:6]' | r[2:7]' | v'; writetable(T,labc=["TB","TA","frek"])
```

TB	TA	frek
20.5	25.5	1
25.5	30.5	8
30.5	35.5	10
35.5	40.5	5
40.5	45.5	4
45.5	50.5	2

Berdasarkan data, modus berada pada kelas 30.5-35.5.

```
>Tb=30.5, p=5, d1=2, d2=5
```

```
30.5  
5  
2  
5
```

```
>Tb+p*d1/(d1+d2)
```

```
31.929
```

Jadi modus dari data hasil pengukuran berat badan 30 siswa di SDN 5 Jember adalah 31.92.

Contoh 8

Diketahui data sebagai berikut.

11,44,34,51,36,21,23,24,26,27,15,14,16,18,19,20,33,39,45,41,43

Tentukan kuartil, desil, dan persentil dari data tersebut!

Penyelesaian :

```
>data=[11,44,34,51,36,21,23,24,26,27,15,14,16,18,19,20,33,39,45,41,43];  
>urut=sort(data)
```

```
[11, 14, 15, 16, 18, 19, 20, 21, 23, 24, 26, 27, 33, 34,  
36, 39, 41, 43, 44, 45, 51]
```

```
>quartiles(data)
```

```
[11, 18.5, 26, 40, 51]
```

Dalam output hitung yang dihasilkan dari 'quartiles(data)' dapat diketahui bahwa nilai Q1(kuartil bawah) = 18.5 , Q2(kuartil tengah(median)) = 26, dan Q3(kuartil atas)= 40. Lalu untuk nilai paling kanan dan paling kiri merupakan minimum dan maximum dari suatu data yang diketahui.

```
>quantile(urut,0.1)
```

```
15
```

Dari hasil tersebut dapat diketahui bahwa nilai desil ke-1 dan persentil ke-10 adalah 15.

Diketahui data sebagai berikut.

87,78,88,77,66,67,76

Tentukan varians dan simpangan baku dari data berikut.

```
>data=[87,78,88,77,66,67,76];  
>urut=sort(data)
```

[66, 67, 76, 77, 78, 87, 88]

```
>a=mean(urut)
```

77

```
>dev=urut-a
```

[-11, -10, -1, 0, 1, 10, 11]

```
>varians=mean(dev^2)
```

63.429

```
>simpanganBaku= sqrt(varians)
```

```
7.9642
```

Jadi, variansnya adalah 63.42 dan simpangan bakunya adalah 7.96.