

Nama: Nasywa Azizah Luthfia Hardanis  
NIM: 22305141051  
Kelas: Matematika B

---

## EMT untuk Statistika

---

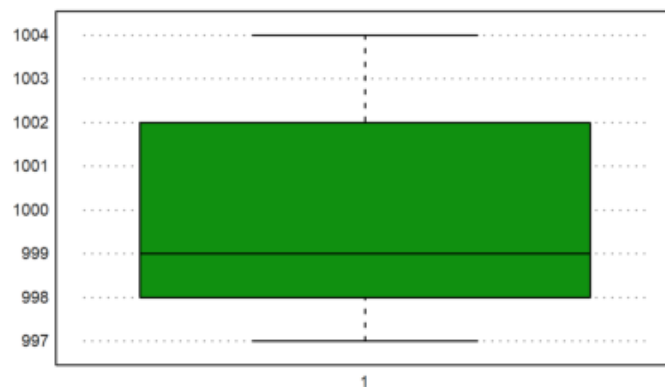
Dalam buku catatan ini, kami mendemonstrasikan plot statistik utama, uji, dan distribusi dalam Euler. Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi Anda mungkin perlu beberapa pengetahuan dasar untuk memahami detailnya. Asumsikan pengukuran berikut. Kami ingin 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 box-and-whiskers untuk data ini. Dalam kasus kita, tidak ada pencilan (outliers).

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



Kita menghitung probabilitas bahwa sebuah nilai lebih besar dari 1005, dengan asumsi bahwa nilai-nilai yang diukur berasal dari distribusi normal.

Semua fungsi untuk distribusi dalam Euler berakhir dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

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

Kita akan mencetak hasilnya dalam bentuk persentase dengan akurasi 2 digit menggunakan fungsi print.

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

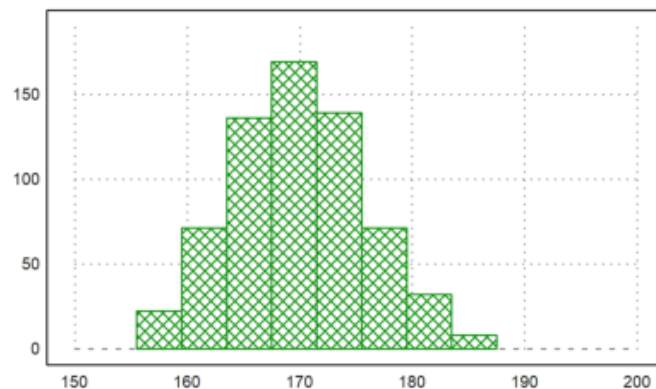
```
3.07 %
```

Untuk contoh selanjutnya, kita akan berasumsi dengan jumlah pria dalam rentang ukuran tertentu adalah sebagai berikut.

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

Berikut adalah plot dari distribusi tersebut.

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



Kita dapat memasukkan data mentah seperti ini ke dalam sebuah tabel.

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

Tabel dapat dicetak dengan menggunakan header. Kita dapat menggunakan vektor string untuk menentukan header.

```
>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 memerlukan nilai rata-rata dan statistik lainnya dari ukuran-ukuran tersebut, kita perlu menghitung titik tengah dari rentangnya. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk ini.

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

```
>(T[,1]+T[,2])/2 // the midpoint of each interval
```

```

157.5
161.5
165.5
169.5
173.5
177.5
181.5
185.5

```

Tetapi lebih mudah jika kita menggunakan vektor [1/2,1/2] untuk menggabungkan (fold) rentangnya.

```
>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 rata-rata dan deviasi dari sampel dengan frekuensi yang diberikan.

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

```

169.901234568
5.98912964449

```

Mari kita tambahkan distribusi normal dari nilai ke plot batang di atas. Rumus untuk distribusi normal dengan mean (m) dan standar deviasi(d) adalah:

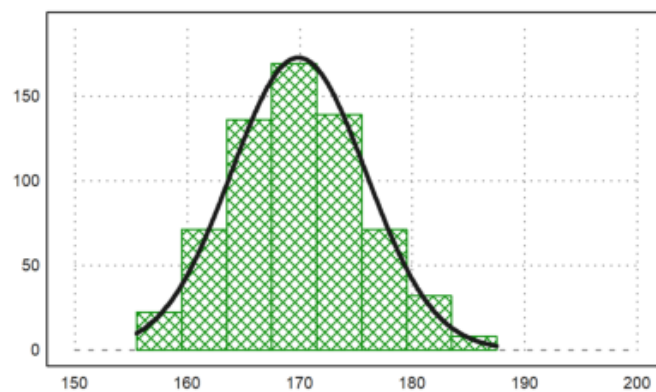
$$y = \frac{1}{d\sqrt{2\pi}} e^{-\frac{(x-m)^2}{2d^2}}.$$

Karena nilainya berada antara 0 dan 1, untuk memplotnya pada plot batang, kita harus mengalikannya dengan 4 kali jumlah total data.

```

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

```



## Tabel

---

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

```
>printfile("table.dat",4);
```

```
Person Sex Age Titanic Evaluation Tip Problem
1 m 30 n . 1.80 n
2 f 23 y g 1.80 n
3 f 26 y g 1.80 y
```

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

Untuk ini, kita akan mendefinisikan kumpulan token. Fungsi `strtokens()` akan mendapatkan vektor string dari token-token dari 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 dalam tabel. Argumen-argumen ini tidak ada dalam daftar parameter fungsi `readtable()`, sehingga Anda perlu menyediakannya dengan `:=`.

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

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n

Tanda titik "." mewakili nilai yang tidak tersedia.

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

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

Fungsi `readtable()` sekarang mengembalikan kumpulan token.

```
>tok
```

```
m  
n  
f  
y  
g  
vg
```

Tabel berisi entri dari file dengan token diterjemahkan menjadi angka.

String khusus NA="." diinterpretasikan sebagai "Not Available" dan diberikan nilai NAN (bukan angka) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA dan NAval.

```
>MT[1]
```

```
[1, 1, 30, 2, NAN, 1.8, 2]
```

Berikut ini adalah isi tabel dengan angka yang belum diterjemahkan.

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

1	1	30	2	.	1.8	2
2	3	23	4	5	1.8	2
3	3	26	4	5	1.8	4
4	1	33	2	.	2.8	2
5	1	37	2	.	1.8	2
6	1	28	4	5	2.8	4
7	3	31	4	6	2.8	2
8	1	23	2	.	0.8	2
9	3	24	4	6	1.8	4
10	1	26	2	.	1.8	2
11	3	23	4	6	1.8	4
12	1	32	4	5	1.8	2
13	1	29	4	6	1.8	4
14	3	25	4	5	1.8	4
15	3	31	4	5	0.8	2
16	1	26	4	5	2.8	2
17	1	37	2	.	3.8	2
18	1	38	4	5	.	2
19	3	29	2	.	3.8	2
20	3	28	4	6	1.8	2
21	3	28	4	1	2.8	4
22	3	28	4	6	1.8	4
23	3	38	4	5	2.8	2
24	3	27	4	1	1.8	4
25	1	27	2	.	2.8	4

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

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

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

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

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
1	m	30	n	.	1.8	n
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
4	m	33	n	.	2.8	n
5	m	37	n	.	1.8	n
6	m	28	y	g	2.8	y
7	f	31	y	vg	2.8	n
8	m	23	n	.	0.8	n
9	f	24	y	vg	1.8	y
10	m	26	n	.	1.8	n
11	f	23	y	vg	1.8	y
12	m	32	y	g	1.8	n
13	m	29	y	vg	1.8	y
14	f	25	y	g	1.8	y
15	f	31	y	g	0.8	n
16	m	26	y	g	2.8	n
17	m	37	n	.	3.8	n
18	m	38	y	g	.	n
19	f	29	n	.	3.8	n
20	f	28	y	vg	1.8	n
21	f	28	y	m	2.8	y
22	f	28	y	vg	1.8	y
23	f	38	y	g	2.8	n
24	f	27	y	m	1.8	y
25	m	27	n	.	2.8	y

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

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

Kita dapat menggunakan ini untuk mengekstrak kolom dari tabel ke dalam tabel baru.

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

Person	Evaluation	Tip
2	g	1.8
3	g	1.8
6	g	2.8
7	vg	2.8
9	vg	1.8
11	vg	1.8
12	g	1.8
13	vg	1.8
14	g	1.8
15	g	0.8
16	g	2.8

20	vg	1.8
21	m	2.8
22	vg	1.8
23	g	2.8
24	m	1.8

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

```
>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))
```

2.175

Fungsi `getstatistics()` mengembalikan elemen-elemen dalam vektor dan hitungannya. Kita menerapkannya pada nilai-nilai "m" dan "f" dalam kolom kedua tabel kita.

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

```
[1, 3]
[12, 13]
```

Kita dapat mencetak hasilnya dalam tabel baru.

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

m	12
f	13

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

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

```
[5, 6]
```

Sekarang kita dapat memilih baris-baris dari tabel yang memiliki salah satu dari nilai dalam v di baris ke-5.

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

Kemudian kita dapat mencetak tabel dengan nilai yang diekstrak dan diurutkan dalam kolom ke-5.

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

Person	Sex	Age	Titanic	Evaluation	Tip	Problem
2	f	23	y	g	1.8	n
3	f	26	y	g	1.8	y
6	m	28	y	g	2.8	y
18	m	38	y	g	.	n
16	m	26	y	g	2.8	n
15	f	31	y	g	0.8	n
12	m	32	y	g	1.8	n
23	f	38	y	g	2.8	n
14	f	25	y	g	1.8	y
9	f	24	y	vg	1.8	y
7	f	31	y	vg	2.8	n
20	f	28	y	vg	1.8	n
22	f	28	y	vg	1.8	y
13	m	29	y	vg	1.8	y
11	f	23	y	vg	1.8	y

Untuk statistik berikutnya, kita ingin menghubungkan dua kolom dalam tabel. Jadi kita akan mengekstrak kolom 2 dan 4, lalu mengurutkan tabel.

