

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

Mari kita mulai dengan beberapa statistik deskriptif. Ini bukanlah sebuah pengantar statistik. Jadi, Anda mungkin memerlukan beberapa latar belakang untuk memahami detailnya.

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

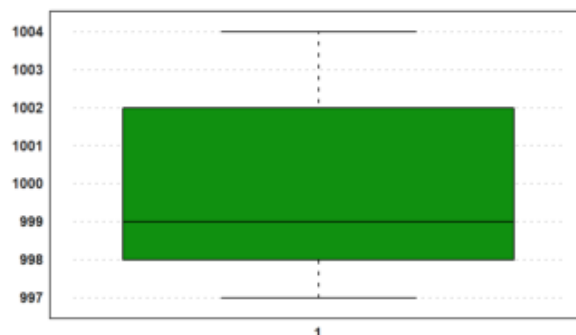
*** Translated with www.DeepL.com/Translator (free version) ***

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

```
999  
999.9  
2.72641400622
```

Kita dapat memplot plot kotak dan kumis untuk data tersebut. Dalam kasus kami, tidak ada pencilan.

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



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

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

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

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

*** Translated with www.DeepL.com/Translator (free version) ***

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

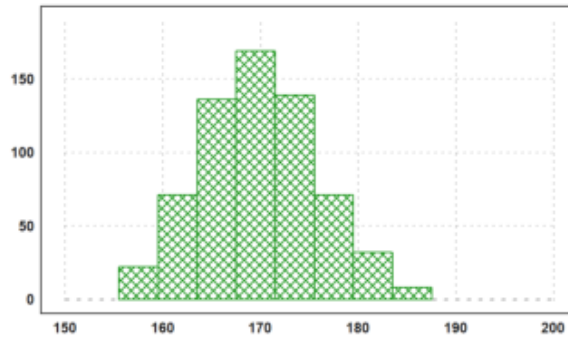
```
3.07 %
```

Untuk contoh berikutnya, kami mengasumsikan jumlah pria berikut dalam rentang ukuran tertentu.

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

Here is a plot of the distribution.

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



Kita dapat memasukkan data mentah tersebut ke dalam tabel.

Tabel adalah sebuah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal rentang, akhir rentang, jumlah orang 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 membutuhkan nilai rata-rata dan statistik lain dari ukuran, kita perlu menghitung titik tengah rentang. Kita dapat menggunakan dua kolom pertama dari tabel kita untuk ini.

Symbol "|" digunakan untuk memisahkan kolom, 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 akan lebih mudah, untuk melipat 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 bisa menghitung rata-rata dan deviasi sampel dengan frekuensi yang diberikan.

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

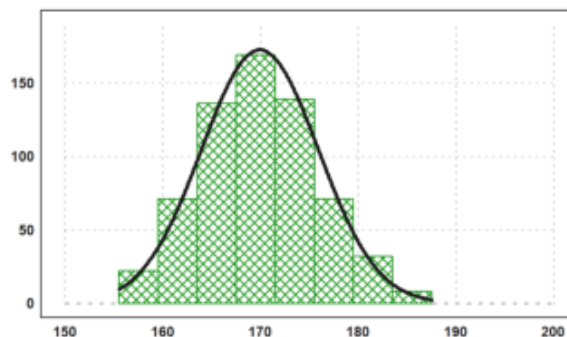
```
169.901234568
5.98912964449
```

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

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

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

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



Tabel

Dalam direktori buku catatan ini, Anda akan menemukan file dengan tabel. Data tersebut mewakili hasil survei. Berikut adalah empat baris pertama dari file tersebut. Data berasal dari buku online berbahasa 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 berisi 7 kolom angka atau token (string). Kita ingin membaca tabel tersebut dari file. Pertama, kita menggunakan terjemahan kita sendiri untuk token-token tersebut.

Untuk itu, kita 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`, dan seterusnya adalah terjemahan dari kolom-kolom pada tabel. Argumen-argumen ini tidak ada dalam daftar parameter `readtable()`, jadi Anda harus 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. Kami mencetak empat baris pertama saja.

```
>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-titik "." mewakili nilai yang tidak tersedia.

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

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

Fungsi `readtable()` sekarang mengembalikan satu set token.

```
>tok
```

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

Tabel berisi entri dari file dengan token yang diterjemahkan ke angka.

String khusus `NA=""` ditafsirkan sebagai "Tidak Tersedia", dan mendapatkan `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 tidak 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 token yang sama dan token yang dibaca dari file, kita dapat mencetak tabel. Kita 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 dari tabel, melewati setiap baris dengan nilai NAN (". " dalam file), dan indeks kolom, yang berisi nilai-nilai ini.

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

Kita dapat menggunakan ini untuk mengekstrak kolom dari tabel untuk 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 Tabel dalam kasus ini.

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

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

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

2.175

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

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

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

Kita bisa 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 dari dua nilai kita dalam tabel token.

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

```
[5, 6]
```

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

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

Sekarang kita dapat mencetak tabel, dengan nilai yang diekstrak dan diurutkan di 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 tabel. Jadi kita mengekstrak 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
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
f	y
f	y
f	y

Dengan `getstatistics()`, kita juga dapat menghubungkan hitungan 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 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

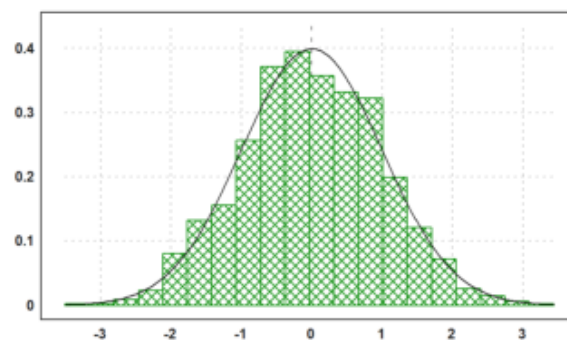
Dan hapus file tersebut.

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

Distribusi

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

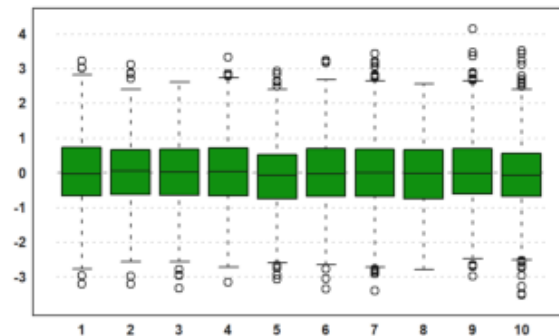
```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="\"); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution plot
```



Harap perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi nyata). Masukkan kembali ketiga perintah tersebut untuk melihat hasil pengambilan sampel yang lain.

Berikut ini adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal dengan menggunakan apa yang disebut box plot. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta pencilan.

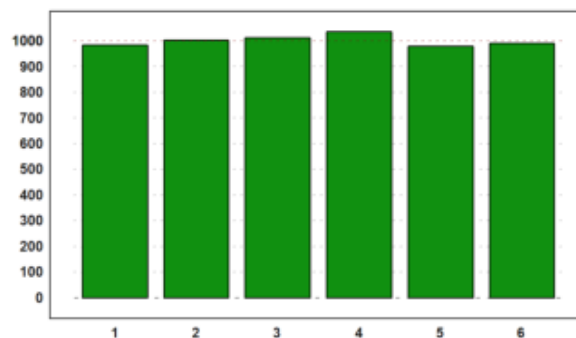
```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki intrandom. Mari kita simulasikan pelemparan dadu dan plot distribusinya.

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

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



Walaupun `intrandom(n,m,k)` mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 sampai `k`, adalah mungkin untuk menggunakan distribusi bilangan bulat yang lain dengan `randpint()`.

Pada contoh berikut, probabilitas untuk 1,2,3 adalah 0.4, 0.1, 0.5 secara berurutan.

```
>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 ke dalam referensi.

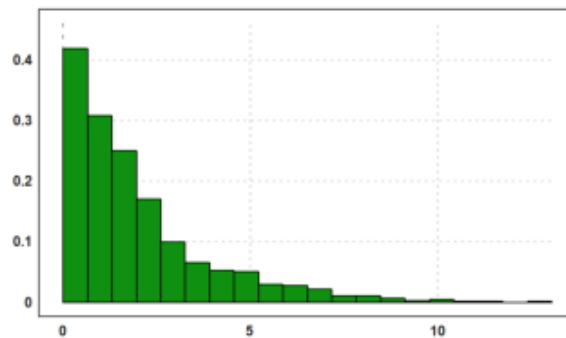
Misalnya, kita mencoba distribusi eksponensial. Sebuah variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

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

dengan parameter

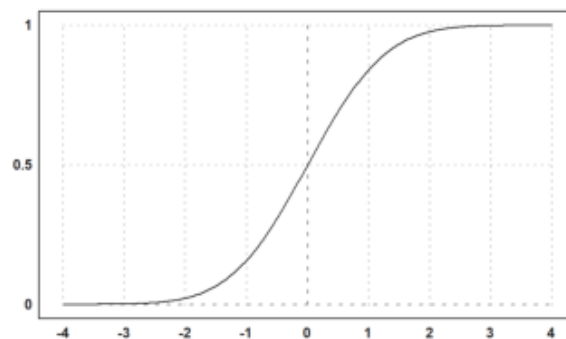
lateks: $\lambda = \frac{1}{\mu}$, μ adalah mean, dan dilambangkan dengan $X \sim \text{Eksponensial}(\lambda)$
*** Translated with www.DeepL.com/Translator (free version) ***

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



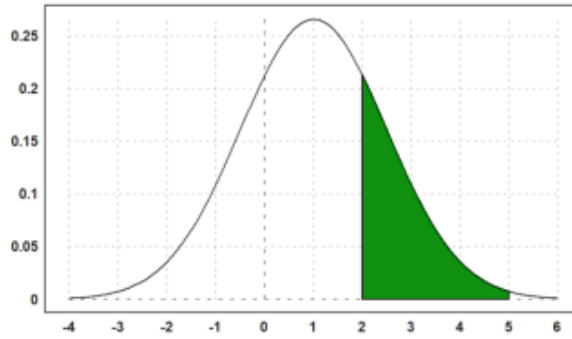
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan kebalikannya.

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



The following is one way to plot a quantile.

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



lateks: $\text{normaldis}(x,m,d)=\int_{-\infty}^x \frac{1}{d\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-m}{d})^2} dt$.
 Probabilitas untuk berada di area hijau adalah sebagai berikut.

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

0.248662156979

Hal ini dapat dihitung secara numerik dengan integral berikut ini.

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

```
>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 bilangan bulat.

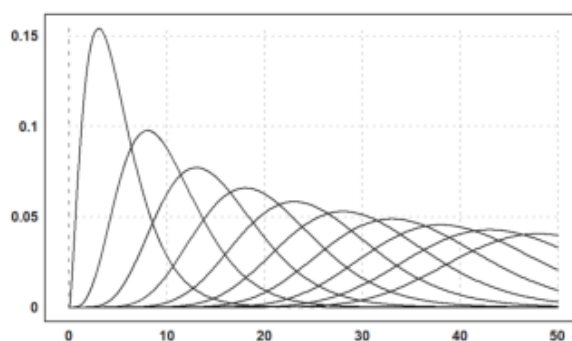
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

525.516721219

526.007419394

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

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



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

Penamaannya diusahakan untuk konsisten. Sebagai contoh,

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

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

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

```
0.527633447259  
0.527633447259
```

Distribusi Diskrit

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

Pertama, kita tetapkan fungsi distribusinya.

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

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

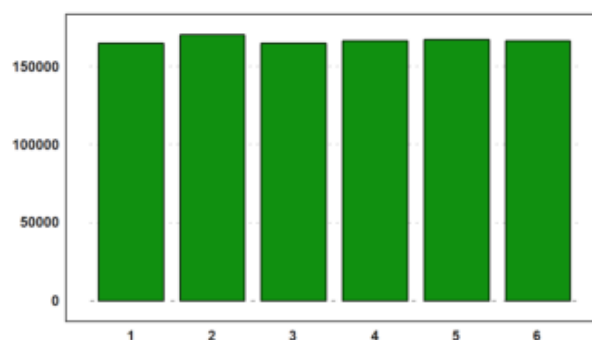
Artinya, dengan probabilitas $wd[i+1]-wd[i]$ kita menghasilkan nilai acak i .

Ini hampir merupakan distribusi yang seragam. Mari kita tentukan sebuah generator angka acak untuk ini. Fungsi `find(v,x)` menemukan nilai x dalam vektor v . Fungsi ini juga dapat digunakan untuk vektor x .

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

Kesalahan ini sangat halus sehingga kita hanya bisa melihatnya setelah melakukan iterasi yang sangat banyak.

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



Berikut ini adalah fungsi sederhana untuk memeriksa distribusi seragam dari nilai 1... K dalam v. Kami menerima hasilnya, jika untuk semua frekuensi

lateks: $|\frac{f_i}{K} - \frac{1}{K}| < \frac{\delta}{\sqrt{n}}$.

```
>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 `binomialsum()`, yang mengembalikan probabilitas i atau kurang dari n percobaan.

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

0.751401349654

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

Arti dari interval ini adalah jika p berada di luar interval, hasil yang diamati sebesar 410 dalam 1000 jarang terjadi.

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

[0.37932, 0.441212]

The following commands are the direct way to get the above result. But for large n , the direct summation is not accurate and slow.

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

0.751401349655

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

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

```
409.932733047
```

Dalam Bridge, kami mengasumsikan 5 kartu yang beredar (dari 52 kartu) di dua tangan (26 kartu). Mari kita hitung 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

Merencanakan Data

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

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

Untuk pesta, kami menggunakan serangkaian nama.

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

Mari kita cetak persentasenya dengan baik.

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

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

Kemudian kita mencetak statistik dalam bentuk tabel. Kita menggunakan nama sebagai judul kolom, dan tahun sebagai judul baris. Lebar default untuk kolom adalah $wc = 10$, tetapi kami lebih suka output yang lebih padat. Kolom-kolom akan diperluas untuk label-label kolom, jika perlu.

```
>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

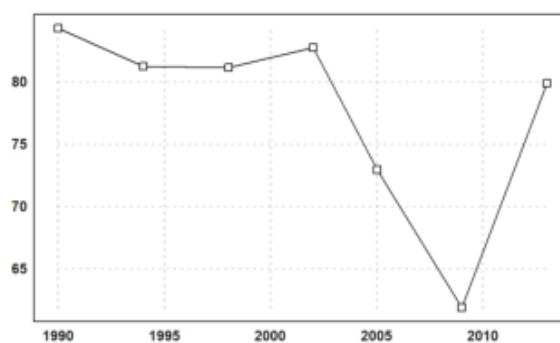
Perkalian matriks berikut ini mengekstrak jumlah persentase dua partai besar yang menunjukkan bahwa partai-partai kecil telah memperoleh suara di parlemen hingga tahun 2009.

```
>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. Alternatif lainnya adalah memanggil `plot2d` dua kali dengan `>add`.

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

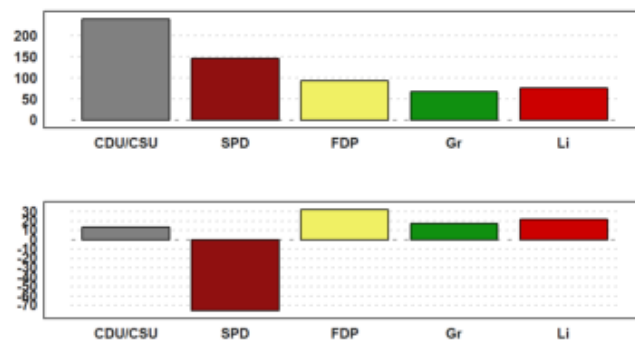


Tentukan beberapa warna untuk masing-masing pihak.

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

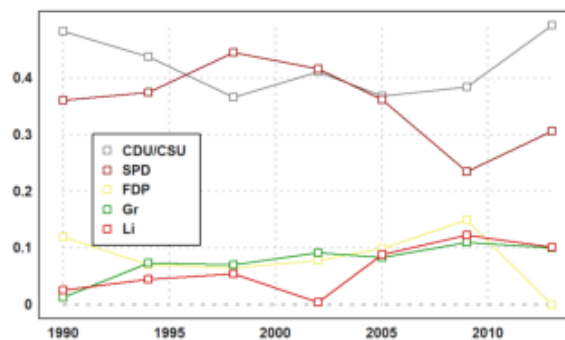

Sekarang kita dapat memplot hasil pemilu 2009 dan perubahannya ke dalam satu plot menggunakan gambar. Kita dapat menambahkan vektor kolom pada setiap plot.

```
>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):
```



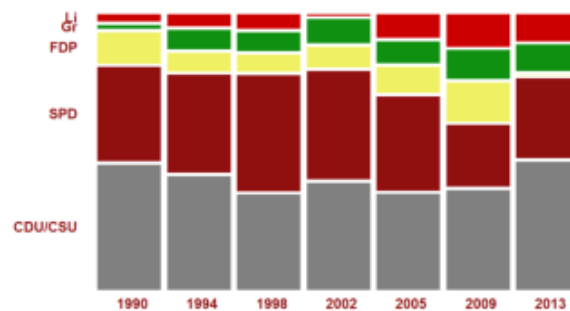
Plot kolom 3D menunjukkan deretan data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. angle adalah sudut pandang.

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



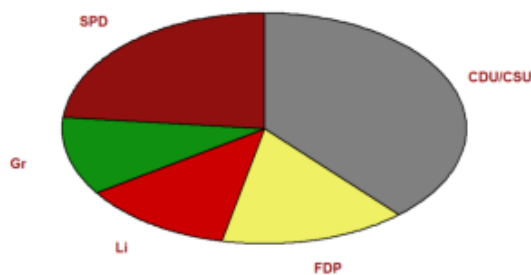
Representasi lainnya adalah plot mosaik. Perhatikan bahwa kolom-kolom pada plot mewakili kolom-kolom pada matriks di sini. Karena panjangnya label CDU/CSU, kita mengambil jendela yang lebih kecil dari biasanya.

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



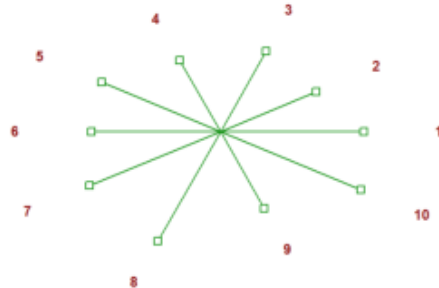
Kita juga bisa membuat diagram lingkaran. Karena warna hitam dan kuning membentuk sebuah koalisi, kita menyusun ulang elemennya.

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



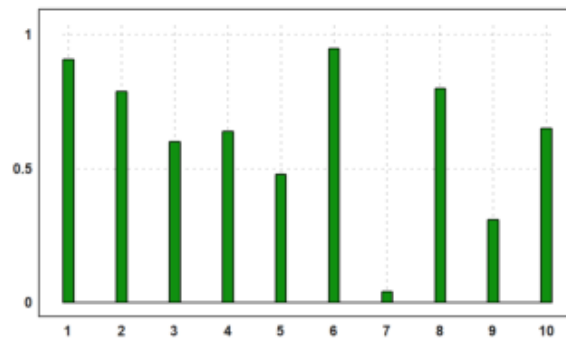
Berikut ini jenis plot yang lain.

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



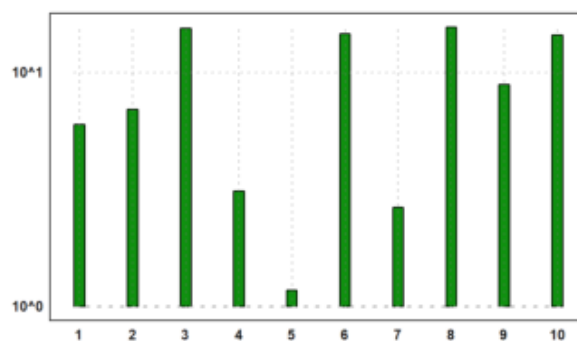
Beberapa plot di plot2d bagus untuk statika. Berikut ini adalah plot impuls dari data acak, yang terdistribusi secara seragam dalam [0,1].

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



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

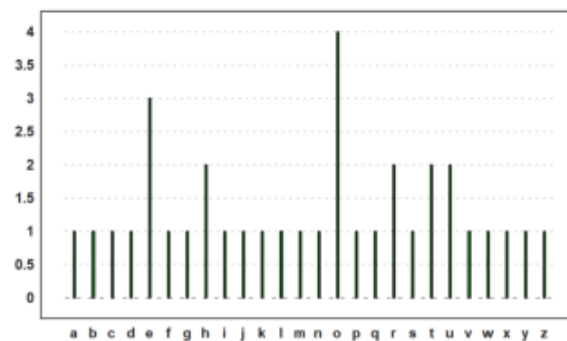
```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi `columnplot()` lebih mudah digunakan, karena hanya membutuhkan sebuah vektor nilai. Selain itu, fungsi ini dapat mengatur labelnya menjadi apa pun yang kita inginkan, kita telah mendemonstrasikan hal ini dalam tutorial ini.

