

Tugas Aplikasi Komputer

Nama : Theresia Selvina Vanny M.
NIM : 22305141029
Kelas : Matematika B

EMT untuk Statistika

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

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukan pengantar statistik. Jadi Anda mungkin memerlukan beberapa pengetahuan dasar untuk memahami detailnya.

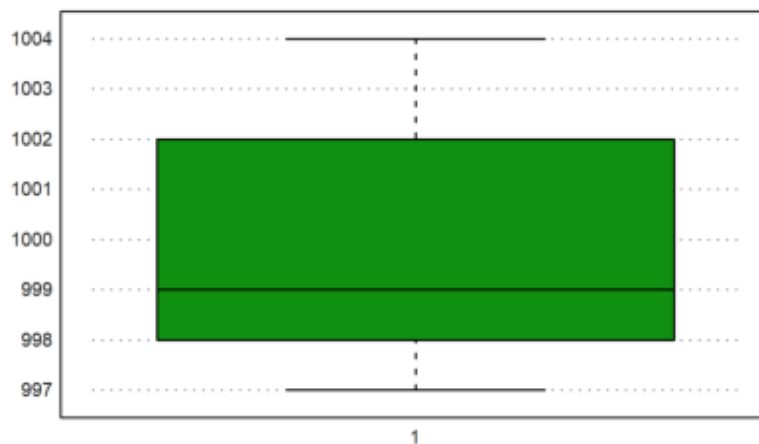
Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan deviasi standar yang diukur.

```
>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-whisker untuk data. Dalam kasus kami, tidak ada outlier.

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



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

Semua fungsi untuk distribusi di Euler diakhiri dengan ...dis dan menghitung distribusi probabilitas kumulatif (CPF).

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

Kami mencetak hasilnya dalam % dengan akurasi 2 digit menggunakan fungsi cetak.

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

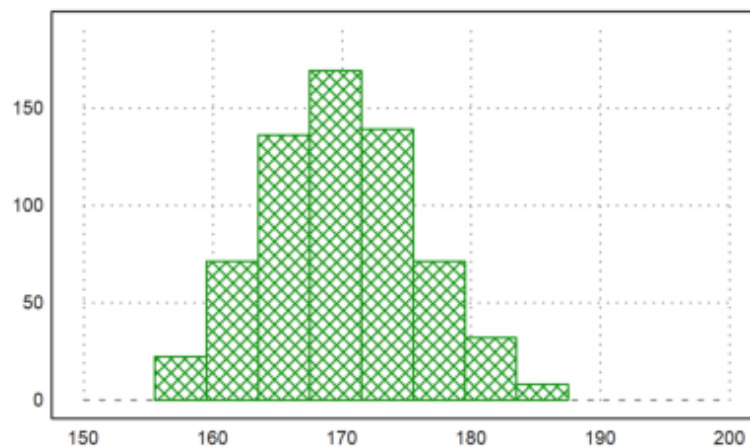
3.07 %

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

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

Berikut adalah plot distribusi.

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



Kami dapat memasukkan data mentah seperti ini ke dalam tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kami harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah pria dalam rentang.

Tabel dapat dicetak dengan header. Kami menggunakan vektor string untuk mengatur 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, kita perlu menghitung titik tengah dari rentang-rentang tersebut. Kami dapat menggunakan dua kolom pertama dari tabel kami untuk ini.

Symbol "|" digunakan untuk memisahkan kolom, dan fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" adalah untuk menentukan header 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
```

Tapi lebih mudah, untuk melipatgandakan rentang-rentang 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,
```

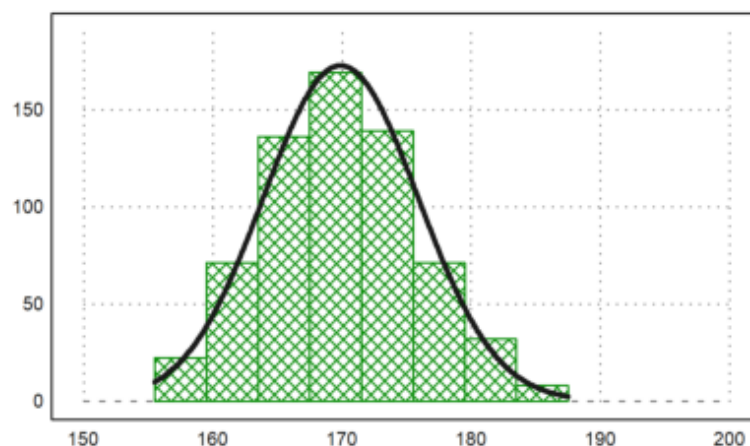
```
169.901234568
5.98912964449
```

Mari kita tambahkan distribusi normal dari nilai-nilai di atas ke plot batang di atas. Formula 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 nilainya antara 0 dan 1, untuk menggambarnya pada plot batang, itu 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):
```



Tabel

Di direktori catatan ini, Anda akan menemukan file dengan tabel. Data tersebut mewakili hasil dari survei. Berikut adalah empat baris pertama dari file tersebut. Data berasal dari buku online Jerman "Einführung in die Statistik mit R" oleh 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 angka atau token (string). Kami ingin membaca tabel dari file tersebut. Pertama, kami menggunakan terjemahan sendiri untuk token-token tersebut.

Untuk ini, kami mendefinisikan set token. Fungsi strtokens() mendapatkan vektor string token dari string yang diberikan.

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

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen tok2, tok4, dll adalah terjemahan kolom tabel. Argumen-argumen ini tidak ada dalam daftar parameter readtable(), jadi Anda perlu memberikannya dengan ":=".

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

Untuk mencetaknya, kita perlu menentukan set token yang sama. Kita hanya 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

Titik "." mewakili nilai yang tidak tersedia.

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

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

Fungsi readtable() sekarang mengembalikan set token.

```
>tok
```

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

Tabel ini berisi entri dari file dengan token yang diterjemahkan menjadi angka.

String khusus NA="." diinterpretasikan sebagai "Not Available", dan mendapatkan NAN (not a number) dalam tabel. Terjemahan ini dapat diubah dengan parameter NA dan NAval.

```
>MT[1]
```

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

Berikut ini konten 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 menempatkan keluaran dari `readtable()` ke dalam daftar.

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

Dengan kolom token yang sama dan token yang dibaca dari file, kami dapat mencetak tabel. Kami dapat menentukan `ctok`, `tok`, dll. atau menggunakan daftar Tabel.

```
>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 kolom tabel, melompati baris dengan nilai NAN (". " dalam file), dan indeks kolom yang berisi nilai-nilai ini.

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

Kami dapat menggunakan ini untuk mengekstrak kolom dari tabel untuk tabel baru.

```
>j=[1,5,6]; writetable(MT[i,j],labr=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 Tabel dalam kasus ini.

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

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

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

```
2.175
```

Fungsi `getstatistics()` mengembalikan elemen dalam vektor, dan jumlahnya. Kami mengaplikasikannya pada nilai "m" dan "f" dalam kolom kedua tabel kami.

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

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

Kami dapat mencetak hasilnya dalam tabel baru.

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

m	12
f	13

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

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

```
[5, 6]
```

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

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

Sekarang kita bisa 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 mengaitkan dua kolom dalam tabel. Jadi kita ekstrak kolom 2 dan 4, dan 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 mengaitkan 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 file.

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

Kemudian kita bisa membaca tabel dari file.

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

	n	y
m	7	5
f	1	12

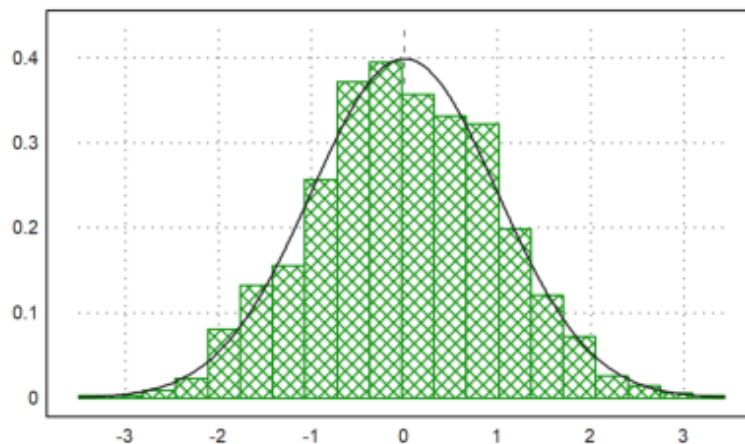
Dan menghapus file.