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

m	n
m	n
m	n
m	n
m	n
m	n
m	n
m	y
m	y
m	y
m	y
m	y
f	n
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y
f	y

Dengan `getstatistics()`, kita juga dapat menghubungkan jumlah 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])
```



	n	y
m	7	5
f	1	12

Tabel dapat ditulis ke dalam sebuah file.

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

Kemudian kita dapat membaca tabel dari file tersebut.

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

	n	y
m	7	5
f	1	12

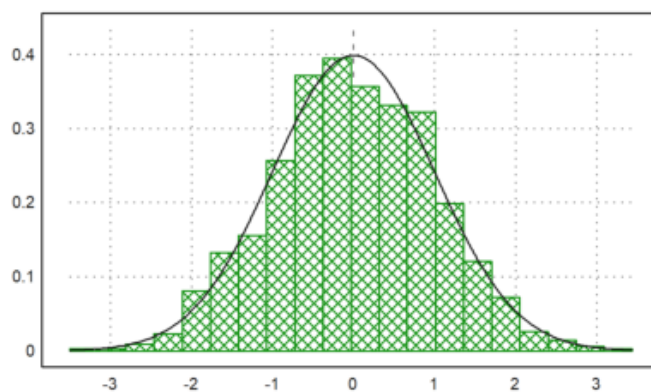
Dan menghapus file tersebut.

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

## Distribusi

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

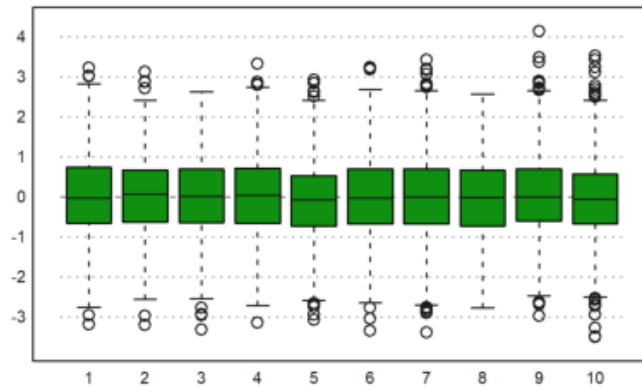
```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="\|"); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution plot
```



Perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi sebenarnya). Coba lagi tiga perintah tersebut untuk melihat hasil sampel yang berbeda.

Berikut adalah perbandingan dari 10 simulasi 1000 nilai yang terdistribusi normal menggunakan plot kotak (box plot). Plot ini menunjukkan median, kuartil ke-25 dan ke-75, nilai minimum dan maksimum, serta pencilan (outliers).

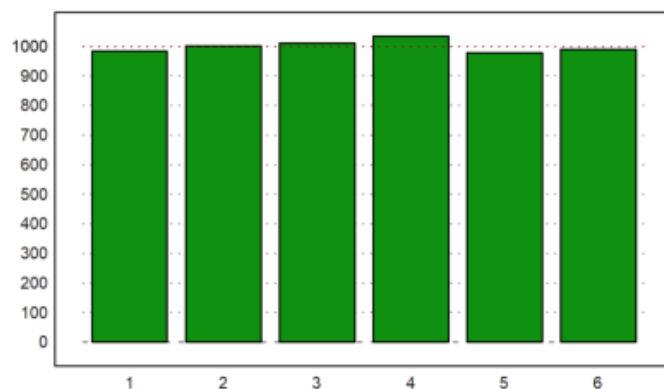
```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan angka acak, Euler memiliki `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 `columnplot()`.

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



Meskipun `intrandom(n,m,k)` menghasilkan bilangan bulat yang terdistribusi seragam antara 1 hingga `k`, kita juga bisa menggunakan distribusi bilangan bulat lainnya dengan `randpint()`.

Pada contoh berikut, probabilitas untuk 1,2,3 adalah masing-masing 0.4, 0.1, 0.5.

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

```
[378, 102, 520]
```

Euler dapat menghasilkan nilai acak dari berbagai distribusi lainnya. Anda dapat melihat referensi untuk lebih lanjut.

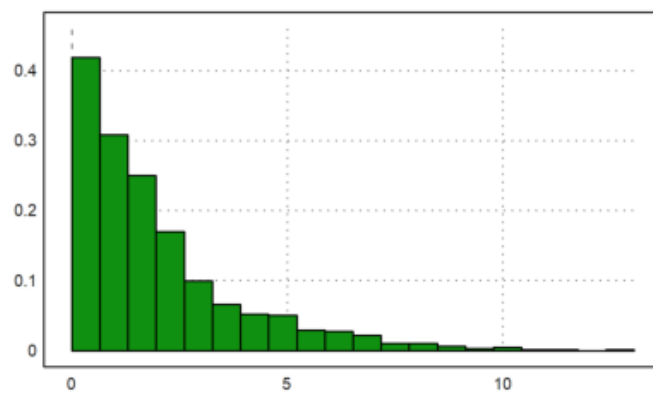
Misalnya, kita mencoba distribusi eksponensial. Variabel acak kontinu  $X$  dikatakan memiliki distribusi eksponensial jika PDF-nya diberikan oleh... (informasi tambahan tidak disediakan).

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

dengan parameter

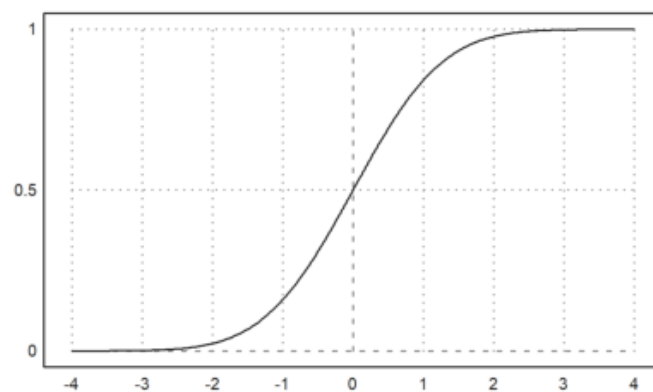
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

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



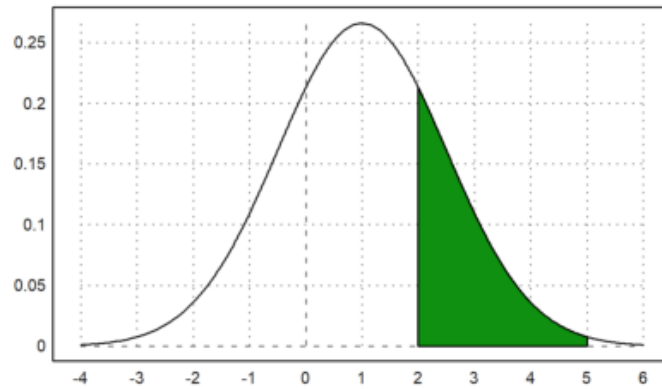
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversenya.

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



Berikut adalah salah satu cara untuk membuat plot quantile.

```
>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 dalam area hijau dapat dihitung dengan metode 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 kita bandingkan distribusi binomial dengan distribusi normal dengan mean dan deviasi yang sama. Fungsi `invbindis()` menyelesaikan interpolasi linear antara nilai-nilai bulat.

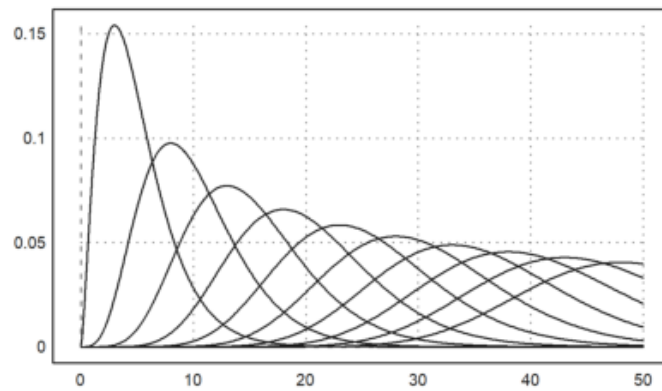
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219
```

```
526.007419394
```

Fungsi `qdis()` adalah densitas dari distribusi chi-kuadrat. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian, kita dapat dengan mudah mendapatkan plot semua distribusi chi-kuadrat 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 integral. Penamaan fungsi mencoba untuk konsisten. Misalnya,

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

Komplemen dari distribusi (upper tail) adalah `chicdis()`.

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

```
0.527633447259
0.527633447259
```

## Distribusi Diskrit

Untuk mendefinisikan distribusi diskrit sendiri, Anda dapat menggunakan metode berikut. Pertama, kita akan menetapkan 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]
```

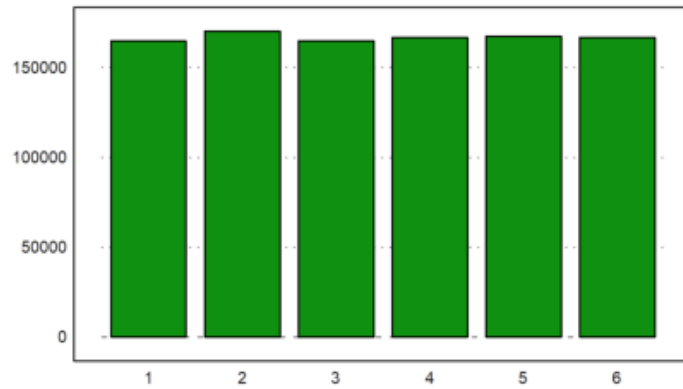
Dalam arti tersebut, dengan probabilitas `wd[i+1]-wd[i]` kita menghasilkan nilai acak `i`.

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

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

Kesalahan ini sangat halus sehingga kita hanya melihatnya dengan iterasi yang sangat banyak.

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



Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai-nilai 1...K dalam v. Kami 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 ini menolak distribusi seragam.

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

0

Dan ia menerima generator acak bawaan.

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

1

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

```
>bindis(410,1000,0.4)
```

0.751401349654

Fungsi invers Beta digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter  $p$ . Level defaultnya adalah  $\alpha$ .

Arti dari interval ini adalah bahwa jika  $p$  berada di luar interval, hasil yang diamati 410 dari 1000 adalah jarang.

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

```
[0.37932, 0.441212]
```

Perintah-perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Tetapi untuk  $n$  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
```

Omong-omong, `invbinsum()` menghitung invers dari `binomialsum()`.

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

```
409.932733047
```

Dalam permainan Bridge, kita mengasumsikan 5 kartu luar biasa (dari 52) dalam dua tangan (26 kartu). Mari kita menghitung probabilitas distribusi yang lebih buruk dari 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])
```

381	100	519
376	91	533
417	80	503
440	94	466
406	112	482
408	94	498
395	107	498
399	96	505
428	87	485
400	99	501

## Plot Data

---

Untuk mem-plotting data, kita mencoba hasil-hasil pemilu di Jerman sejak tahun 1990, diukur dalam jumlah 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 rangkaian nama-nama dalam bentuk string.

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

Mari kita cetak persentase dengan rapi.

Pertama, kita ekstrak kolom-kolom yang diperlukan. Kolom 3 hingga 7 adalah jumlah kursi untuk setiap partai, dan kolom 2 adalah total jumlah kursi. Kolom adalah tahun dari pemilu.

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

Kemudian kita cetak statistik dalam bentuk tabel. Kita menggunakan nama-nama sebagai header kolom, dan tahun sebagai header baris. Lebar default untuk kolom adalah wc=10, tetapi kita lebih suka tampilan 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

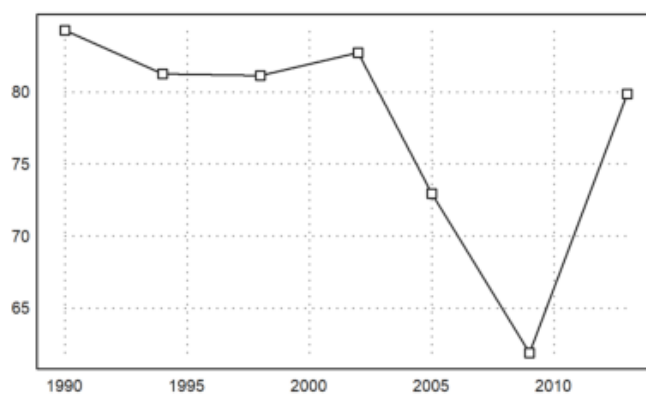
Perkalian matriks berikut mengekstrak jumlah persentase dari dua partai besar yang menunjukkan bahwa partai-partai kecil telah mendapatkan kursi di parlemen hingga tahun 2009.

```
>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 memanggil plot2d dua kali dengan >add.

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



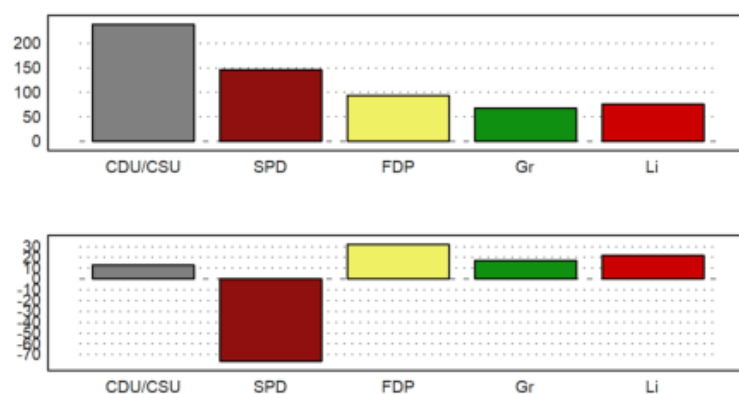


Tentukan beberapa warna untuk setiap partai.

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

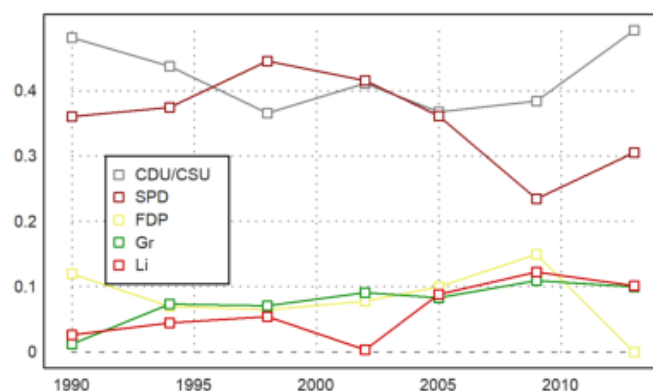
Sekarang kita dapat membuat plot hasil pemilu tahun 2009 dan perubahannya dalam satu plot menggunakan figure. Kita dapat menambahkan vektor kolom ke setiap plot.

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



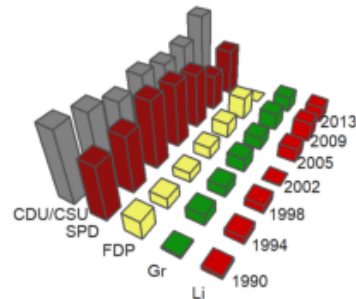
Plot data menggabungkan baris data statistik dalam satu plot.

```
>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):
```



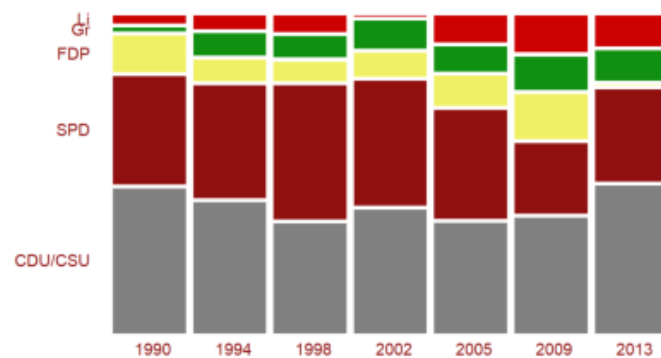
Plot kolom 3D menunjukkan baris data statistik dalam bentuk kolom. Kita menyediakan label-label untuk baris dan kolom-kolom. Angle adalah sudut pandang.

```
>columnsplot3d(BT,scols=P,srows=YT, ...
> angle=30°,ccols=CP):
```



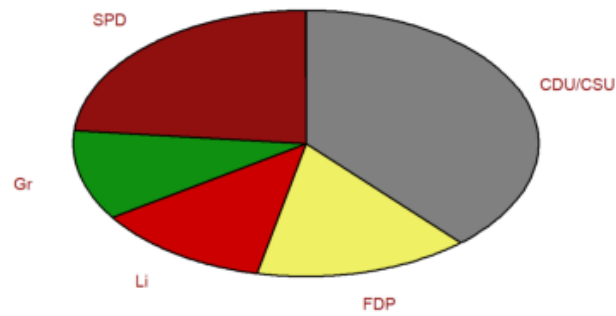
Representasi lainnya adalah plot mozaik. Perhatikan bahwa kolom-kolom plot ini mewakili kolom-kolom matriks di sini. Karena panjang label CDU/CSU, kita menggunakan jendela yang lebih kecil dari biasanya.

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



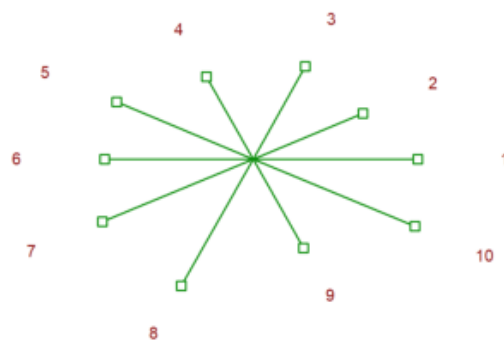
Kita juga dapat membuat diagram lingkaran (pie chart). Karena hitam dan kuning membentuk koalisi, kita mengurutkan ulang elemen-elemennya.

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



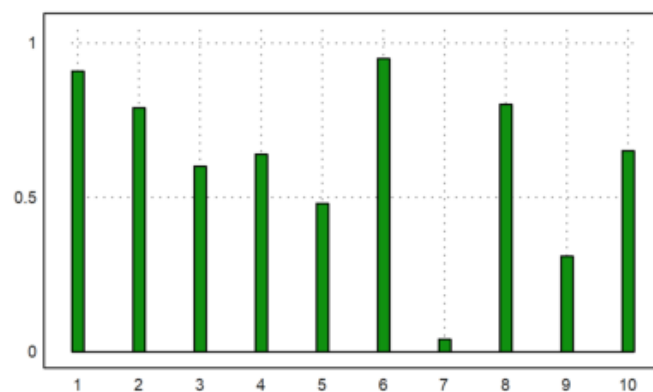
Berikut adalah jenis plot lainnya.

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



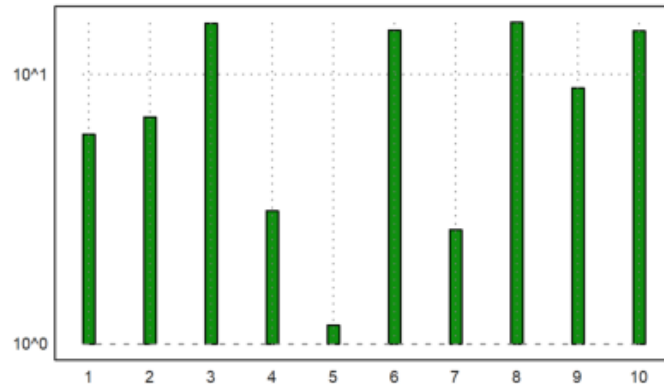
Beberapa plot dalam plot2d bagus untuk statistik. Berikut adalah plot impuls dari data acak yang terdistribusi secara seragam di  $[0,1]$ .

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



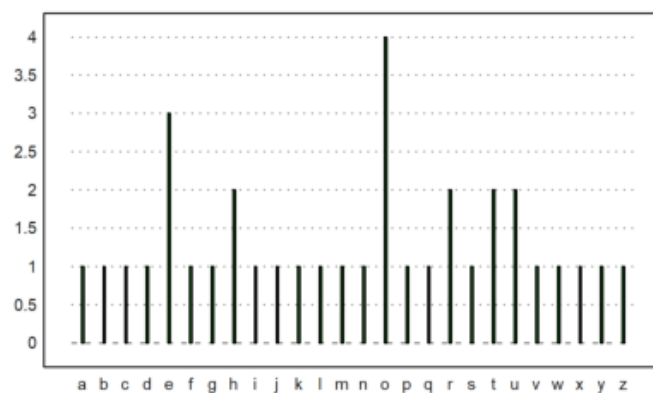
Tetapi untuk data yang terdistribusi secara eksponensial, kita mungkin memerlukan plot logaritma.