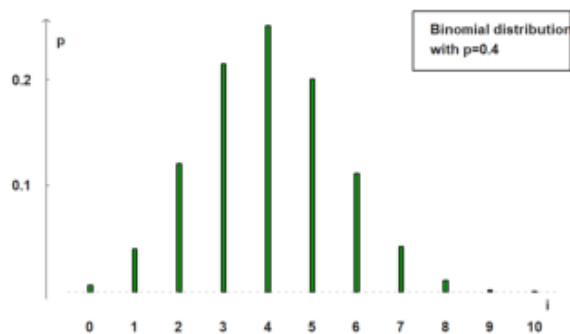
Berikut ini adalah aplikasi lain, di mana kita menghitung karakter dalam sebuah kalimat dan memplot statistik.

```
>v=strochar("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):
```



Anda juga dapat menetapkan sumbu secara manual.

```
>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 ini adalah cara untuk memplot frekuensi angka dalam vektor.

Kami membuat vektor angka acak bilangan bulat 1 hingga 6.

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

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

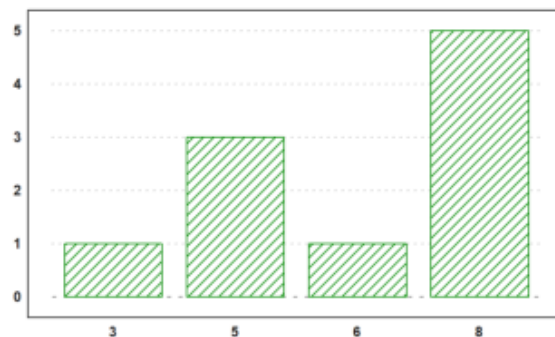
Kemudian ekstrak nomor unik dalam v.

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

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

Dan memplot frekuensi dalam plot kolom.

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



Kami ingin mendemonstrasikan fungsi untuk distribusi nilai empiris.

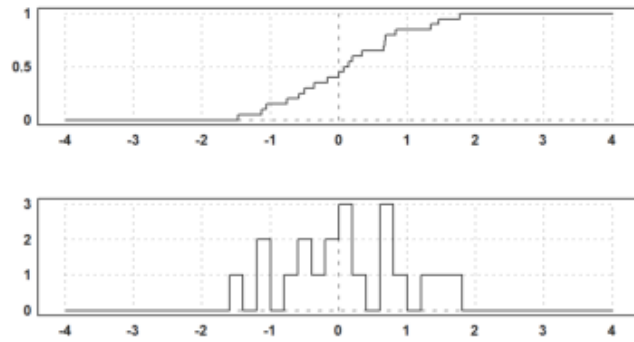
```
>x=normal(1,20);
```

Fungsi `empdist(x,vs)` membutuhkan larik nilai yang diurutkan. Jadi kita harus mengurutkan x sebelum dapat menggunakannya.

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

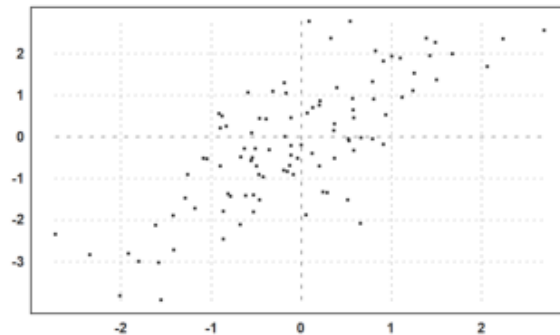
Kemudian kami memplot distribusi empiris dan beberapa batang kepadatan ke dalam satu plot. Alih-alih plot batang untuk distribusi, kali ini kami menggunakan plot gigi gergaji.

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



Plot sebar mudah dilakukan di Euler dengan plot titik biasa. Grafik berikut ini menunjukkan bahwa X dan X+Y berkorelasi positif secara jelas.

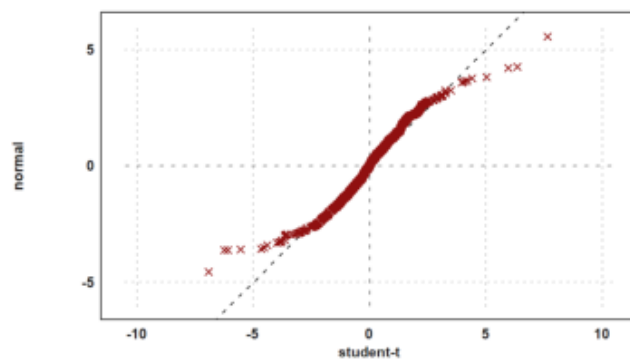
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style="..") :
```



Sering kali, kita ingin membandingkan dua sampel dari distribusi yang berbeda. Hal ini dapat dilakukan dengan plot kuantil-kuantil.

Untuk pengujian, kami 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) :
```



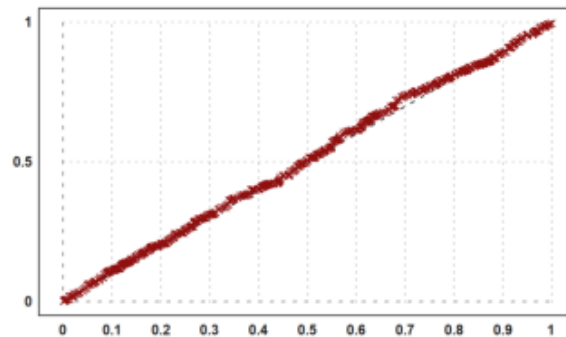
Plot ini dengan jelas menunjukkan bahwa nilai yang terdistribusi normal cenderung lebih kecil pada ujung yang ekstrim.

Jika kita memiliki dua distribusi dengan ukuran yang berbeda, kita dapat memperluas distribusi yang lebih kecil atau memperkecil distribusi yang lebih besar. Fungsi berikut ini bagus untuk keduanya. 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 linier dapat dilakukan dengan fungsi polyfit() atau berbagai fungsi kecocokan. Sebagai permulaan, kita 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

Kami ingin membandingkan kecocokan tanpa bobot dan dengan bobot. Pertama, koefisien dari kecocokan linier.

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

```
[0.733333, 0.812121]
```

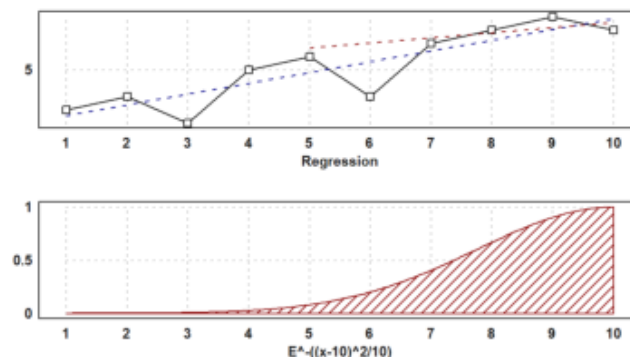
Sekarang, koefisien dengan bobot yang menekankan nilai terakhir.

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

```
[4.71566, 0.38319]
```

Kami menempatkan semuanya ke 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 survei tentang siswa, usia mereka, usia orang tua mereka, dan jumlah saudara kandung dari sebuah file.

Tabel ini berisi "m" dan "f" di kolom kedua. Kita menggunakan variabel tok2 untuk mengatur terjemahan yang tepat dan bukannya membiarkan readtable() mengumpulkan terjemahan.

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

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

Bagaimana usia saling bergantung satu sama lain? Kesan pertama datang dari scatterplot berpasangan.


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

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

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

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

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

Ini jelas merupakan model yang salah. Garis regresinya adalah $s = 17 + 0,74t$, di mana t adalah usia ibu dan s adalah usia ayah. Perbedaan usia mungkin sedikit bergantung pada usia, tetapi tidak terlalu banyak.

Sebaliknya, kita menduga fungsi seperti $s = a + t$. Maka a adalah rata-rata dari $s-t$. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

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

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

Mari kita plotkan ini ke dalam satu scatter plot.

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

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

Berikut ini adalah plot kotak dari kedua usia tersebut. Ini hanya menunjukkan, bahwa usia keduanya berbeda.

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

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

Sangat menarik bahwa perbedaan dalam median tidak sebesar perbedaan dalam mean.

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

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

Koefisien korelasi menunjukkan korelasi positif.

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

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

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

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

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

Membuat Fungsi baru

Tentu saja, bahasa EMT dapat digunakan untuk memprogram fungsi baru. Misalnya, kita mendefinisikan fungsi kemiringan.

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

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 singkat dan efisien. Mari kita coba fungsi ini.

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

```
-0.198710316203
```

Berikut ini adalah fungsi lain, yang disebut koefisien kemencengan Pearson.

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

-0.0801873249135

Simulasi Monte Carlo

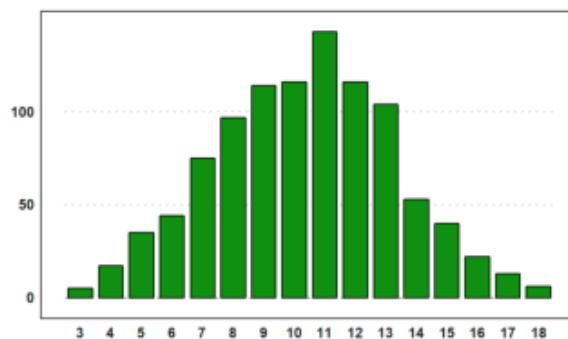
Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Berikut ini adalah contoh lainnya, yang mensimulasikan 1000 kali pelemparan 3 dadu, dan menanyakan distribusi dari jumlah tersebut.

```
>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]
```

Kita bisa merencanakan ini sekarang.

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



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

Fungsi berikut ini menghitung jumlah cara angka k dapat direpresentasikan sebagai jumlah 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 ini adalah hasil dari tiga lemparan dadu.

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

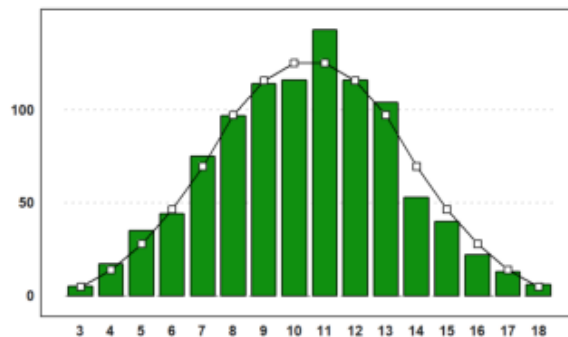
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,  
255, 185, 121, 70, 35, 15, 5, 1]
```

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

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

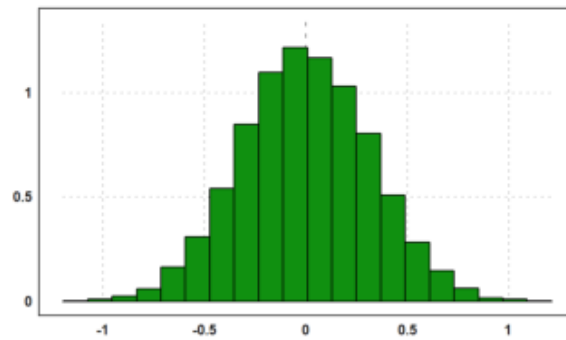
```
0.316227766017
```

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

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

```
0.319493614817
```

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



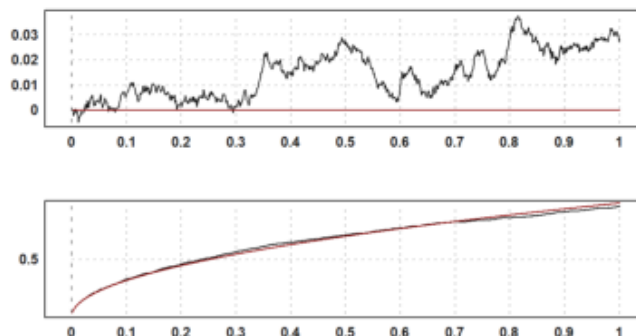
Median dari 10 bilangan acak berdistribusi normal 0-1 memiliki deviasi yang lebih besar.

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

0.374460271535

Karena kita dapat dengan mudah menghasilkan jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot deviasi standar dan rata-rata dari langkah ke-n dari proses-proses ini bersama dengan 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):
```



Tes

Tes adalah alat yang penting dalam statistik. Dalam Euler, banyak tes yang diterapkan. Semua tes ini mengembalikan kesalahan yang kita terima jika kita menolak hipotesis nol.

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

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

0.498830517952

Uji chi-square 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 bisa menolak distribusi seragam. Ini tidak membuktikan bahwa dadu kita adil. Namun kita tidak bisa menolak hipotesis kita.

Selanjutnya kita buat 1000 lemparan dadu dengan menggunakan generator bilangan acak, dan lakukan pengujian yang sama.

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

0.528028118442

Mari kita uji 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 untuk diuji.

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

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

0.38722000942

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

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

5.60009101758e-07

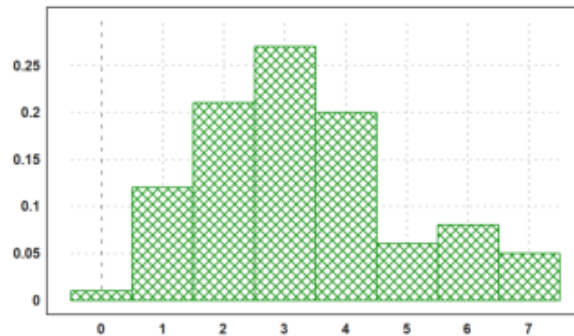
Pada contoh berikut, kami menghasilkan 20 lemparan dadu acak sebanyak 100 kali dan menghitung jumlah dadu yang muncul. Rata-rata harus ada $20/6 = 3,3$ mata dadu.

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

3.28

Sekarang kita bandingkan jumlah tersebut dengan distribusi binomial. Pertama, kita memplot distribusi angka 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-square menolak hipotesis bahwa distribusi kita adalah distribusi binomial, jika hasilnya $<0,05$.

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

0.53921579764

Contoh berikut ini berisi hasil dari dua kelompok orang (laki-laki dan perempuan, katakanlah) yang memberikan suara untuk 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 χ^2 melakukan hal ini. Hasilnya terlalu besar untuk menolak independensi. Jadi kita tidak dapat mengatakan, jika pemungutan suara tergantung pada jenis kelamin dari data ini.

```
>tabletest (A)
```

```
0.990701632326
```

Berikut ini adalah tabel yang diharapkan, jika kita mengasumsikan frekuensi pemungutan suara 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 terkoreksi. Karena koefisien ini sangat dekat dengan 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
>contingency (A)
```

```
0.0427225484717
```

Beberapa Tes Lainnya

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

```
>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, kami menolak hipotesis nilai rata-rata yang sama. Kami melakukan ini 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 gabungan.


```
>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. Uji 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

Sekarang mari kita coba mensimulasikan dua perawatan a dan b yang diterapkan pada 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 daripada b.

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

0.0546875

Ini adalah kesalahan yang terlalu besar. 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 pengujian lagi dengan menggunakan rangkaian yang dihasilkan.

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

0.0068706451766

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

0.275145971064

Berikut ini adalah tes untuk generator bilangan acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu mengharapka adanya masalah.

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

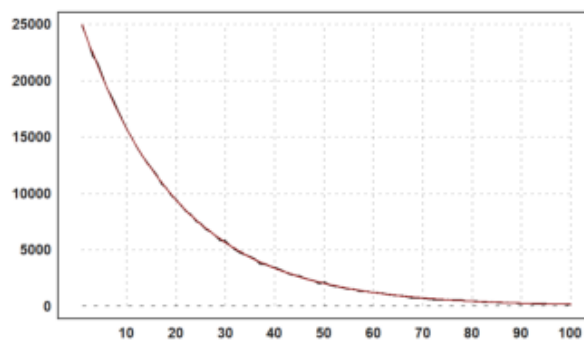
```
>n:=10000000; r:=random(1,n);
```

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

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

Terakhir, kami memplot berapa kali, setiap jarak yang 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):
```



Menghapus data.

```
>remvalue n;
```

Pengantar untuk Pengguna Proyek R

Jelas, EMT tidak bersaing dengan R sebagai sebuah paket statistik. Namun, ada banyak prosedur dan fungsi statistik yang tersedia di EMT juga. Jadi EMT dapat memenuhi kebutuhan dasar. Bagaimanapun, EMT hadir dengan paket numerik dan sistem aljabar komputer.

Buku ini diperuntukkan bagi Anda yang sudah terbiasa dengan R, tetapi perlu mengetahui perbedaan sintaks EMT dan R. Kami mencoba memberikan gambaran umum mengenai hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami juga membahas cara-cara untuk bertukar data di antara kedua sistem tersebut.

*** Translated with www.DeepL.com/Translator (free version) ***

Perhatikan bahwa ini adalah pekerjaan yang sedang berlangsung.

Sintaks Dasar

Hal pertama yang Anda pelajari dalam R adalah membuat sebuah vektor. Dalam EMT, perbedaan utamanya adalah bahwa operator `:` dapat mengambil ukuran langkah. Selain itu, operator ini memiliki daya ikat 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. Anda dapat menggunakan vektor untuk menggabungkan beberapa hal.

Contoh berikut ini, seperti banyak contoh lainnya, berasal dari "Interduksi ke R" yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti alurnya 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 ukuran langkah EMT digantikan oleh fungsi `seq()` dalam R. Kita dapat 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()` dari R tidak ada dalam EMT. Untuk input vektor, dapat dituliskan 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 dalam EMT.

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

```
77.6713990297 miles
```

Operator `"<-"` untuk penugasan juga menyematkan, dan bukan ide yang baik untuk R. Berikut ini akan membandingkan `a` dan `-4` dalam EMT.

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

```
0
```

Dalam R, "a<-4<3" bisa digunakan, tetapi "a<-4<-3" tidak. Saya juga mengalami ambiguitas yang sama di EMT, tetapi saya mencoba untuk menghilangkannya.

EMT dan R memiliki vektor dengan tipe boolean. Tetapi dalam EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Dalam R, nilai benar dan salah tetap dapat digunakan dalam aritmatika biasa seperti dalam EMT.

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

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

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

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

```
NAN
1
```

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

Dalam R ada paket-paket untuk Unicode. Dalam EMT, sebuah string dapat berupa string Unicode. Sebuah string Unicode dapat diterjemahkan ke pengkodean lokal dan sebaliknya. Selain itu, u"... " dapat berisi entitas HTML.

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

```
© René Grothmann
```

Berikut ini mungkin atau mungkin tidak ditampilkan dengan benar pada sistem Anda sebagai A dengan titik dan tanda hubung di atasnya. Hal ini tergantung pada jenis huruf yang Anda gunakan.

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

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

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

```
pi = 3.14159265359
```

Pengindeksan

Sebagian besar waktu, ini akan bekerja seperti di R.

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

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

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

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

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

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

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

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

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

Sama seperti 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 dimungkinkan dalam EMT. Untuk paket statistik, hal ini mungkin diperlukan untuk memudahkan 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]
```

Tipe Data

EMT memiliki lebih banyak tipe data yang tetap dibandingkan R. Jelas, dalam R terdapat vektor yang berkembang. Anda bisa mengatur sebuah vektor numerik kosong `v` dan memberikan sebuah nilai pada elemen `v[17]`. Hal ini tidak mungkin dilakukan dalam EMT.

Hal berikut ini sedikit tidak efisien.

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

EMT sekarang akan membuat vektor dengan v dan i yang ditambahkan pada tumpukan dan menyalin vektor tersebut kembali ke variabel global v.

Semakin efisien mendefinisikan vektor.

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

Untuk mengubah jenis tanggal di EMT, Anda dapat menggunakan fungsi seperti `complex()`.

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

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

Konversi ke string hanya dapat dilakukan untuk tipe data dasar. Format saat ini digunakan untuk penggabungan string sederhana. Tetapi ada fungsi-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 terdapat sebuah 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 mencari indeks dari wilayah-wilayah di dalam daftar unik dari wilayah-wilayah tersebut.

```
>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 kita dapat mencetak rata-rata pendapatan negara di wilayah dengan cara yang ramah.

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

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

Kita juga dapat mencoba meniru perilaku R sepenuhnya.

Faktor-faktor tersebut harus disimpan dengan jelas dalam sebuah koleksi dengan jenis dan kategorinya (negara bagian dan wilayah 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()` dengan cara berikut. Ini akan mengembalikan sebuah 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 bisa dicetak sebagai sebuah 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

Larik

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau sebuah pustaka C untuk ini.

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

Dalam EMT, sebuah vektor adalah sebuah matriks dengan satu baris. Ini bisa dibuat menjadi sebuah matriks dengan `redim()`.

*** Translated with www.DeepL.com/Translator (free version) ***

```
>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, sama seperti di R.