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

Distribusi

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

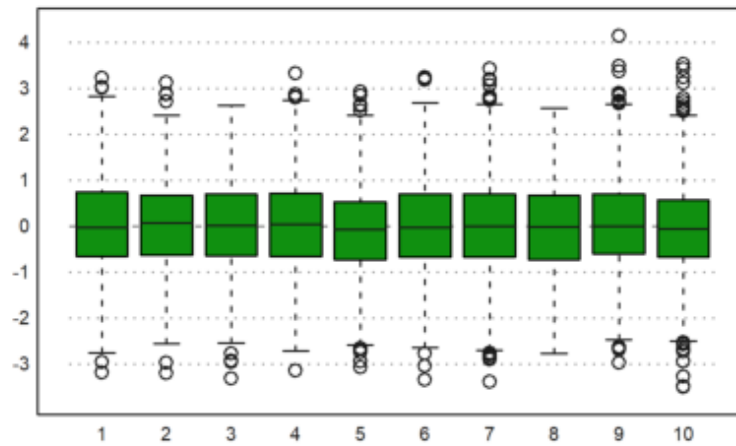
```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="\"); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution plot
```



Harap perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi sebenarnya). Masukkan kembali tiga perintah ini untuk melihat hasil sampel lain.

Berikut perbandingan dari 10 simulasi dari 1000 nilai yang didistribusikan secara normal menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, dan outlier.

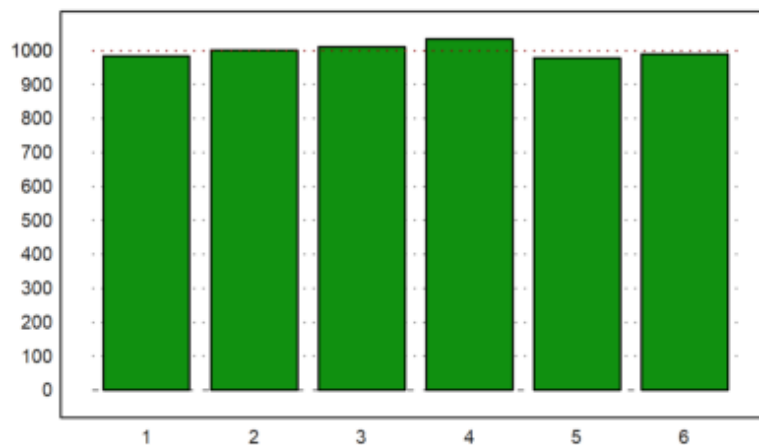
```
>p=normal(10,1000); boxplot(p):
```

Untuk menghasilkan angka acak, Euler memiliki `inrandom`. Mari kita simulasi lemparan dadu dan gambar distribusinya.

Kita menggunakan fungsi `getmultiplicities(v,x)`, yang menghitung seberapa sering elemen-elemen v muncul dalam x . Kemudian kami menggambar hasilnya menggunakan `columnplot()`.

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



Sementara `inrandom(n,m,k)` mengembalikan bilangan bulat yang didistribusikan secara merata dari 1 hingga k , mungkin ada kemungkinan menggunakan distribusi bilangan bulat lainnya dengan `randpint()`.

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

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

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

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Lihatlah di referensi.

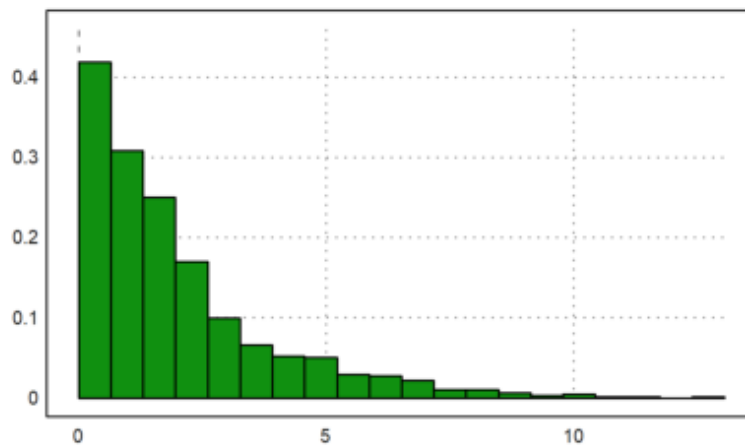
Misalnya, kita mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

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

dengan parameter

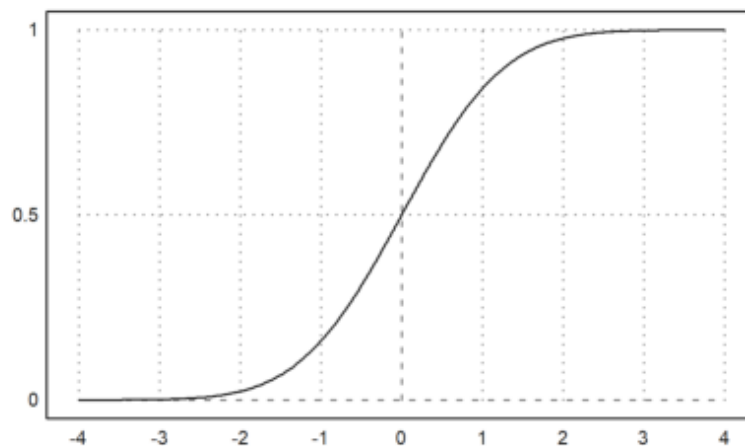
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ is the mean, and denoted by } X \sim \text{Exponential}(\lambda).$$

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



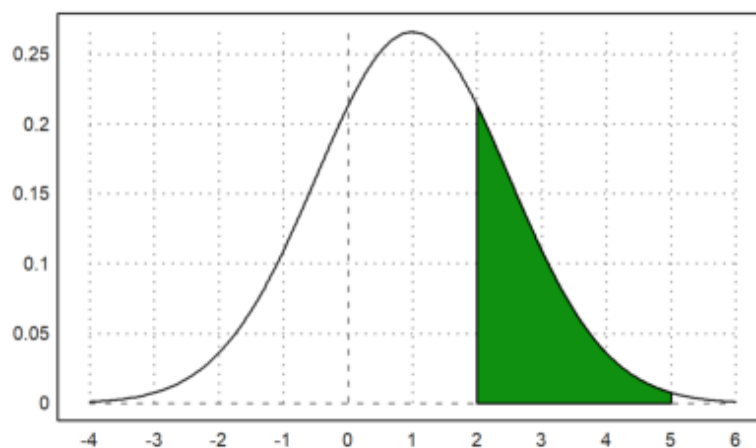
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

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



Berikut adalah salah satu cara untuk menggambar 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
```

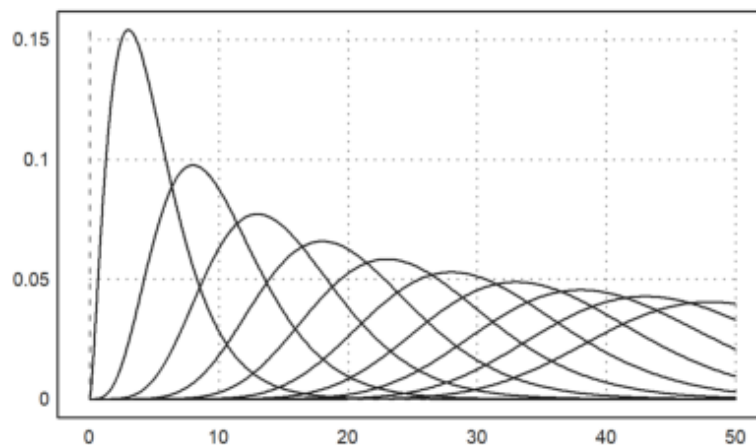
Mari kita bandingkan distribusi binomial dengan distribusi normal dengan rata-rata dan deviasi yang sama. Fungsi `invbindis()` menyelesaikan interpolasi linier antara nilai-nilai bulat.

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

```
525.516721219
526.007419394
```

Fungsi `qdis()` adalah kepadatan distribusi chi-square. Seperti biasa, Euler memetakan vektor ke fungsi ini. Dengan demikian, kita mendapatkan plot semua distribusi chi-square dengan derajat 5 hingga 30 dengan mudah 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.

Nama-nama mencoba untuk konsisten. Misalnya,

- distribusi chi-square adalah `chidis()`,
- fungsi invers adalah `invchidis()`,
- kepadatan adalah `qchidis()`.

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

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

```
0.527633447259
0.527633447259
```

Distribusi Diskrit

Untuk mendefinisikan distribusi diskrit Anda sendiri, Anda dapat menggunakan metode berikut.

Pertama, kita mengatur 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]
```

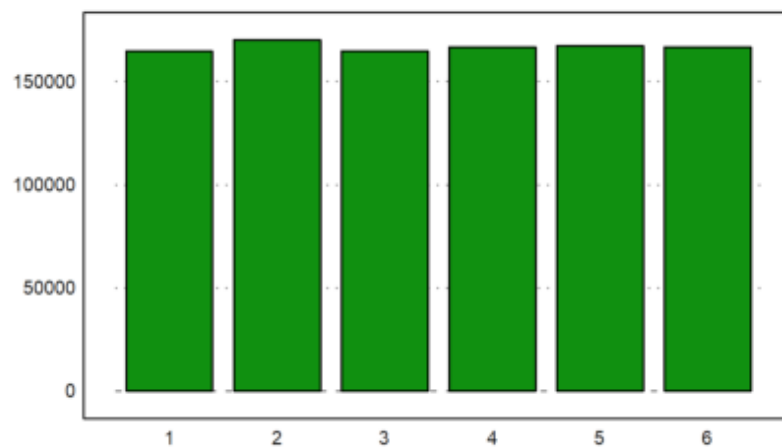
Artinya adalah bahwa dengan probabilitas $wd[i+1]-wd[i]$, kita menghasilkan nilai acak i .