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



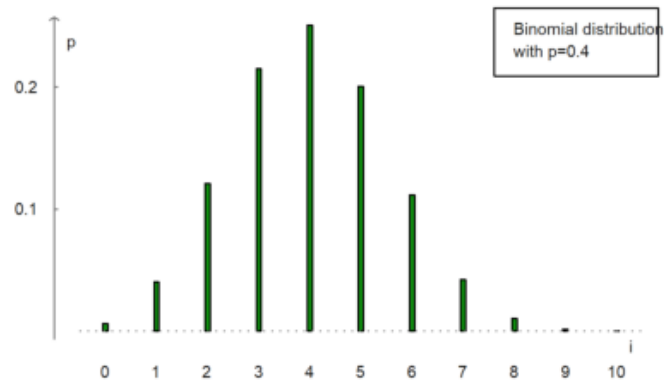
Fungsi `columnplot()` lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, kita dapat menentukan label-labelnya sesuai keinginan, yang telah kita demonstrasikan sebelumnya dalam tutorial ini. Berikut adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan membuat plot statistik.

```
>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 menetapkan sumbu-sumbu pada plot.

```
>n=10; p=0.4; i=0:n; x=bin(n,i)*p^i*(1-p)^(n-i); ...
>columnplot(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 acak integer 1 hingga 6.

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

```
[8, 5, 8, 8, 6, 8, 8, 3, 5, 5]
```

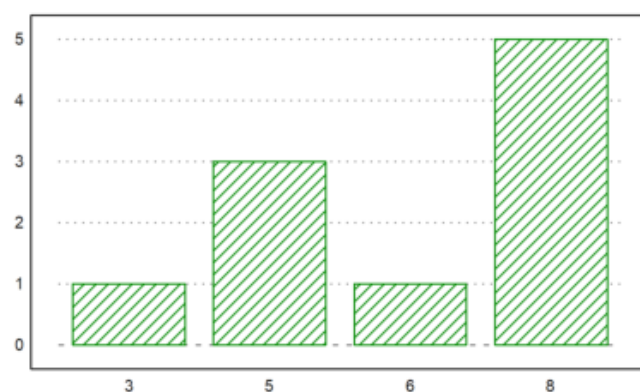
Kemudian ekstrak angka-angka unik dalam v.

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

```
[3, 5, 6, 8]
```

Dan plot frekuensinya dalam plot kolom.

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



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

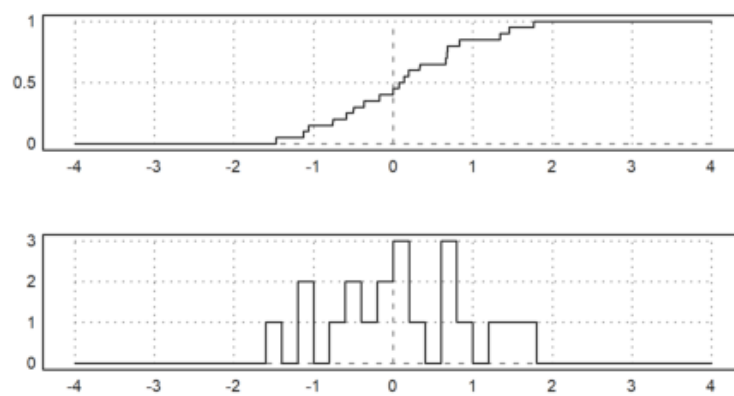
```
>x=normal(1,20);
```

Fungsi `empdist(x,vs)` membutuhkan array nilai yang terurut. Oleh karena itu, kita harus mengurutkan `x` sebelum kita dapat menggunakannya.

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

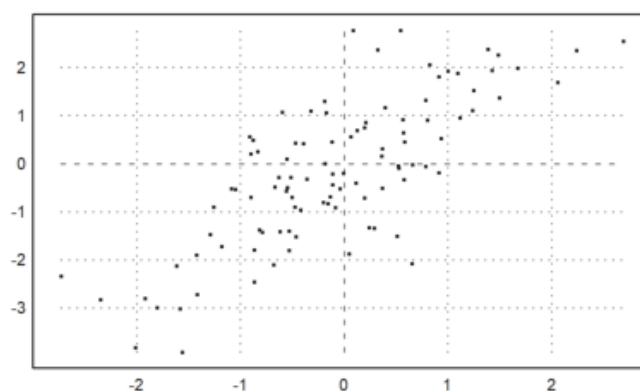
Kemudian kita membuat plot distribusi empiris dan beberapa batang densitas ke dalam satu plot. Kali ini, kita menggunakan plot sawtooth sebagai alternatif plot batang untuk distribusi.

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



Plot sebaran (scatter plot) mudah dilakukan dalam Euler dengan plot titik biasa. Grafik berikut menunjukkan bahwa `X` dan `X+Y` jelas berkorelasi positif.

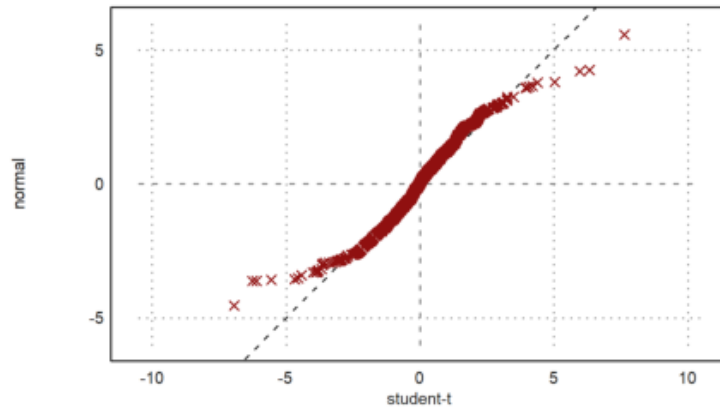
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



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

Untuk uji coba, 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):
```



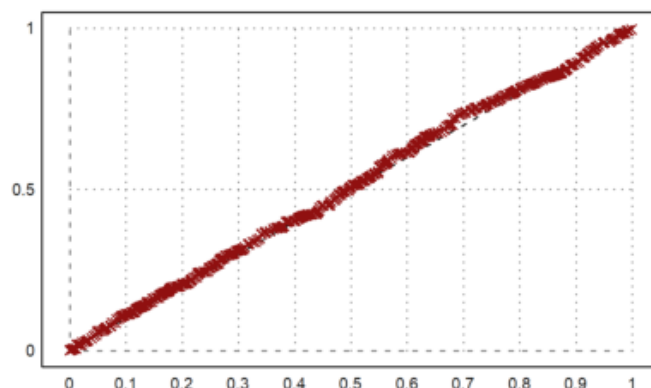
Plot tersebut dengan jelas menunjukkan bahwa nilai-nilai yang terdistribusi normal cenderung lebih kecil di ujung-ujung ekstrem.

Jika kita memiliki dua distribusi dengan ukuran yang berbeda, kita dapat memperluas yang lebih kecil atau merapatkan yang lebih besar. Fungsi berikut baik untuk kedua hal tersebut. Fungsi 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 fungsi `polyfit()` atau berbagai fungsi fit lainnya. Untuk memulai, kita akan mencari 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 regresi yang tidak tertimbang dan yang tertimbang. Pertama, koefisien dari regresi linear.

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

```
[0.733333, 0.812121]
```

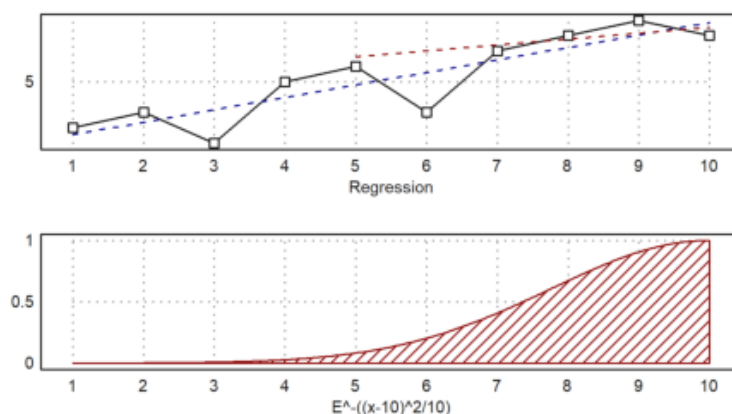
Sekarang koefisien dengan bobot yang menekankan pada nilai-nilai terakhir.

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

```
[4.71566, 0.38319]
```

Kita menggabungkan semuanya ke 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 akan membaca sebuah survei mahasiswa, usia mereka, usia orang tua mereka, dan jumlah saudara dari sebuah file.

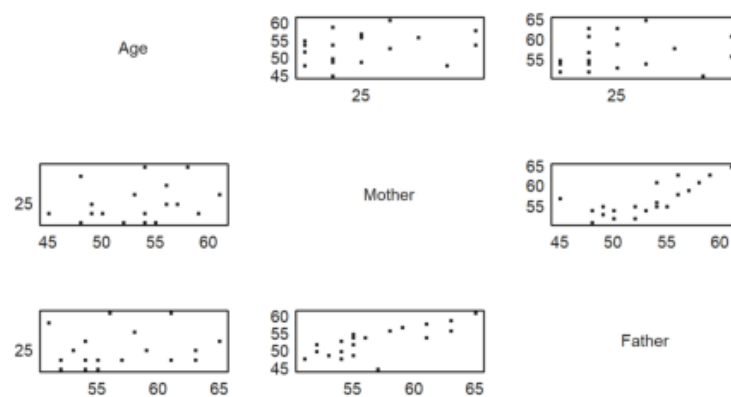
Tabel ini berisi "m" dan "f" dalam kolom kedua. Kita menggunakan variabel tok2 untuk menetapkan terjemahan yang tepat daripada membiarkan readtable() mengumpulkan terjemahan.

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

Person	Sex	Age	Mother	Father	Siblings
1	m	29	58	61	1
2	f	26	53	54	2
3	m	24	49	55	1
4	f	25	56	63	3
5	f	25	49	53	0
6	f	23	55	55	2
7	m	23	48	54	2
8	m	27	56	58	1
9	m	25	57	59	1
10	m	24	50	54	1
11	f	26	61	65	1
12	m	24	50	52	1
13	m	29	54	56	1
14	m	28	48	51	2
15	f	23	52	52	1
16	m	24	45	57	1
17	f	24	59	63	0
18	f	23	52	55	1
19	m	24	54	61	2
20	f	23	54	55	1

Bagaimana usia berkaitan satu sama lain? Kesan awal datang dari scatterplot berpasangan.

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



Jelas bahwa usia ayah dan ibu saling berkaitan. Mari kita tentukan dan plotkan garis regresi.

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