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

2	3
7	8
12	13
17	18

Namun, dalam R dimungkinkan untuk menetapkan daftar indeks tertentu dari vektor ke suatu nilai. Hal yang sama juga dapat dilakukan dalam EMT hanya dengan sebuah 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 dalam EMT hanya dapat dilakukan di antara vektor. Hal ini dilakukan secara 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

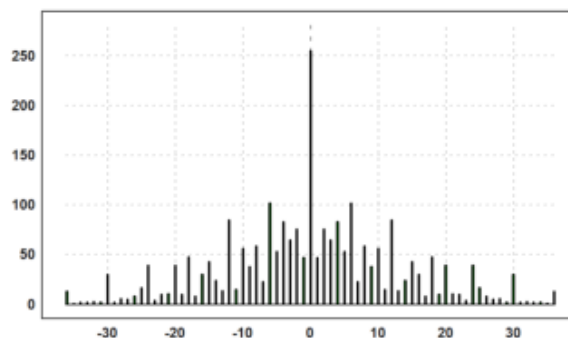
Dalam pengantar PDF untuk R ada sebuah contoh, yang menghitung distribusi $ab-cd$ untuk a, b, c, d yang dipilih dari 0 sampai n secara acak. Solusinya dalam R adalah membentuk sebuah matriks 4 dimensi dan menjalankan `table()` di atasnya.

Tentu saja, ini bisa dicapai dengan sebuah perulangan. Tetapi perulangan tidak efektif dalam EMT atau R. Dalam EMT, kita bisa menulis perulangan dalam C dan itu adalah solusi tercepat.

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

*** Translated with www.DeepL.com/Translator (free version) ***

```
>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"):
```



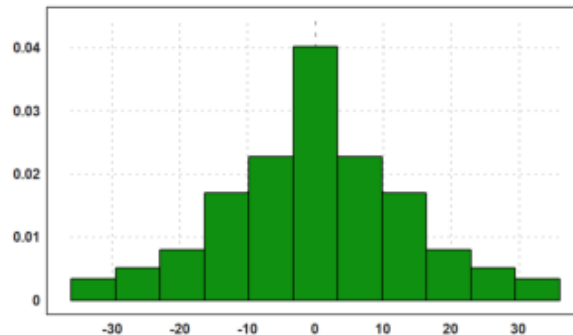
Selain kelipatan yang tepat, EMT dapat menghitung frekuensi dalam vektor.

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

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

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

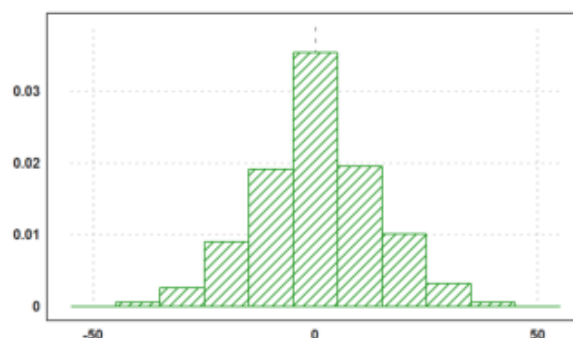
```
>plot2d(q,distribution=11):
```



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

Karena fungsi `histo()` mengembalikan frekuensi, kita perlu menskalakannya 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="/"):
```



Daftar

EMT memiliki dua jenis daftar. Yang pertama adalah daftar global yang dapat diubah, dan yang kedua adalah jenis daftar yang tidak dapat diubah. Kita tidak peduli dengan daftar global di sini.

Tipe daftar yang tidak dapat diubah disebut koleksi dalam EMT. Ia berperilaku seperti struktur dalam C, tetapi elemen-elemennya hanya diberi nomor dan tidak diberi nama.

```
>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 sering ingin mengimpor matriks data dari sumber lain ke EMT. Tutorial ini akan menjelaskan kepada Anda tentang berbagai cara untuk melakukan hal tersebut. Fungsi yang sederhana adalah `writematrix()` dan `readmatrix()`.

Mari kita tunjukkan bagaimana cara membaca dan menulis sebuah vektor real ke sebuah file.

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

```
0.49815  
0.28037
```

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

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

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

Sekarang kita menulis vektor kolom `a'` ke 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 bahasa 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 koma). File "test.csv" berikut ini akan muncul di folder current Anda.

```
>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 tajuk token.

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

	f	m
1	6	7
2	5	2

Untuk statika, 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 bentuknya yang paling sederhana dapat membaca ini dan mengembalikan kumpulan nilai dan baris judul.

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

Koleksi ini dapat dicetak dengan `writetable()` ke buku catatan, 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 menulis matriks ke dalam sebuah file. Untuk keluarannya, kami membuat file di direktori kerja saat ini.

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

Berikut ini 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 pada sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal. Namun, 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 ditulis dengan parameter "w". Standarnya adalah "r" untuk membaca.

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

Matriks dipisahkan oleh garis 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 "format lain", 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 desimal. Anda dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak perlu untuk membaca matriks ke dalam EMT.

Cara termudah untuk membaca ini ke dalam Euler adalah `readmatrix()`. Semua koma digantikan oleh titik-titik dengan parameter `>comma`. Untuk CSV bahasa Inggris, hilangkan saja parameter ini.

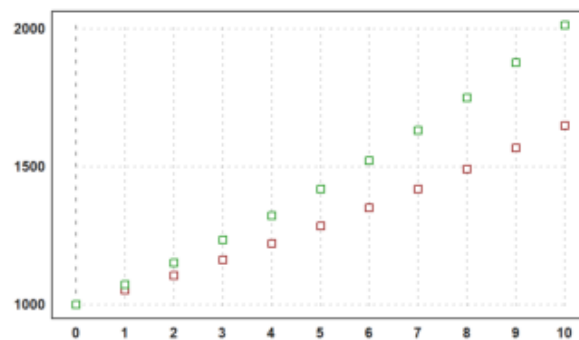
*** Translated with www.DeepL.com/Translator (free version) ***

```
>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

Let us plot this.

```
>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 adalah contoh untuk hal ini. Fungsi ini akan berhenti pada akhir file atau baris kosong.

*** Translated with www.DeepL.com/Translator (free version) ***

```
>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 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 garis dengan tangan. Misalkan, kita memiliki baris dengan format berikut.

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

- tanda kurung tutup >,
- setiap string yang tidak mengandung tanda kurung dengan

sub-pencocokan "(...)",

- kurung pembuka dan kurung penutup menggunakan solusi terpendek,
- sekali lagi, semua string yang tidak mengandung tanda kurung,
- dan sebuah kurung pembuka <.

Ekspresi reguler agak sulit untuk dipelajari tetapi sangat kuat.
*** Translated with www.DeepL.com/Translator (free version) ***

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

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

```
>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, kami membaca versi saat ini dari situs EMT. Kami menggunakan ekspresi reguler untuk mendandai "Versi ..." dalam judul.

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

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

```
>readversion
```

```
Version 2024-01-12
```

Input dan Output 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
```

To remove any files use `fileremove()`.

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

Vektor baris dalam file tidak memerlukan koma, jika setiap angka berada di 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]
```

Tambahan Materi

Subtopik 1

Menyimpan data dalam bentuk matriks

Pada subtopik 1 ini menjelaskan tentang menyimpan data dalam bentuk matriks untuk dianalisis lebih lanjut. Untuk itu akan dibahas :

- Penjelasan umum mengenai matriks dalam statistika
- Penggunaan operasi matriks pada persoalan statistika
- Penghitungan lanjutan dengan matriks

Array

Dalam statistika, matriks adalah array (daftar) bilangan yang terdiri dari baris-baris dan kolom-kolom. Aljabar matriks adalah aljabar khusus untuk array tersebut. Setiap array diperlakukan sebagai satu entitas yang membuatnya sangat berguna dalam menganalisa data.

EMT hanya memiliki dua dimensi untuk array. Tipe datanya disebut matriks. Akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau library C untuk hal ini.

Dalam EMT, sebuah vektor adalah sebuah matriks dengan satu baris.

Ini bisa dibuat menjadi sebuah 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

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

2	3
7	8
12	13
17	18

Namun, dalam R dimungkinkan untuk mengatur daftar indeks tertentu dari vektor ke suatu nilai. Hal yang sama juga dapat dilakukan dalam EMT hanya dengan sebuah perulangan.

Contoh Soal 1 :

Apabila diketahui matriks Gross Income beberapa Negara (US, Canada, Australia, UK) tahun 1981 dan 1982 adalah sebagai berikut.

```
>A=[27,15,18,21;32,14,21,30]
```

27	15	18	21
32	14	21	30

```
>B=[19,9,11,17;22,10,13,24]
```

19	9	11	17
22	10	13	24

Berapakah Gross Profit dalam tahun 1981 dan 1982 untuk keempat negara

=> untuk menghitung Gross Profit adalah mengurangkan matriks Gross Income dengan matriks pengeluaran. Maka di dapat matriks Gross Profit sebagai berikut :

```
>A-B
```

8	6	7	4
10	4	8	6

Contoh Soal 2 :

Perhitungan jumlah uang yang diperlukan dalam masalah pembelian tikus, katak, dan kelinci untuk percobaan di departemen biologi dapat menggunakan perkalian matriks.

Misal harga hewan berturut-turut 3, 1, 10 ribu rupiah. Banyak hewan yang diperlukan berturut-turut 50, 100, dan 30 ekor. Berapa jumlah uang yang diperlukan.

```
>a=[3;1;10]
```

```
3
1
10
```

```
>b=[50;100;30]
```

```
50
100
30
```

```
>a*b
```

```
150
100
300
```

maka uang yang diperlukan adalah $150+100+300 = 550$ ribu rupiah.

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

```
endfunction
```

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

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

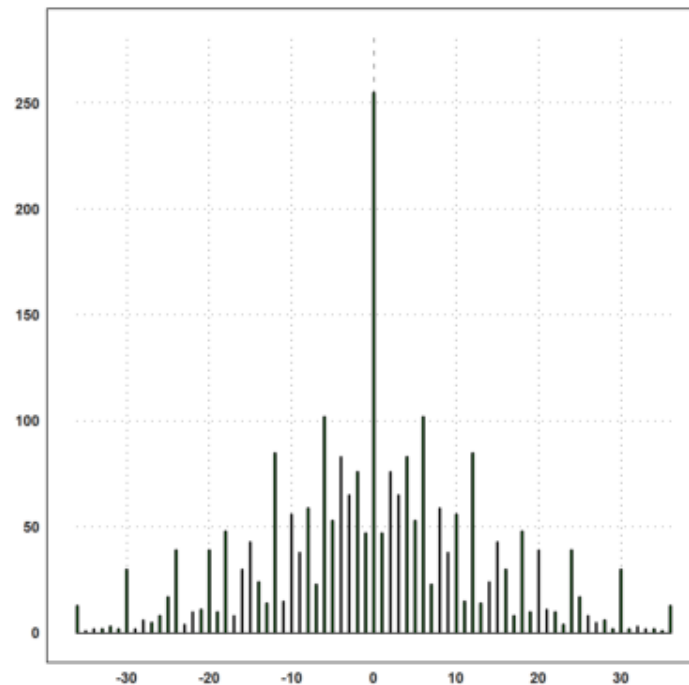
```
1      2      3      4      5
6      7      8      9     10
11     12     13     14     15
16     17     18     19     20
```

Hasil kali luar dalam 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
```

```
>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"):
```



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

Tentu saja, hal ini bisa dicapai dengan perulangan. Tetapi perulangan tidak efektif dalam EMT. Dalam EMT, kita dapat menulis perulangan dalam C dan itu akan menjadi solusi tercepat.

Tetapi kita ingin meniru perilaku R. Untuk itu, kita perlu meratakan perkalian ab dan membuat matriks $ab-cd$.

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

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

Subtopik 2

Data acak dengan fungsi distribusi

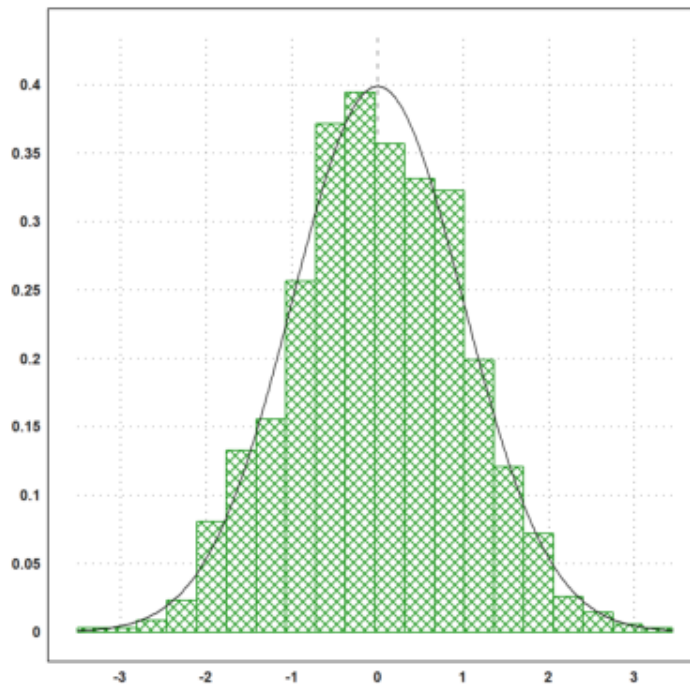
Pada subtopik 2 ini menjelaskan tentang menghasilkan data acak (simulasi) dengan menggunakan fungsi distribusi tertentu

Untuk itu akan dibahas :

- memplot data distribusi eksperimen acak
- distribusi normal
- distribusi diskrit

Dengan `plot2d`, terdapat metode yang sangat mudah untuk memplot sebaran data eksperimen.

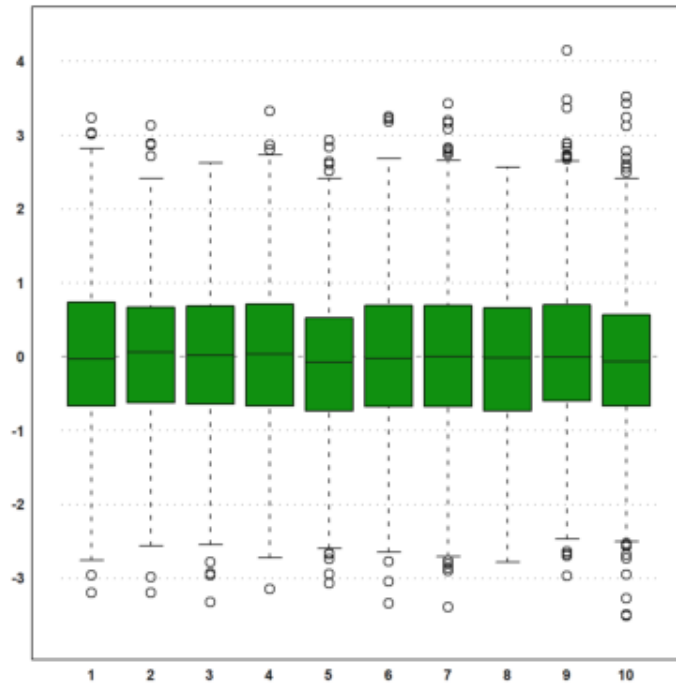
```
>p=normal(1,1000); //1000 sampel acak berdistribusi normal p
>plot2d(p,distribution=20,style="\"); // plot sampel acak p
>plot2d("qnormal(x,0,1)",add=1): // menambahkan plot distribusi normal standar
```



Perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi sesungguhnya). Masukkan kembali ketiga perintah tersebut untuk melihat hasil pengambilan sampel yang lain.

Berikut ini adalah perbandingan 10 simulasi dari 1000 nilai terdistribusi normal dengan menggunakan apa yang disebut plot kotak. Plot ini menunjukkan median, kuartil 25% dan 75%, nilai minimal dan maksimal, serta pencilan (the outliers).

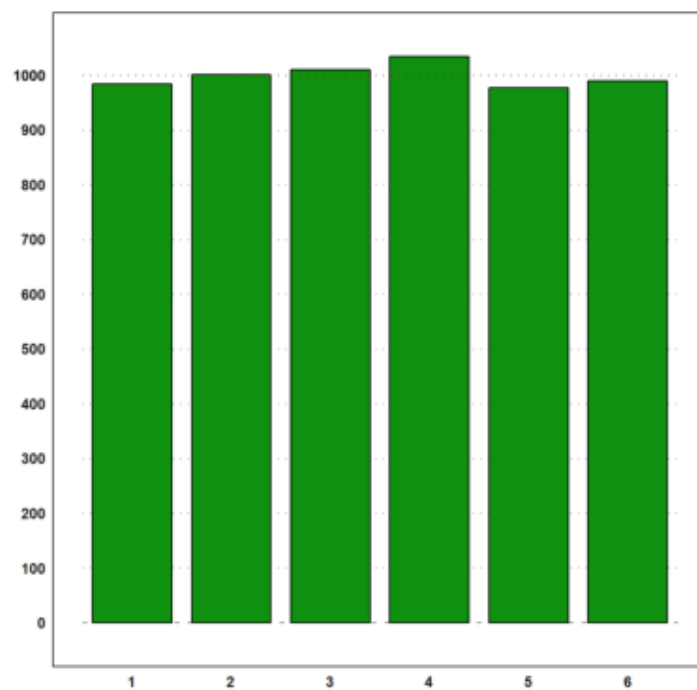
```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan bilangan bulat acak, Euler memiliki `intrandom`. Misal kita simulasikan pelemparan dadu dan plot distribusinya.

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

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



Meskipun `inrandom(n,m,k)` mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 sampai k, adalah mungkin untuk menggunakan distribusi bilangan bulat yang lain dengan `randpint()`. Pada contoh berikut, probabilitas untuk 1,2,3 masing-masing adalah 0.4,0.1,0.5.

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

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

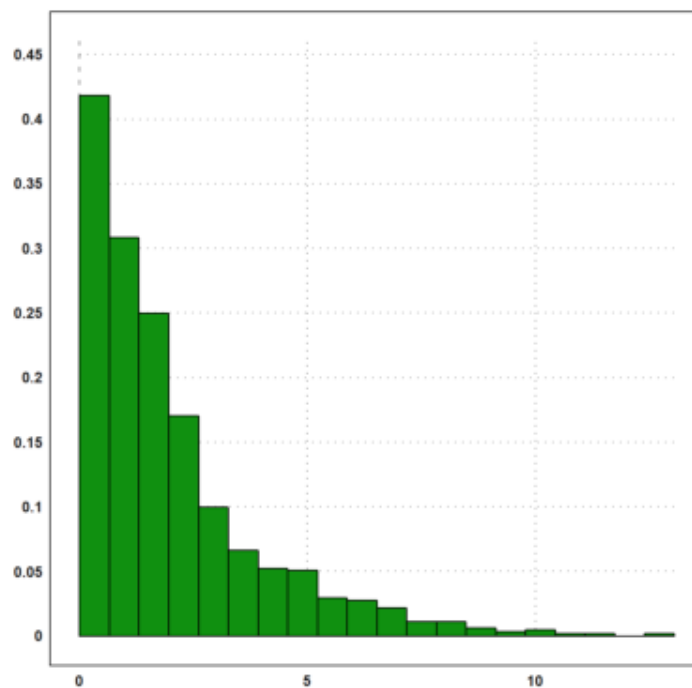
Sebagai contoh, kita mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

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

dengan parameter

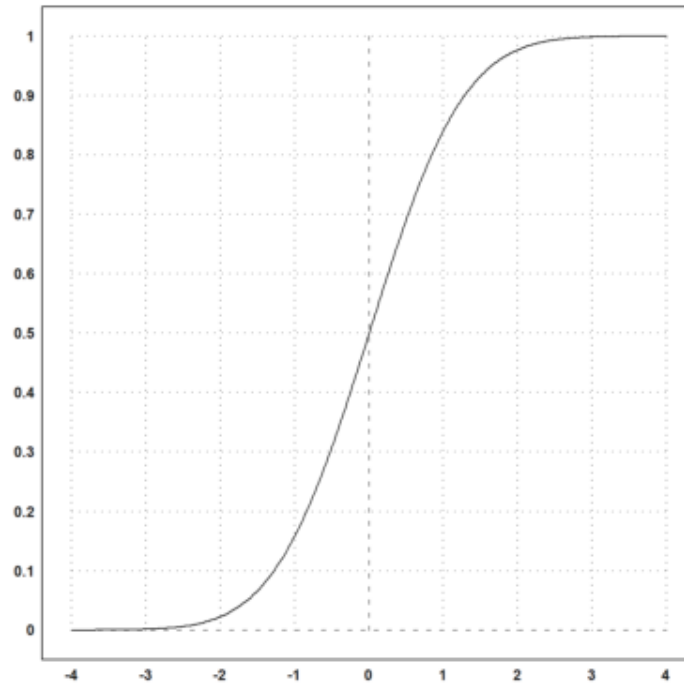
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ adalah rata-rata, dan dilambangkan dengan } X \sim \text{Eksponensial}(\lambda).$$

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



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

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



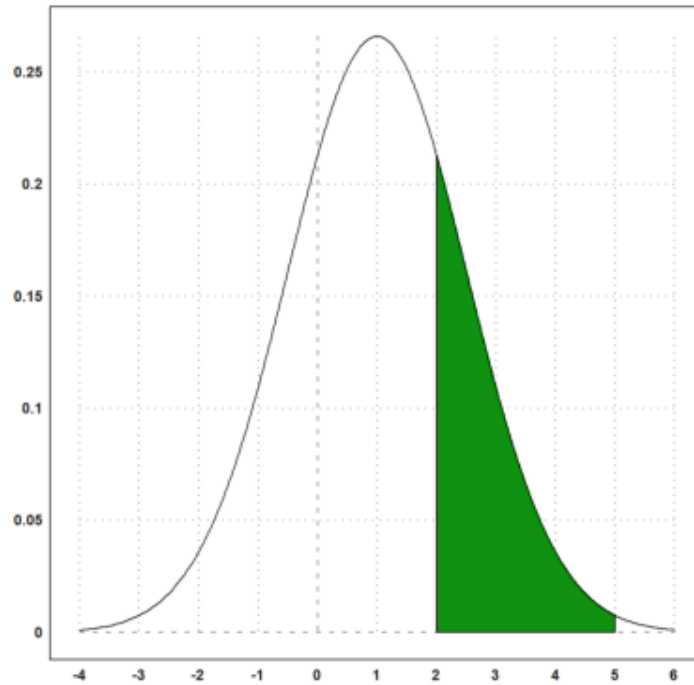
Distribusi Normal

Distribusi normal adalah jenis distribusi probabilitas yang sering digunakan dalam statistik dan ilmu data untuk menggambarkan data yang terpusat di sekitar nilai tengah. Distribusi ini sering dikenal sebagai distribusi Gaussian dan memiliki bentuk simetris berbentuk lonceng yang disebut kurva lonceng (bell curve).

EMT dapat menghitung banyak distribusi dan inversnya, terutama distribusi normal.

Berikut ini adalah salah satu cara untuk memplot kuantil.

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



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

Probabilitas untuk berada di area hijau adalah sebagai berikut.

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

0.24866

Hal ini dapat dihitung secara numerik dengan integral berikut ini.

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

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

0.24866

CONTOH :

Pada 1000 lemparan koin, jumlah gambar yang diharapkan terdistribusi dengan nilai rata-rata 500 dan deviasi standar

$$\sigma = \sqrt{1000 \times 0.5 \times 0.5}$$

Hitunglah probabilitas untuk mendapatkan lebih dari 520 muncul gambar, dan ketika probabilitasnya kurang dari 0,1% approximating distribusi binomial dengan distribusi normal.