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

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

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

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



Berikut adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai $1 \dots K$ 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 ini menerima generator acak bawaan.

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

1

Kita dapat menghitung distribusi binomial. Pertama ada `binomsum()`, yang mengembalikan probabilitas i atau lebih hasil dari n percobaan.

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

0.751401349654

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

Makna interval ini adalah jika p berada di luar interval, hasil yang diamati sebesar 410 dari 1000 adalah langka.

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

[0.37932, 0.441212]

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

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

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

409.932733047

Dalam Bridge, kita mengasumsikan 5 kartu terbuka (dari 52) dalam dua tangan (26 kartu). Mari kita hitung probabilitas 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])
```

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 memplot data, kita mencoba hasil pemilihan 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, kita menggunakan string nama-nama.

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

Mari kita mencetak persentasenya dengan baik.

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

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

Kemudian kita mencetak statistik dalam bentuk tabel. Kami menggunakan nama-nama sebagai header kolom, dan tahun-tahun sebagai header untuk baris. Lebar default untuk kolom adalah wc=10, tetapi kita lebih suka keluaran yang lebih padat. Kolom akan diperluas untuk 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

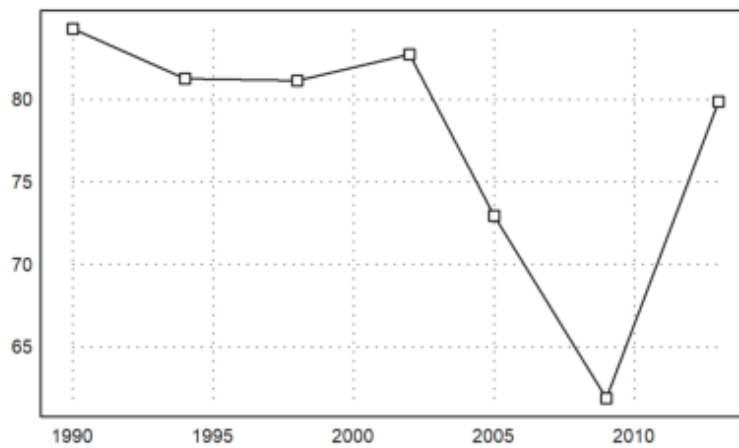
Perkalian matriks berikutnya mengekstraksi jumlah persentase dua partai besar, menunjukkan bahwa partai-partai kecil telah mendapatkan kursi di parlemen hingga tahun 2009.

```
>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. Kami 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)];
```

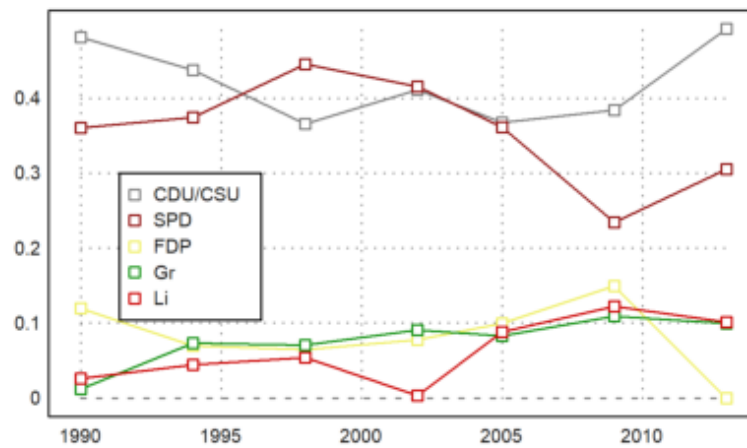
Sekarang kita bisa memplot hasil pemilihan tahun 2009 dan perubahan dalam satu plot menggunakan figure. Kita bisa menambahkan vektor kolom ke masing-masing plot.

```
>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):
```



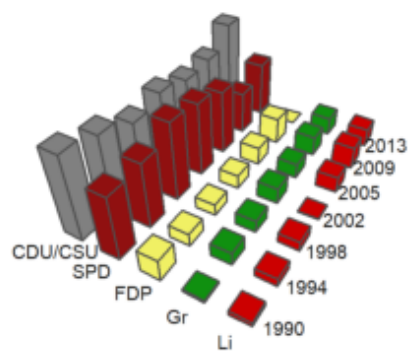
Plot data menggabungkan baris data statistik dalam satu plot.

```
>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):
```



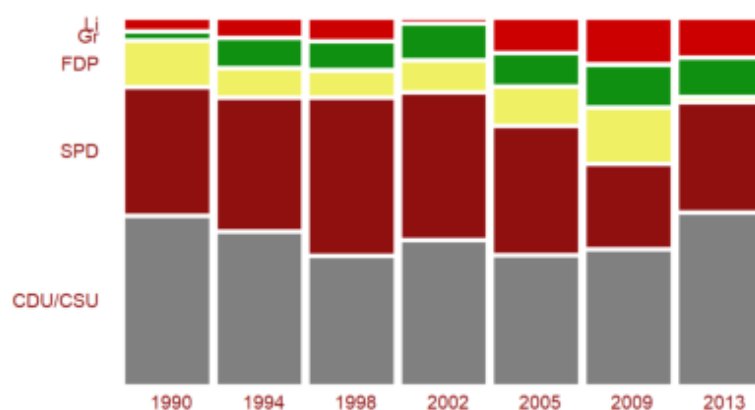
Plot kolom 3D menampilkan baris data statistik dalam bentuk kolom. Kami memberikan label untuk baris dan kolom. Sudut adalah sudut pandangan.

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



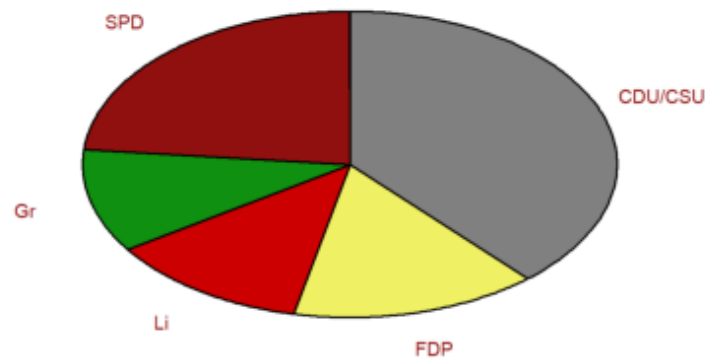
Representasi lainnya adalah plot mozaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kita mengambil jendela yang lebih kecil dari biasanya.

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



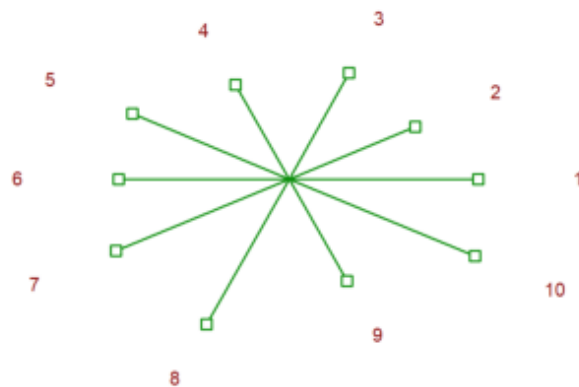
Kita juga bisa membuat pie chart. Karena hitam dan kuning membentuk koalisi, kita mengurutkan ulang elemen-elemen.

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



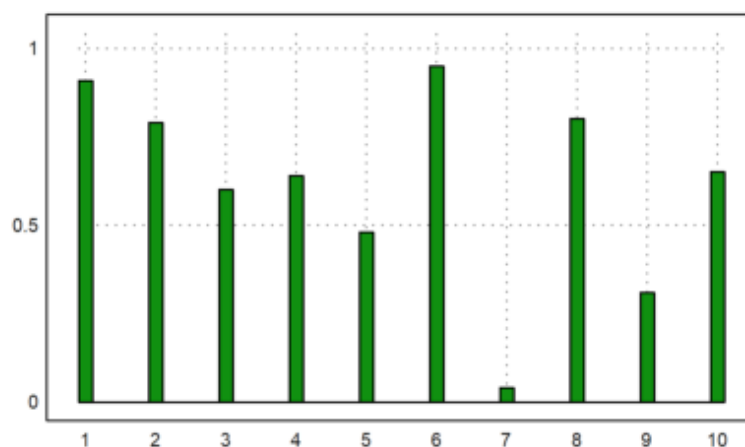
Berikut adalah jenis plot lainnya.

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



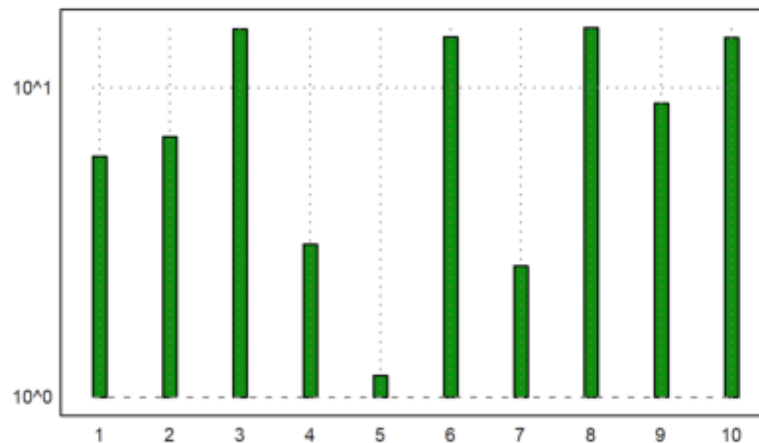
Beberapa plot dalam plot2d cocok untuk statistik. Berikut adalah plot impuls dari data acak, didistribusikan secara seragam dalam [0,1].

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



Tetapi untuk data yang didistribusikan secara eksponensial, kita mungkin memerlukan plot logaritmik.

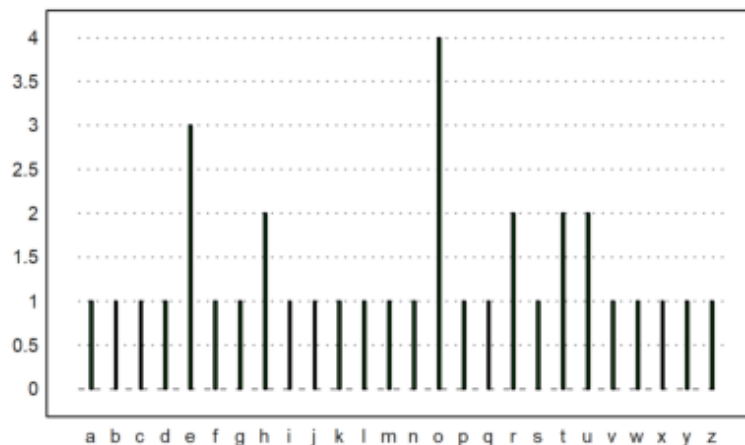
```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi `columnplot()` lebih mudah digunakan, karena hanya memerlukan vektor nilai. Selain itu, ia dapat mengatur labelnya menjadi apa pun yang kita inginkan, seperti yang telah kami tunjukkan dalam tutorial ini.