```
[17.3789, 0.740964]
```

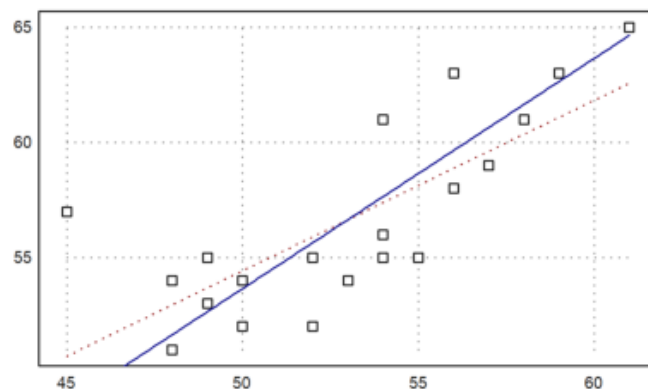
Ini jelas adalah model yang salah. Garis regresinya akan menjadi  $s=17+0.74t$ , di mana  $t$  adalah usia ibu dan  $s$  adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak sebanyak itu. Lebih baik, kita curigai fungsi seperti  $s=a+t$ . Kemudian  $a$  adalah rata-rata dari  $s-t$ . Itu adalah perbedaan usia rata-rata antara ayah dan ibu.

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

3.65

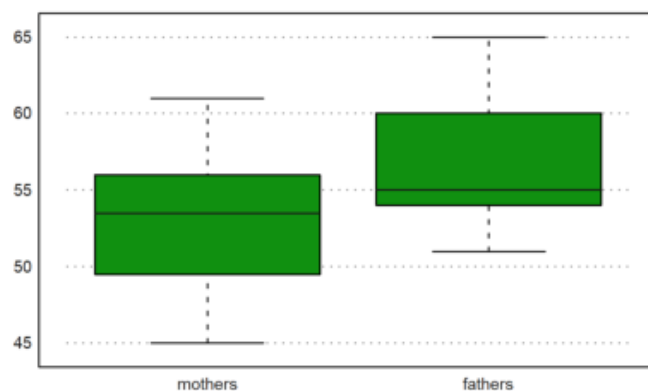
Mari kita plotkan ini dalam satu scatter plot.

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



Berikut adalah box plot dari kedua usia. Ini hanya menunjukkan bahwa usia mereka berbeda.

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



Menariknya, perbedaan median tidak sebesar perbedaan rata-rata.

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

1.5

Koefisien korelasi menunjukkan korelasi positif.

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

0.7588307236

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

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

0.758925292358

## Membuat Fungsi Baru

---

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi-fungsi baru. Sebagai contoh, kita mendefinisikan fungsi skewness.

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

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 dapat dengan mudah menggunakan bahasa matriks untuk mendapatkan implementasi yang sangat pendek dan efisien. Mari kita coba fungsi ini.

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

-0.198710316203

Berikut adalah fungsi lainnya, disebut koefisien skewness Pearson.

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

-0.0801873249135

## Simulasi Monte Carlo

---

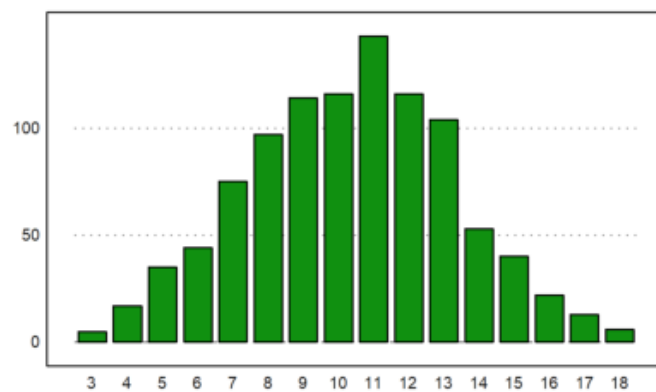
Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita sudah melihat contoh-contoh sederhana di atas. Berikut adalah contoh lain yang mensimulasikan 1000 kali lemparan 3 dadu, dan menanyakan distribusi jumlahnya.

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

```
[5, 17, 35, 44, 75, 97, 114, 116, 143, 116, 104, 53, 40,  
22, 13, 6]
```

Sekarang kita dapat membuat plot dari hasil simulasi ini.

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



Untuk menentukan distribusi yang diharapkan tidaklah mudah. Kita menggunakan rekursi yang lebih canggih untuk ini.

Fungsi berikut menghitung jumlah cara di mana angka k dapat diwakili sebagai jumlah dari n angka dalam rentang 1 hingga m. Fungsi 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 hasil untuk tiga lemparan dadu.

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

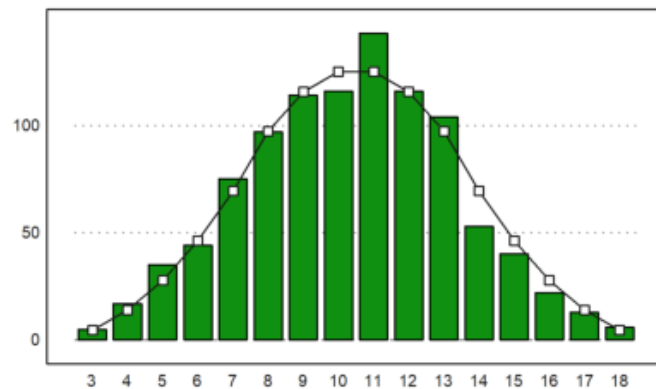
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,  
255, 185, 121, 70, 35, 15, 5, 1]
```

```
>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 dari nilai rata-rata dari  $n$  variabel acak terdistribusi normal 0-1 adalah  $1/\sqrt{n}$ .

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

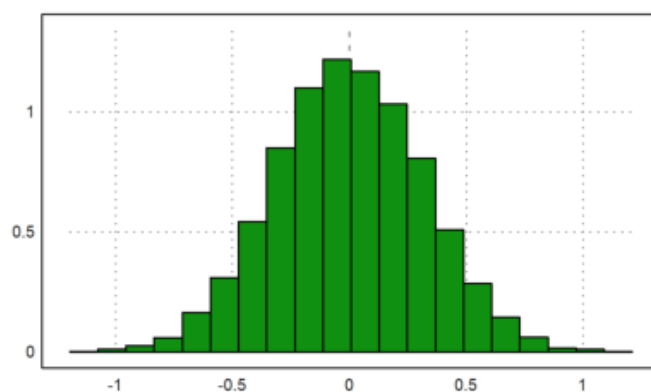
```
0.316227766017
```

Mari kita periksa ini dengan simulasi. Kita menghasilkan 10000 kali 10 vektor acak.

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

```
0.319493614817
```

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



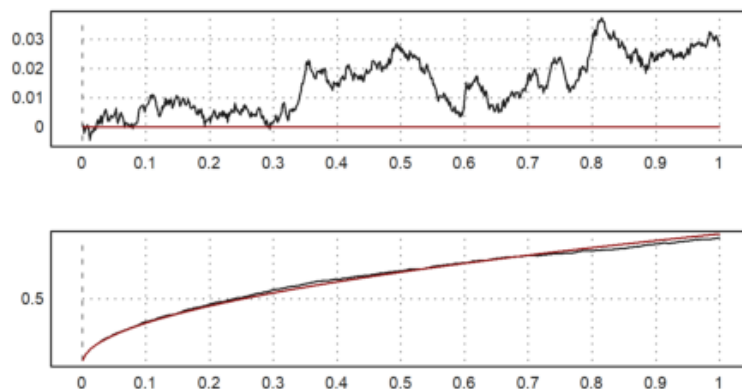
Median dari 10 angka acak terdistribusi normal 0-1 memiliki deviasi yang lebih besar.

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

0.374460271535

Karena kita dapat dengan mudah menghasilkan random walk, kita dapat mensimulasikan proses Wiener. Kita ambil 1000 langkah dari 1000 proses. Kemudian kita plot deviasi standar 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

Uji merupakan alat penting dalam statistik. Di Euler, banyak uji yang 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)
```

0.526

Kesalahan ini terlalu besar. Jadi kita tidak dapat menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Tetapi kita tidak dapat menolak hipotesis kita.

Selanjutnya kita menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan uji yang sama.

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

0.528028118442

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

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

0.0218365848476

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

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

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

0.38722000942

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

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

5.60009101758e-07

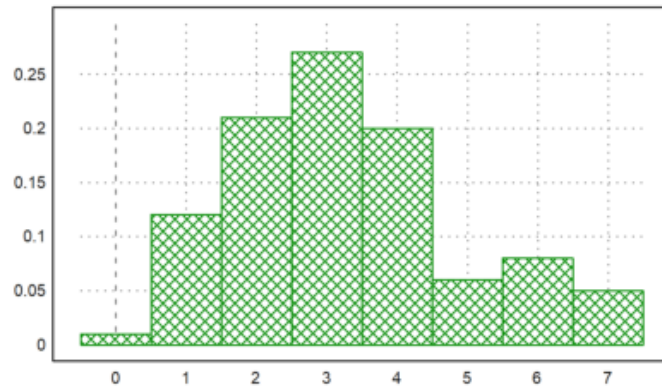
Pada contoh berikutnya, kita menghasilkan 20 lemparan dadu acak sebanyak 100 kali dan menghitung jumlah angka satu di dalamnya. Rata-ratanya seharusnya ada  $20/6=3,3$  mata dadu.

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

3.28

Sekarang kita bandingkan jumlah angka satu dengan distribusi binomial. Pertama kita 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 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)
```

```
0.53921579764
```

Contoh berikutnya berisi hasil dari dua kelompok orang (laki-laki dan perempuan, 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 untuk independensi dari suara terhadap jenis kelamin. Uji tabel chi-kuadrat melakukan hal ini. Hasilnya terlalu besar untuk menolak independensi. Jadi kita tidak dapat mengatakan apakah pemilihan bergantung pada jenis kelamin dari data ini.

```
>tabletest(A)
```

```
0.990701632326
```



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

```
>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 sangat dekat dengan 0, kita 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). Di 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

Ini berarti kita menolak hipotesis nilai rata-rata yang sama. Kita lakukan 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 digabung.

```
>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 kesamaan lainnya adalah uji peringkat. Ini jauh lebih tajam dibandingkan uji median.

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

```
0.00199969612469
```

Pada contoh berikutnya, kedua distribusi memiliki rata-rata yang sama.

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

```
0.129608141484
```

Mari kita mencoba mensimulasikan dua perlakuan a dan b yang diterapkan pada 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 memutuskan apakah a lebih baik dari b.

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

```
0.0546875
```

Ini terlalu banyak kesalahan. Kita tidak dapat 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 coba dua uji lainnya menggunakan deret yang dihasilkan.

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

```
0.0068706451766
```

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

```
0.275145971064
```

## Bilangan Acak

---

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

Pertama, kita menghasilkan sepuluh juta bilangan acak dalam [0,1].

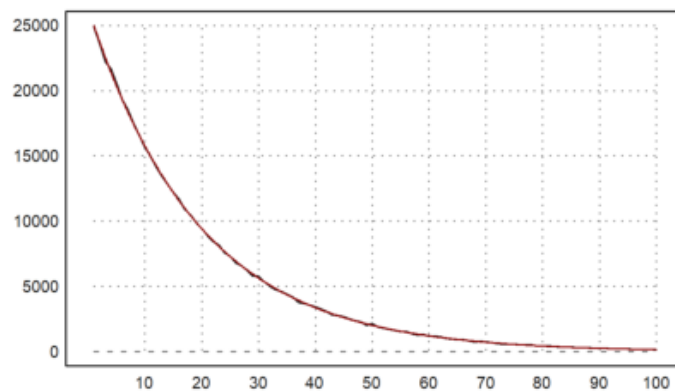
```
>n:=10000000; r:=random(1,n);
```

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

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

Terakhir, kita plot jumlah kali, setiap jarak terjadi, dan membandingkan 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 bagi Pengguna Proyek R

---

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

Buku catatan ini adalah untuk Anda jika Anda sudah familiar dengan R, tetapi perlu mengetahui perbedaan sintaksis EMT dan R. Kami mencoba memberikan gambaran tentang hal-hal yang perlu Anda ketahui yang jelas maupun yang kurang jelas.

Selain itu, kita akan melihat cara bertukar data antara kedua sistem tersebut.

Perhatikan bahwa ini masih dalam tahap pengembangan. **Sintaks Dasar**

---

Hal pertama yang Anda pelajari di R adalah membuat vektor. Di EMT, perbedaan utamanya adalah operator ":" bisa memiliki langkah. Selain itu, operator ini memiliki prioritas pengikatan 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. Tetapi, memungkinkan untuk menggunakan vektor untuk menggabungkan beberapa hal.

Contoh berikut, seperti banyak contoh lainnya, berasal dari "Interoduction to R" yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti jalannya 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 dengan langkah ukuran EMT diganti dengan 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()` dari R tidak ada di EMT. Untuk input vektor, bisa 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]
```

Perhatikan bahwa "=" atau ":=" digunakan untuk penugasan. Operator ">" digunakan untuk unit di EMT.

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

```
77.6713990297 miles
```

Operator "<-" untuk penugasan adalah menyesatkan dan bukan ide yang baik di R. Berikut adalah perbandingan a dan -4 di EMT.

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

```
0
```

Di R, "`a<-4<3`" bekerja, tetapi "`a<-4<-3`" tidak. Saya memiliki ambigü serupa di EMT juga, tetapi mencoba untuk menghilangkannya sedikit demi sedikit.

EMT dan R memiliki vektor tipe boolean. Tetapi di EMT, angka 0 dan 1 digunakan untuk merepresentasikan false dan true. Di R, nilai true dan false dapat digunakan dalam perhitungan biasa seperti di EMT.

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

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

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

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

```
NAN  
1
```

String sama di R dan EMT. Keduanya berada dalam locale saat ini, bukan Unicode.

Di R, ada paket untuk Unicode. Di EMT, string dapat menjadi string Unicode. String unicode dapat diterjemahkan ke encoding lokal dan sebaliknya. Selain itu, u"..." dapat berisi entitas HTML.

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

© René Grothmann

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

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

Penggabungan string dilakukan dengan "+" atau "|". Ini dapat mencakup angka, yang akan dicetak dalam format saat ini.

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

```
pi = 3.14159265359
```

## Indeks

---

Seringkali, 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 di EMT dengan menggunakan `drop()`.

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

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

Vektor logika tidak diperlakukan secara berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu mengekstrak elemen non-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 di R, vektor indeks dapat berisi pengulangan.

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

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

Namun, pemberian nama untuk indeks tidak memungkinkan dalam EMT. Untuk paket statistik, ini seringkali menjadi hal yang diperlukan untuk mempermudah akses ke elemen-elemen vektor. Untuk meniru perilaku ini, kita dapat mendefinisikan sebuah fungsi sebagai 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)]; ... ...  
      ^  
[10.4,  3.1]
```

## Jenis Data

EMT memiliki lebih banyak tipe data yang tetap dibandingkan dengan R. Jelas, dalam R terdapat vektor yang berkembang. Anda dapat menetapkan vektor numerik kosong `v` dan memberikan nilai pada elemen `v[17]`. Hal ini tidak mungkin dilakukan di EMT. Berikut adalah contoh yang agak tidak efisien.

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

EMT sekarang akan membuat vektor dengan `v` dan `i` yang ditambahkan ke dalam tumpukan dan menyalin vektor tersebut kembali ke variabel global `v`. Lebih efisien untuk mendefinisikan vektor sebelumnya.

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

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

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

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

Konversi ke string hanya mungkin dilakukan 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 Anda 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 komunikasi dengan Maxima, ada fungsi `convertmxm()`, yang juga dapat digunakan untuk memformat vektor untuk output.

```
>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 R, ada contoh dengan apa yang disebut faktor.

Berikut ini 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"];
```

Asumsikan, 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 tersebut. Sebagai sebuah program statistik, R memiliki fungsi `factor()` dan `tapply()` untuk hal ini.

EMT dapat melakukan hal ini dengan menemukan indeks wilayah dalam daftar wilayah yang unik.

```
>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 titik ini, kita dapat menulis fungsi perulangan kita sendiri untuk melakukan berbagai hal 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 sedikit tidak efisien, karena menghitung wilayah unik untuk setiap *i*, tetapi berfungsi.

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

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

Perhatikan bahwa ini bekerja untuk setiap vektor wilayah.

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

```
[44.5, 57.3333333333]
```

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

Jadi kami dapat mencetak pendapatan negara rata-rata di wilayah tersebut dengan cara yang bersahabat.

```
>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 bisa mencoba meniru perilaku R sepenuhnya.

Faktor-faktor tersebut harus disimpan dengan jelas dalam sebuah koleksi dengan jenis dan kategorinya (negara bagian dan teritori dalam contoh kita). Untuk EMT, kita menambahkan 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 ini 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 dapat meniru `tapply()` sebagai berikut. Ini akan mengembalikan tabel sebagai kumpulan 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 kategori (faktor) yang tidak memiliki data. Tetapi kita harus memeriksa panjang `t` yang benar dan ketepatan koleksi `tf`.

Tabel ini dapat dicetak sebagai sebuah tabel dengan `writetable()`.  
yang agak tidak efisien.

```
>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

## Array

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Bagaimanapun, akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau pustaka C untuk hal ini.

R memiliki lebih dari dua dimensi. Dalam R, array adalah vektor dengan bidang berdimensi.

Dalam EMT, vektor adalah sebuah matriks dengan satu baris. Ini bisa dibuat 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

Ekstraksi baris dan kolom, atau sub-matriks, mirip seperti di R.

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

2	3
7	8
12	13
17	18

Namun, di R dimungkinkan untuk mengatur daftar indeks tertentu dari vektor ke suatu nilai. Hal yang sama juga dapat dilakukan dalam 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 mendemonstrasikan hal ini untuk menunjukkan bahwa matriks diteruskan dengan referensi dalam 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

Hasil kali luar di EMT hanya dapat dilakukan di antara vektor. Hal ini otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan vektor baris.

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

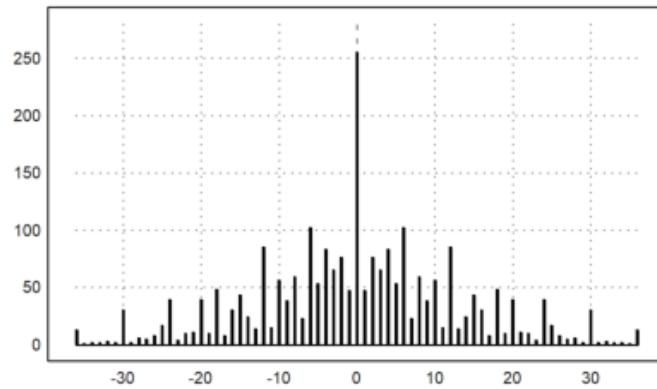
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

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

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

Tetapi kita ingin meniru perilaku R. Untuk ini, kita perlu meratakan perkalian ab dan membuat sebuah 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"):
```



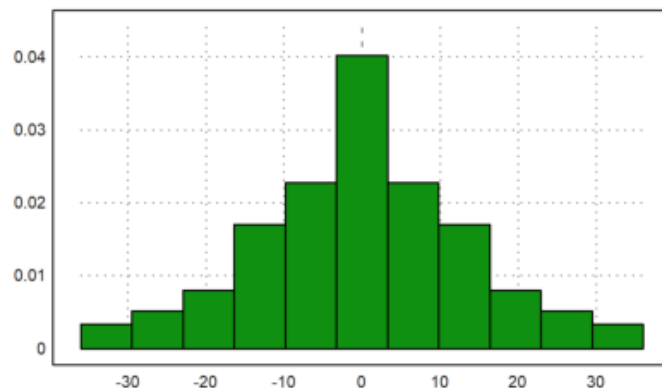
Selain multiplikasi yang tepat, EMT dapat menghitung frekuensi dalam vektor.

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

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

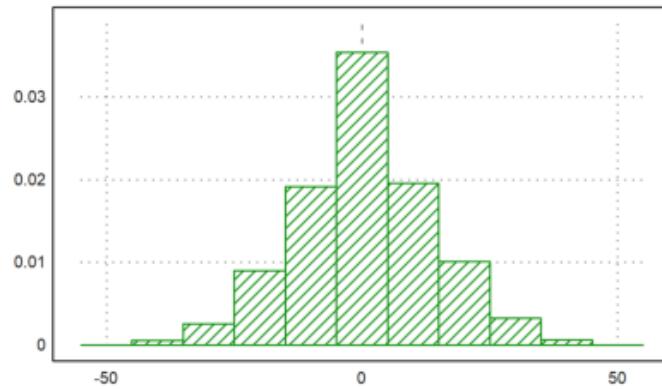
Cara paling mudah untuk memplot ini sebagai distribusi adalah sebagai berikut.

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



Namun, juga memungkinkan untuk melakukan perhitungan jumlah dalam interval yang dipilih sebelumnya. Tentu saja, hal berikut menggunakan `getfrequencies()` secara internal. Karena fungsi `histo()` mengembalikan frekuensi, kita perlu menyesuaikan agar integral di bawah diagram batang menjadi 1.

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



## Daftar (List)

EMT memiliki dua jenis daftar. Yang pertama adalah daftar global yang dapat diubah, dan yang kedua adalah tipe daftar yang tidak dapat diubah. Kami tidak akan membahas tentang daftar global di sini.

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

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

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

Saat ini elemen-elemen tersebut tidak memiliki nama, meskipun nama dapat ditetapkan untuk tujuan khusus. Elemen-elemen tersebut diakses dengan angka.

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

```
1992
```

## Input dan Output File (Membaca dan Menulis Data)

Anda mungkin akan sering mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberitahu Anda tentang banyak cara untuk mencapai hal ini. Fungsi yang sederhana adalah `writematrix()` dan `readmatrix()`. Mari kita tunjukkan cara membaca dan menulis vektor real ke sebuah file.

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

```
0.49815
0.28037
```

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

Karena pengenalan ini kemungkinan besar berada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk notebook sendiri, hal ini tidak perlu dilakukan, karena file data akan ditulis ke direktori yang sama.

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

Sekarang kita tuliskan vektor kolom a' ke dalam file. Hal ini akan menghasilkan satu angka pada setiap baris file.

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

Untuk membaca data, kita menggunakan readmatrix().

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

Dan hapus file tersebut.

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

```
0.49815  
0.28037
```

Fungsi writematrix() atau writetable() dapat dikonfigurasi untuk bahasa lain. Misalnya, jika Anda memiliki sistem Indonesia (titik desimal dengan koma), Excel Anda membutuhkan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan dengan koma). File "test.csv" berikut akan muncul di folder Anda saat ini.

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

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

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

Terkadang kita memiliki string dengan token seperti berikut ini.

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

Untuk menandai ini, kita mendefinisikan vektor token.

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

```
f  
m
```

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

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

Tulis 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 akan terlihat seperti ini.

```
>printfile(file)
```

```
A,B,C
0.7003664386138074,0.1875530821001213,0.3262339279660414
0.5926249243193858,0.1522927283984059,0.368140583062521
0.8065535209872989,0.7265910840408142,0.7332619844597152
```

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

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

Kumpulan ini dapat dicetak dengan `writetable()` ke notebook atau ke sebuah file

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

A	B	C
0.70037	0.18755	0.32623
0.59262	0.15229	0.36814
0.80655	0.72659	0.73326

Matriks nilai adalah elemen pertama dari L. Perhatikan bahwa `mean()` dalam EMT menghitung nilai rata-rata dari baris-baris matriks.

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

```
0.40472
0.37102
0.75547
```

## File CSV

Pertama, mari kita tulis sebuah matriks ke dalam sebuah file. Untuk outputnya, kita buat sebuah file di working directory saat ini.

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

Berikut adalah isi file ini.

```
>printfile(file)
```

```
0.8221197733097619,0.821531098722547,0.7771240608094004  
0.8482947121863489,0.3237767724883862,0.6501422353377985  
0.1482301827518109,0.3297459716109594,0.6261901074210923
```

CVS ini dapat dibuka di sistem bahasa Inggris ke Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal. Tetapi titik desimal juga merupakan format default untuk EMT. Anda dapat membaca sebuah matriks dari sebuah file dengan readmatrix().

```
>readmatrix(file)
```

```
0.82212    0.82153    0.77712  
0.84829    0.32378    0.65014  
0.14823    0.32975    0.62619
```

Dimungkinkan untuk menulis beberapa matriks ke dalam satu file. Perintah open() dapat membuka file untuk menulis dengan parameter "w". Standarnya adalah "r" untuk membaca.

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

Matriks-matriks tersebut dipisahkan oleh sebuah baris kosong. Untuk membaca matriks, buka file dan panggil 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 matriks sebagai CSV (nilai yang dipisahkan dengan koma). Pada Excel 2007, gunakan "save as" dan "other format", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut ini adalah contohnya.

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

```

0;1000;1000
1;1051,271096;1072,508181
2;1105,170918;1150,273799
3;1161,834243;1233,67806
4;1221,402758;1323,129812
5;1284,025417;1419,067549
6;1349,858808;1521,961556
7;1419,067549;1632,31622
8;1491,824698;1750,6725
9;1568,312185;1877,610579
10;1648,721271;2013,752707

```

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

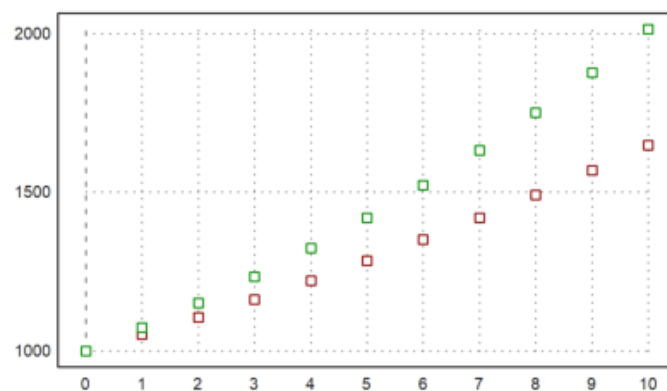
Cara termudah untuk membaca ini ke dalam Euler adalah menggunakan `readmatrix()`. Semua koma akan diganti dengan titik dengan parameter `>comma`. Untuk CSV berbahasa Inggris, cukup abaikan parameter ini.

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

0	1000	1000
1	1051.3	1072.5
2	1105.2	1150.3
3	1161.8	1233.7
4	1221.4	1323.1
5	1284	1419.1
6	1349.9	1522
7	1419.1	1632.3
8	1491.8	1750.7
9	1568.3	1877.6
10	1648.7	2013.8

Mari kita plotkan.

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





Ada beberapa cara yang lebih mendasar untuk membaca data dari file. Anda dapat membuka file dan membaca angka baris demi baris. Fungsi `getvectorline()` akan membaca angka dari sebuah baris data. Secara default, fungsi ini mengharapkan sebuah titik desimal. Tetapi fungsi ini juga dapat menggunakan koma desimal, jika Anda memanggil `setdecimaldot(",")` sebelum menggunakan fungsi ini. Fungsi berikut ini adalah contohnya. Fungsi ini akan berhenti di 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)
```

```
0.82212    0.82153    0.77712
0.84829    0.32378    0.65014
0.14823    0.32975    0.62619
```

Anda juga dapat membaca semua angka di dalam file tersebut dengan `getvector()`.

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

```
0.82212    0.82153    0.77712
0.84829    0.32378    0.65014
0.14823    0.32975    0.62619
```

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

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

```
0.50303
```

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

```
0.50303
```

## Menggunakan tabel

---

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

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

```
one,two,three
0.09,    0.39,    0.86
0.39,    0.86,    0.71
0.2,     0.02,    0.83
```

Ini dapat diimpor ke Excel.

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

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

```
one    two    three
0.09    0.39    0.86
0.39    0.86    0.71
0.2     0.02    0.83
```

## Menganalisis Garis

---

Anda bahkan dapat mengevaluasi setiap baris dengan tangan. Misalkan, kita memiliki baris dengan format berikut ini.

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

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

Pertama, kita dapat memberi tanda pada garis.

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

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

Kemudian, kita dapat mengevaluasi setiap elemen garis dengan 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 ekspresi reguler, Anda dapat mengekstrak hampir semua informasi dari sebuah baris data.

Anggaplah kita memiliki baris dokumen HTML berikut ini.

```
>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, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung penutup > ,
- string apa pun yang tidak mengandung tanda kurung dengan sub-match "...)",
- braket pembuka dan penutup menggunakan solusi terpendek,
- lagi semua string yang tidak mengandung tanda kurung,
- dan braket pembuka < .

Ekspresi reguler agak sulit untuk dipelajari, tetapi sangat kuat.

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

Hasilnya adalah kecocokan posisi, string yang cocok, dan vektor string untuk sub-match.

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

```
1145.5  
5.6
```

Berikut ini adalah 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

---

Situs web atau file dengan URL dapat dibuka di EMT dan dapat dibaca baris demi baris.

Dalam contoh, kita membaca versi saat ini dari situs EMT. Kami menggunakan ekspresi reguler untuk mendandai "Version..." dalam sebuah judul.

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

```
>readversion
```

## Input dan Output dari Variabel

---

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

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

```
    mypi = 3.141592653589793;
```

Untuk pengujian, kami membuat file Euler di direktori kerja EMT.

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

```
M = [ ..  
    0.5991820585590205, 0.7960280262224293;  
    0.5167243983231363, 0.2996684599070898];
```

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

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

```
M =  
    0.59918    0.79603  
    0.51672    0.29967
```

Sebagai catatan, jika `writevar()` digunakan pada sebuah variabel, maka ia akan mencetak definisi variabel dengan nama variabel tersebut.

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

```
M = [ ..
0.5991820585590205, 0.7960280262224293;
0.5167243983231363, 0.2996684599070898];
inch$ = 0.0254;
```

Kita juga dapat membuka file baru atau menambahkan ke file yang sudah ada. Dalam contoh ini, kami menambahkan ke file yang telah dibuat sebelumnya.

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

```
M1 =
  0.30287    0.15372
  0.7504     0.75401
M2 =
  0.27213
  0.053211
  0.70249
```

Untuk menghapus file, gunakan `fileremove()`.

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

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

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

```
M = [
0.344851384551
0.0807510017715
0.876519562911
0.754157709472
0.688392638934
];
```

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

```
[0.34485, 0.080751, 0.87652, 0.75416, 0.68839]
```

## Contoh Soal

---

1. Diketahui bahwa jumlah uang yang diperlukan untuk membeli hewan kelinci, tikus, dan katak dalam percobaan di prodi biologi dapat menggunakan perkalian matriks. Jika harga hewan berturut-turut adalah 15 ribu, 3 ribu, dan 2 ribu rupiah. Banyak hewan yang diperlukan berturut turut sebanyak 40, 50, dan 80 ekor. Berapakah jumlah uang yang diperlukan untuk membeli keperluan dalam percobaan tersebut?

Penyelesaian:

Misalkan

m = harga hewan

n = banyak hewan