```
>n=1000; p=0.5;...  
>m=n*p; s=sqrt(n*p*(1-p));...  
>1-normaldis(520,m,s)
```

0.10295

```
>ceil(invnormaldis(99.9%,m,s))
```

549

Perhatikan bahwa fungsi normaldis dalam skala EMT dengan cara yang berbeda dari fungsi erf, yang juga tersedia.

Semua distribusi dalam EMT diimplementasikan sebagai fungsi distribusi, dari 0 hingga 1.

Perkiraan untuk distribusi binomial juga dapat dihitung

```
>1-bindis(520,1000,0.5)
```

0.097383

```
>invbindis(99.9%,1000,0.5)
```

548.35

Fungsi invbindis() menyelesaikan interpolasi linier antara nilai bilangan bulat.

Mari kita bandingkan distribusi binomial dengan distribusi normal mean dan deviasi yang sama. Fungsi invbindis() memecahkan interpolasi linier antara nilai bilangan bulat.

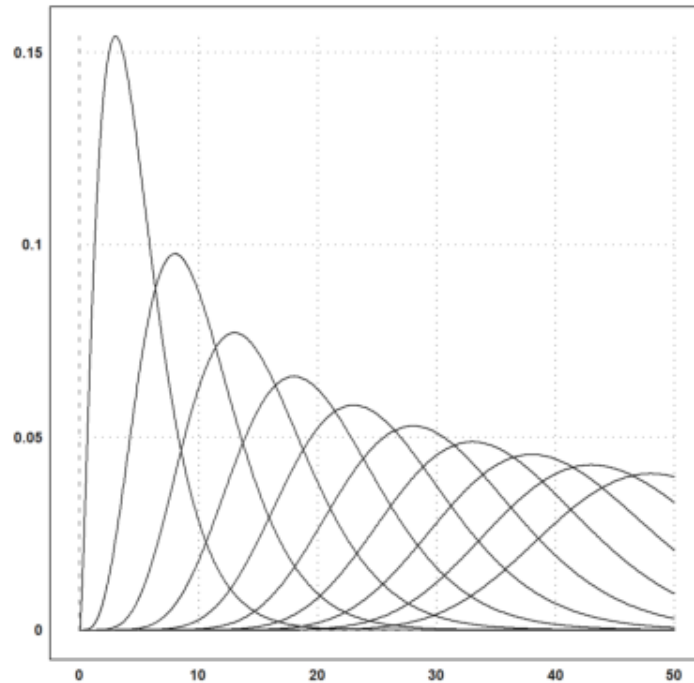
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

525.52

526.01

Fungsi qdis() adalah kepadatan distribusi chi-kuadrat. Seperti biasa, Euler memetakan vektor ke fungsi ini. Jadi kita mendapatkan plot dari semua distribusi chi-kuadrat dengan derajat 5 sampai 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.

Penamaan mencoba untuk konsisten. Misalnya.,

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

Pelengkap distribusi (ekor atas) adalah `chicdis()`.

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

```
0.52763
0.52763
```

Distribusi Diskrit

Distribusi diskrit adalah jenis distribusi probabilitas yang menggambarkan kemungkinan dari nilai-nilai tertentu dalam variabel acak diskrit. Variabel acak diskrit adalah variabel yang hanya bisa mengambil nilai tertentu, seperti bilangan bulat atau kategori, bukan nilai kontinu. Distribusi diskrit digunakan untuk situasi di mana hasilnya adalah bilangan terbatas atau terhitung, misalnya jumlah kemunculan dalam lemparan koin, jumlah kelahiran per keluarga, atau jumlah kendaraan yang melewati suatu jalan.

Untuk menentukan distribusi diskrit, dapat menggunakan metode berikut.

Pertama, tetapkan fungsi distribusinya terlebih dahulu.

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

```
[0, 0.165, 0.335, 0.5, 0.66667, 0.83333, 1]
```

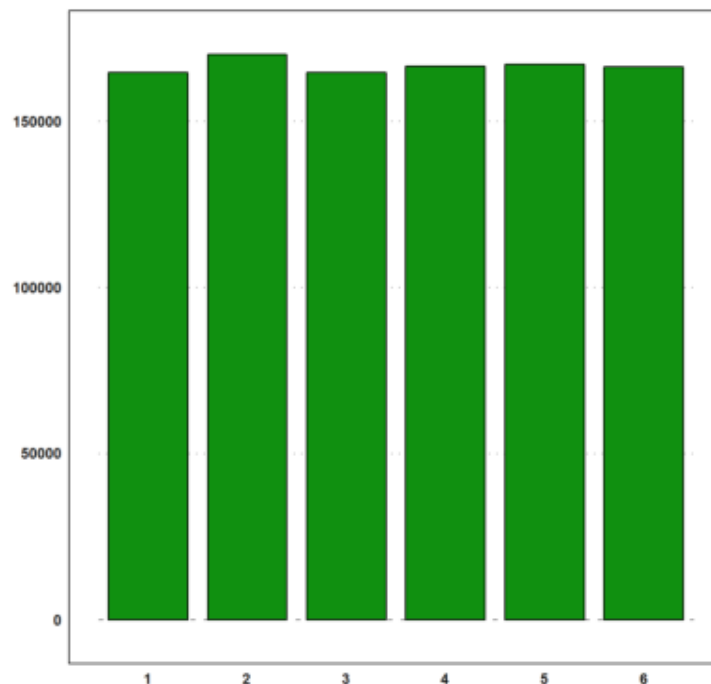
Artinya, dengan probabilitas $wd[i+1]-wd[i]$ kita menghasilkan nilai acak i .

Ini hampir merupakan distribusi yang seragam. Mari kita definisikan sebuah generator bilangan acak untuk ini. Fungsi $find(v,x)$ menemukan nilai x dalam vektor v . Fungsi ini juga dapat digunakan untuk vektor x .

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

Kesalahan ini sangat halus sehingga kita hanya bisa melihatnya setelah melakukan iterasi yang sangat banyak.

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



Berikut ini 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(intrandom(1,1000000,6))
```

1

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

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

0.7514

Invers Beta function digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p . Tingkat defaultnya adalah α .

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

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

[0.37932, 0.44121]

Perintah berikut ini adalah cara langsung untuk mendapatkan hasil di atas. Tetapi untuk n yang besar, penjumlahan langsung tidak akurat dan lambat.

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

0.7514

`invbinsum()` menghitung invers dari `binomialsum()`.

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

409.93

Dalam Bridge, kita mengasumsikan 5 kartu yang terbuka (dari 52 kartu) di dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (misalnya 0:5, 1:4, 4:1, atau 5:0).

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

0.32174

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

Input dan Output File (Membaca dan Menulis Data) Euler Math Toolbox

(EMT) dapat membaca data yang tersimpan di dalam berkas dengan berbagai format (teks biasa, CSV, dsb.) untuk melakukan pemrosesan atau analisis lebih lanjut. Berikut ini adalah beberapa metode umum:

1. Membaca Data dari File CSV

File CSV (Comma-Separated Values) adalah format file teks sederhana yang digunakan untuk menyimpan data dalam bentuk tabel (baris dan kolom). File CSV menggunakan koma sebagai pemisah (delimiter) untuk memisahkan nilai dalam kolom, dan setiap baris dalam file mewakili satu baris dalam tabel.

Karakteristik Utama CSV

1. Pemisah Kolom: Biasanya menggunakan koma untuk memisahkan nilai antar kolom. Namun, terkadang pemisah lain seperti titik koma (;) atau tab digunakan.
2. Baris Data: Setiap baris dalam file CSV mewakili satu baris dalam tabel. Baris diakhiri dengan baris baru.
3. Format Teks Sederhana: CSV adalah format teks biasa, sehingga mudah dibaca oleh manusia dan aplikasi pengolah data.
4. Kesesuaian Luas: Format CSV banyak digunakan dalam aplikasi spreadsheet seperti Microsoft Excel, Google Sheets, dan aplikasi database lainnya.

Kelebihan dan Kekurangan CSV

- Kelebihan: Mudah dibaca, digunakan secara luas, kompatibel dengan banyak perangkat lunak.
- Kekurangan: Tidak mendukung format data yang kompleks (seperti hierarki), dan tidak cocok untuk data besar yang memerlukan struktur yang lebih kompleks.

CSV adalah format yang populer karena kesederhanaannya, sehingga sering digunakan untuk mentransfer data antar aplikasi.

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

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

```
0.8908550127980653,0.5587551936014515,0.6488123555430445  
0.5630568786802863,0.8514487256170911,0.908283057251973  
0.7889076527474217,0.6002760729861754,0.6402181435462749
```

```
>readmatrix(file)
```

```
0.89086 0.55876 0.64881
0.56306 0.85145 0.90828
0.78891 0.60028 0.64022
```

Dimungkinkan untuk menulis beberapa matriks ke satu file. Perintah "open ()" dapat membuka file untuk ditulis dengan parameter "w".

Standarnya adalah "r" untuk membaca.

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

Matriks dipisahkan oleh garis kosong. Untuk membaca matriks, buka file dan panggil "readmatrix()" beberapa kali.

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

```
0.89086 0.56306 0.78891
0.55876 0.85145 0.60028
0.64881 0.90828 0.64022
```

Di Excel atau spreadsheet serupa, kita dapat mengekspor matriks sebagai CSV (nilai dipisahkan dengan koma). Di Excel 2007, gunakan "save as" dan "other format", lalu pilih "CSV". Pastikan tabel saat ini hanya berisi data yang ingin diekspor.

Berikut ini contohnya.

```
>reset;
>file="excel-data.csv"; ...
>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
```

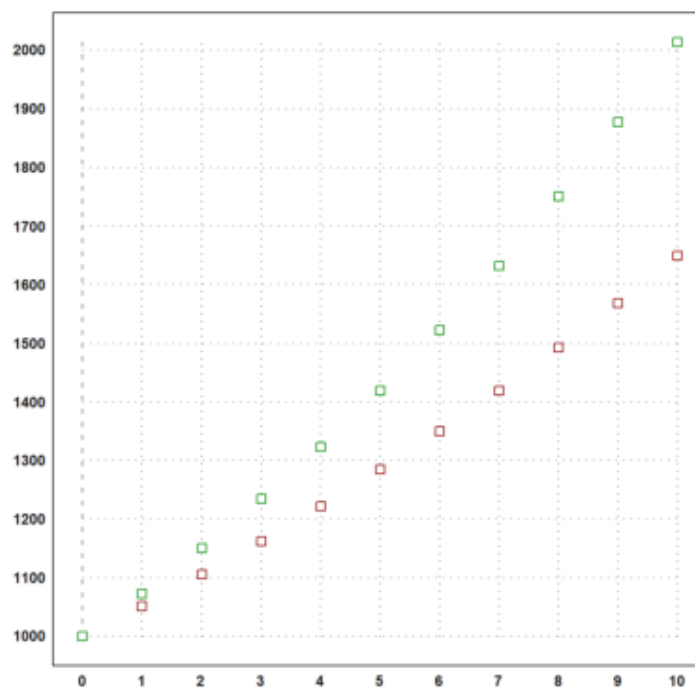
Seperti yang bisa dilihat, dalam sistem Jerman menggunakan titik koma sebagai pemisah dan koma desimal. Kita dapat mengubahnya di pengaturan sistem atau di Excel, tetapi tidak perlu membaca matriks ke EMT.

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

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

0	1000	1000
1	1051.27	1072.51
2	1105.17	1150.27
3	1161.83	1233.68
4	1221.4	1323.13
5	1284.03	1419.07
6	1349.86	1521.96
7	1419.07	1632.32
8	1491.82	1750.67
9	1568.31	1877.61
10	1648.72	2013.75

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



Ada cara yang lebih mendasar untuk membaca data dari sebuah file. Kita dapat membuka file dan membaca angka baris demi baris. Fungsi "getvectorline ()" akan membaca angka dari sebaris data. Secara default, ini mengharapkan titik desimal. Tapi itu juga bisa menggunakan koma desimal, jika kita memanggil "setdecimal-dot (",")" sebelum kita menggunakan fungsi ini.

Fungsi berikut adalah contoh untuk ini. Ini akan berhenti di akhir file atau baris kosong.

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

```
endfunction
```

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

```
0          1000          1000
1          1051          271096
1072       508181          2
```

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

```
0.504815975079
```

Contoh Lain

Kita akan membaca file csv yang bernama "GOOG"

```
>reset;
>file="GOOG.csv"; ...
>printfile("GOOG.csv")
```

```
Date,Open,High,Low,Close,Adj Close,Volume
2019-11-04,1276.449951,1323.739990,1276.354980,1311.369995,1311.369995,7217800
2019-11-11,1303.180054,1334.880005,1293.510010,1334.869995,1334.869995,5900600
2019-11-18,1332.219971,1335.529053,1291.150024,1295.339966,1295.339966,6446400
2019-11-25,1299.180054,1318.359985,1298.130005,1304.959961,1304.959961,3688500
2019-12-02,1301.000000,1344.000000,1279.000000,1340.619995,1340.619995,6719700
2019-12-09,1338.040039,1359.449951,1336.040039,1347.829956,1347.829956,6129400
2019-12-16,1356.500000,1365.000000,1348.984985,1349.589966,1349.589966,9558800
2019-12-23,1355.869995,1364.530029,1342.780029,1351.890015,1351.890015,2936500
2019-12-30,1350.000000,1372.500000,1329.084961,1360.660034,1360.660034,4605700
2020-01-06,1350.000000,1434.928955,1350.000000,1429.729980,1429.729980,8084600
2020-01-13,1436.130005,1481.295044,1426.020020,1480.390015,1480.390015,8063800
2020-01-20,1479.119995,1503.213989,1465.250000,1466.709961,1466.709961,6783300
2020-01-27,1431.000000,1470.130005,1421.199951,1434.229980,1434.229980,8166900
2020-02-03,1462.000000,1490.000000,1426.300049,1479.229980,1479.229980,11826100
2020-02-10,1474.319946,1529.630005,1474.319946,1520.739990,1520.739990,6059400
2020-02-17,1515.000000,1532.105957,1480.439941,1485.109985,1485.109985,4898300
2020-02-24,1426.109985,1438.140015,1271.000000,1339.329956,1339.329956,14316700
2020-03-02,1351.609985,1410.150024,1261.050049,1298.410034,1298.410034,11969000
2020-03-09,1205.300049,1281.150024,1113.300049,1219.729980,1219.729980,16512100
2020-03-16,1096.000000,1157.969971,1037.280029,1072.319946,1072.319946,19600200
2020-03-23,1061.319946,1169.969971,1013.536011,1110.709961,1110.709961,18250300
2020-03-30,1125.040039,1175.310059,1079.810059,1097.880005,1097.880005,11683000
2020-04-06,1138.000000,1225.569946,1130.939941,1211.449951,1211.449951,9202500
2020-04-13,1209.180054,1294.430054,1187.598022,1283.250000,1283.250000,10349000
2020-04-20,1271.000000,1293.310059,1209.709961,1279.310059,1279.310059,9148200
2020-04-27,1296.000000,1359.989990,1232.199951,1320.609985,1320.609985,13086900
2020-05-04,1308.229980,1398.760010,1299.000000,1388.369995,1388.369995,7156600
2020-05-11,1378.280029,1416.530029,1323.910034,1373.189941,1373.189941,7924600
2020-05-18,1361.750000,1415.489990,1354.250000,1410.420044,1410.420044,7454400
2020-05-25,1437.270020,1441.000000,1391.290039,1428.920044,1428.920044,7276700
2020-06-01,1418.390015,1446.552002,1404.729980,1438.390015,1438.390015,6970600
2020-06-08,1422.339966,1474.259033,1386.020020,1413.180054,1413.180054,8274100
2020-06-15,1390.800049,1460.000000,1387.920044,1431.719971,1431.719971,9501200
2020-06-22,1429.000000,1475.941040,1351.989990,1359.900024,1359.900024,10226400
2020-06-29,1358.180054,1482.949951,1347.010010,1464.699951,1464.699951,7486900
```

```
2020-07-06,1480.060059,1543.829956,1472.859985,1541.739990,1541.739990,7551500
2020-07-13,1550.000000,1577.131958,1483.500000,1515.550049,1515.550049,8018100
2020-07-20,1515.260010,1586.989990,1488.400024,1511.869995,1511.869995,6879500
2020-07-27,1515.599976,1540.969971,1454.030029,1482.959961,1482.959961,9166000
2020-08-03,1486.640015,1516.844971,1458.650024,1494.489990,1494.489990,9785200
2020-08-10,1487.180054,1537.250000,1473.079956,1507.729980,1507.729980,6991400
2020-08-17,1514.670044,1597.719971,1507.969971,1580.420044,1580.420044,8219400
2020-08-24,1593.979980,1659.219971,1580.569946,1644.410034,1644.410034,11011800
2020-08-31,1647.890015,1733.180054,1547.613037,1591.040039,1591.040039,11877700
2020-09-07,1533.510010,1584.081055,1497.359985,1520.719971,1520.719971,7601300
2020-09-14,1539.005005,1564.000000,1437.130005,1459.989990,1459.989990,9323100
2020-09-21,1440.060059,1469.520020,1406.550049,1444.959961,1444.959961,8902600
2020-09-28,1474.209961,1499.040039,1449.301025,1458.420044,1458.420044,7750300
2020-10-05,1466.209961,1516.520020,1436.000000,1515.219971,1515.219971,6728000
2020-10-12,1543.000000,1593.859985,1532.569946,1573.010010,1573.010010,8989300
2020-10-19,1580.459961,1642.359985,1525.670044,1641.000000,1641.000000,9226500
2020-10-26,1625.010010,1687.000000,1514.619995,1621.010010,1621.010010,11248500
2020-11-02,null,null,null,null,null,null
2020-11-02,1628.160034,1660.674927,1627.652466,1638.349976,1638.349976,1278888
```