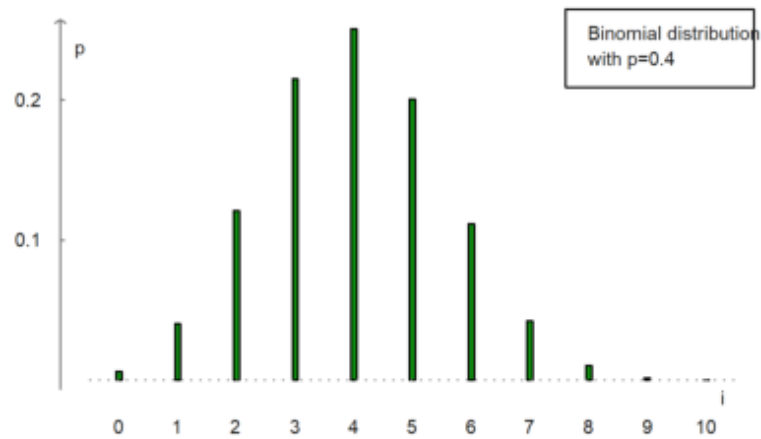
Berikut adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan memplot statistik.

```
>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 memungkinkan untuk secara manual menetapkan sumbu.

```
>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 memplot frekuensi angka dalam vektor.

Kami membuat vektor angka acak bulat 1 hingga 6.

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

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

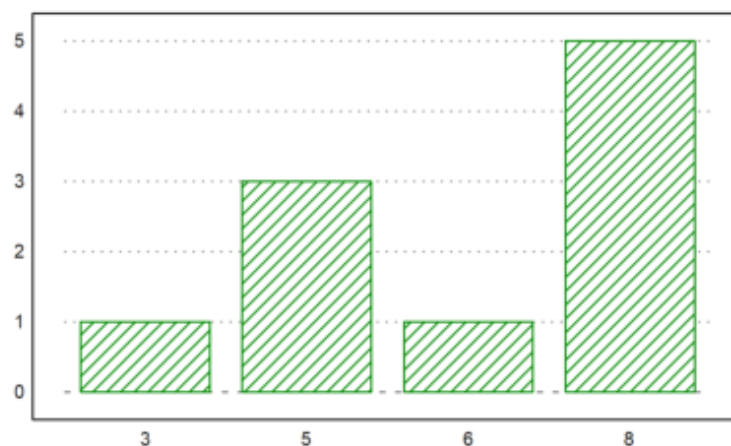
Kemudian ekstrak angka-angka unik dalam v.

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

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

Dan memplot frekuensi dalam plot kolom.

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



Kami ingin mendemonstrasikan fungsi distribusi empiris nilai.

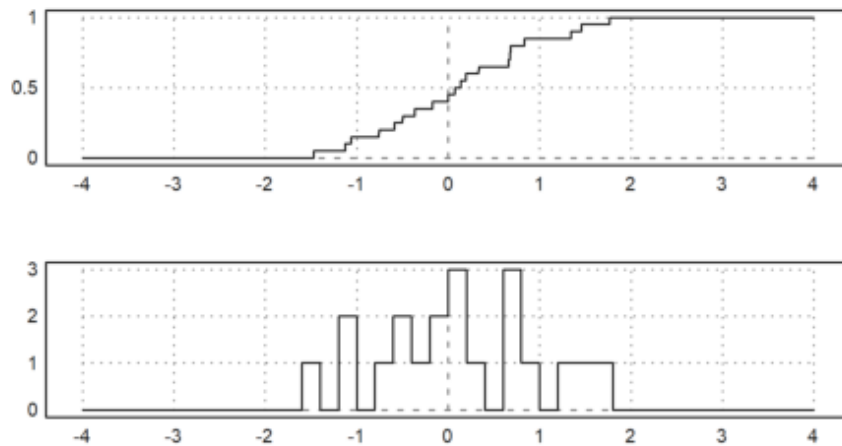
```
>x=normal(1,20);
```

Fungsi `empdist(x,vs)` memerlukan array nilai yang sudah diurutkan. Jadi kita harus mengurutkan x sebelum menggunakannya.

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

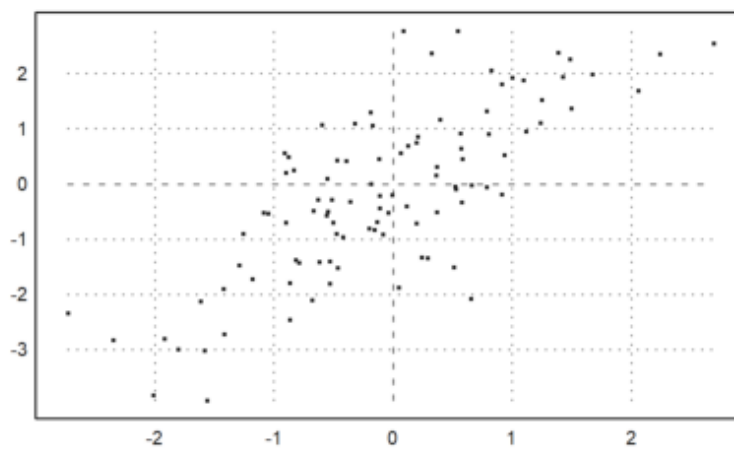
Kemudian kita memplot distribusi empiris dan beberapa bar densitas dalam satu plot. Alih-alih plot kolom untuk distribusi, kita kali ini menggunakan plot gergaji.