```
>m = [15; 3; 2]
```

```
15
 3
 2
```

```
>n = [40; 50; 80]
```

```
40
50
80
```

Menggunakan perkalian matriks, maka diperoleh

```
>m*n
```

```
600
150
160
```

Jumlahkan hasilnya

```
>600+150+160
```

```
910
```

Jadi, uang yang diperlukan untuk membeli keperluan dalam percobaan tersebut adalah 910 ribu rupiah.

---

2. Diketahui matriks Gross Income dan matriks pengeluaran Negara China, Jepang, Prancis, Kanada tahun 1995 dan 1996 adalah sebagai berikut:

```
>mgi = [38,21,22,30;42,23,34,41]
```

```
38      21      22      30
42      23      34      41
```

```
>mp = [25,16,17,24;28,18,20,31]
```

```
25      16      17      24
28      18      20      31
```

Maka berapakah matriks Gross Profit keempat negara tersebut pada tahun 1995 dan 1996?

Penyelesaian:

Gross Profit dapat dihitung dengan mengurangkan matriks Gross Income (mpi) dengan matriks pengeluaran (mp). Sehingga diperoleh matriks Gross Profit berikut:

```
>mpi-mp
```

13	5	5	6
14	5	14	10

3. Kadar nikotin yang berasal dari sebuah sampel acak enam batang rokok merk tertentu yaitu 2.7, 2.9, 2.3, 2.1, 1.9, dan 3.1 miligram. Tentukan mediannya!

Penyelesaian:

Kadar nikotin itu kita urutkan dari yang terkecil sampai terbesar, sehingga diperoleh:

1.9 2.1 2.3 2.7 2.9 3.1

Maka mediannya adalah rata-rata dari 2.3 dan 2.7 yaitu

$$\bar{x} = \frac{2.3 + 2.7}{2} = 2.5 \text{ miligram}$$

atau menggunakan EMT:

```
>mean ([2.3, 2.7])
```

2.5

4. Seorang pelatih tembak akan mengevaluasi nilai ketangkasan sepuluh anak buahnya jenis senapan yang dipakai M-16 dengan jarak 250 meter dan masing-masing mendapat nilai 76,88,74,68,70,80,60,85, 65,dan 50. Berapakah rata-rata nilai ketangkasan sepuluh anak tersebut?

Penyelesaian:

```
>x=[76,88,74,68,70,80,60,85,65,50]; mean(x),
```

71.6

dengan cara manual:

$$\sum x_i = 76 + 88 + 74 + 68 + 70 + 80 + 60 + 85 + 65 + 50 = 716$$

$$n = 10$$

$$\bar{X} = \frac{\sum x_i}{n}$$

$$\bar{X} = \frac{716}{10}$$

$$\bar{X} = 71.6$$

Sehingga rata-rata nilai ketangkasan sepuluh anak tersebut adalah 71.6

---

5. Diberikan data urut sebagai berikut: 45,48,49,50,52,52,52,53,53,54,54,54,54,54,56,56, 56,56,57,57,58,58,58,58,58,58,59,59,62,62,62,63,63,64,64,65,67,68,69,70,70,71,73,74.

Buatlah distribusi frekuensi berdasarkan data diatas beserta histogram!

Penyelesaian:

- Mencari range

range = nilai maks - nilai min

$$= 74 - 45$$

$$= 29$$

- Mencari banyak kelas dengan aturan struges.

banyak kelas =  $1 + 3,3 \log n$ , n banyaknya data}

$$= 1 + 3,3 \log 48$$

$$= 6,64$$

$$= 7$$

- Mencari panjang kelas

$$p = \frac{\text{range}}{\text{banyak kelas}}$$

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

$$p = 4.14 = 5$$

Maka diketahui nilai minimum = 45, nilai maksimum = 74, banyak kelas = 7, dan panjang kelas = 5.

Sehingga dapat dibuat tabel distribusi frekuensi dengan batas bawah kelas pertama yaitu 43 dan batas atas kelas ketujuh yaitu 77.

Tepi bawah kelas pertama yaitu  $43 - 0.5 = 42.5$

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"," Frekuensi"])
```

TB	TA	Frekuensi
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 interval setiap kelas

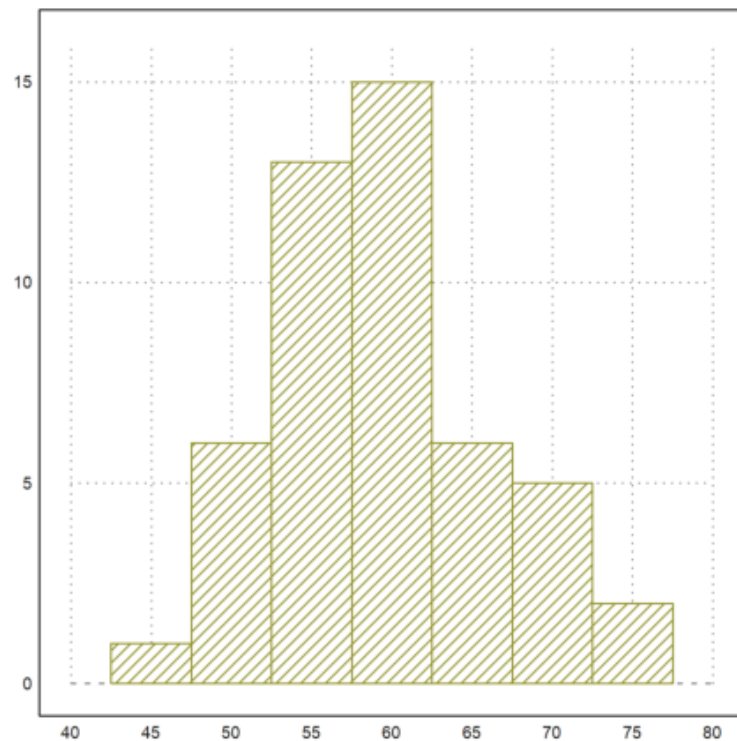
```
> (T[,1]+T[,2])/2 // the midpoint of each interval
```



45  
50  
55  
60  
65  
70  
75

Menggambar plot histogram berdasarkan data di atas

```
>plot2d(r,v,a=40,b=80,c=0,d=16,bar=1,fillcolor=6, style="/"):
```



6. Berikut adalah data hasil dari pengukuran berat badan 30 siswa SMA Budi Utomo. Siswa yang mempunyai berat badan dalam rentang 41-45 kg sebanyak 8 orang, yang mempunyai berat badan dalam rentang 46-50 kg sebanyak 10 orang, yang mempunyai berat badan dalam rentang 51-55 kg sebanyak 7 orang, yang mempunyai berat badan dalam rentang 56-60 kg sebanyak 3 orang, yang mempunyai berat badan dalam rentang 61-65 kg sebanyak 1 orang, dan yang mempunyai berat badan 66-70 kg sebanyak 1 orang. Tentukan modus dari data hasil pengukuran berat badan 30 siswa di SMA tersebut!

Penyelesaian:

- Menentukan tepi bawah kelas

```
>41-0.5
```

40.5

- Menentukan panjang kelas

```
> (45-41) +1
```

5

- Menentukan tepi atas terbesar

```
> 70+0.5
```

70.5

- Membuat tabel distribusi kumulatif

```
> r=40.5:5:70.5; v=[8,10,7,3,1,1];  
> T := r[1:6]' | r[2:7]' | v'; writetable(T,labc=["TB","TA"," frekuensi"])
```

TB	TA	frekuensi
40.5	45.5	8
45.5	50.5	10
50.5	55.5	7
55.5	60.5	3
60.5	65.5	1
65.5	70.5	1

Dari tabel di atas, terlihat bahwa modus berada di kelas 40.5 - 50.5

```
> Tb=45.5; p=5; d1=2; d2=3;  
> Tb+ (d1/ (d1+d2)) *p
```

47.5

Jadi, modus dari data hasil pengukuran berat badan 30 siswa di SMA Budi Utomo adalah 47.5.

---

7. Data berikut menunjukkan nilai yang diperoleh 50 siswa di suatu kelas pada Ujian mata kuliah kalkulus. Siswa yang mendapat nilai dalam rentang 61-65 sebanyak 2 orang, dalam rentang 66-70 sebanyak 5 orang, dalam rentang 71-75 sebanyak 8 orang, dalam rentang 76-80 sebanyak 10 orang, dalam rentang 81-85 sebanyak 12 orang, dalam rentang 86-90 sebanyak 9 orang, dan dalam rentang 91-95 sebanyak 4 orang. Tentukan rata-rata nilai yang diperoleh 50 siswa tersebut!

Penyelesaian:

- Menentukan tepi bawah kelas

```
> 61-0.5
```

60.5

- Menentukan panjang kelas

```
> (65-61)+1
```

5

- Menentukan tepi atas kelas

```
> 95+0.5
```

95.5

```
>r=60.5:5:95.5; v=[2,5,8,10,12,9,4];  
>T:=r[1:7]' | r[2:8]' | v'; writetable(T,labc=["TB","TA"," Frekuensi"])
```

TB	TA	Frekuensi
60.5	65.5	2
65.5	70.5	5
70.5	75.5	8
75.5	80.5	10
80.5	85.5	12
85.5	90.5	9
90.5	95.5	4

- Menentukan titik tengah kelas

```
>(T[,1]+T[,2])/2 // the midpoint of each interval
```

63  
68  
73  
78  
83  
88  
93

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

[63, 68, 73, 78, 83, 88, 93]

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

79.8

Jadi, rata-rata nilai yang diperoleh siswa tersebut adalah 79.8.

---

8. Hitunglah nilai rata-rata data dalam file "test.dat"!

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

```
0.0728752572063342,0.9889661469786067,0.01043758862055607  
0.3566257484784296,0.5214304734587624,0.4288933665752114  
0.168134262921015,0.1827423588992564,0.288048270849559
```

```
>readmatrix(filename)
```

0.0728753	0.988966	0.0104376
0.356626	0.52143	0.428893
0.168134	0.182742	0.288048

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

```
0.357426  
0.43565  
0.212975
```

---