Kita coba lagi untuk membaca file csv bernama "sample".

```
>reset;
>file="sample.csv"; ...
>printfile("sample.csv")
```

```
female,read,write,math,hon,femalexmath
0,57,52,41,0,0
1,68,59,53,0,53
0,44,33,54,0,0
0,63,44,47,0,0
0,47,52,57,0,0
0,44,52,51,0,0
0,50,59,42,0,0
0,34,46,45,0,0
0,63,57,54,0,0
0,57,55,52,0,0
0,60,46,51,0,0
0,57,65,51,1,0
0,73,60,71,0,0
0,54,63,57,1,0
0,45,57,50,0,0
0,42,49,43,0,0
0,47,52,51,0,0
0,57,57,60,0,0
0,68,65,62,1,0
0,55,39,57,0,0
0,63,49,35,0,0
0,63,63,75,1,0
0,50,40,45,0,0
0,60,52,57,0,0
0,37,44,45,0,0
0,34,37,46,0,0
0,65,65,66,1,0
0,47,57,57,0,0
```

0,44,38,49,0,0
0,52,44,49,0,0
0,42,31,57,0,0
0,76,52,64,0,0
0,65,67,63,1,0
0,42,41,57,0,0
0,52,59,50,0,0
0,60,65,58,1,0
0,68,54,75,0,0
0,65,62,68,1,0
0,47,31,44,0,0
0,39,31,40,0,0
0,47,47,41,0,0
0,55,59,62,0,0
0,52,54,57,0,0
0,42,41,43,0,0
0,65,65,48,1,0
0,55,59,63,0,0
0,50,40,39,0,0
0,65,59,70,0,0
0,47,59,63,0,0
0,57,54,59,0,0
0,53,61,61,1,0
0,39,33,38,0,0
0,44,44,61,0,0
0,63,59,49,0,0
0,73,62,73,1,0
0,39,39,44,0,0
0,37,37,42,0,0
0,42,39,39,0,0
0,63,57,55,0,0
0,48,49,52,0,0
0,50,46,45,0,0
0,47,62,61,1,0
0,44,44,39,0,0
0,34,33,41,0,0
0,50,42,50,0,0
0,44,41,40,0,0
0,60,54,60,0,0
0,47,39,47,0,0
0,63,43,59,0,0
0,50,33,49,0,0
0,44,44,46,0,0
0,60,54,58,0,0
0,73,67,71,1,0
0,68,59,58,0,0
0,55,45,46,0,0
0,47,40,43,0,0
0,55,61,54,1,0
0,68,59,56,0,0
0,31,36,46,0,0
0,47,41,54,0,0
0,63,59,57,0,0
0,36,49,54,0,0
0,68,59,71,0,0
0,63,65,48,1,0

```
0,55,41,40,0,0
0,55,62,64,1,0
0,52,41,51,0,0
0,34,49,39,0,0
0,50,31,40,0,0
0,55,49,61,0,0
0,52,62,66,1,0
0,63,49,49,0,0
1,68,62,65,1,65
1,39,44,52,0,52
1,44,44,46,0,46
1,50,62,61,1,61
1,71,65,72,1,72
1,63,65,71,1,71
1,34,44,40,0,40
```

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. Kita menggunakan ekspresi reguler untuk mendai "Versi ..." di sebuah judul.

```
>reset;
>function readversion () ...

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

```
>readversion
```

```
Version 2024-01-12
```

SUB TOPIK 6 : Menggambar Grafik Statistika

Diagram adalah suatu representasi simbolis informasi dalam bentuk geometri 2 dimensi sesuai teknik visualisasi. Kadang teknik yang dipakai memanfaatkan visualisasi tiga dimensi yang kemudian diproyeksikan ke permukaan dua dimensi. Kata grafik dan bagan biasa dipakai sebagai sinonim kata diagram.

Grafik juga diartikan sebagai suatu kombinasi angka, huruf, simbol, gambar, lambang, perkataan dan lukisan yang disajikan dalam suatu media untuk menggambarkan informasi dari data.

Diagram dapat digunakan untuk alasan yang berbeda, seperti untuk menunjukkan bagian dari keseluruhan, langkah-langkah dari suatu proses, dan hubungan. Sebuah bantuan grafis akan menampilkan informasi secara visual sehingga pembaca dapat lebih memahami dan mengingat ide-ide.

Jenis-Jenis Diagram ataupun kurva pada EMT:

- Diagram kotak,

- Diagram batang,
- Diagram lingkaran,
- Diagram bintang,
- Diagram impuls,
- Histogram,
- Kurva fungsi kerapatan probabilitas,
- Kurva fungsi distribusi kumulatif,
- Diagram titik,
- Diagram garis,
- Kurva regresi.

Diagram Kotak

Dalam statistika deskriptif, diagram kotak garis atau boxplot adalah metode grafis untuk menggambarkan kumpulan data numerik berdasarkan nilai kuartilnya. Diagram kotak garis bersifat nonparametrik, artinya diagram ini menampilkan variasi sampel populasi statistik tanpa membuat asumsi apa pun tentang distribusi statistik yang mendasarinya. Jarak antara bagian-bagian kotak yang berbeda menunjukkan derajat dispersi (sebaran), kemiringan, dan pencilan dari data tersebut.

Boxplot merupakan ringkasan distribusi sampel yang disajikan secara grafis yang bisa menggambarkan bentuk distribusi data (skewness), ukuran tendensi sentral dan ukuran penyebaran (keragaman) data pengamatan.

Terdapat 5 ukuran statistik yang bisa kita baca dari boxplot, yaitu:

- nilai minimum: nilai observasi terkecil
- Q1: kuartil terendah atau kuartil pertama
- Q2: median atau nilai pertengahan
- Q3: kuartil tertinggi atau kuartil ketiga
- nilai maksimum: nilai observasi terbesar.
- Selain itu, boxplot juga dapat menunjukkan ada tidaknya nilai

outlier dan nilai ekstrim dari data pengamatan.

Contoh penggunaan diagram kotak di EMT

Misalkan ada sebuah data M sebagai berikut:

Maka, akan divisualisasikan data M dengan diagram kotak

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

```
999.9
2.72641400622
```

$$mean = \frac{sum(data)}{len(data)}$$

Jumlah data dibagi banyak data.

Sedangkan dev() merupakan standart deviasi nya.

Rumusnya:

$$deviasi = [(x - mean)^2 \text{ for } x \text{ in data}]$$

$$variasi = \frac{\text{sum}(deviasi)}{\text{len}(data)}$$

$$\text{standartdeviasi} = \sqrt{\text{variasi}}$$

Pertama-tama data M akan diurut berdasarkan nilainya, dari terendah hingga tertinggi(terbesar).

$$M = [997, 997, 998, 998, 998, 1000, 1001, 1002, 1004, 1004]$$

Mencari Median:

Median=kuartil tengah(Q3)

Median terletak pada data ke-

$$\text{Median} = \frac{1}{2} \cdot (\text{datake} - (\frac{n}{2}) + \text{datake} - (\frac{n}{2}) + 1)$$

Dikarenakan data berjumlah genap maka, digunakan rumus seperti itu.

$$= \frac{1}{2} \cdot (\text{datake} - (\frac{10}{2}) + \text{datake} - (\frac{10}{2}) + 1)$$

$$= \frac{1}{2} \cdot (\text{datake} - (5) + \text{datake} - (5) + 1)$$

$$= \frac{1}{2} \cdot (998 + 1000)$$

$$= 999$$

Ditemukan median-nya adalah 999

Mencari Kuartil atas dan bawah

$$Q1 = \text{datake} - \frac{1}{4} \cdot n$$

$$= \text{datake} - \frac{1}{4} \cdot 10$$

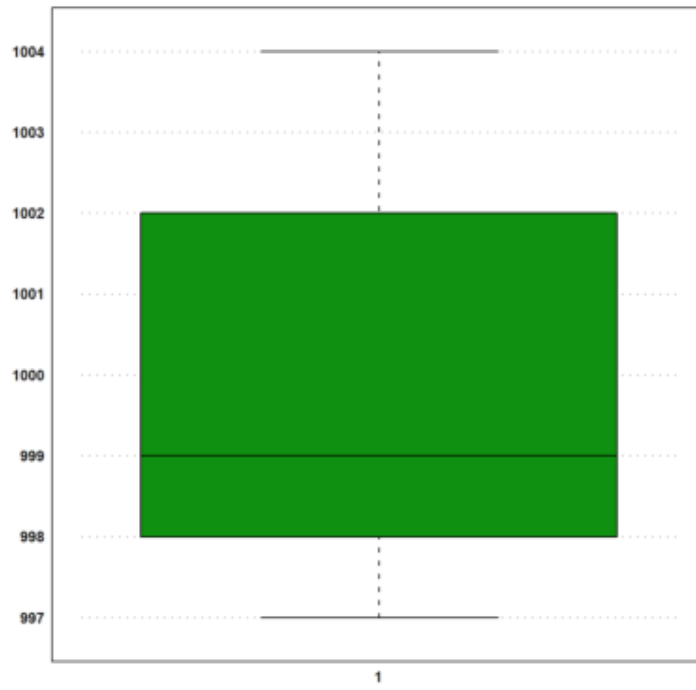
$$= 998$$

$$Q3 = \text{datake} - \frac{3}{4} \cdot n$$

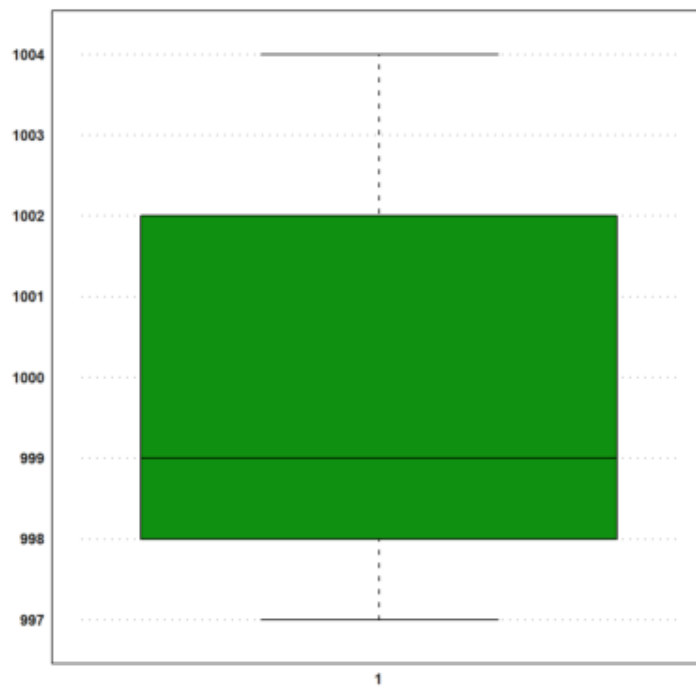
$$= \text{datake} - \frac{3}{4} \cdot 10$$

$$= 1002$$

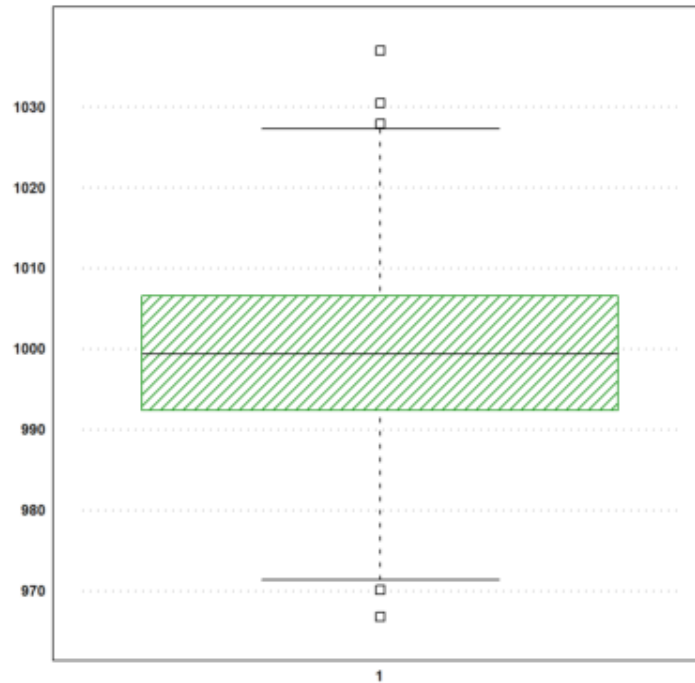
```
>boxplot(M, outliers=1.5):
```



```
>boxplot(M, lab=none, style="0#", textcolor=red, outliers=1.5, pointstyle="o", range=none)
```



```
>x=normal(1000)*10+1000; boxplot(x, lab=none, style="/", textcolor=red, outliers=1.5, poin
```



Penjelasan:

- "lab" disini digunakan untuk melabelkan atau menamai kotak
- "style" disini digunakan untuk mengatur style atau gaya yang

digunakan pada kotak, bisa menggunakan 0, =, +, /, |

- "textcolor" digunakan untuk mewarnai text pada "lab"
- "outliers" Parameter outliers dalam konteks boxplot menentukan

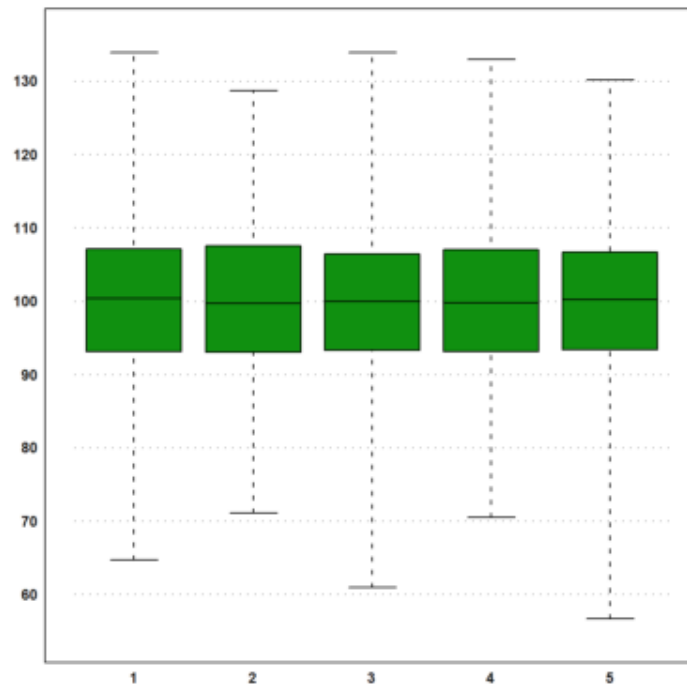
apakah dan bagaimana outlier (nilai-nilai ekstrem yang jauh dari kebanyakan nilai) ditampilkan dalam boxplot. Dalam beberapa fungsi pembuat boxplot, parameter ini digunakan untuk mengontrol penampilan dan penanganan.

- "pointstyle" digunakan untuk mengatur style atau gaya pada

point-point yang ada

- "range" digunakan untuk mengatur jarak

```
>x=randnormal(5,1000,100,10); boxplot(x,outliers=none):
```



Boxplots dapat membantu kita dalam memahami karakteristik dari distribusi data. Selain untuk melihat derajat penyebaran data (yang dapat dilihat dari tinggi/panjang boxplot) juga dapat digunakan untuk menilai kesimetrisan sebaran data. Panjang kotak menggambarkan tingkat penyebaran atau keragaman data pengamatan, sedangkan letak median dan panjang whisker menggambarkan tingkat kesimetrisannya.

Diagram Batang

Diagram batang merupakan jenis grafik yang digunakan untuk

menunjukkan dan membandingkan kuantitas data dalam kategori yang berbeda. Diagram batang umumnya digunakan untuk menggambarkan perkembangan nilai suatu objek penelitian dalam kurun waktu tertentu. Diagram batang menunjukkan keterangan-keterangan dengan batang-batang tegak atau mendatar dan sama lebar dengan batang-batang terpisah

Diagram batang memiliki kelebihan, yaitu diagram batang merupakan

diagram paling sederhana dan paling umum digunakan. Namun diagram batang juga memiliki kekurangan, yaitu diagram batang hanya disajikan data yang telah dikelompokkan atas atribut dan kategori. Dan Diagram batang tidak dapat menampilkan datum dari tiap orang atau benda yang dicatat(sebut saja data individual).

Contoh penggunaan diagram batang di EMT

Untuk membuat suatu diagram atau plot kita memerlukan sebuah data yang nantinya akan diolah. Data yang kita gunakan merupakan hasil pemilu Jerman sejak tahun 1990, diukur dalam kursi.

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

Untuk P disini dimasukkan sebagai nama-nama partai pada data sebelumnya.

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

Kolom pertama = Tahun Terjadinya Pemilu
Kolom kedua = jumlah kursi keseluruhan pada tahun tertentu
kolom ketiga sampai ketujuh = jumlah kursi tiap partai

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

Fungsi BW[3:7] disini mengartikan bahwa fungsi bw akan dipakai pada kolom 3 sampai 7

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

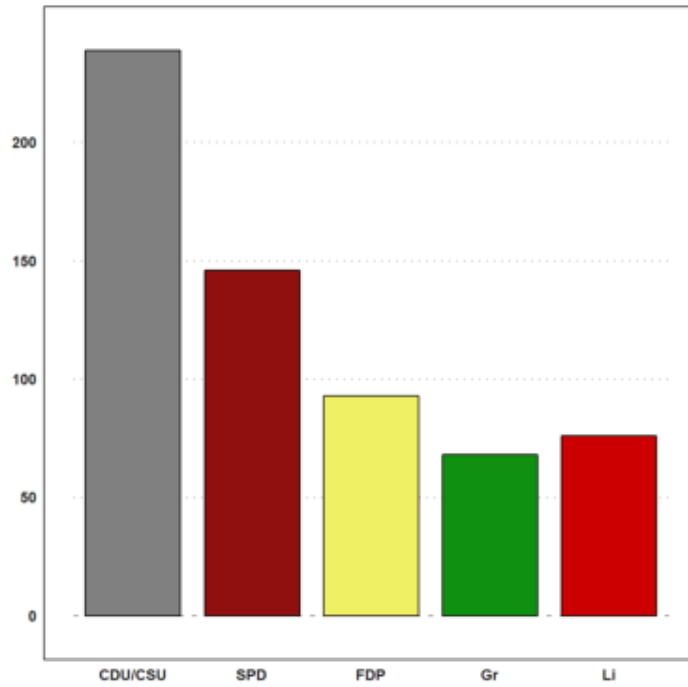
	CDU/CSU	SPD	FDP	Gr	Li
1990	48	36	12	1	3
1994	44	38	7	7	4
1998	37	45	6	7	5
2002	41	42	8	9	0
2005	37	36	10	8	9
2009	38	23	15	11	12
2013	49	31	0	10	10

Memmbaca dari data sebelumnya yang merupakan matriks menjadi sebuah tabel yang tiap kolom nya sudah dinamai.

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

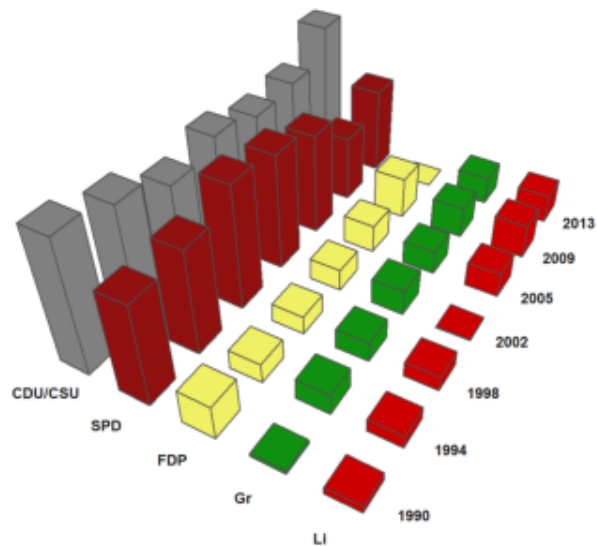
Fungsi CP disini digunakan nantinya untuk membuat perbedaan warna pada setiap batang yang menyesuaikan nama partai

```
>columnplot(BW[6, 3:7], P, color=CP):
```



Berikut merupakan diagram batang, hasil dari memplot data sebelumnya. Ini merupakan diagram batang dalam 2 dimensi.

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



Ini adalah perintah untuk membuat diagram batang(kolom) dalam 3 Dimensi . Plot ini akan menggunakan data dari BT, dengan kolom-kolom yang dipilih pada data BW, yaitu nama-nama partai (scols=p). Selain itu, parameter angle=30° mengatur sudut pandang plot dalam ruang 3D, ccols=CP mungkin mengatur warna kolom.

Diagram Lingkaran

Diagram lingkaran merupakan suatu diagram yang difungsikan untuk

menyajikan data dalam bentuk lingkaran baik menggunakan data absolut maupun relatif. Untuk membuat diagram lingkaran pertama-tama kita harus membuat lingkaran terlebih dahulu lalu dibagi-bagi menjadi beberapa sektor. Tiap sektor melukiskan kategori data yang terlebih dahulu diubah kedalam derajat.

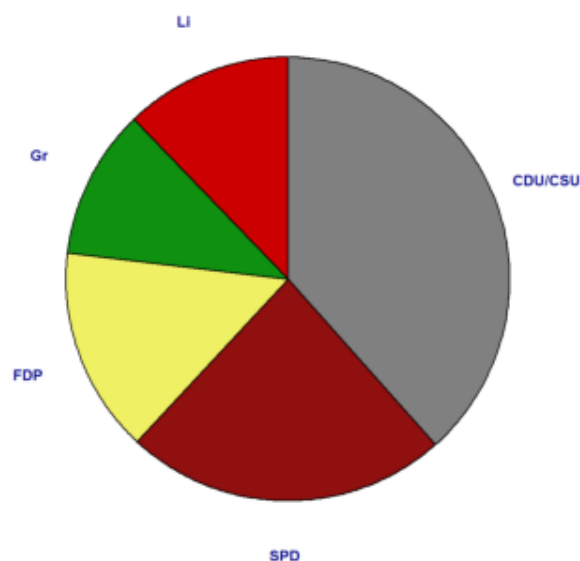
Kelebihan diagram lingkaran, yaitu Tempat untuk membuat diagram

lingkaran tidak terlalu besar. Dan Diagram lingkaran sangat berguna untuk menunjukkan dan membandingkan proporsi dari data. Namun diagram lingkaran juga memiliki Kekurangan, yaitu Diagram lingkaran tidak dapat menunjukkan frekuensinya.

Contoh penggunaan diagram lingkaran di EMT

Data yang dipakai Masih menggunakan data sebelumnya, yaitu data pemilu jerman dari tahun.