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



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

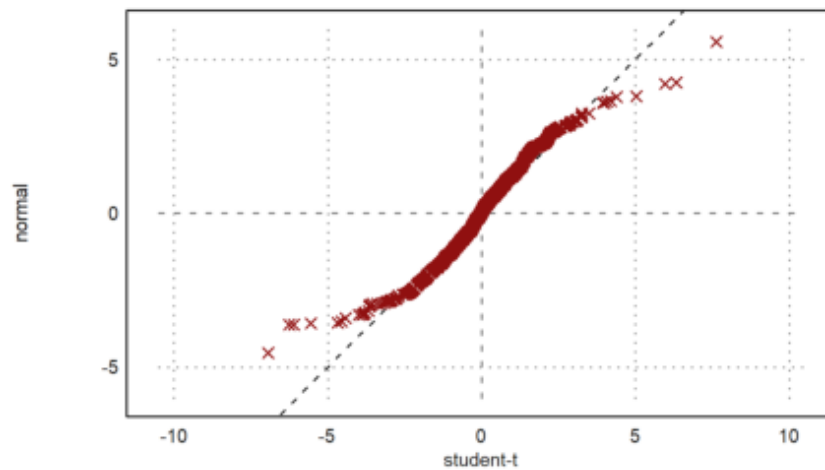
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



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

Untuk pengujian, kita mencoba distribusi student-t 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):
```



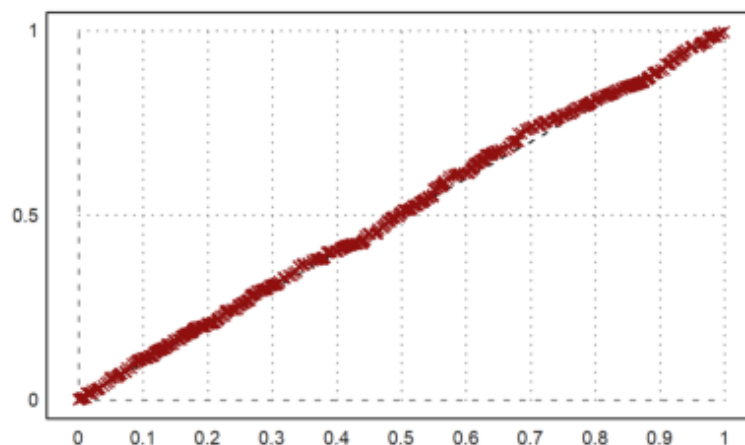
Plot ini dengan jelas menunjukkan bahwa nilai yang didistribusikan secara normal cenderung lebih kecil di ujung-ujung ekstrim.

Jika kita memiliki dua distribusi ukuran yang berbeda, kita dapat memperluas yang lebih kecil atau mengecilkan yang lebih besar. Fungsi berikut baik 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 fungsi `polyfit()` atau berbagai fungsi fit.

Untuk awal, kita 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',labco=["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 fit non-bobot dan fit berbobot. Pertama, koefisien dari fit linier.

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

```
[0.733333, 0.812121]
```

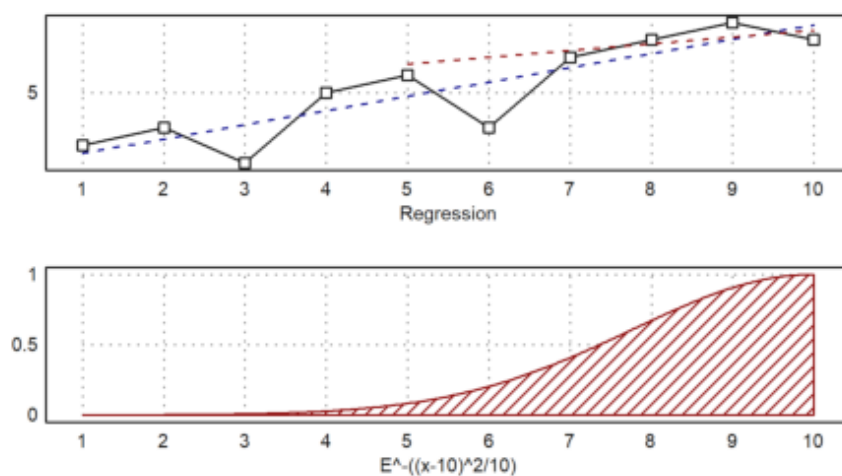
Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

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

```
[4.71566, 0.38319]
```

Kita menempatkan semuanya dalam satu plot untuk titik-titik dan 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 siswa, usia mereka, usia orangtua mereka, dan jumlah saudara kandung dari file.

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

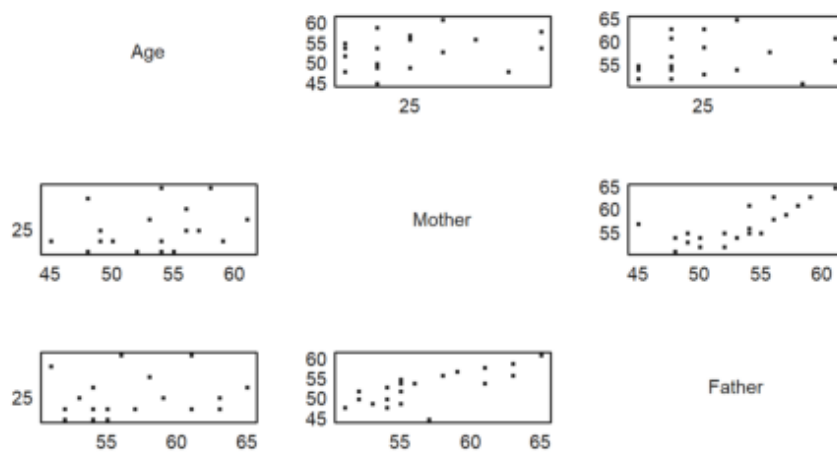
```
>{MS,hd}:=readtable("table1.dat",tok2:["m","f"]); ...
writetable(MS,labc=hd,tok2:["m","f"]);
```

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 saling bergantung? Kesimpulan pertama datang dari pairwise scatterplot.

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



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

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

```
[17.3789, 0.740964]
```

Ini jelas adalah model yang salah. Garis regresi 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.

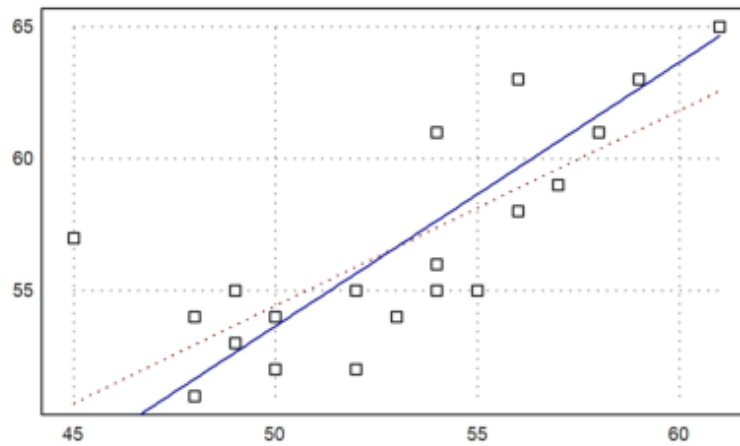
Lebih tepatnya, kita menduga fungsi seperti $s=a+t$. Maka a adalah rata-rata dari $s-t$. Itu adalah selisih usia rata-rata antara ayah dan ibu.

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

```
3.65
```

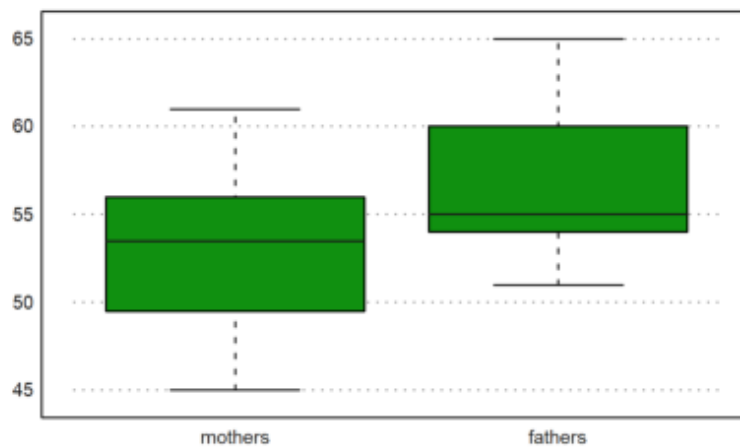
Mari kita plot ini dalam satu scatter plot.

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



Berikut adalah plot kotak dari dua usia tersebut. Ini hanya menunjukkan bahwa usia mereka berbeda.

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



Menarik bahwa 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 urutan yang sama dalam kedua vektor. Ini juga cukup positif.

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

```
0.758925292358
```

Membuat Fungsi Baru

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi baru. Misalnya, kita mendefinisikan fungsi 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 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 singkat dan efisien. Mari kita coba fungsi ini.

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

```
-0.198710316203
```

Berikut adalah fungsi lain, yang disebut koefisien skewness Pearson.

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

```
-0.0801873249135
```

Simulasi Monte Carlo

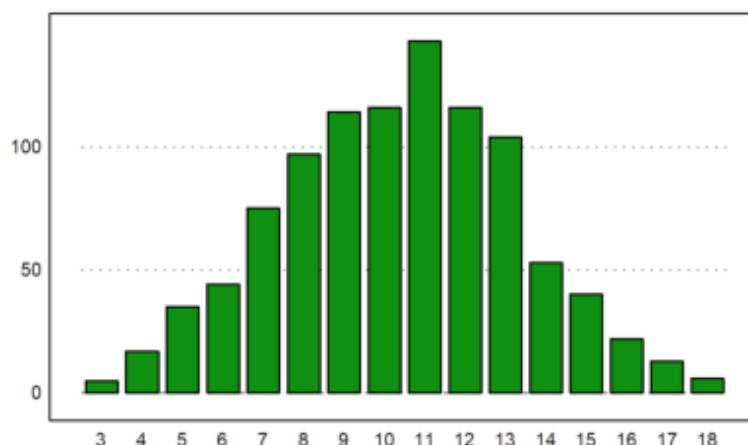
Euler dapat digunakan untuk mensimulasikan peristiwa acak. Kita telah melihat contoh-contoh sederhana di atas. Berikut adalah contoh lain, yang mensimulasikan 1000 kali lemparan tiga dadu, dan bertanya tentang distribusi jumlahnya.

```
>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 bisa memplotnya.

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



Untuk menentukan distribusi yang diharapkan tidak semudah itu. Kami menggunakan rekursi lanjutan untuk ini.

Fungsi berikut menghitung jumlah cara di mana angka k dapat direpresentasikan sebagai jumlah 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 hasilnya untuk tiga lemparan dadu.

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

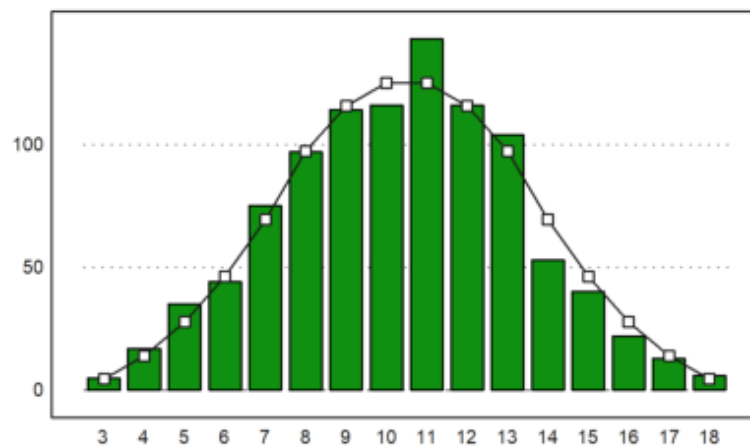
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,
255, 185, 121, 70, 35, 15, 5, 1]
```

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

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

Kami menambahkan nilai yang diharapkan ke plot.

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



Untuk simulasi lain, deviasi dari nilai rata-rata n variabel acak yang didistribusikan secara normal 0-1 adalah $1/\sqrt{n}$.

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

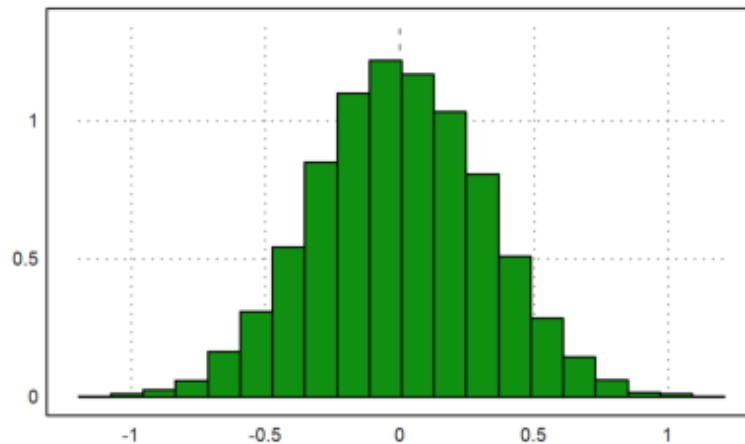
```
0.316227766017
```

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

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

0.319493614817

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



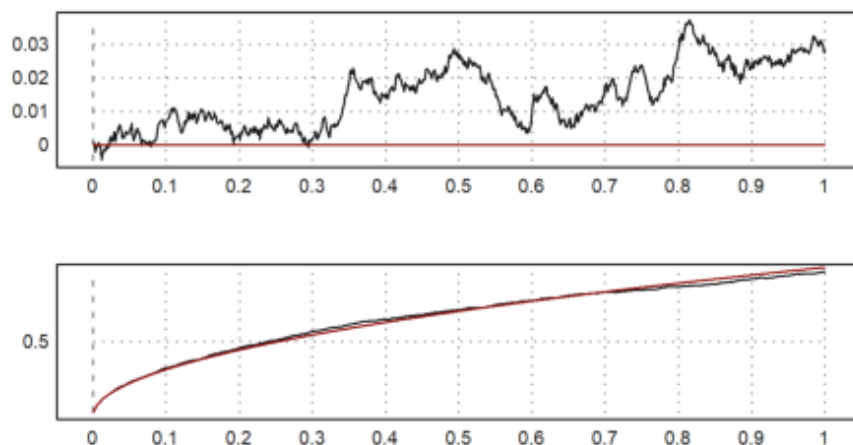
Median dari 10 angka acak yang didistribusikan secara normal 0-1 memiliki deviasi yang lebih besar.

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

0.374460271535

Karena kita dapat dengan mudah menghasilkan perjalanan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kemudian kita memplot deviasi standar dan rata-rata langkah ke-n dari proses-proses ini bersama dengan nilai yang diharapkan dalam 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 adalah alat penting dalam statistik. Di Euler, banyak uji 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, kami mendapatkan nilai-nilai berikut, yang kami masukkan ke 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 mengartikan vektor y sebagai vektor probabilitas.

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

```
0.526
```

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

Selanjutnya kami 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. Kami menolak hipotesis bahwa mereka memiliki rata-rata yang sama, jika hasilnya <0,05.

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

```
0.38722000942
```

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

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

```
5.60009101758e-07
```

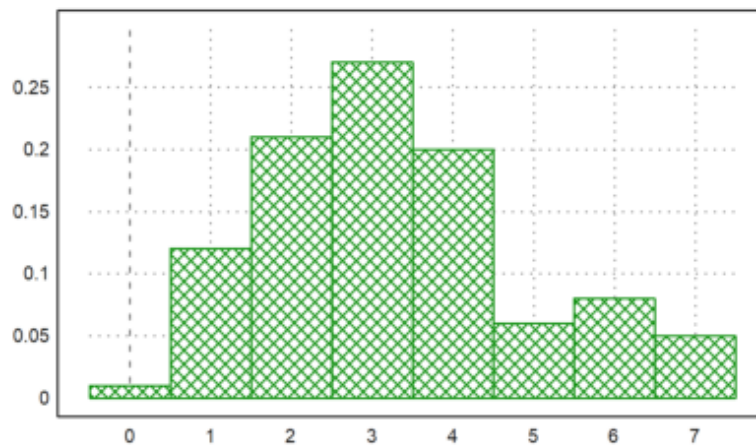
Dalam contoh berikut, kami menghasilkan 20 lemparan dadu acak 100 kali dan menghitung jumlah satunya. Harus ada $20/6=3,3$ satu rata-rata.

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

```
3.28
```

Sekarang kami membandingkan jumlah satu dengan distribusi binomial. Pertama kami memplot distribusi satu.

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



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

Kemudian kami menghitung nilai yang diharapkan.

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

Kami 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 kami adalah distribusi binomial, jika hasilnya <0,05.

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

```
0.53921579764
```

Contoh berikut berisi hasil dari dua kelompok orang (laki-laki dan perempuan, katakanlah) yang memilih 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

Kami ingin menguji independensi suara dari jenis kelamin. Uji tabel chi-kuadrat melakukan ini. Hasilnya terlalu besar untuk menolak independensi. Jadi kami tidak dapat mengatakan, apakah pemilihan tergantung 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

Kami dapat menghitung koefisien kontingensi yang diperbaiki. Karena sangat dekat dengan 0, kami menyimpulkan bahwa pemilihan tidak tergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```

Beberapa Uji Lainnya

Selanjutnya kami menggunakan analisis variansi (uji F) untuk menguji tiga sampel data yang didistribusikan secara normal untuk nilai rata-rata yang sama. Metodenya disebut ANOVA (analisis variansi). 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. Kami melakukan ini dengan kemungkinan kesalahan sebesar 1,3%.

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

```
>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 tentang kesetaraan adalah uji peringkat. Ini jauh lebih tajam daripada uji median.

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

```
0.00199969612469
```

Dalam contoh berikut, 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 diberikan kepada 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 menentukan apakah a lebih baik daripada b.

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

```
0.0546875
```

Ini terlalu besar kesalahan. Kami tidak dapat menolak bahwa a sebaik b.

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

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

```
0.0296680599405
```

Mari kita mencoba dua uji lain menggunakan rangkaian yang dihasilkan.

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

```
0.0068706451766
```

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

```
0.275145971064
```

Angka Acak

Berikut adalah tes untuk generator angka acak. Euler menggunakan generator yang sangat baik, sehingga kita tidak perlu mengharapakan masalah apa pun.

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

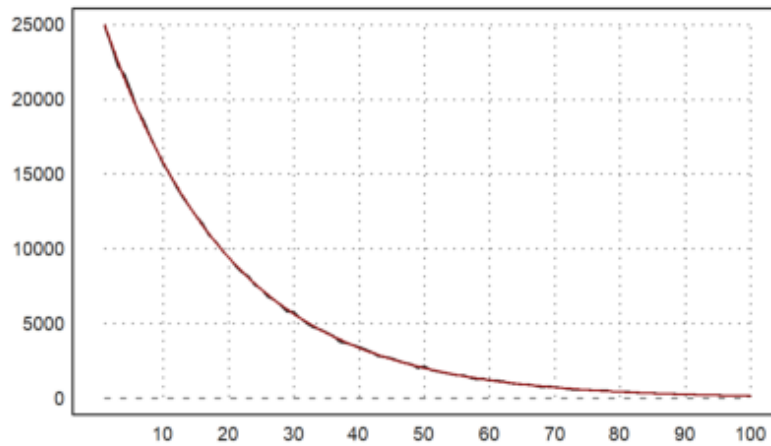
```
>n:=10000000; r:=random(1,n);
```

Selanjutnya kami menghitung jarak antara dua angka yang kurang dari 0,05.

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

Terakhir, kami memplot 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

elas, EMT tidak bersaing dengan R sebagai paket statistik. Namun, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi EMT mungkin 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 sintaksis EMT dan R. Kami mencoba memberikan gambaran tentang hal-hal yang perlu Anda ketahui, baik yang terlihat maupun yang kurang terlihat.

Selain itu, kami akan melihat cara pertukaran data antara kedua sistem tersebut.

Perhatikan bahwa ini masih dalam tahap pengembangan.

Basic Syntax

Hal pertama yang Anda pelajari di R adalah membuat vektor. Di EMT, perbedaan utamanya adalah operator ":" dapat mengambil langkah. Selain itu, ini memiliki kekuatan ikatan 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. Memungkinkan untuk menggunakan vektor untuk menggabungkan hal-hal.

Contoh berikut adalah, seperti banyak lainnya, dari "Pengenalan ke R" yang disertakan dalam 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 EMT digantikan oleh fungsi `seq()` di R. Kita bisa menulis fungsi ini dalam 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() R tidak ada dalam EMT. Untuk input vektor, bisa ditulis seperti 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 satuan di EMT.

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

```
77.6713990297 miles
```

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

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

```
0
```

Di R, "a<-4<3" " berfungsi, tetapi "a<-4<-3" tidak. Saya memiliki ambiguitas serupa di EMT juga, tetapi mencoba untuk menghilangkannya perlahan-lahan.

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

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

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

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

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

```
NAN
1
```

String sama di R dan EMT. Keduanya dalam bahasa lokal saat ini, bukan dalam 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 akan atau tidak akan tampil dengan benar di sistem Anda sebagai A dengan titik dan garis atas. Itu tergantung pada font yang Anda gunakan.

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

Ä

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

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

```
pi = 3.14159265359
```

Indexing

Sebagian besar waktu, ini akan berfungsi seperti di R.

Tetapi 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 bisa dicapai di EMT dengan `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 yang tidak 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]
```

Seperti di R, vektor indeks dapat berisi pengulangan.

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

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

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

Untuk meniru perilaku ini, kita bisa 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)]; ... ..
```

[10.4, 3.1]

Data Types

EMT memiliki lebih banyak tipe data yang tetap daripada R. Jelas, di R ada vektor yang berkembang. Anda bisa mengatur vektor numerik kosong `v` dan memberikan nilai pada elemen `v[17]`. Ini tidak mungkin di EMT.

Berikut adalah sedikit tidak efisien.

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

EMT sekarang akan membuat vektor dengan `v` dan `i` ditambahkan ke tumpukan dan menyalin kembali vektor itu ke variabel global `v`.

Yang lebih efisien lebih didefinisikan sebelumnya.

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

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

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

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

Konversi menjadi 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 bisa 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 berkomunikasi 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]
```

Factors and Tables

Dalam pengenalan ke R ada contoh dengan faktor yang disebut faktor.

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, kami 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, kami ingin menghitung rata-rata pendapatan di wilayah. Sebagai program statistik, R memiliki faktor() dan tapply() untuk ini.

EMT dapat melakukannya dengan menemukan indeks wilayah dalam daftar wilayah 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 saat itu, kami bisa menulis fungsi loop kami sendiri untuk melakukan hal-hal untuk satu faktor saja.

Atau kita bisa 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 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]
```

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

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

[44.5, 57.3333333333]
```

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

Jadi kita bisa mencetak rata-rata pendapatan negara bagian dalam 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 bisa mencoba meniru perilaku R sepenuhnya.

Faktor jelas harus tetap dalam koleksi dengan tipe dan kategori (negara bagian dan wilayah dalam contoh kami). Untuk EMT, kami 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 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 berkaitan dengan kategori (faktor) tanpa data. Tetapi Anda harus memeriksa panjang t yang benar 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 disebut matriks. Akan mudah untuk menulis fungsi untuk dimensi lebih tinggi atau perpustakaan C untuk ini, bagaimanapun.

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

Dalam EMT, 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 di R.

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

2	3
7	8
12	13
17	18

Namun, di R mungkin untuk mengatur daftar indeks tertentu dari vektor ke nilai. Hal yang sama hanya mungkin di EMT dengan loop.

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

Kami mendemonstrasikannya untuk menunjukkan bahwa matriks diteruskan 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

Perkalian luar di EMT hanya bisa dilakukan antara vektor. Ini otomatis karena bahasa matriks. Salah satu vektor harus menjadi vektor kolom dan yang lainnya vektor baris.

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

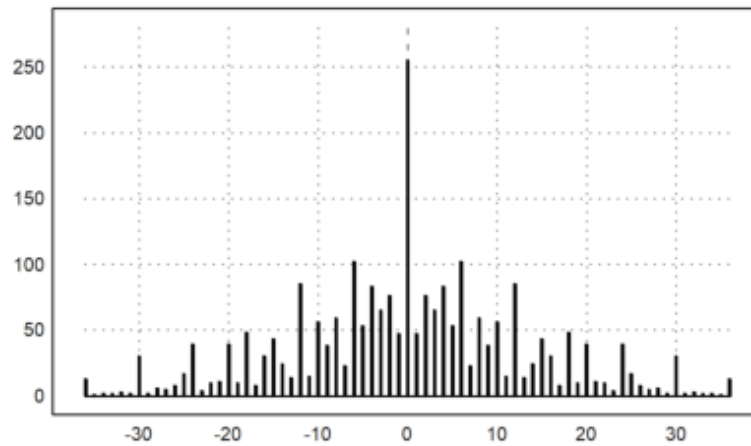
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

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

Tentu saja, ini bisa dicapai dengan loop. Tetapi loop tidak efektif dalam EMT atau R. Dalam EMT, kita bisa menulis loop dalam C dan itu akan menjadi solusi yang paling cepat.

Tetapi kami 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"):
```



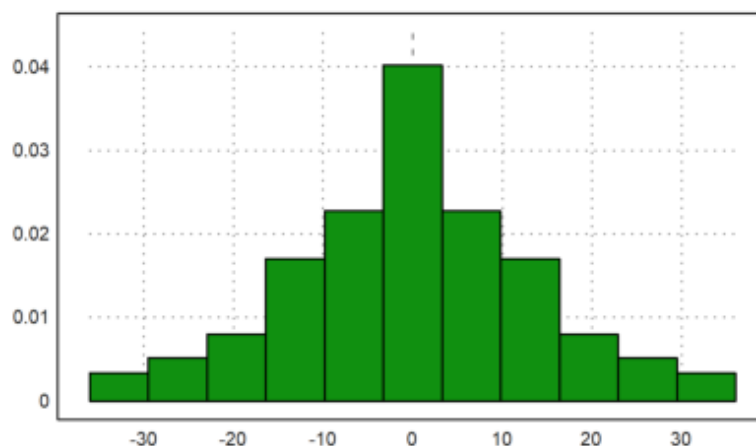
Selain kelipatan yang tepat, EMT dapat menghitung frekuensi dalam vektor.

```
>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.

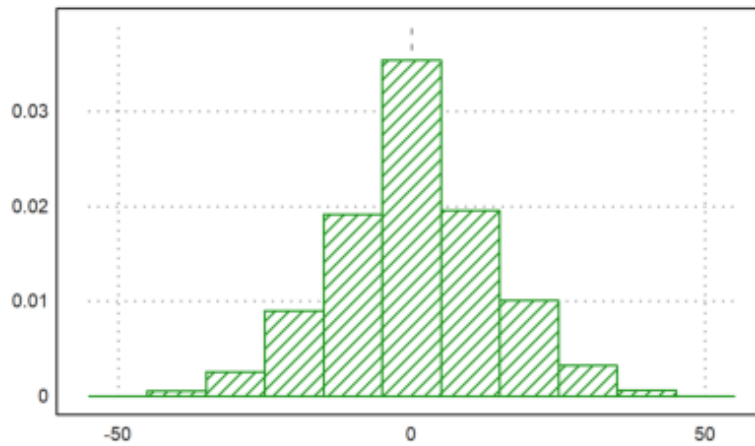
```
>plot2d(q,distribution=ll):
```



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

Karena fungsi `histo()` mengembalikan frekuensi, kami perlu menskalakan frekuensi ini sehingga integral di bawah grafik batang adalah 1.

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



Lists

EMT memiliki dua jenis daftar. Satu adalah daftar global yang dapat diubah, dan yang lain adalah tipe daftar yang tidak dapat diubah. Kami tidak memperhatikan daftar global di sini.

Tipe daftar yang tidak dapat diubah disebut koleksi dalam EMT. Ini berperilaku seperti struktur dalam C, tetapi elemennya hanya dinomori 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 File (Membaca dan Menulis Data)

Anda sering ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini memberi tahu Anda tentang banyak cara untuk mencapainya. Fungsi sederhana adalah `writematrix()` dan `readmatrix()`.

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

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

```
0.49815
0.28037
```

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

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

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

Sekarang kita menulis vektor kolom `a'` ke file. Ini menghasilkan satu angka dalam setiap baris file.


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

Untuk membaca data, kita gunakan 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 berbahasa Indonesia (tanda desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan oleh koma). File berikut, "test.csv," harus muncul di folder saat ini.

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

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

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

Terkadang kita memiliki string dengan token seperti berikut.

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

Untuk membagi string ini, kita mendefinisikan vektor token.

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

```
f
m
```

Kemudian kita dapat menghitung jumlah kemunculan masing-masing token dalam string tersebut, dan menempatkan hasilnya ke dalam sebuah tabel.

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

Tulis tabel dengan header 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 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 bentuk yang paling sederhana dapat membaca ini dan mengembalikan koleksi nilai dan baris judul.

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

Koleksi ini dapat dicetak dengan `writetable()` ke notebook, atau ke 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 matriks.

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

```
0.40472
0.37102
0.75547
```

File CSV

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

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

Berikut ini adalah isi dari file tersebut.

```
>printfile(file)
```

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

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

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

```
>readmatrix(file)
```

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

Mungkin kita ingin menulis beberapa matriks ke dalam satu file. Perintah `open()` dapat membuka file untuk penulisan dengan parameter "w". Defaultnya adalah "r" untuk membaca.

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

Matriks dipisahkan oleh 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 oleh koma). Di Excel 2007, gunakan "simpan sebagai" dan "format lain," lalu pilih "CSV". Pastikan tabel saat ini hanya berisi data yang ingin Anda ekspor.

Berikut 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 bisa Anda lihat, sistem berbahasa Jerman saya telah menggunakan titik koma sebagai pemisah dan koma desimal. Anda dapat mengubah ini di pengaturan sistem atau di Excel, tetapi itu tidak diperlukan untuk membaca matriks ke EMT.

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

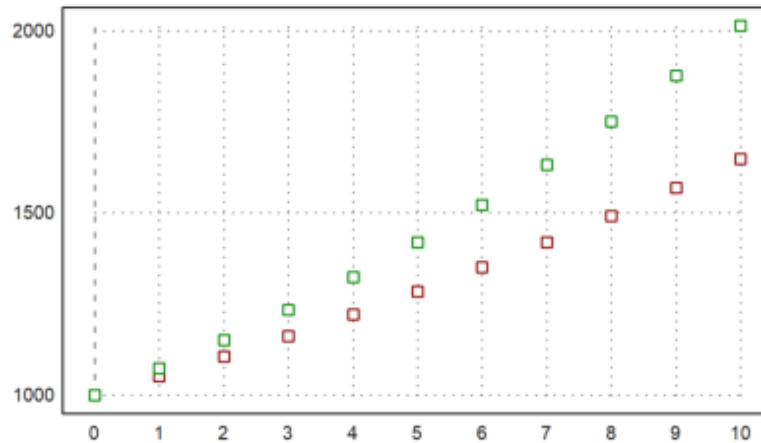
```
>M=readmatrix("excel-data.csv",>comma)
```

```
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 gambarkan ini.

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



Ada cara yang lebih sederhana untuk membaca data dari file. Anda dapat membuka file dan membaca angka-angka baris demi baris. Fungsi `getvectorline()` akan membaca angka dari baris data. Secara default, ia mengharapkan titik desimal. Tetapi ia juga dapat menggunakan koma desimal, jika Anda memanggil `setdecimaldot(",")` sebelum Anda menggunakan fungsi ini.

Fungsi berikut adalah contoh untuk ini. Ia 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

Juga mungkin membaca semua angka dalam file 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

Jadi sangat mudah untuk menyimpan vektor nilai, satu nilai dalam setiap baris, dan membacanya kembali.

```
>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 baris dan judul kolom ke dalam 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 dalam 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 Sebuah Baris

Anda bahkan bisa mengevaluasi setiap baris secara manual. Misalkan kita memiliki baris dengan format berikut.

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

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

Pertama-tama, kita dapat memecah baris tersebut menjadi token.

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

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

Kemudian kita dapat mengevaluasi setiap elemen baris 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 baris data.

Misalkan kita memiliki baris berikut dalam 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

- a closing bracket >,
- any string not containing brackets with a sub-match "...)",
- an opening and a closing bracket using the shortest solution,
- again any string not containing brackets,
- and an opening bracket <.

Ekspresi reguler agak sulit untuk dipelajari tetapi sangat kuat.

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

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

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

```
1145.5
5.6
```

Berikut 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

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

Dalam contoh ini, kita membaca versi terbaru dari situs EMT. Kami menggunakan ekspresi reguler untuk mencari "Versi ..." dalam 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
```

```
Version 2022-05-18
```

Input dan Output Variabel

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

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

```
mypi = 3.141592653589793;
```

Sebagai tes, kita menghasilkan 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];
```

Kita sekarang dapat memuat file tersebut. Ini akan mendefinisikan matriks M.

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

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

Selain itu, jika `writevar()` digunakan pada variabel, itu akan mencetak definisi variabel dengan nama variabel ini.

```
>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, kita menambahkan ke file yang sudah 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 file tidak memerlukan koma, jika setiap angka ada di baris yang berbeda. Mari kita menghasilkan file seperti itu, dengan 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 data berat badan dari siswa kelas B
56, 55, 60, 47, 65, 56, 57, 62, 55, 40, 47, 45, 42, 68, 54, 48, 49, 38, 54, 47.

Carilah rata-rata dan standar deviasinya.

```
>TB=[56, 55, 60, 47, 65, 56, 57, 62, 55, 40, 47, 45, 42, 68, 54, 48, 49, 38, 54, 47]; ..
mean(TB), dev(TB)
```

```
52.25
8.14264278837
```

Jadi, rata-rata berat badan dari siswa kelas B adalah 52,25 dan standar deviasinya 8,14264278837

2. Diketahui rata-rata nilai ulangan matematika kelas A adalah 78.5, kelas B 79.7, kelas C 76.4, kelas D 72.7, kelas E 70.9, kelas F 71.2, dan kelas G 80.2. Hitunglah rata-rata nilai matematika untuk seluruh kelas.

```
>N=[78.5, 79.7, 76.4, 72.7, 70.9, 71.2, 80.2]; ...
```


mean (N)

75.6571428571

Jadi, rata-rata nilai ulangan matematika dari kelas A sampai G adalah 75.65.

3. Data berikut menunjukkan nilai yang diperoleh 50 siswa kelas 10 pada ujian mata pelajaran biologi. Siswa yang mendapat nilai dalam rentang 61-65 sebanyak 2 orang, dalam rentang 66-70 sebanyak 3 orang, dalam rentang 71-75 sebanyak 9 orang, dalam rentang 76-80 sebanyak 10 orang, dalam rentang 81-85 sebanyak 16 orang, dalam rentang 86-90 sebanyak 7 orang, dan dalam rentang 91-95 sebanyak 3 orang. Tentukan rata-rata, median, dan histogram dari data nilai ujian biologi kelas 10!

Penyelesaian :

Menentukan tepi bawah kelas yang terkecil

$>61-0.5$

60.5

Menentukan panjang kelas

$>(65-61)+1$

5

Menentukan tepi atas kelas yang terbesar

$>95+0.5$

95.5

```
>r=60.5:5:95.5; v=[2,3,9,10,16,7,3];
>T:=r[1:7]' | r[2:8]' | v'; writetable(T,labc=["TB","TA","Frek"])
```

TB	TA	Frek
60.5	65.5	2
65.5	70.5	3
70.5	75.5	9
75.5	80.5	10
80.5	85.5	16
85.5	90.5	7
90.5	95.5	3

Mencari Rata-rata

Menentukan nilai tengah

$>(T[,1]+T[,2])/2$

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
```

Mencari Median

Berdasarkan data, median berada pada urutan ke 25, maka median berada pada kelas 80.5-85.5

```
>Tb=80.5, p=5, n=50, Fks=24, fm=16
```

```
80.5
```

```
5
```

```
50
```

```
24
```

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

```
80.8125
```

Jadi, rata-rata untuk nilai ujian biologi kelas 10 adalah 79.8 dan mediannya adalah 80.81.

Histogram dari data nilai biologi kelas 10

```
>plot2d(r,v,a=61,b=95,c=0,d=20,bar=1,style="/") :
```