```
>i=[1,2,3,4,5]; piechart(BW[6,3:7][i], style="O#", color=CP[i],lab=P[i], textcolor=blue, r
```



"i" digunakan untuk iterasi kolom, atau mengurutkan kolom sesuai potongan pada diagram lingkaran
 "textcolor" digunakan untuk mewarnai text
 "color" digunakan untuk mewarnai potongan lingkaran
 "style" digunakan untuk mengatur gradasi style atau gaya pada diagram lingkaran
 "lab" digunakan untuk melabelkan atau menamai setiap potongan lingkaran
 "r" digunakan untuk mengatur radius pada diagram lingkaran

```
>i=[1,5,3,2,4]; piechart(BW[6,3:7][i], style="/", color=CP[i],lab=P[i], textcolor=green, r
```

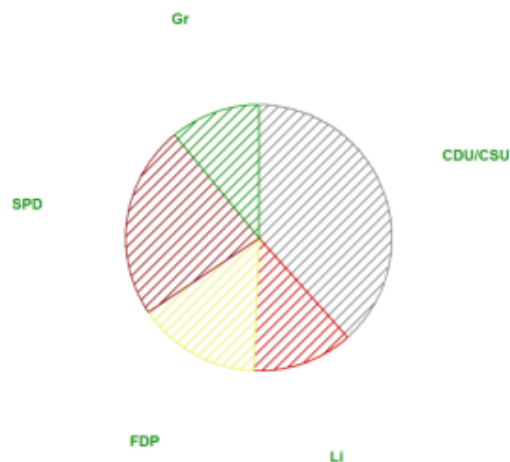


Diagram Bintang

Diagram bintang adalah jenis diagram yang terdiri dari titik-titik

yang terhubung oleh garis-garis sehingga membentuk pola seperti bintang. Grafik ini memiliki serangkaian garis yang menjulur dari titik pusat, menyerupai bentuk bintang. Setiap lengan dari bintang mewakili dimensi atau variabel yang berbeda, dan panjang garis atau area yang diisi dapat mencerminkan nilai atau kontribusi relatif dari setiap dimensi.

Diagram ini sering digunakan dalam matematika, statistik, dan ilmu

komputer untuk memvisualisasikan hubungan antara titik-titik atau entitas dalam sebuah sistem. Diagram bintang dapat memiliki berbagai bentuk dan ukuran tergantung pada jumlah titik yang digunakan.

Diagram bintang paling sederhana adalah diagram bintang dengan lima

titik yang membentuk pola bintang lima sudut. Namun, diagram bintang juga dapat memiliki lebih banyak titik dan membentuk pola yang lebih kompleks. Diagram bintang dapat digunakan untuk berbagai tujuan, seperti memvisualisasikan hubungan antara entitas dalam jaringan sosial, menggambarkan hubungan antara variabel dalam analisis multivariat, atau menggambarkan struktur hierarki dalam organisasi.

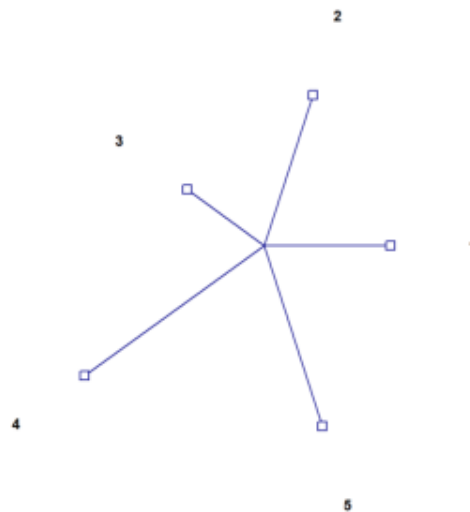
Contoh penggunaan diagram bintang di EMT

Misalkan ada sebuah data V:

```
>v = [4, 5, 3, 7, 6];
```

Selanjutnya akan di ilustrasikan data V kebentuk diagram bintang, sebagai berikut:

```
>starplot(v, style="+", color=blue, lab=1:5, rays=10, pstyle="+", textcolor=black, r=1.5)
```



Penjelasan:

- "lab" disini digunakan untuk melabelkan atau menamai kotak
- "style" disini digunakan untuk mengatur style atau gaya yang

digunakan pada garis bintang.

- "textcolor" digunakan untuk mewarnai text pada "lab".
- "rays" digunakan untuk menunjukkan jumlah "rays" atau "spokes"

pada radar plot.

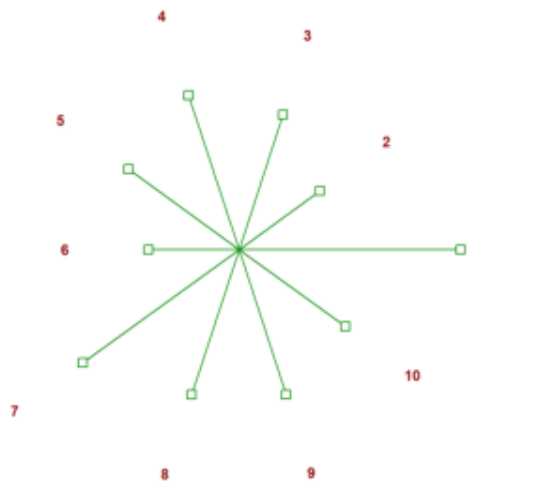
- "pstyle" digunakan untuk mengatur style atau gaya pada point-point

yang ada.

- "r" digunakan untuk mengatur radius.

#Contoh penggunaan diagram bintang pada distribusi normal statistika

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



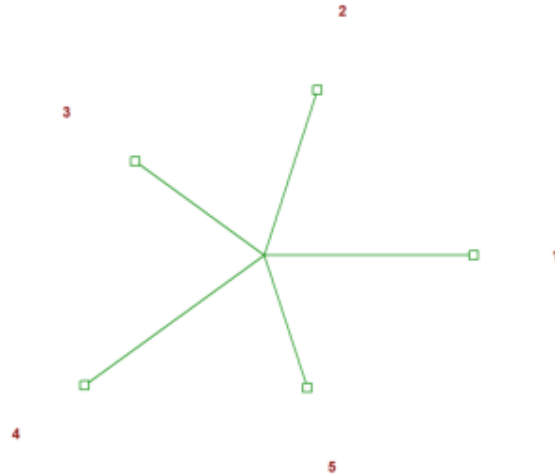
#Contoh soal lainnya

Misalkan ada sebuah data nilai matematika dari latihan UTBK di suatu tempat les. Data tersebut berisikan 5 orang siswa dengan nilai 87 untuk siswa pertama, 72 untuk siswa kedua, 66 untuk siswa ketiga, 92 untuk siswa keempat, dan 58 untuk siswa kelima. Ilustrasikan data tersebut dengan diagram bintang!

```
>A=[87,72,66,92,58]
```

```
[87, 72, 66, 92, 58]
```

```
>starplot(A,lab=1:5,>rays):
```



#Contoh soal lainnya

PT SukaMaju memiliki tim penjualan yang terdiri dari lima anggota.

Perusahaan tersebut ingin mengevaluasi kinerja masing-masing anggota tim berdasarkan Penjualan tiap bulan. Berikut adalah data evaluasi kinerja untuk setiap anggota tim pada bulan september:

- John: Penjualan (85)
- Maria: Penjualan (78)
- David: Penjualan (92)
- Lisa: Penjualan (85)
- Kevin: Penjualan (80)

Akan diilustrasikan dengan diagram bintang, sebagai berikut:

```
>Nama=["Johan","Maria","David","Lisa","Kevin"];  
>Penjualan=[85,78,92,85,80];  
>starplot(Penjualan, style="+", color=red, lab>Nama, rays=10, pstyle="+", textcolor=blue,
```

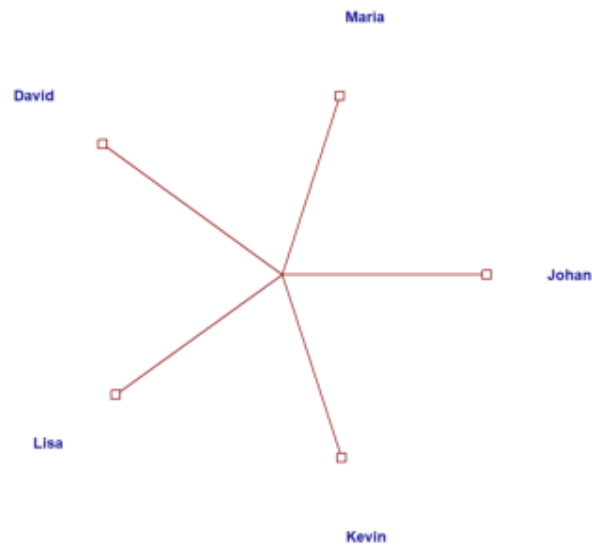


Diagram Impuls

Dalam statistik, istilah "grafik impuls" mungkin tidak umum digunakan. Namun, jika Anda merujuk pada grafik distribusi fungsi massa probabilitas diskrit, di mana probabilitas terkonsentrasi pada nilai tertentu, maka histogram atau diagram batang mungkin sesuai dengan konsep ini.

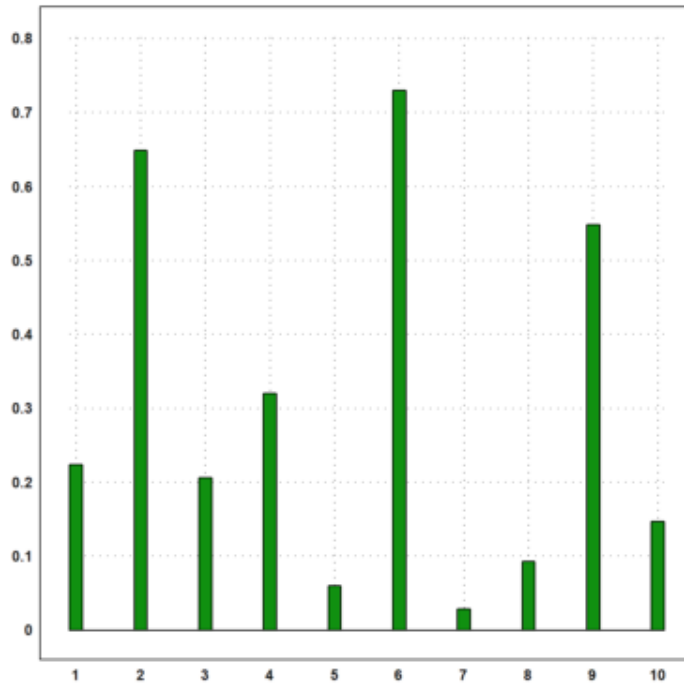
Sebagai contoh, jika Anda memiliki distribusi probabilitas diskrit seperti distribusi Poisson atau distribusi binomial, Anda dapat membuat grafik yang menunjukkan probabilitas pada setiap nilai yang mungkin. Ini akan menunjukkan "puncak" pada nilai-nilai tertentu, yang mungkin tampak mirip dengan bentuk grafik impuls.

Dalam Euler Math Toolbox, dalam membuat grafik statistik, termasuk grafik impuls, kita dapat menggunakan fungsi perencanaan yang dibangun. Berikut salah satu contoh bagaimana anda dapat membuat bagan impuls untuk distribusi probabilitas diskrit

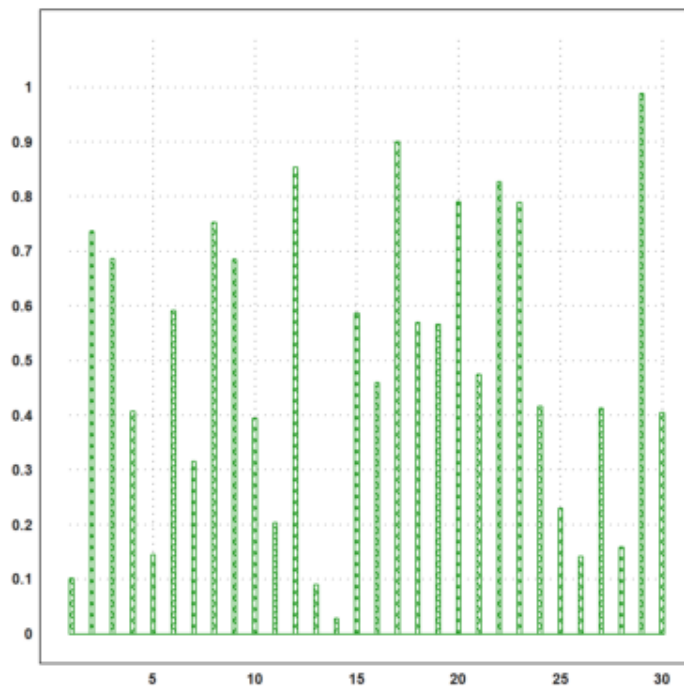
Contoh penggunaan diagram impuls di EMT

Berikut adalah plot impuls dari sebuah data acak yang terdistribusi secara merata didalam interval $[0,1]$.

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



```
>plot2d(makeimpulse(1:30,random(1,30)),>bar, style="\/" ) :
```



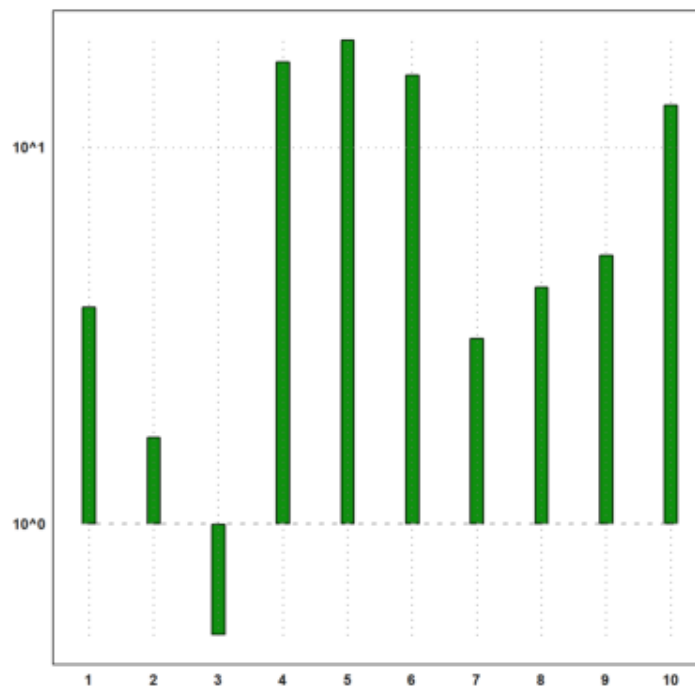
Jika data tersebut merupakan data yang terdistribusi secara

eksponensial, maka diperlukan plot logaritmik untuk membuat diagram impuls nya. Distribusi eksponensial adalah jenis distribusi di mana banyak data memiliki nilai yang kecil, tetapi ada beberapa data yang memiliki nilai yang sangat besar.

Pada sumbu logaritmik, perbedaan skala nilai yang besar dapat

diurutkan. Sehingga akan ada perbedaan kecil yang dapat dilihat lebih jelas. Dikarenakan dapat terlihat perbedaan antara nilai-nilai yang mendekati nol dan nilai yang lebih besar pada grafik logaritmik. Berikut adalah penggunaan diagram impuls dengan fungsi logaritma.

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



#Contoh soal lainnya

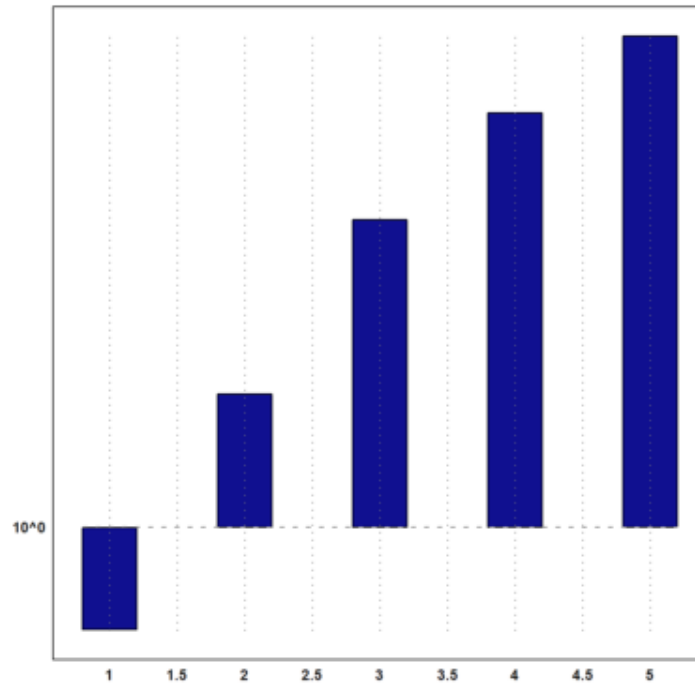
Sebuah koloni bakteri tumbuh secara eksponensial. Pada suatu

eksperimen, populasi bakteri diukur setiap jam selama 5 jam. Data pertumbuhan populasi bakteri (dalam ribuan) adalah sebagai berikut:

```
>Jam=[1:5];
```

Populasi ini diatur dalam ribuan

```
>Populasi=[2,5,20,80,320];  
>logimpulseplot(Jam,log(Populasi), style="O#", color=blue, d=0.2):
```



Penjelasan:

- "Jam" disini digunakan sebagai data pada x bar
- "log(populasi)" disini digunakan untuk menglogaritmakan data yang

ada di populasi sebagai daya y bar dalam diagram

- "style" disini digunakan untuk mengatur style atau gaya yang

digunakan dalam kotak/batang.

- "color" disini digunakan untuk mengatur warna pada kotak/batang

diagram.

- "d'" disini digunakan untuk mengatur ketebalan kotak/batang diagram.

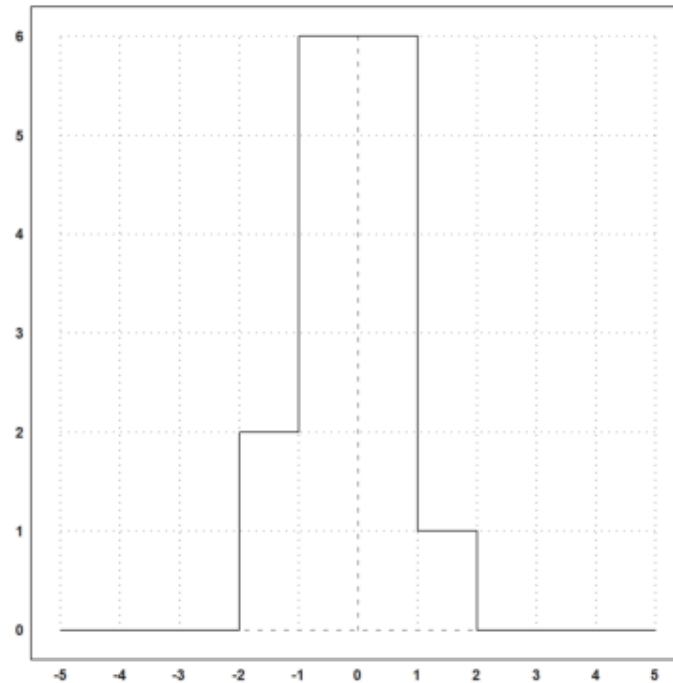
Histogram

Histogram adalah representasi grafis untuk distribusi warna dari citra digital. Sumbu ordinat vertikal merupakan representasi piksel dengan nilai tonal dari tiap-tiap deret bin pada sumbu axis horizontalnya. Sumbu axis terdiri dari deret logaritmik bin densitometry yang membentuk rentang luminasi atau exposure range yang mendekati respon spectral sensitivity visual mata manusia. Deret bin pada density yang terpadat mempunyai interval yang relatif sangat linear dengan variabel mid-tone terletak tepat di tengahnya.

Contoh penggunaan Histogram di EMT

Berikut merupakan contoh histogram pada sebuah data yang terdistribusi normal.

```
>plot2d(histo(normal(1,15),v=-5:5,<bar)):
```



#Contoh soal lainnya

Sebuah universitas melakukan penelitian untuk mengidentifikasi

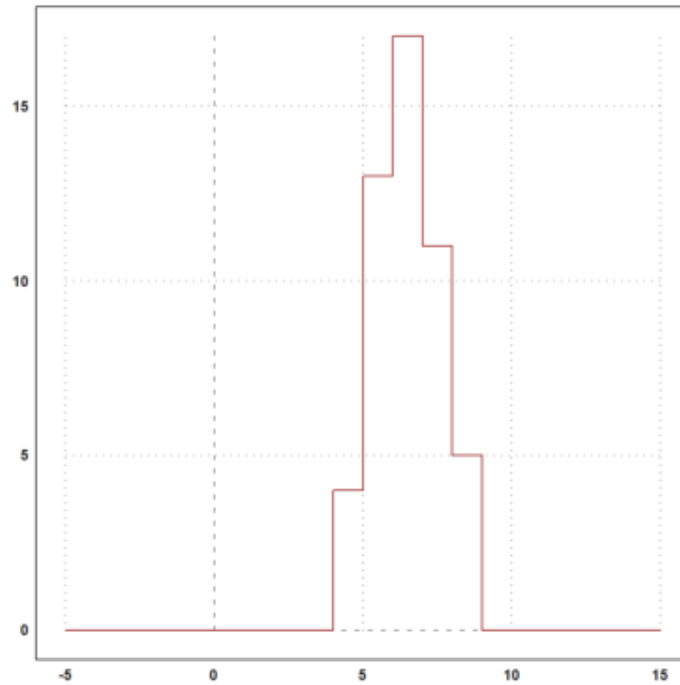
distribusi waktu studi mahasiswa di suatu jurusan. Data waktu studi (dalam tahun) dari 100 mahasiswa lulusan jurusan tersebut telah dikumpulkan. Berikut adalah data tersebut:

Data M:

```
>M=[5,4,6,7,5,6,7,8,4,6, ...  
>5,6,7,5,6,5,7,6,8,6, ...  
>4,5,6,7,5,6,5,6,7,8, ...  
>6,7,5,6,7,5,6,7,8,6, ...  
>7,8,6,5,4,5,6,7,5,6];
```

Akan diplotkan histogram

```
>plot2d(histo(M, v=-5:15, <bar, ),color=red):
```



Penjelasan:

- "histo" disini digunakan untuk memanggil fungsi histogram yang ada

pada plot 2d di EMT.

- "M" disini merupakan data yang akan diplotkan.
- "v" disini digunakan untuk mengatur interval pada grafik

histogram yang diperlihatkan.

- "<bar" digunakan untuk membuat bar pada grafik.
- "color" disini digunakan untuk mengatur warna grafik

#Contoh soal lainnya

Seorang pemilik toko buku ingin menganalisis pola penjualan harian

untuk memahami frekuensi pembelian buku oleh pelanggan. Data penjualan harian selama 30 hari terakhir telah dikumpulkan, dan berikut adalah jumlah buku yang terjual setiap hari:

```
>T=[35, 28, 42, 18, 24, 35, 40, 22, 29, 33, ...
>26, 38, 32, 19, 31, 25, 37, 29, 24, 27, ...
>36, 20, 28, 33, 31, 26, 39, 30, 23, 34];
>plot2d(histo(T, v=10:50, <bar),color=blue):
```

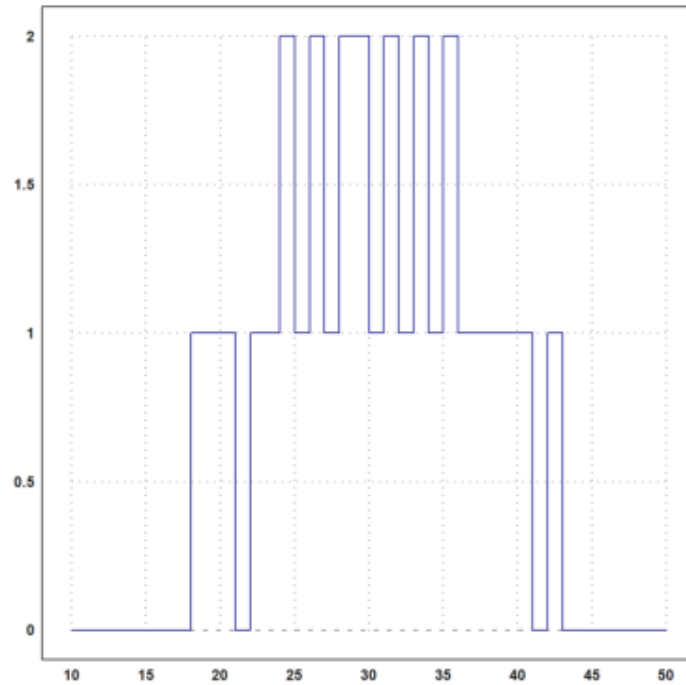


Diagram titik

Diagram titik atau disebut juga sebagai scatter plot, adalah jenis diagram statistik yang menggunakan titik-titik untuk merepresentasikan nilai dari dua variabel yang berbeda. Setiap titik dalam diagram pencar mewakili satu pengamatan atau data dengan nilai-nilai yang sesuai untuk kedua variabel tersebut.

Scatter plot sangat berguna untuk menemukan pola atau hubungan antara dua variabel, serta untuk mengevaluasi distribusi data.

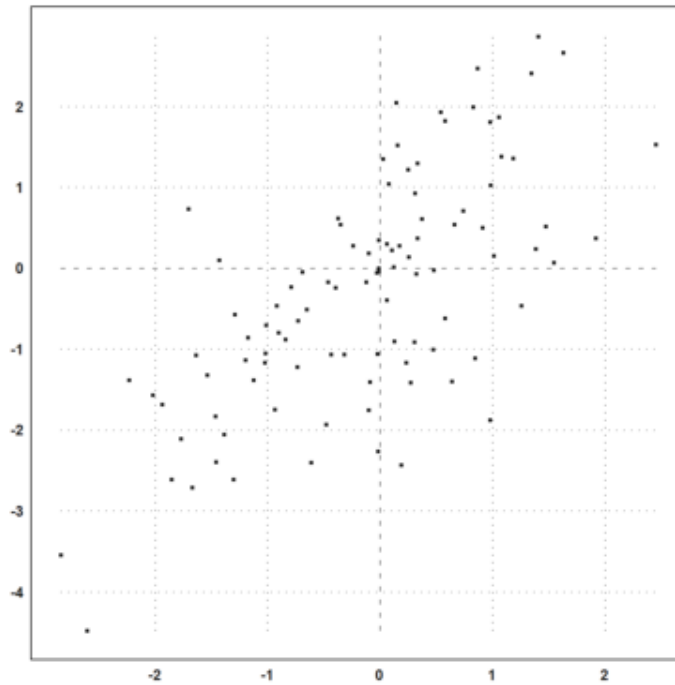
Dalam scatter plot, sumbu horizontal umumnya digunakan untuk variabel independen (bebas), sedangkan sumbu vertikal digunakan untuk variabel dependen (bergantung). Dengan mengamati pola penyebaran titik-titik, kita mendapatkan wawasan tentang apakah ada korelasi antara dua variabel dan jenis korelasi apa yang mungkin ada (positif, negatif, atau tidak ada korelasi).

Berikut contoh menggambar diagram titik di EMT

Contoh 1

Pada contoh ini, kita akan menggambar diagram titik dengan menggunakan fungsi `plot2d()`.

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



Terdapat banyak style titik yang dapat kita gunakan, yaitu:

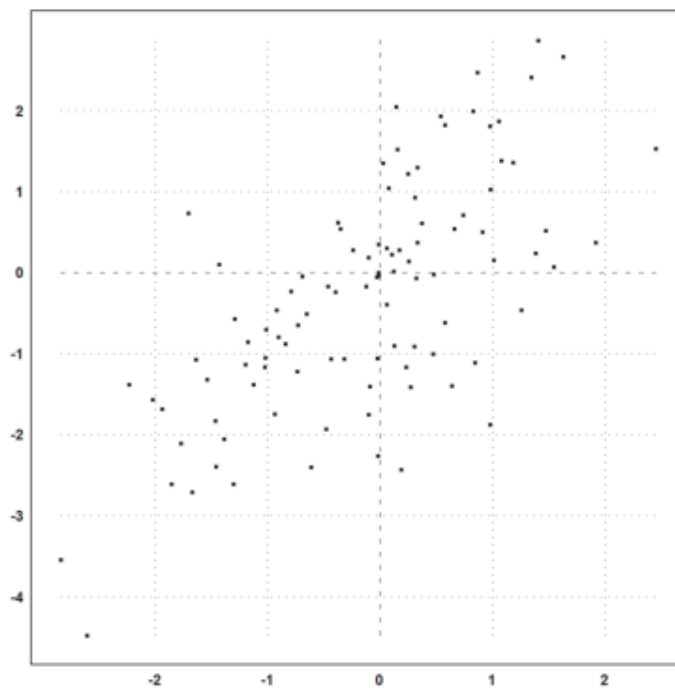
"[]", "<>", ":", ":", "...", "*", "+", "|", "-", "o",

"[]", "<>", "o" (bentuk terisi)

"[]w", "<>w", "ow" (tidak transparan)

Selanjutnya, akan kita coba gambarkan diagram tersebut menggunakan fungsi `statplot()`, pilih `plottype="p"` (karena kita akan menggambar plot titik).

```
>statplot(x,x+rotright(x),plottype="p",pstyle="..") :
```



Contoh 2

Pada contoh ini, akan kita gambarkan diagram titik menggunakan fungsi scatterplot().

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

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

```
>writetable(MS,labc=hd,tok2:["m","f"]);
```

```
Variable or function MS not found.
Error in:
writetable(MS,labc=hd,tok2:["m","f"]); ...
      ^
```

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

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

Diagram Garis

Diagram garis adalah penyajian data yang digunakan untuk menggambarkan suatu keadaan berupa data berkala atau berkelanjutan.

Selain itu, diagram ini juga bisa dikatakan berhubungan dengan kurun waktu dan untuk menunjukkan perkembangan suatu keadaan.

Diagram ini sangat tepat untuk menyajikan data untuk mengetahui kecenderungan kelakuan atau tren, seperti produksi minyak tiap tahun, jumlah kelahiran tiap tahun, jumlah produksi tiap jam, dan lain-lain.

Dalam diagram garis, terdapat sumbu vertikal (sumbu y) untuk menunjukkan frekuensi dan sumbu horizontal (sumbu x) untuk menunjukkan variabel tertentu.

Berikut contoh menggambar diagram garis di EMT

Contoh 1

Akan digambarkan diagram garis data banyaknya pelanggan di toko A tahun 2015-2023.

Kita deskripsikan terlebih dahulu matriks x dan y, kemudian akan kita buat tabel datanya.

```
>x=[2015,2016,2017,2018,2019,2020,2021,2022,2023]; y=[600,500,900,1000,800,850,900,1000,1200];
>writetable(x'|y',labc=["Tahun","Banyak Pelanggan"])
```

Tahun	Banyak Pelanggan
2015	600
2016	500
2017	900
2018	1000
2019	800
2020	850
2021	900
2022	1000
2023	1200

Selanjutnya, akan digambarkan diagram garis dengan menggunakan fungsi `statplot`, dengan format:
`statplot(x, y, plottype="l", lstyle="-", xl="", yl="", color=None, vertical=0)`

`x` : data untuk sumbu x

`y` : data untuk sumbu y

`plotstyle` : "l" (kita pilih style "l" karena berupa plot garis)

`lstyle` : style garis

`xl` : label sumbu x

`yl` : label sumbu y

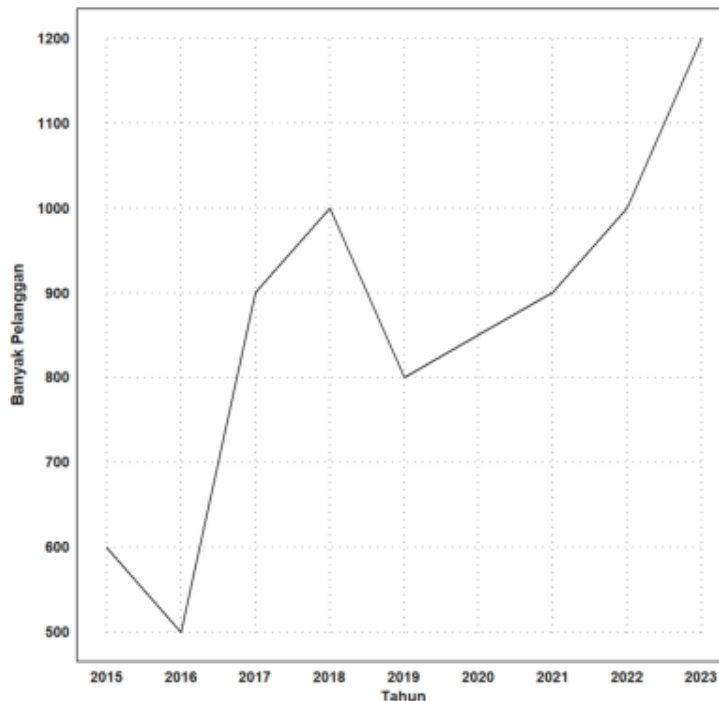
`color` : warna garis

`vertikal` : vertikal

Style garis:

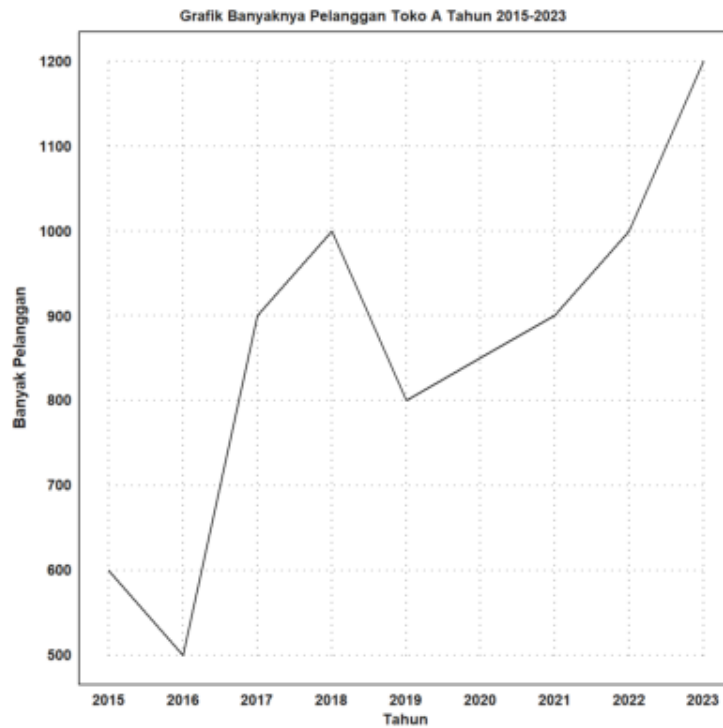
"-", "-", "-.", ".", "-.-", "-.-", "->"

```
>statplot(x,y,plottype="l",lstyle="-",xl="Tahun",yl="Banyak Pelanggan",vertical=50):
```



Kita juga bisa menambahkan judul pada grafik dengan menggunakan fungsi `title()`.

```
>title("Grafik Banyaknya Pelanggan Toko A Tahun 2015-2023"):
```



Menyajikan data dalam bentuk diagram dapat memudahkan pembaca untuk memahami data yang disajikan.

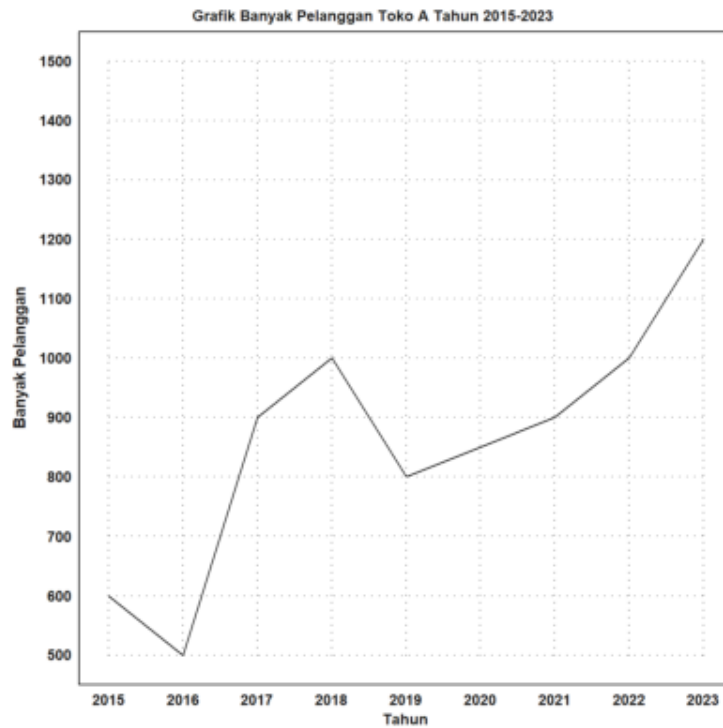
Dari diagram garis diatas, dapat kita peroleh informasi bahwa:

Banyaknya pelanggan di toko A tidak tetap (naik-turun) untuk setiap tahunnya.

Banyaknya pelanggan paling sedikit pada tahun 2016 yaitu sebanyak 500 pelanggan, sedangkan banyak pelanggan paling banyak pada tahun 2023 yaitu sebanyak 1200 pelanggan.

Selain menggunakan fungsi `statplot`, kita juga dapat menggambar diagram garis menggunakan fungsi `plot2d()` seperti yang sudah pernah kita pelajari sebelumnya, yaitu sebagai berikut.

```
>plot2d(x,y,a=2015,b=2023,c=500,d=1500,style="_",xl="Tahun",yl="Banyak Pelanggan",vertical
```



a dan b : batas untuk sumbu x
 c dan d : batas untuk sumbu y
 style : gaya garis
 xl : label untuk sumbu x
 yl : label untuk sumbu y

Selain kita dapat menggambarkan diagram garis saja atau diagram titik saja, kita juga dapat menggambarkan diagram keduanya.

Contoh 2

Akan kita gambar diagram titik dan garis data hasil pemilu Jerman dari tahun 1990 sampai 2013, diukur dalam kursi.

```
>BW := [ ...
>1990,662,319,239,79,8,17; ...
>1994,672,294,252,47,49,30; ...
>1998,669,245,298,43,47,36; ...
>2002,603,248,251,47,55,2; ...
>2005,614,226,222,61,51,54; ...
>2009,622,239,146,93,68,76; ...
>2013,631,311,193,0,63,64];
>P:=["CDU/CSU","SPD","FDP","Gr","Li"];
>BT:=BW[,3:7]; BT:=BT/sum(BT); YT:=BW[,1]';
>writetable(BT*100,wc=6,dc=0,>fixed,labc=P,labr=YT)
```

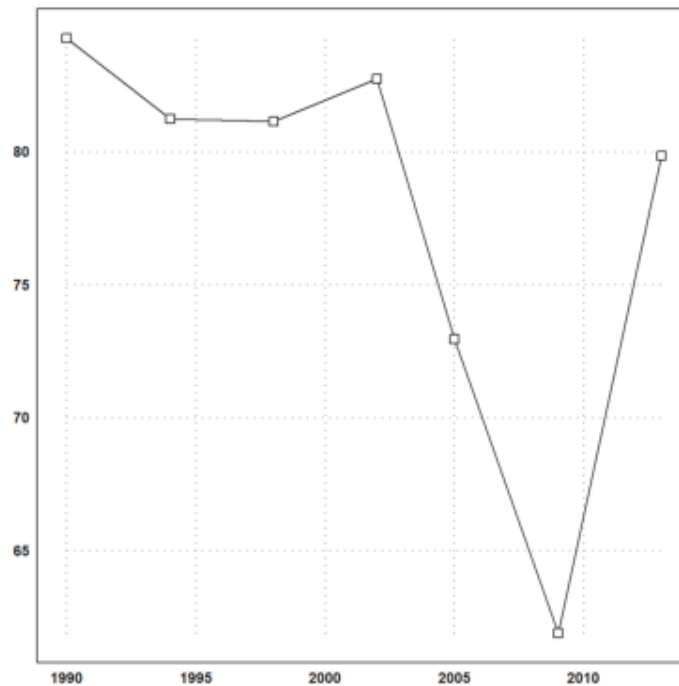
	CDU/CSU	SPD	FDP	Gr	Li
1990	48	36	12	1	3
1994	44	38	7	7	4
1998	37	45	6	7	5
2002	41	42	8	9	0
2005	37	36	10	8	9
2009	38	23	15	11	12
2013	49	31	0	10	10


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

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

Akan kita gambarkan plot statistik sederhana, yaitu plot titik dan garis secara bersamaan dengan menggunakan fungsi `statplot` dan pilih `plotype="b"`.

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



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

Untuk menggabungkan deretan data statistik dalam satu plot, dapat kita digunakan fungsi `dataplot()`.

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



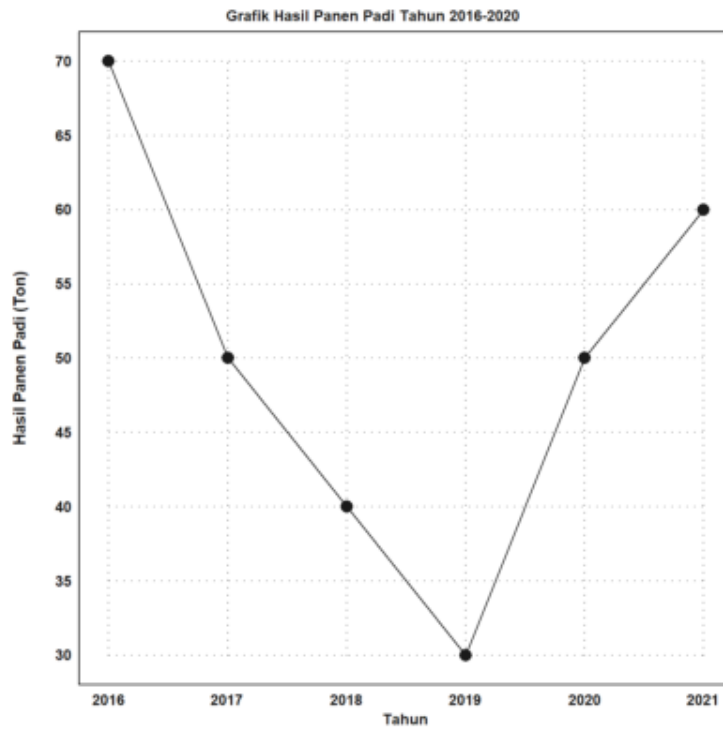
Contoh 3

Akan digambarkan diagram titik dan garis dari data hasil panen padi pada tahun 2016 sampai 2020.

```
>T=[2016,2017,2018,2019,2020,2021]; P=[70,50,40,30,50,60];
>writetable(T|P',labc=["Tahun","Hasil Panen Padi (Ton)"])
```

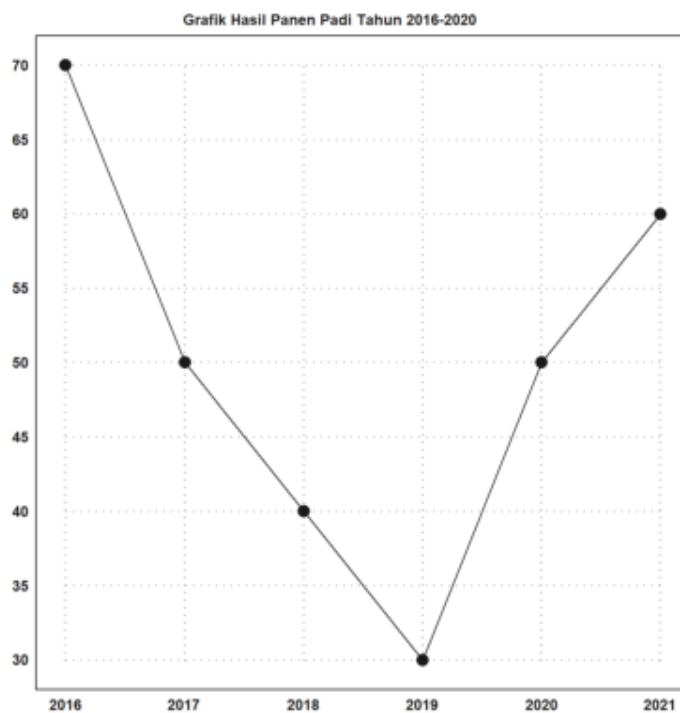
Tahun	Hasil Panen Padi (Ton)
2016	70
2017	50
2018	40
2019	30
2020	50
2021	60

```
>statplot(T,P,"b",pstyle="o#",lstyle="-",xl="Tahun",yl="Hasil Panen Padi (Ton)",vertical=5
>title("Grafik Hasil Panen Padi Tahun 2016-2020"):
```



Kita juga menggambar diagram titik dan garis secara bersama dengan menggunakan fungsi `plot2d()`, yaitu sebagai berikut.

```
>plot2d(T,P,2016,2020); plot2d(T,P,>points,style="o#",>add); title("Grafik Hasil Panen Pad
```



Dari grafik di atas, dapat dengan mudah kita ketahui bahwa panen padi paling banyak yaitu pada tahun 2016 (70 ton) dan paling sedikit yaitu pada tahun 2019 (30 ton).

Kurva regresi adalah representasi grafis dari model regresi yang digunakan untuk memodelkan hubungan antara satu atau lebih variabel independen (biasanya dilambangkan sebagai (X) dan variabel dependen (Y). Kurva regresi ini mencoba untuk menunjukkan pola atau tren dalam data dan memungkinkan kita untuk membuat prediksi atau estimasi berdasarkan model tersebut.

Secara umum, terdapat dua jenis kurva regresi yang umum digunakan yaitu regresi linier dan regresi non-linier.

Regresi Linier adalah garis lurus yang digunakan untuk memodelkan hubungan antara variabel independen (X) dan variabel dependen (Y).

Persamaan regresi linier umumnya ditulis sebagai

$$Y = mX + b$$

di mana m adalah kemiringan (slope) dan b adalah perpotongan sumbu-y (intercept).

Regresi linier dapat dilakukan dengan fungsi `polyfit()` atau berbagai fungsi fit.

Sebagai permulaan kita menemukan garis regresi untuk data univariat dengan `polyfit(x, y, 1)`.

Berikut contoh menggambar kurva regresi di EMT

Contoh 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

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

```
[0.733333, 0.812121]
```

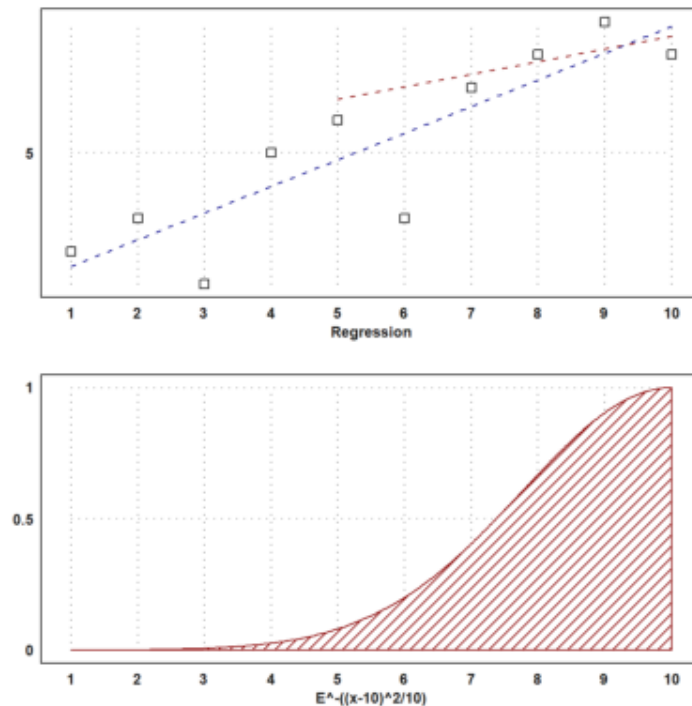
```
>w &= "exp(-(x-10)^2/10)"; pw=polyfit(x,y,1,w=w(x))
```

```
[4.71566, 0.38319]
```

```

>figure(2,1); ...
>figure(1); statplot(x,y,"p",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):

```



Kurva Fungsi Kerapatan Probabilitas

Fungsi kerapatan/kepadatan probabilitas adalah fungsi yang memberikan kemungkinan bahwa nilai suatu variabel acak akan berada di antara rentang nilai tertentu.

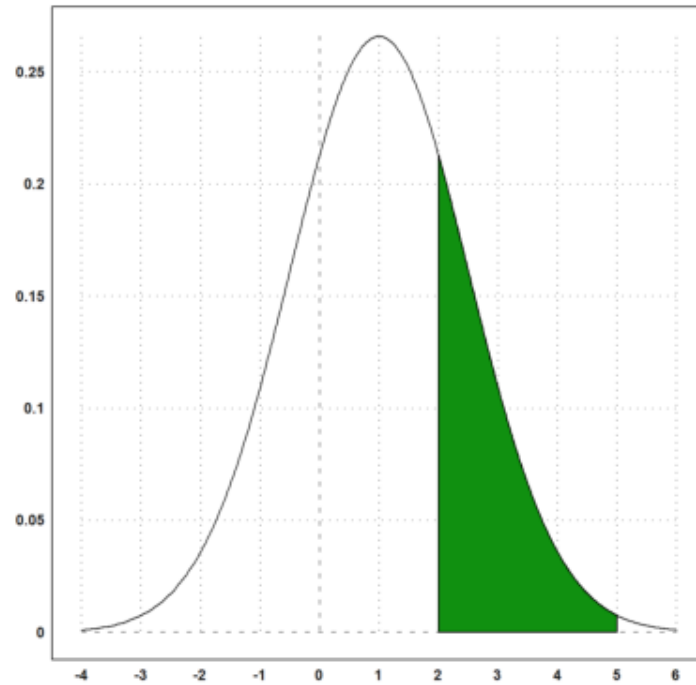
Grafik fungsi kepadatan probabilitas berbentuk kurva lonceng. Area yang terletak di antara dua nilai tertentu memberikan probabilitas hasil observasi yang ditentukan.

Berikut contoh kurva fungsi kepadatan probabilitas.

```

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

```



Kurva Distribusi Kumulatif

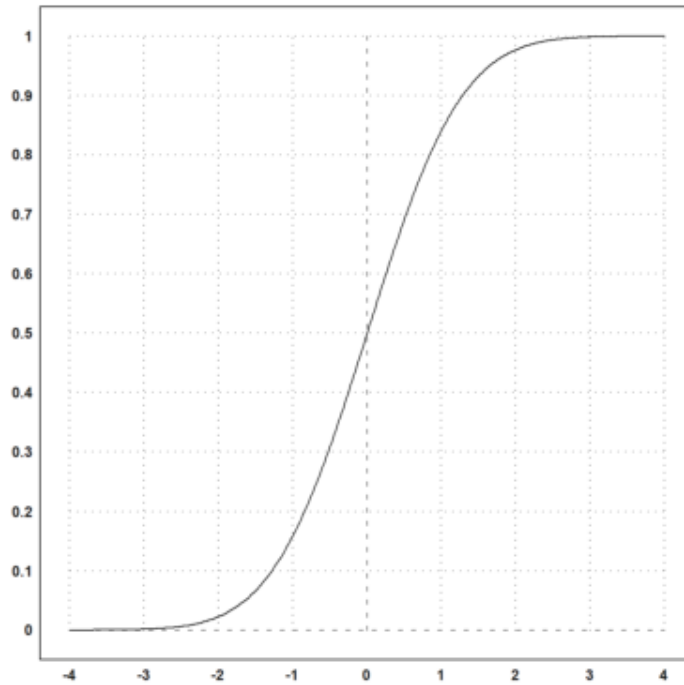
Kurva distribusi kumulatif (CDF) adalah representasi kumulatif dari fungsi distribusi probabilitas suatu variabel acak. CDF memberikan probabilitas bahwa variabel acak tersebut kurang dari atau sama dengan suatu nilai tertentu.

Seringkali, grafik CDF disajikan dalam bentuk kurva monotonik yang terus meningkat, dan ini memberikan gambaran visual yang baik tentang distribusi probabilitas variabel acak.

Berikut menggambar kurva distribusi kumulatif di EMT

Contoh 1

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



Dapat kita lihat dalam kurva CDF kontinu di atas dibagi menjadi 3 bagian, yaitu:

1. Bernilai 0 untuk x kurang dari batas bawah daerah rentang.
2. Merupakan fungsi monoton naik pada daerah rentang.
3. Bernilai konstan 1 untuk x lebih dari batas atas daerah rentangnya.

Contoh 2

Diberikan variabel acak dengan PDF sebagai berikut:

$$f(x) = \begin{cases} \frac{6}{5}(x^2 + x) & 0 \leq x \leq 1 \\ 0 & x \text{ yang lain.} \end{cases}$$

Untuk menggambar grafik CDF-nya, pertama kita cari terlebih dahulu CDF dari fungsi tersebut.

Untuk x pada interval

$$(-\infty, x)$$

$$F(x) = \int_{-\infty}^x f(x) dx$$

$$F(x) = \int_{-\infty}^x 0 dx$$

$$F(x) = 0$$

Untuk x pada interval $[0,1]$

$$F(x) = \int_{-\infty}^x f(x) dx$$

$$F(x) = \int_{-\infty}^0 f(x) dx + \int_0^x f(x) dx$$

$$F(x) = \int_{-\infty}^0 0 dx + \int_0^x \frac{6}{5}(x^2 + x) dx$$

$$F(x) = 0 + \frac{6}{5}\left(\frac{x^3}{3} + \frac{x^2}{2}\right)$$

$$F(x) = \frac{6}{5}\left(\frac{x^3}{3} + \frac{x^2}{2}\right)$$

Untuk x pada interval

$$(1, \infty)$$

$$F(x) = \int_{-\infty}^x f(x) dx$$

$$F(x) = \int_{-\infty}^0 f(x) dx + \int_0^1 f(x) dx + \int_1^x f(x) dx$$

$$F(x) = \int_{-\infty}^0 0 dx + \int_0^1 \frac{6}{5}(x^2 + x) dx + \int_1^{\infty} 0 dx$$

$$F(x) = 0 + \frac{6}{5}\left(\frac{1^3}{3} + \frac{1^2}{2}\right) + 0$$

$$F(x) = \frac{6}{5} \frac{5}{6} = 1$$

Sehingga, diperoleh CDF :

$$F(x) = \begin{cases} 0 & x < 0 \\ \frac{6}{5}\left(\frac{x^3}{3} + \frac{x^2}{2}\right) & 0 \leq x \leq 1 \\ 1 & x > 1. \end{cases}$$

Selanjutnya, akan kita gambarkan grafik CDF tersebut di EMT.

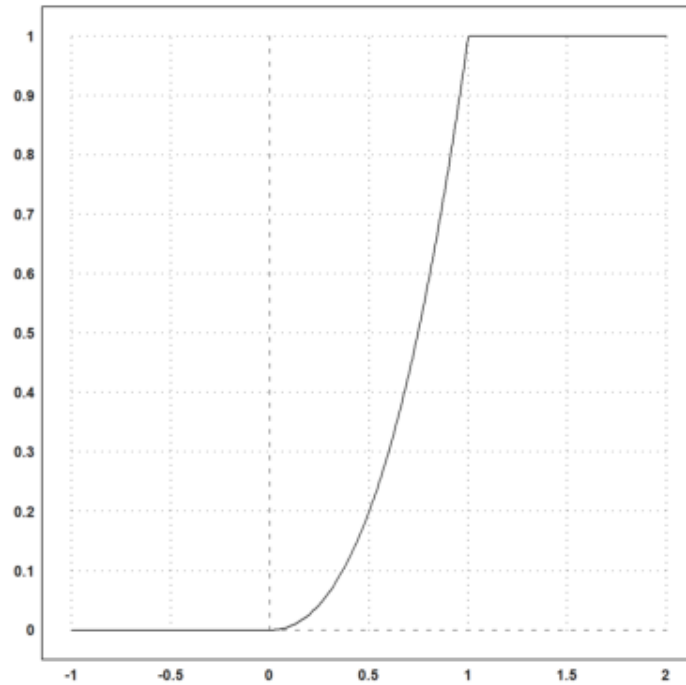
Pertama, kita definisikan terlebih dahulu fungsi f(x) dengan menggunakan fungsi function map.

```
>function map f(x) ...
```

```
    if x<0 then return 0
    else if x>=0 and x<=1 then return (6/5)*((x^3/3)+(x^2/2))
    else return 1
    endif;
endfunction
```

Kemudian, kita akan menggambar grafik fungsi di atas dengan menggunakan fungsi plot2d() pada interval x dari -1 sampai 2.

```
>plot2d ("f(x)", -1, 2):
```

Plot diatas merupakan diagram kotak, Kotak yang berwarna hijau tersebut merupakan hasil dari perintah EMT yang diberikan dimana kotak tersebut dibatasi dengan Kuartil terendah dan kuartil tertinggi, dan terdapat garis hitam di dalam kotak merupakan Median atau nilai tengah.

image: BoxPlot.png

Di bawah ini diperlihatkan rincian detail boxplot beserta cara penentuan batas-batasnya.

Bagian utama boxplot adalah kotak berbentuk persegi (Box) yang

merupakan bidang yang menyajikan interquartile range (IQR), dimana 50 % dari nilai data pengamatan terletak di sana.

- Panjang kotak sesuai dengan jangkauan kuartil dalam (inner

Quartile Range, IQR) yang merupakan selisih antara Kuartil ketiga (Q3) dengan Kuartil pertama (Q1). IQR menggambarkan ukuran penyebaran data. Semakin panjang bidang IQR menunjukkan data semakin menyebar. Pada Gambar, $IQR = UQ - LQ = Q3 - Q1$

- Garis bawah kotak (LQ) = Q1 (Kuartil pertama), dimana 25% data

pengamatan lebih kecil atau sama dengan nilai Q1

- Garis tengah kotak = Q2 (median), dimana 50% data pengamatan

lebih kecil atau sama dengan nilai ini

- Garis atas kotak (UQ) = Q3 (Kuartil ketiga) dimana 75% data

pengamatan lebih kecil atau sama dengan nilai Q1

Garis yang merupakan perpanjangan dari box(baik ke arah atas

ataupun ke arah bawah) dinamakan dengan whiskers.

- Whiskers bawah menunjukkan nilai yang lebih rendah dari kumpulan

data yang berada dalam IQR

- Whiskers atas menunjukkan nilai yang lebih tinggi dari kumpulan

data yang berada dalam IQR

- Panjang whisker = $1.5 \times \text{IQR}$. Masing-masing garis whisker dimulai

dari ujung kotak IQR, dan berakhir pada nilai data yang bukan dikategorikan sebagai outlier (Pada gambar, batasnya adalah garis UIF dan LIF). Dengan demikian, nilai terbesar dan terkecil dari data pengamatan (tanpa termasuk outlier) masih merupakan bagian dari Boxplot yang terletak tepat di ujung garis tepi whiskers.

Nilai yang berada di atas atau dibawah whisker dinamakan nilai

outlier atau ekstrim.

- Nilai outlier adalah nilai data yang letaknya lebih dari $1.5 \times$

panjang kotak (IQR), diukur dari UQ (atas kotak) atau LQ (bawah kotak). Pada Gambar di atas, ada 2 data pengamatan yang merupakan outlier, yaitu data pada case 33 dan case 55 (ada pada baris ke 33 dan baris 35)

=> $Q3 + (1.5 \times \text{IQR}) < \text{outlier atas} = Q3 + (3 \times \text{IQR})$

=> $Q1 - (1.5 \times \text{IQR}) > \text{outlier bawah} = Q1 - (3 \times \text{IQR})$

- Nilai ekstrim adalah nilai-nilai yang letaknya lebih dari $3 \times$

panjang kotak (IQR), diukur dari UQ (atas kotak) atau LQ (bawah kotak). Pada gambar di atas, ada 1 data yang merupakan nilai ekstem, yaitu data pada case 15.

=> Ekstrim bagian atas apabila nilainya berada di atas $Q3 + (3$

$\times \text{IQR})$ dan

=> Ekstrim bagian bawah apabila nilainya lebih rendah dari $Q1 -$

$(3 \times \text{IQR})$

Berikut penggunaan boxplot pada distribusi normal: