

Nama : Siti Nurkhofifah ALiyah
NIM : 23030130017
Kelas : Pendidikan Matematika A 2023

EMT untuk Statistika

Dalam buku catatan ini, kami mendemonstrasikan plot statistik utama, pengujian, dan distribusi di Euler.

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

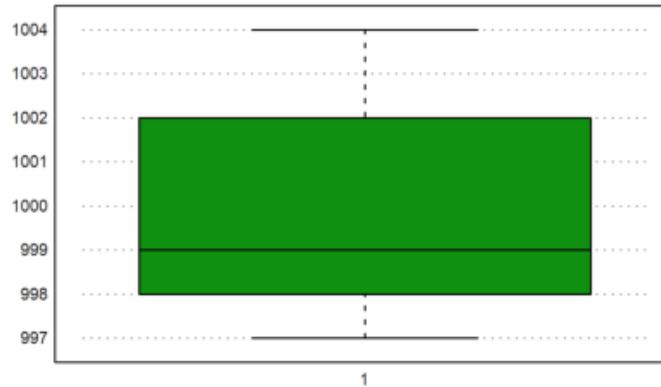
Asumsikan pengukuran berikut. Kami ingin menghitung nilai rata-rata dan standar deviasi yang diukur.

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

```
999  
999.9  
2.72641400622
```

Kita dapat memplot plot kotak-dan-kumis untuk data. Dalam kasus kami tidak ada outlier.

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



Kami menghitung probabilitas bahwa suatu nilai lebih besar dari 1005, dengan asumsi nilai terukur dan distribusi normal.

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

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

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

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

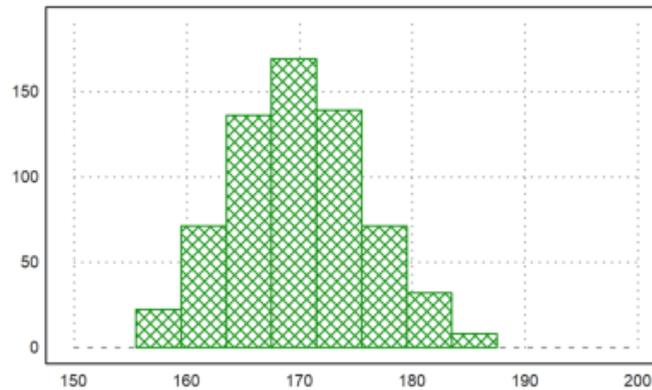
3.07 %

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

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

Berikut adalah plot distribusinya.

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



Kita bisa memasukkan data mentah tersebut ke dalam sebuah tabel.

Tabel adalah metode untuk menyimpan data statistik. Tabel kita harus berisi tiga kolom: Awal jangkauan, akhir jangkauan, jumlah orang dalam jangkauan.

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.

Sumbul "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan opsi "labc" adalah untuk menentukan header kolom.

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

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

Tetapi 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 dapat menghitung mean dan deviasi sampel dengan frekuensi yang diberikan.

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

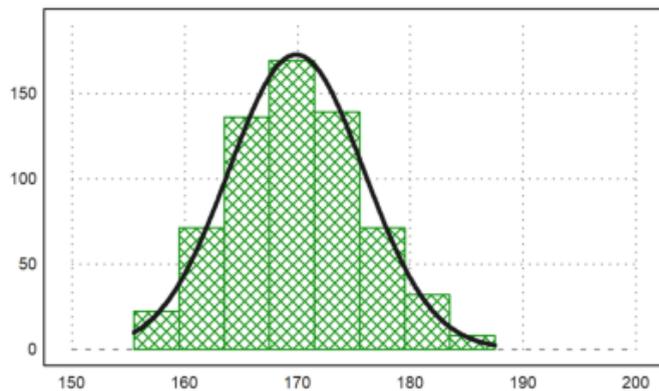
```
169.901234568
5.98912964449
```

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

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

Karena nilainya antara 0 dan 1, untuk memplotnya pada bar plot harus dikalikan dengan 4 kali jumlah total data.

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



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

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

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

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

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

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

Sekarang kita membaca tabel dengan terjemahan ini.

Argumen tok2, tok4 dll. adalah terjemahan dari kolom tabel. 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[1:4],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

Titik "." mewakili nilai-nilai, yang tidak tersedia.

Jika kita tidak ingin menentukan token untuk terjemahan terlebih dahulu, 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 isi tabel dengan angka yang belum diterjemahkan.

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

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

Untuk kenyamanan, Anda dapat memasukkan output `readtable()` ke dalam daftar.

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

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 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],lab=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 dalam vektor, dan jumlahnya. Kami menerapkannya pada nilai "m" dan "f" di kolom kedua tabel kami.

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

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

Kami dapat mencetak hasilnya dalam tabel baru.

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

```
      m      12  
      f      13
```

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

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

```
[5, 6]
```

Sekarang kita dapat memilih baris 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

Dengan `getstatistics()`, kita juga bisa 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 file.

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

Kemudian kita bisa membaca tabel dari file tersebut.

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

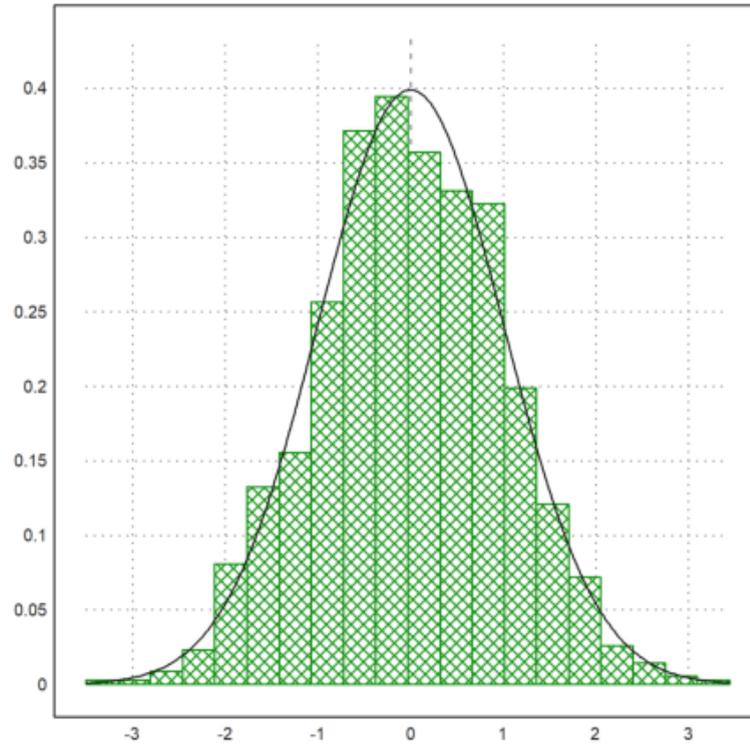
	n	y
m	7	5
f	1	12

Dan hapus file tersebut.

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

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

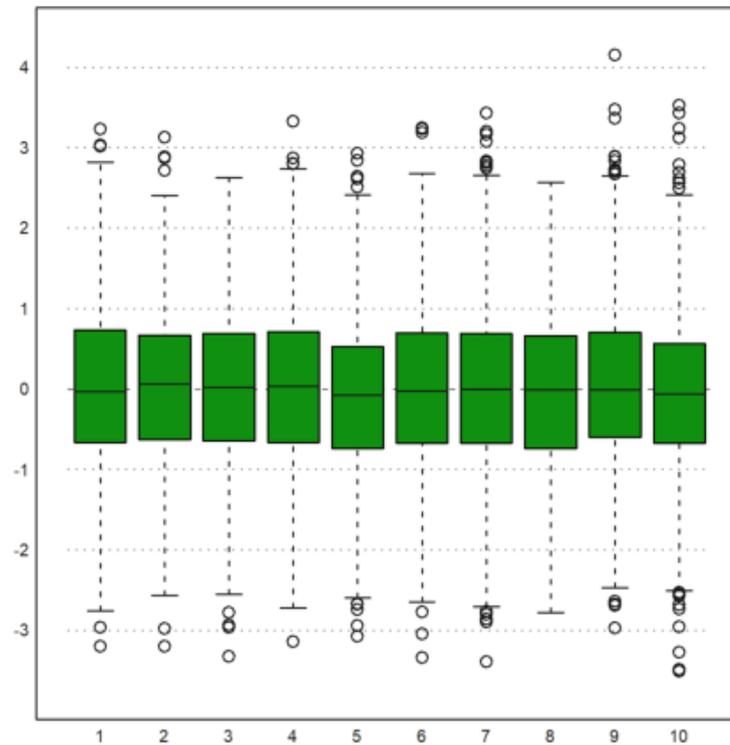
```
>p=normal(1,1000); //1000 random normal-distributed sample p
>plot2d(p,distribution=20,style="\"); // plot the random sample p
>plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution plot
```



Harap perhatikan perbedaan antara plot batang (sampel) dan kurva normal (distribusi nyata). Masukkan kembali tiga perintah untuk melihat hasil pengambilan sampel lainnya.

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

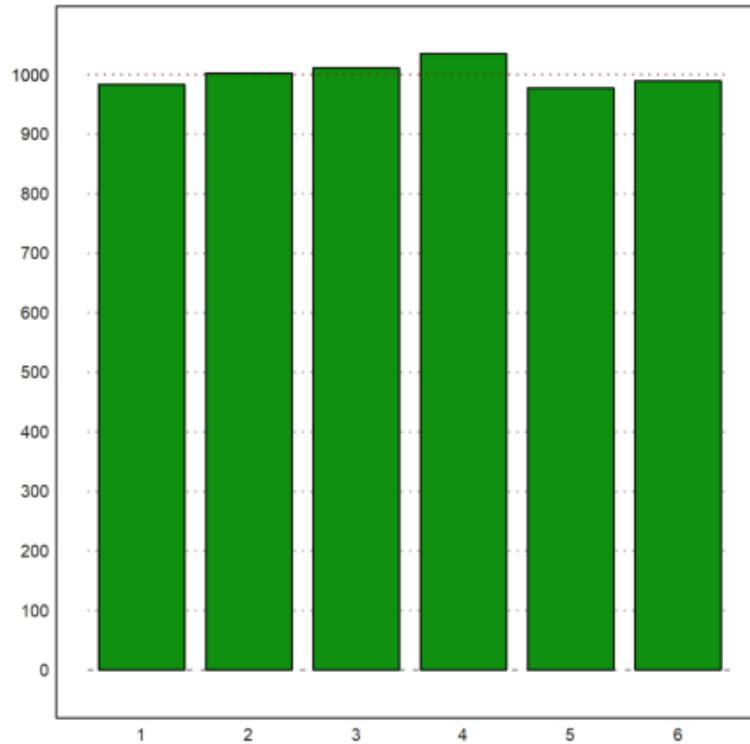
```
>p=normal(10,1000); boxplot(p):
```



Untuk membangkitkan bilangan bulat acak, Euler memiliki `intrarandom`. Mari kita simulasikan lemparan dadu dan plot distribusinya.

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

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



Sementara `inrandom(n,m,k)` mengembalikan bilangan bulat yang terdistribusi secara seragam dari 1 ke k, dimungkinkan untuk menggunakan distribusi bilangan bulat lain yang diberikan dengan `randpint()`.

Dalam 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]

Euler dapat menghasilkan nilai acak dari lebih banyak distribusi. Coba lihat referensinya.

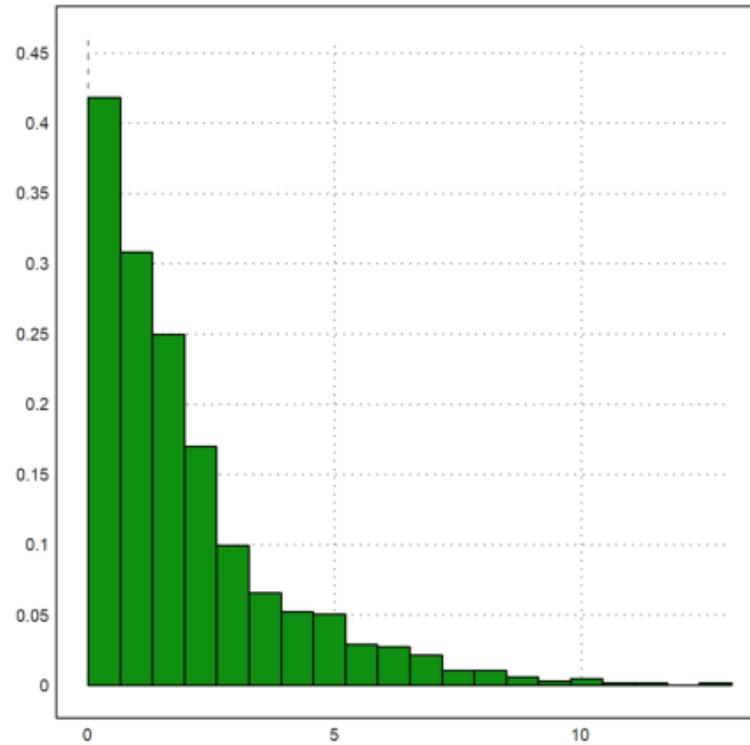
Misalnya, kami mencoba distribusi eksponensial. Variabel acak kontinu X dikatakan memiliki distribusi eksponensial, jika PDF-nya diberikan oleh

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

dengan parameter

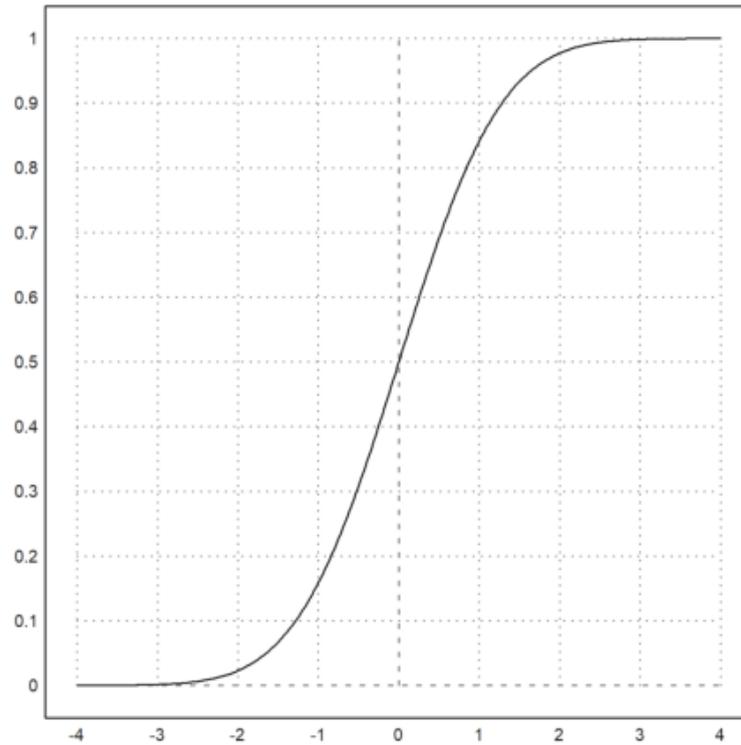
$$\lambda = \frac{1}{\mu}, \quad \mu \text{ adalah rata-rata, dan dilambangkan dengan } X \sim \text{Eksponensial}(\lambda).$$

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



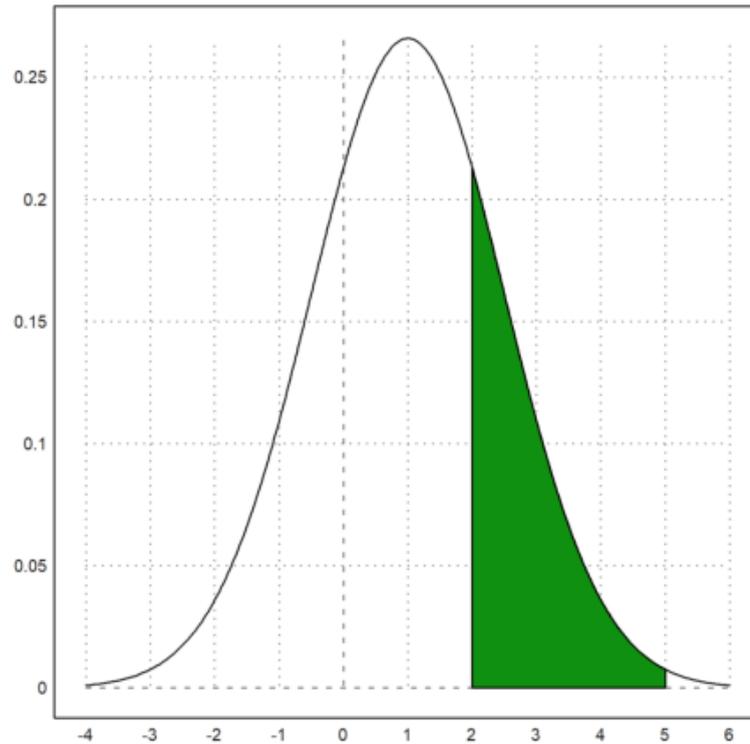
Untuk banyak distribusi, Euler dapat menghitung fungsi distribusi dan inversnya.

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



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)
```

0.248662156979

Ini dapat dihitung secara numerik dengan integral berikut.

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

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

0.248662156979

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

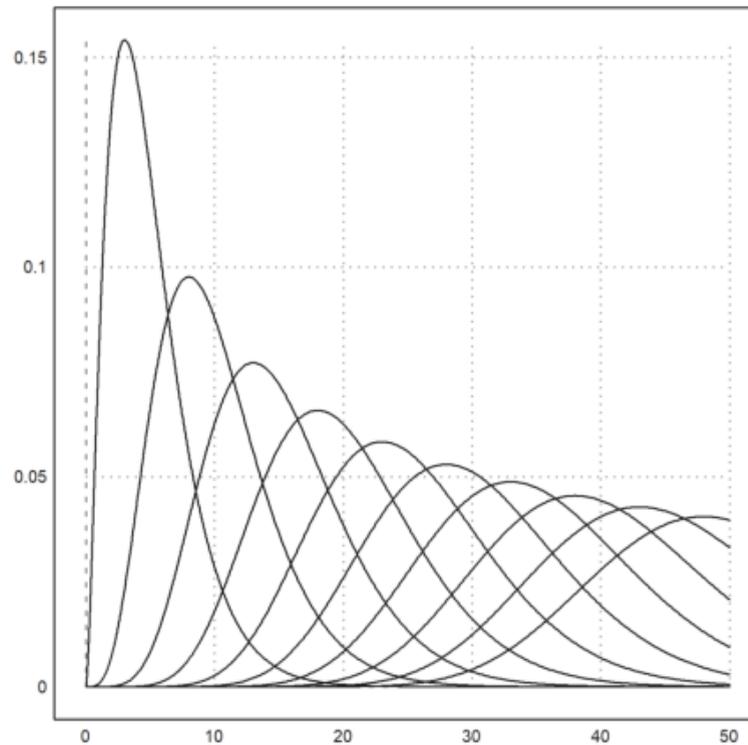
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

525.516721219

526.007419394

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 `chidis()`.

```
>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 mengatur fungsi distribusi.

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

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

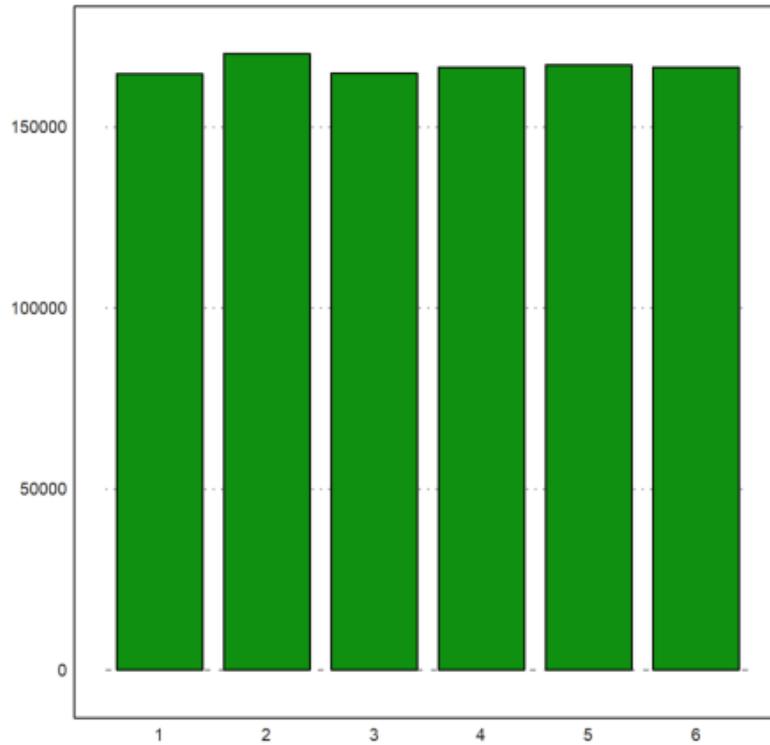
Artinya dengan probabilitas $wd[i+1]-wd[i]$ kita menghasilkan nilai acak i .

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

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

Kesalahannya sangat halus sehingga kami melihatnya hanya dengan iterasi yang sangat banyak.

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



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

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

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

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

Memang fungsi menolak distribusi seragam.

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

0

Dan itu menerima generator acak bawaan.

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

1

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

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

```
0.751401349654
```

Fungsi Beta terbalik digunakan untuk menghitung interval kepercayaan Clopper-Pearson untuk parameter p . Level default adalah alfa.

Arti interval ini adalah jika p berada di luar interval, hasil pengamatan 410 dalam 1000 jarang terjadi.

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

```
[0.37932, 0.441212]
```

Perintah berikut adalah cara langsung untuk mendapatkan hasil di atas. Tapi untuk n besar, penjumlahan langsungnya tidak akurat dan lambat.

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

```
0.751401349655
```

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

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

```
409.932733047
```

Di Bridge, kami mengasumsikan 5 kartu beredar (dari 52) dengan dua tangan (26 kartu). Mari kita hitung probabilitas distribusi yang lebih buruk dari 3:2 (mis. 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 pemilu Jerman sejak 1990, yang diukur dalam jumlah kursi.

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

Untuk partai, kami menggunakan rangkaian nama.

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

Mari kita cetak persentasenya dengan baik.

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

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

Kemudian kami mencetak statistik dalam bentuk tabel. Kami menggunakan nama sebagai tajuk kolom, dan tahun sebagai tajuk untuk baris. Lebar default untuk kolom adalah $wc=10$, tetapi kami lebih memilih hasil yang lebih padat. Kolom akan diperluas untuk 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

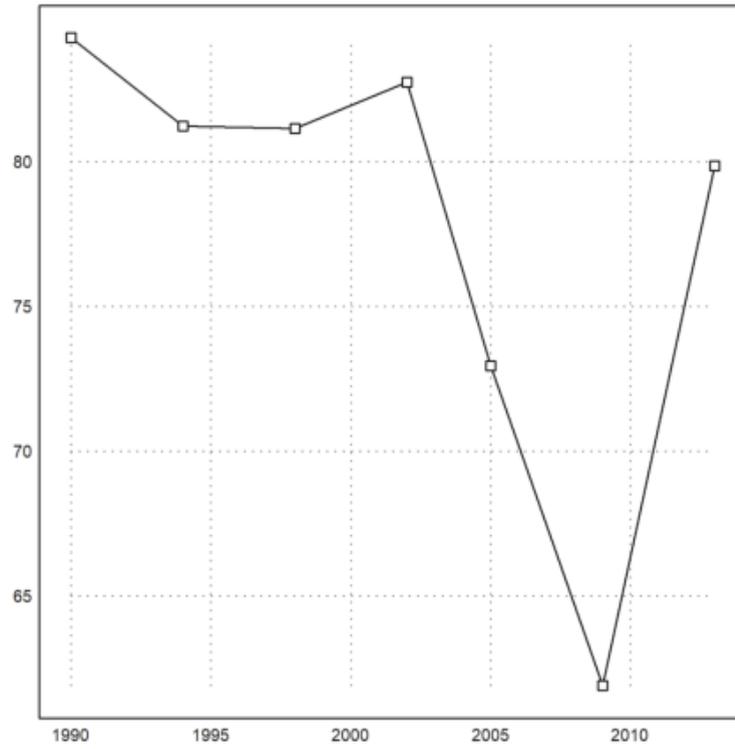
Perkalian matriks berikut mengekstrak jumlah persentase dari dua partai besar yang menunjukkan bahwa partai-partai kecil telah mendapatkan rekaman di parlemen hingga tahun 2009.

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

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

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

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

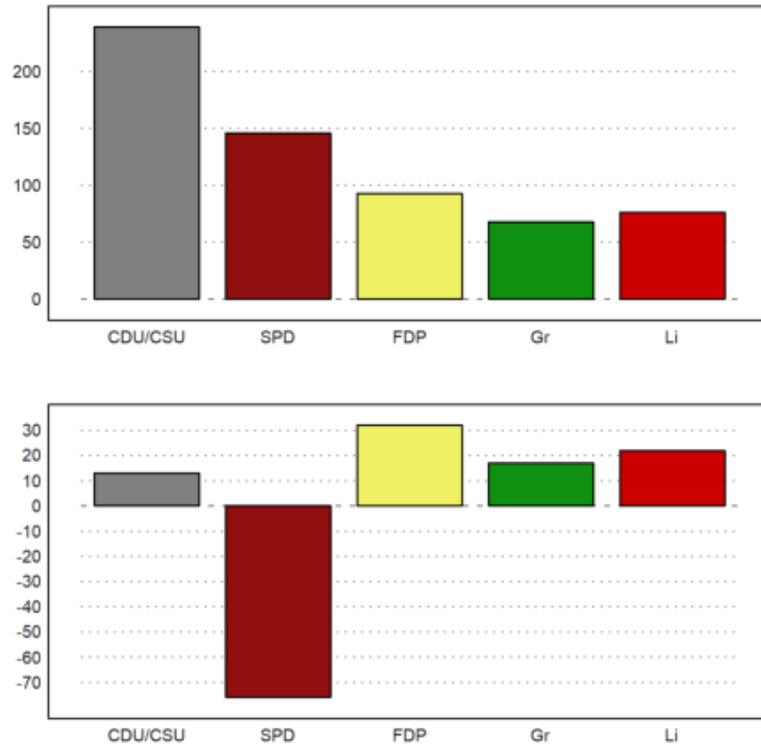


Tentukan beberapa warna untuk masing-masing pihak.

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

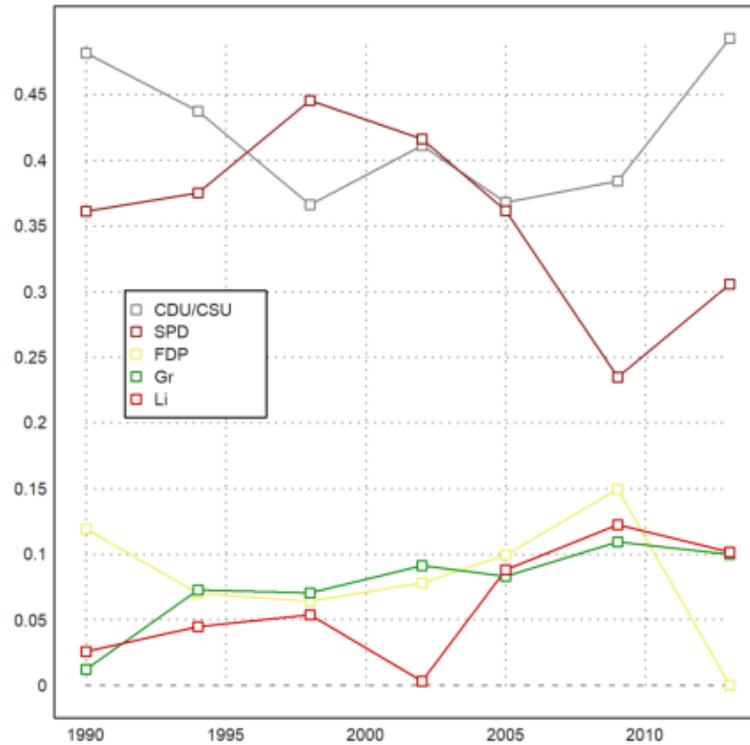
Sekarang kita bisa memplot hasil pemilu 2009 dan mengubahnya menjadi satu plot menggunakan gambar. Kita dapat menambahkan vektor kolom ke setiap plot.

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



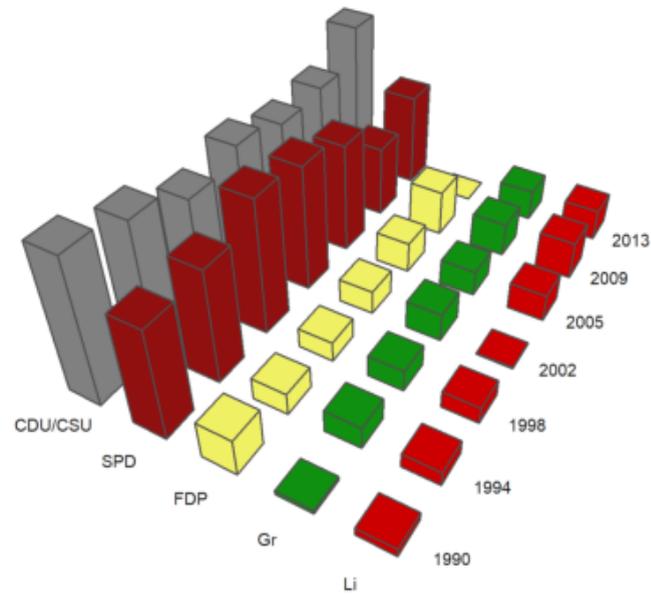
Plot data menggabungkan deretan data statistik dalam satu plot.

```
>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):
```



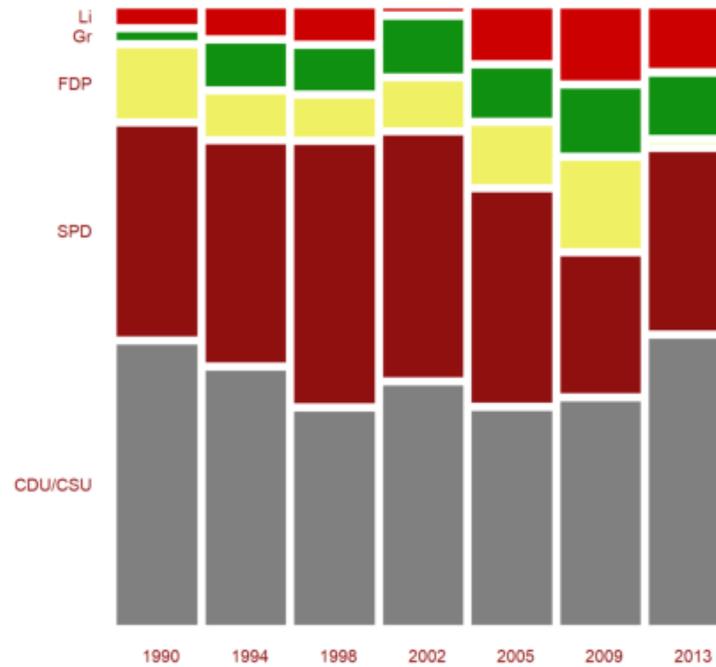
Plot kolom 3D menunjukkan deretan data statistik dalam bentuk kolom. Kami menyediakan label untuk baris dan kolom. sudut adalah sudut pandang.

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



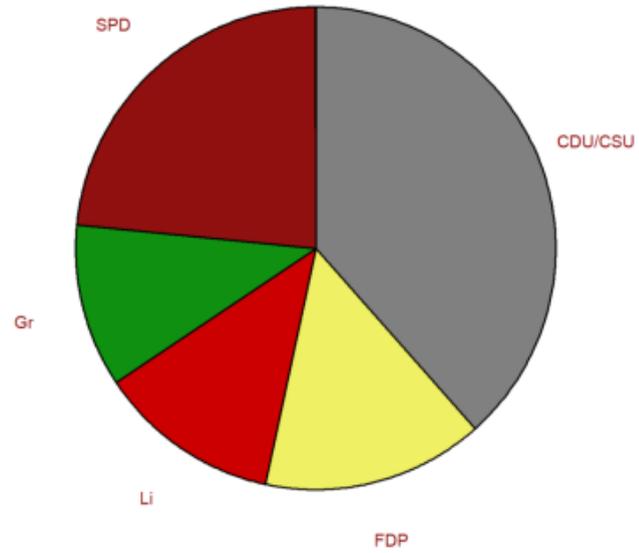
Representasi lainnya adalah plot mozaik. Perhatikan bahwa kolom plot mewakili kolom matriks di sini. Karena panjang label CDU/CSU, kami mengambil jendela yang lebih kecil dari biasanya.

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



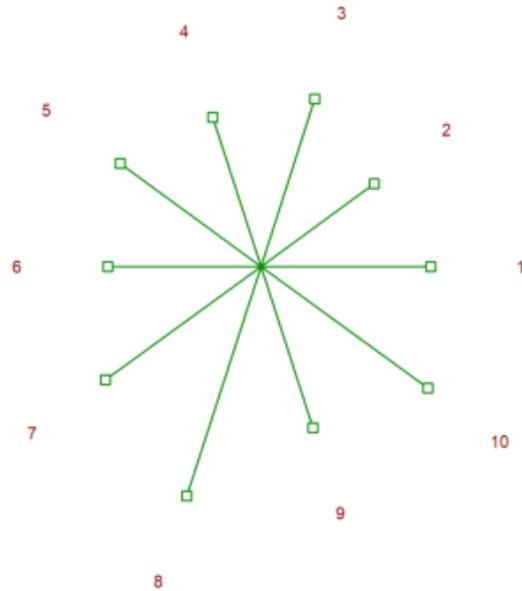
Kita juga bisa membuat diagram lingkaran. Karena hitam dan kuning membentuk koalisi, kami menyusun ulang elemennya.

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



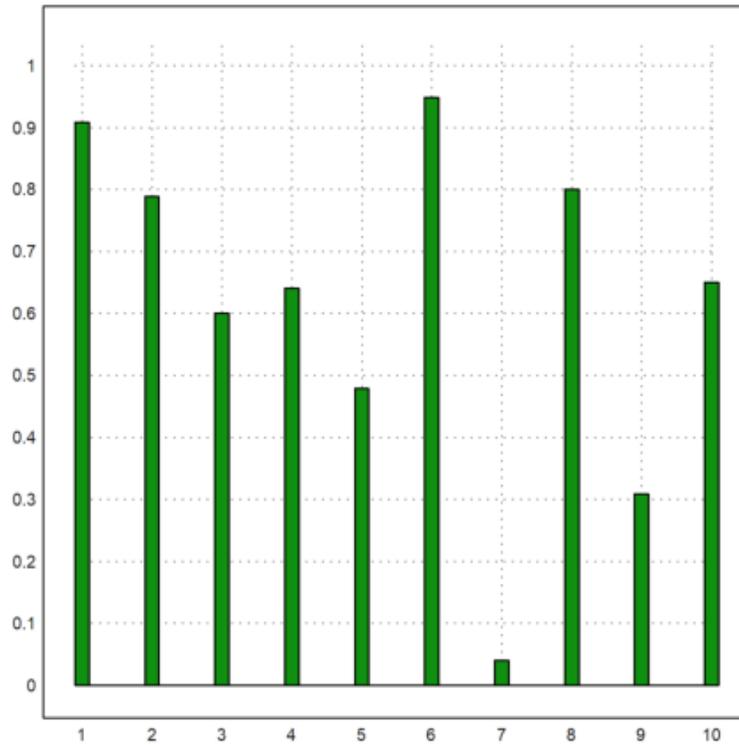
Ini jenis plot lainnya.

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



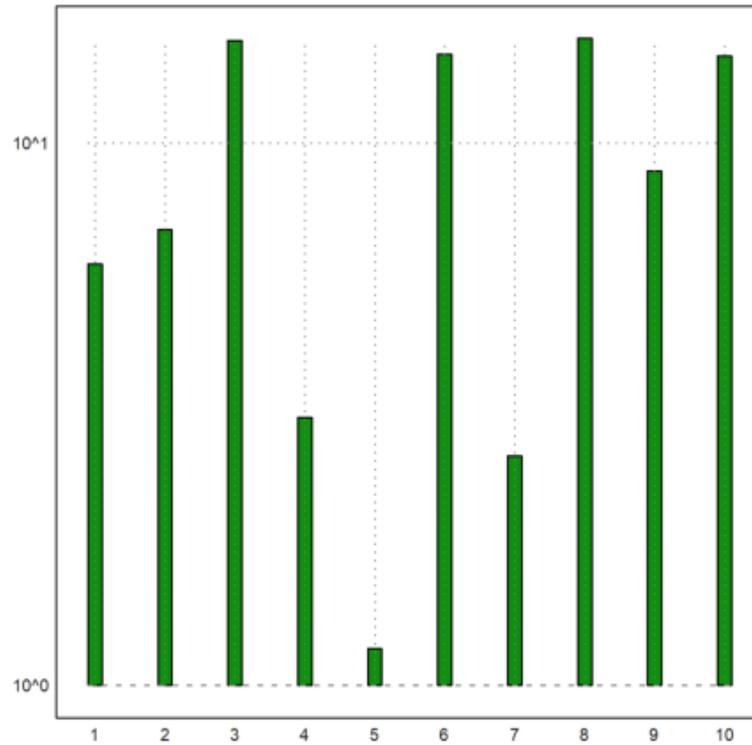
Beberapa plot di plot2d bagus untuk statika. Berikut adalah plot impuls dari data acak, terdistribusi secara seragam di $[0,1]$.

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



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

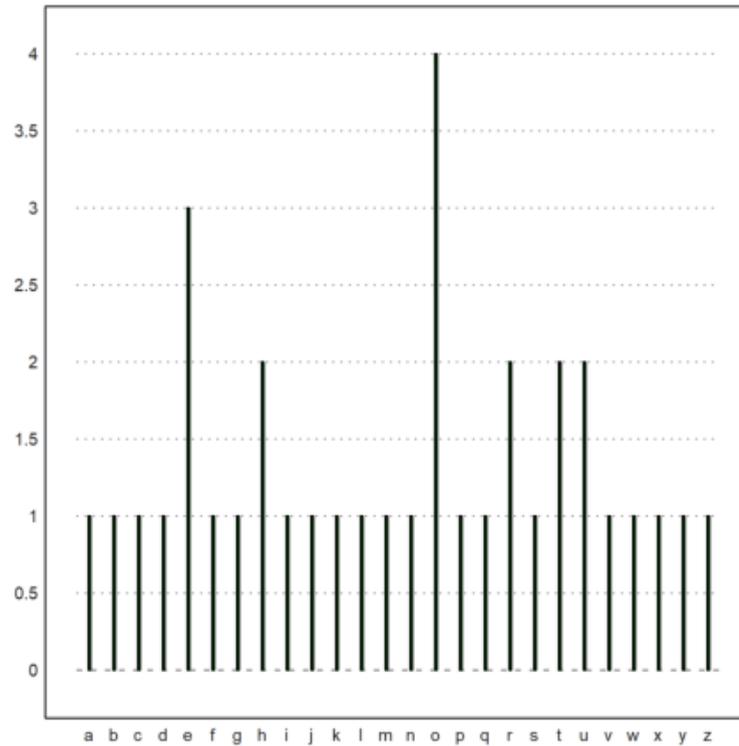
```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi `columnplot()` lebih mudah digunakan, karena hanya membutuhkan vektor nilai. Selain itu, ia dapat mengatur labelnya ke apa pun yang kita inginkan, kita telah mendemonstrasikannya di tutorial ini.

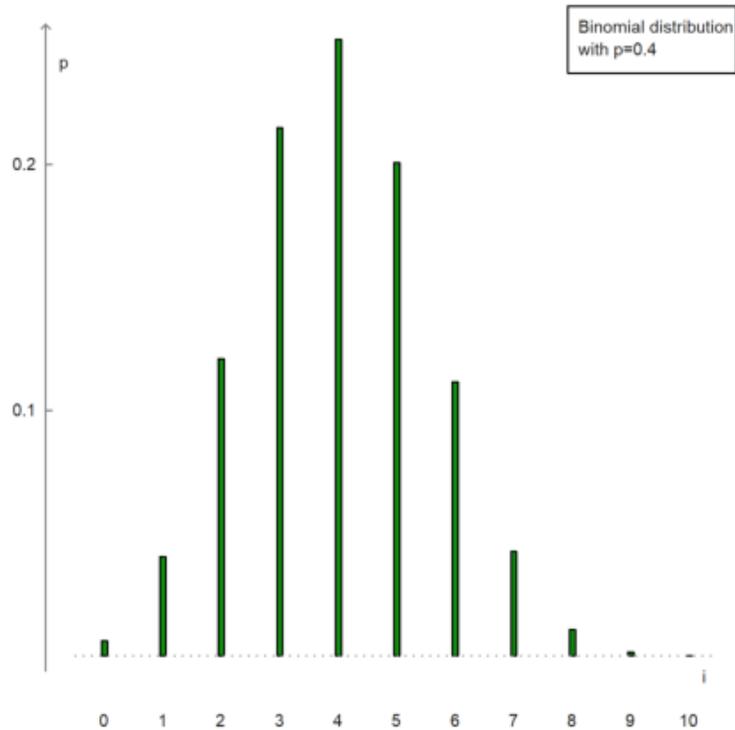
Ini adalah aplikasi lain, di mana kami menghitung karakter dalam sebuah kalimat dan memplot statistik.

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



Dimungkinkan juga untuk mengatur 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 memplot frekuensi bilangan dalam vektor.
Kami membuat vektor bilangan acak bilangan bulat 1 hingga 6.

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

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

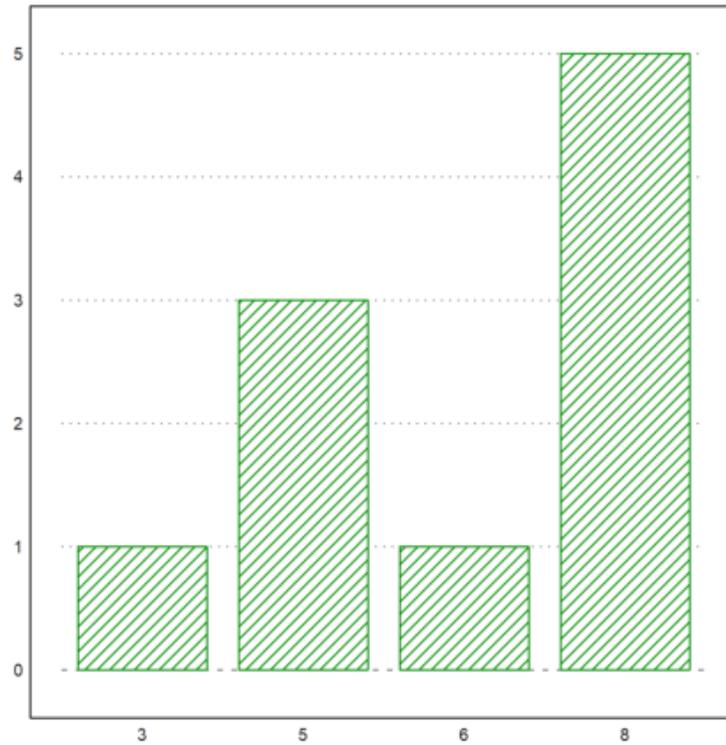
Kemudian ekstrak angka unik di v.

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

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

Dan plot frekuensi dalam plot kolom.

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



Kami ingin menunjukkan fungsi untuk distribusi nilai empiris.

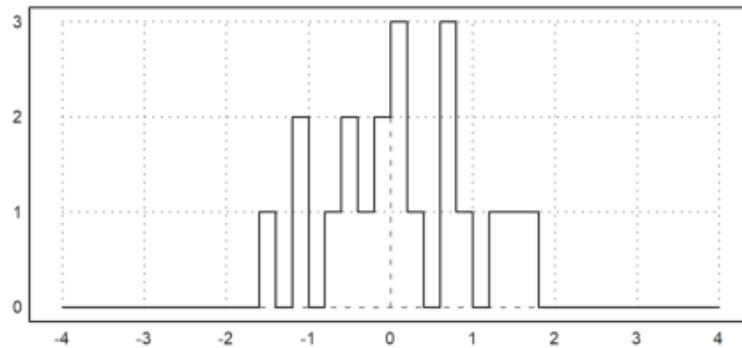
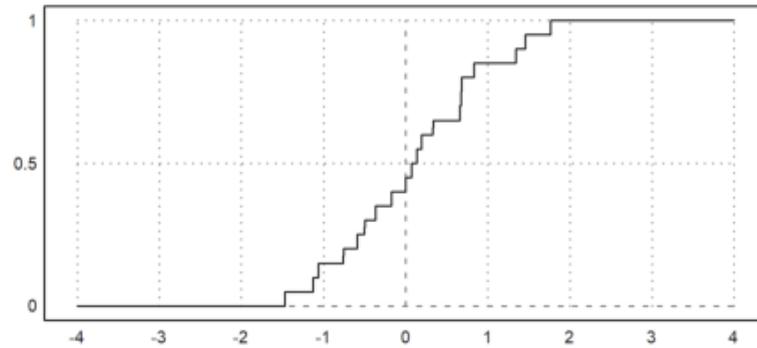
```
>x=normal(1,20);
```

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

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

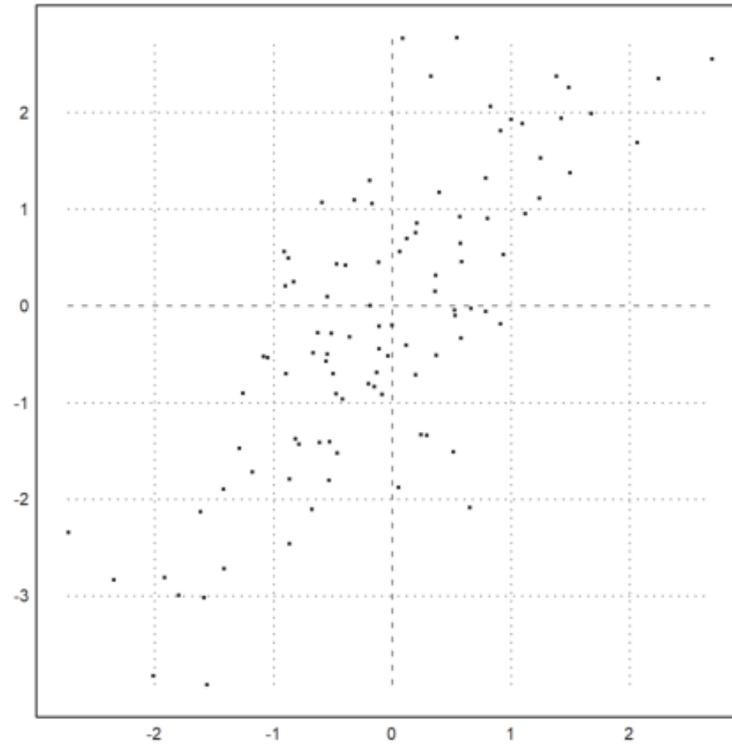
Kemudian kami memplot distribusi empiris dan beberapa batang kepadatan menjadi satu plot. Alih-alih plot batang untuk distribusi, kali ini kami menggunakan plot gigi gergaji.

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



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

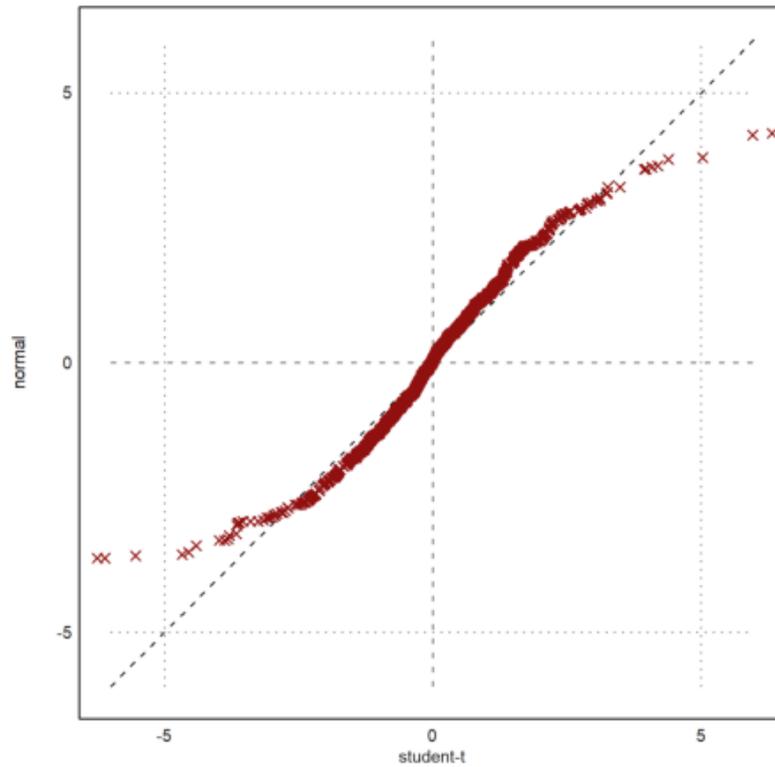
```
>x=normal(1,100); plot2d(x,x+rotright(x),>points,style=".."):
```



Seringkali, kami ingin membandingkan dua sampel dari distribusi yang berbeda. 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):
```



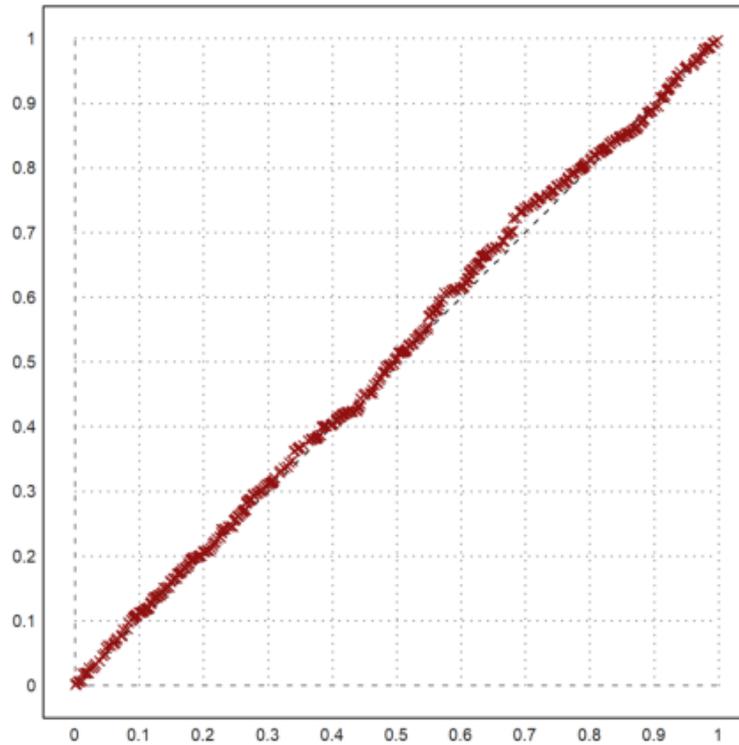
Plot jelas menunjukkan bahwa nilai terdistribusi normal cenderung lebih kecil di ujung ekstrim.

Jika kita memiliki dua distribusi dengan ukuran berbeda, kita dapat memperluas yang lebih kecil atau mengecilkan yang lebih besar. Fungsi berikut ini baik untuk keduanya. Dibutuhkan 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 fit.

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

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

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

Kami ingin membandingkan kecocokan yang tidak berbobot dan berbobot. Pertama koefisien fit linier.

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

```
[0.733333, 0.812121]
```

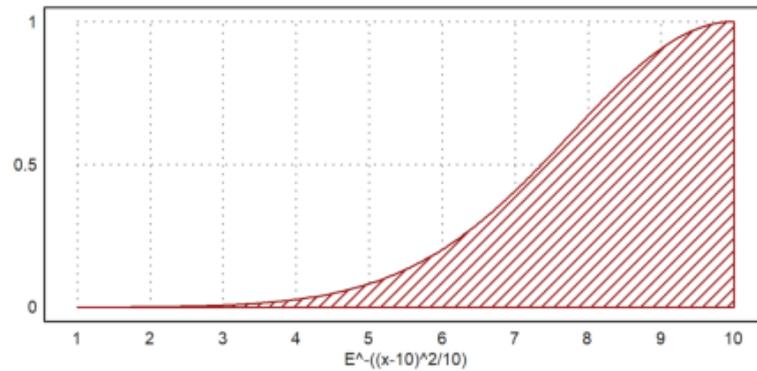
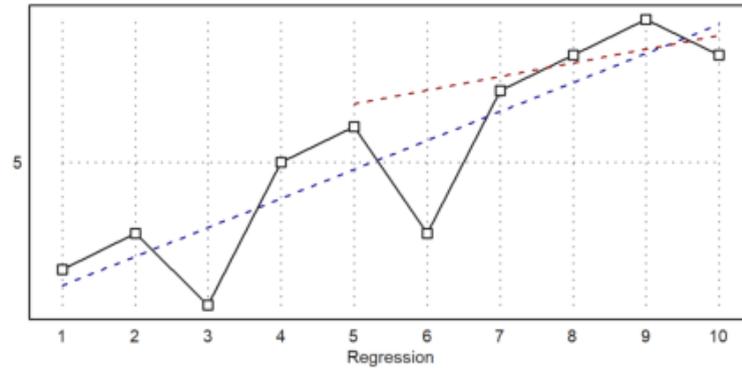
Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

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

```
[4.71566, 0.38319]
```

Kami memasukkan semuanya ke dalam satu plot untuk poin 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):
```



Sebagai contoh lain kita membaca survei siswa, umur mereka, umur orang tua mereka dan jumlah saudara dari sebuah file.

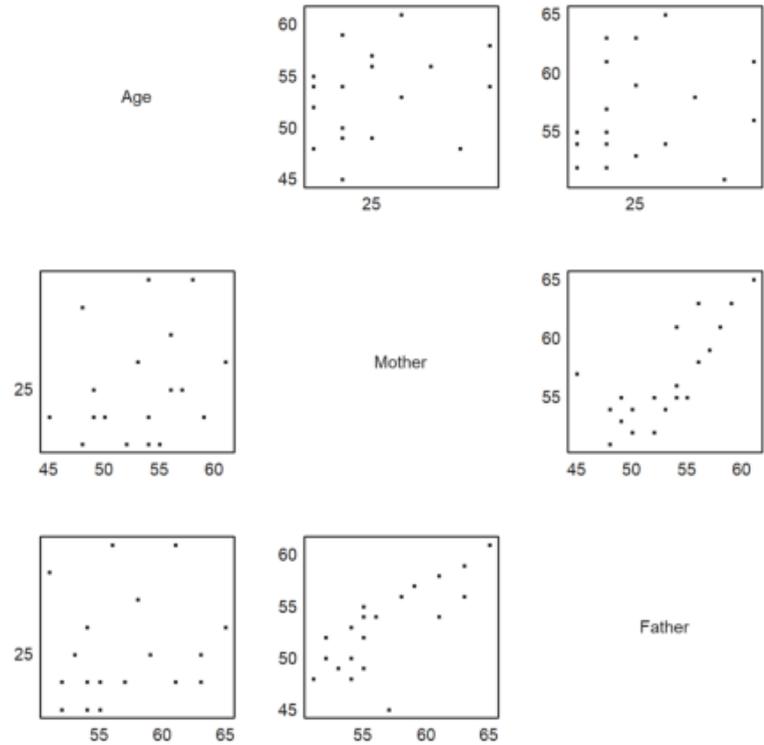
Tabel ini berisi "m" dan "f" di kolom kedua. Kami menggunakan variabel tok2 untuk menyetel terjemahan yang tepat alih-alih membiarkan readtable() mengumpulkan terjemahan.

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

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

Bagaimana usia bergantung satu sama lain? Kesan pertama berasal dari sebar berpasangan.

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



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

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

[17.3789, 0.740964]

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

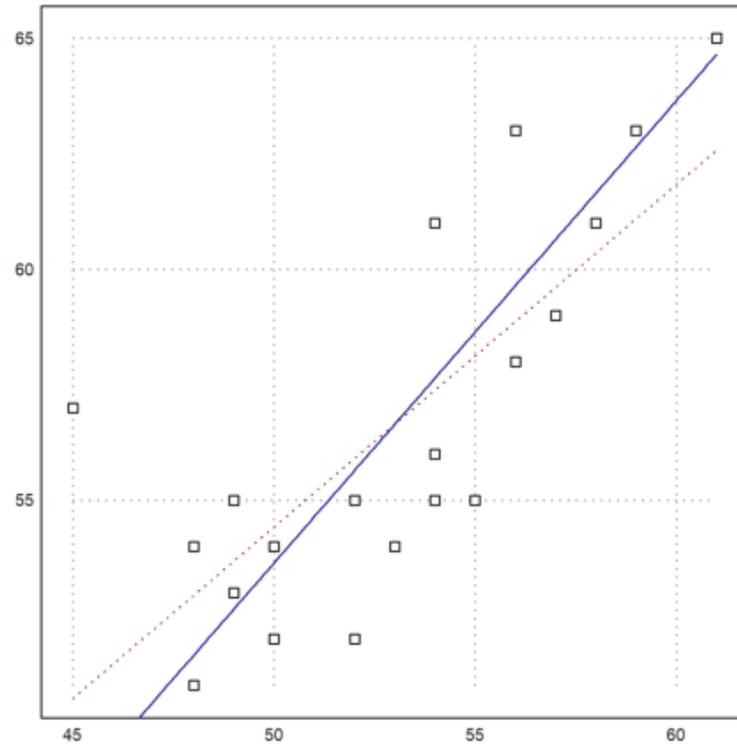
Sebaliknya, kami mencurigai fungsi seperti $s=a+t$. Maka a adalah rata-rata dari $s-t$. Ini adalah perbedaan usia rata-rata antara ayah dan ibu.

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

3.65

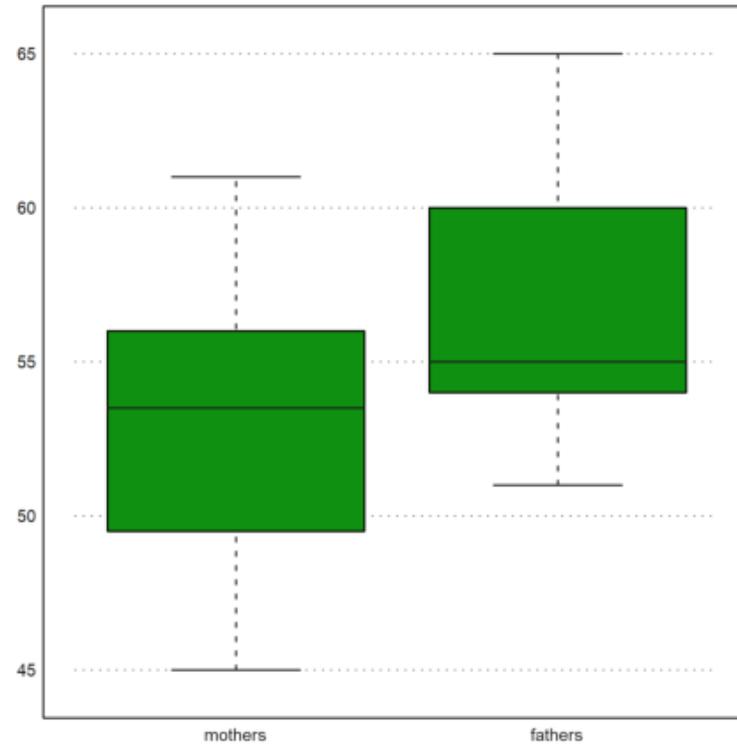
Mari kita plot ini menjadi satu plot pencar.

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



Berikut adalah plot kotak dari dua zaman. Ini hanya menunjukkan, bahwa usianya berbeda.

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



Menariknya, perbedaan median tidak sebesar perbedaan rata-rata.

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

Koefisien korelasi menunjukkan korelasi positif.

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

```
0.7588307236
```

Korelasi peringkat adalah ukuran untuk urutan yang sama di kedua vektor. Ini juga cukup positif.

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

```
0.758925292358
```

Membuat Fungsi baru

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

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

di mana m adalah rata-rata dari x .

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

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

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

```
-0.198710316203
```

Ini adalah fungsi lain, yang disebut koefisien kemiringan Pearson.

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

-0.0801873249135

Simulasi Monte Carlo

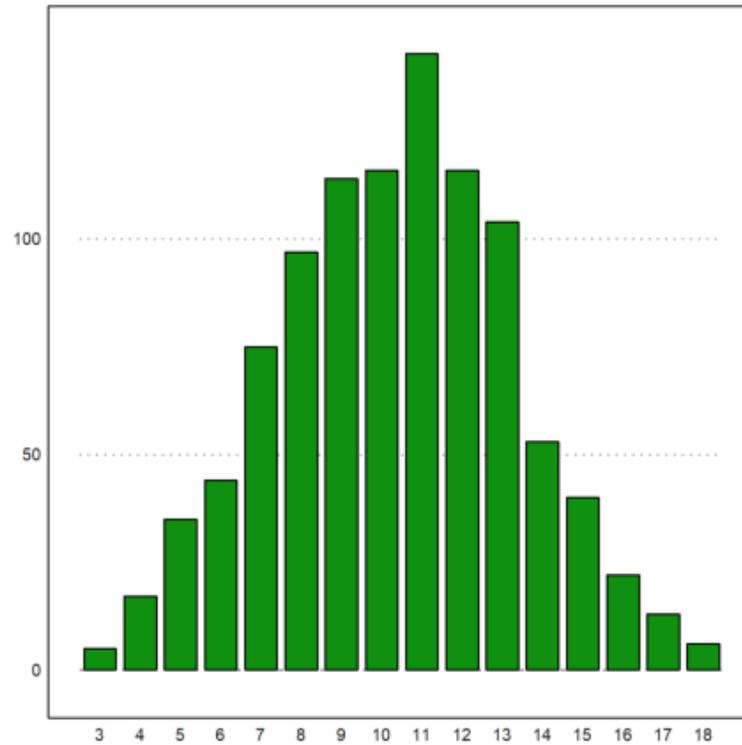
Euler dapat digunakan untuk mensimulasikan kejadian acak. Kita telah melihat contoh sederhana di atas. Ini satu lagi, yang mensimulasikan 1000 kali 3 lemparan dadu, dan meminta distribusi jumlahnya.

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

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

Kita bisa merencanakan ini sekarang.

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



Untuk menentukan distribusi yang diharapkan tidak begitu mudah. Kami menggunakan rekursi lanjutan untuk ini.

Fungsi berikut menghitung banyaknya cara bilangan k dapat dinyatakan sebagai jumlah dari n bilangan dalam rentang 1 sampai m . Ini bekerja secara rekursif dengan cara yang jelas.

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

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

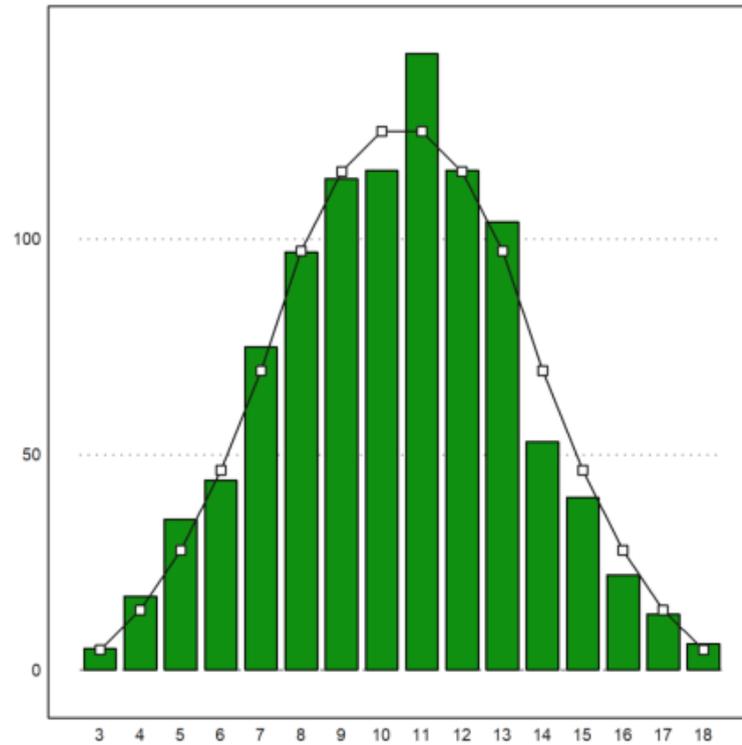
Inilah hasil lemparan dadu sebanyak tiga kali.

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

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

Kami menambahkan nilai yang diharapkan ke plot.

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



Untuk simulasi lain, penyimpangan nilai rata-rata n 0-1-variabel acak terdistribusi normal adalah $1/\sqrt{n}$.

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

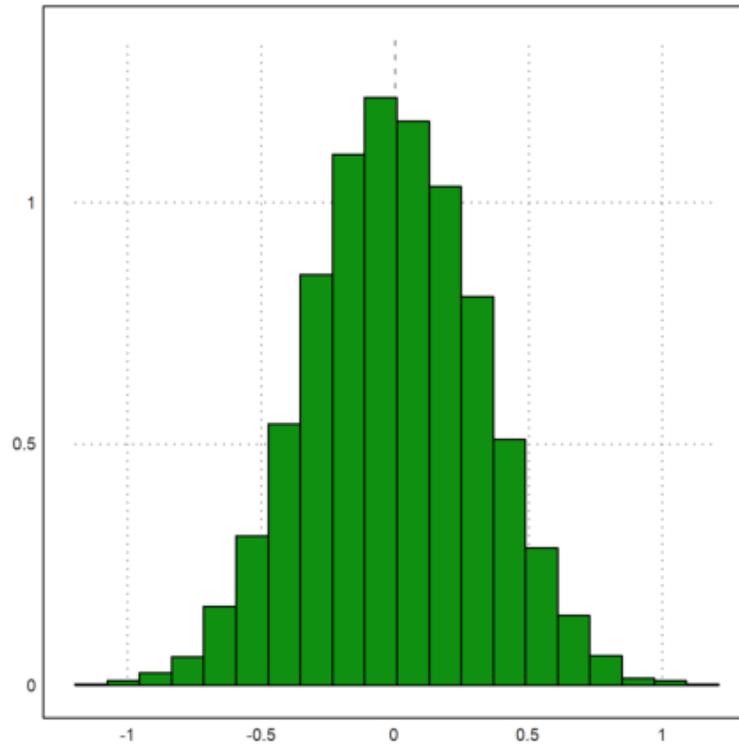
0.316227766017

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

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

0.319493614817

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



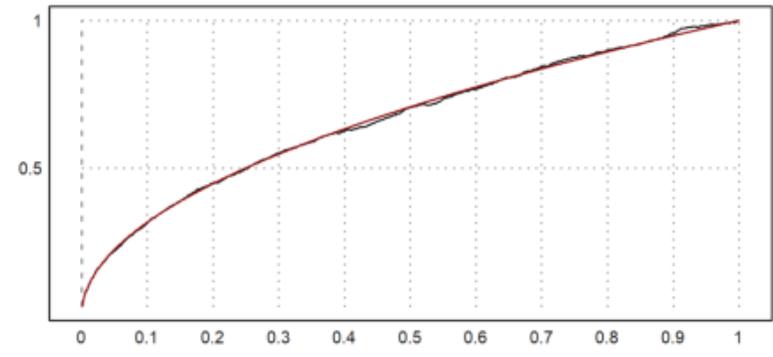
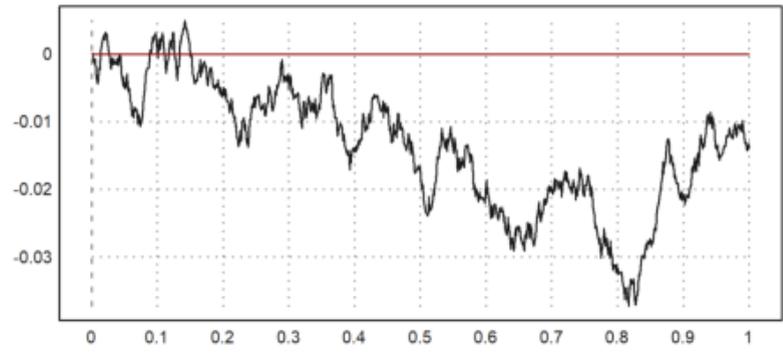
Median dari 10 bilangan acak terdistribusi 0-1-normal memiliki deviasi yang lebih besar.

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

0.374460271535

Karena kita dapat dengan mudah membuat jalan acak, kita dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami kemudian memplot standar deviasi dan rata-rata langkah ke-n dari 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 adalah alat penting dalam statistik. Di Euler, banyak tes yang diterapkan. Semua tes ini mengembalikan kesalahan yang kami terima jika kami menolak hipotesis nol.

Sebagai contoh, kami menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kami mendapat nilai berikut, yang kami masukkan ke uji chi-square.

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

```
0.498830517952
```

Tes chi-kuadrat juga memiliki mode yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya harus 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 pemerataan distribusi. Ini tidak membuktikan bahwa dadu kami adil. Tapi kita tidak bisa menolak hipotesis kita.

Selanjutnya kami menghasilkan 1000 lemparan dadu menggunakan generator angka acak, dan melakukan 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, simpangan, jumlah data, dan nilai rata-rata untuk diuji.

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

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

0.38722000942

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

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

5.60009101758e-07

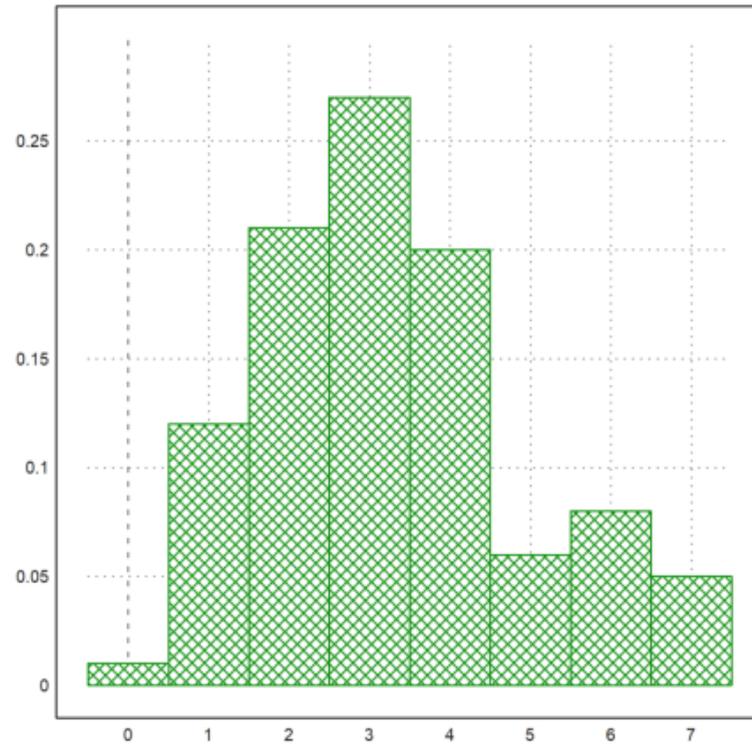
Dalam contoh berikutnya, kami menghasilkan 20 lemparan dadu acak 100 kali dan menghitungnya. Harus ada rata-rata $20/6=3,3$.

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

3.28

Kami sekarang membandingkan jumlah satu dengan distribusi binomial. Pertama kita memplot distribusi satuan.

```
>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;
```

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

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

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

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

0.53921579764

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

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

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

Kami ingin menguji independensi suara dari jenis kelamin. Tes tabel χ^2 melakukan ini. Hasilnya terlalu besar untuk menolak kemerdekaan. Jadi kami tidak bisa mengatakan, jika pemungutan suara tergantung pada jenis kelamin dari data tersebut.

```
>tabletest(A)
```

0.990701632326

Berikut 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 yang dikoreksi. Karena sangat mendekati 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```

Beberapa Tes Lagi

Selanjutnya kami menggunakan analisis varians (F-test) untuk menguji tiga sampel data yang terdistribusi normal untuk nilai rata-rata yang sama. Metode tersebut dinamakan ANOVA (analysis of variance). Di 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 dengan nilai rata-rata yang sama. Kami melakukan ini dengan probabilitas kesalahan 1,3%.

Ada juga uji median yang menolak sampel data dengan distribusi rata-rata yang berbeda menguji median sampel bersatu.

```
>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

Tes lain tentang kesetaraan adalah tes peringkat. Ini jauh lebih tajam daripada tes median.

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

0.00199969612469

Dalam contoh berikut, kedua distribusi memiliki rata-rata yang sama.

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

0.129608141484

Mari kita coba mensimulasikan dua perlakuan 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];
```

Tes signum memutuskan, jika a lebih baik dari b.

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

```
0.0546875
```

Ini terlalu banyak kesalahan. Kita tidak dapat menolak bahwa a sama baiknya dengan b.

Tes Wilcoxon lebih tajam dari tes ini, tetapi bergantung pada nilai kuantitatif perbedaannya.

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

```
0.0296680599405
```

Mari kita coba dua tes lagi menggunakan rangkaian yang dihasilkan.

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

0.0068706451766

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

0.275145971064

Angka Acak

Berikut ini adalah tes untuk generator angka acak. Euler menggunakan generator yang sangat bagus, jadi kita tidak perlu berharap ada masalah.

Pertama kami menghasilkan sepuluh juta angka acak di $[0,1]$.

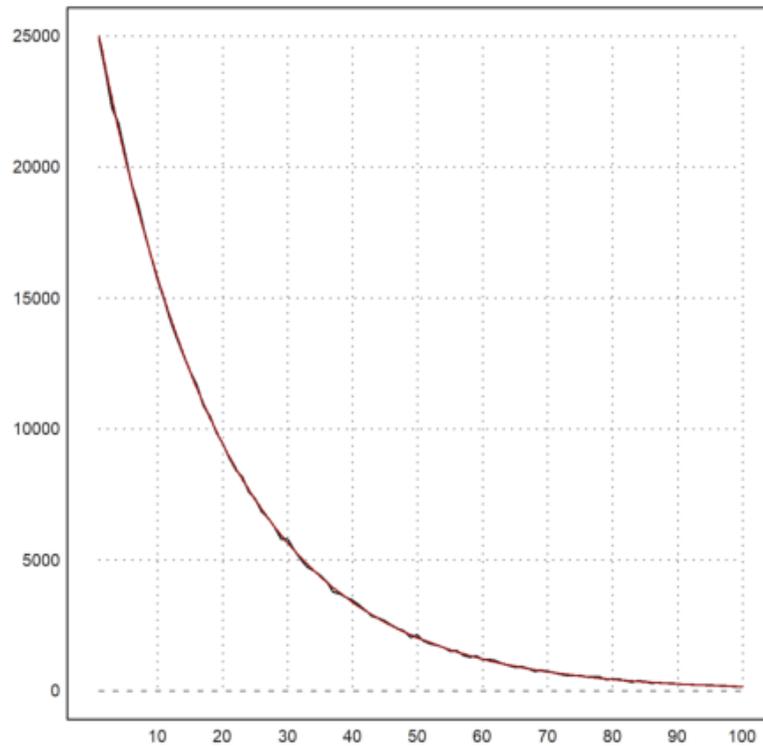
```
>n:=10000000; r:=random(1,n);
```

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

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

Akhirnya, kami memplot berapa kali, setiap jarak terjadi, dan membandingkannya dengan nilai yang diharapkan.

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



Hapus datanya.

```
>remvalue n;
```

Pengantar untuk Pengguna Proyek R

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

Notebook ini cocok untuk Anda jika sudah familiar dengan R, namun perlu mengetahui perbedaan sintaks EMT dan R. Kami mencoba memberikan gambaran umum tentang hal-hal yang jelas dan kurang jelas yang perlu Anda ketahui.

Selain itu, kami mencari cara untuk bertukar data antara kedua sistem.

Perhatikan bahwa ini adalah pekerjaan yang sedang berjalan.

Sintaks Dasar

Hal pertama yang Anda pelajari di R adalah membuat vektor. Di EMT, perbedaan utamanya adalah operator : dapat mengambil ukuran langkah. Apalagi daya ikatnya 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. Dimungkinkan untuk menggunakan vektor untuk menggabungkan berbagai hal.

Contoh berikut, seperti banyak lainnya, dari "Introduction to R" yang disertakan dengan proyek R. Jika Anda membaca PDF ini, Anda akan menemukan bahwa saya mengikuti jalannya dalam tutorial ini.

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

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

Operator titik dua dengan ukuran langkah EMT diganti dengan fungsi `seq()` di R. Kita bisa menulis fungsi ini di EMT.

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

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

Fungsi `rep()` dari R tidak ada di EMT. Untuk input vektor, dapat ditulis sebagai berikut.

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

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

Perhatikan bahwa "=" atau "==" digunakan untuk tugas. Operator "->" digunakan untuk unit di EMT.

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

```
77.6713990297 miles
```

The "<-" operator for assignment is misleading anyway, and not a good idea of R. The following will compare a and -4 in EMT.

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

```
0
```

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

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

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

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

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

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

```
NAN  
1
```

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

Di R ada paket untuk Unicode. Di EMT, sebuah string dapat berupa string Unicode. 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 di sistem Anda sebagai A dengan titik dan garis di atasnya. Itu tergantung pada font yang Anda gunakan.

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

Penggabungan string dilakukan dengan "+" atau "|". Itu bisa termasuk angka, yang akan dicetak dalam format saat ini.

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

```
pi = 3.14159265359
```

Pengindeksan

Sebagian besar waktu, ini akan berfungsi seperti di R.

Tetapi EMT akan menginterpretasikan indeks negatif dari belakang vektor, sedangkan 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 logis tidak diperlakukan berbeda sebagai indeks di EMT, berbeda dengan R. Anda perlu mengekstraksi elemen 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]
```

Tetapi nama untuk indeks tidak dimungkinkan di EMT. Untuk paket statistik, hal ini sering diperlukan untuk memudahkan akses ke elemen vektor.

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

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

```
[10.4, 3.1]
```

Tipe Data

EMT memiliki lebih banyak tipe data tetap daripada R. Jelas, di R terdapat vektor yang tumbuh. Anda dapat menyetel vektor numerik kosong v dan menetapkan nilai ke elemen $v[17]$. Ini tidak mungkin di EMT.

Berikut ini agak tidak efisien.

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

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

Semakin efisien pra-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 dimungkinkan untuk tipe data dasar. Format saat ini digunakan untuk penggabungan string sederhana. Tapi ada fungsi seperti `print()` atau `frac()`.

Untuk vektor, Anda dapat dengan mudah menulis fungsi Anda sendiri.

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

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

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

Untuk komunikasi dengan Maxima, terdapat 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 Lateks, perintah tex dapat digunakan untuk mendapatkan perintah Lateks.

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

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

Faktor dan Tabel

Dalam pengantar R ada contoh dengan apa yang disebut faktor.

Berikut ini adalah daftar wilayah dari 30 negara bagian.

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

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

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

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

EMT dapat melakukannya dengan menemukan indeks wilayah di daftar unik wilayah.

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

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

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

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

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

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

Ini sedikit tidak efisien, karena menghitung wilayah unik untuk setiap i, tetapi berhasil.

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

```
[44.5, 57.333, 55.5, 53.6, 55, 60.5, 56, 52.25]
```

Perhatikan bahwa ini berfungsi untuk setiap vektor wilayah.

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

```
[44.5, 57.333]
```

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

Sehingga kita bisa mencetak rata-rata pendapatan negara di daerah dengan cara yang bersahabat.

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

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

Kami juga dapat mencoba meniru perilaku R sepenuhnya.

Faktor jelas harus disimpan dalam kumpulan dengan jenis dan kategori (negara bagian dan teritori dalam contoh kita). Untuk EMT, kami menambahkan indeks yang telah dihitung sebelumnya.

```
>function makef (t) ...
```

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

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

Sekarang elemen ketiga dari koleksi akan berisi indeks.

```
>statef[3]
```

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

Sekarang kita bisa meniru `tapply()` dengan cara berikut. Ini akan mengembalikan tabel sebagai 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 pengecekan tipe di sini. Satu-satunya tindakan pencegahan menyangkut kategori (faktor) tanpa data. Tetapi orang harus memeriksa panjang `t` yang benar dan kebenaran koleksi `tf`.

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

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

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

Array

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

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

Dalam EMT, vektor adalah matriks dengan satu baris. Itu dapat dibuat menjadi matriks dengan `redim()`.

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

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

Ekstraksi baris dan kolom, atau sub-matriks, sangat mirip dengan R.

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

2	3
7	8
12	13
17	18

Namun, dalam R dimungkinkan untuk menetapkan daftar indeks spesifik vektor ke suatu nilai. Hal yang sama dimungkinkan di EMT hanya dengan satu putaran.

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

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

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

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

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

Produk luar di EMT hanya dapat dilakukan di antara vektor. Ini otomatis karena bahasa matriks. Satu vektor harus berupa vektor kolom dan yang lainnya vektor baris.

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

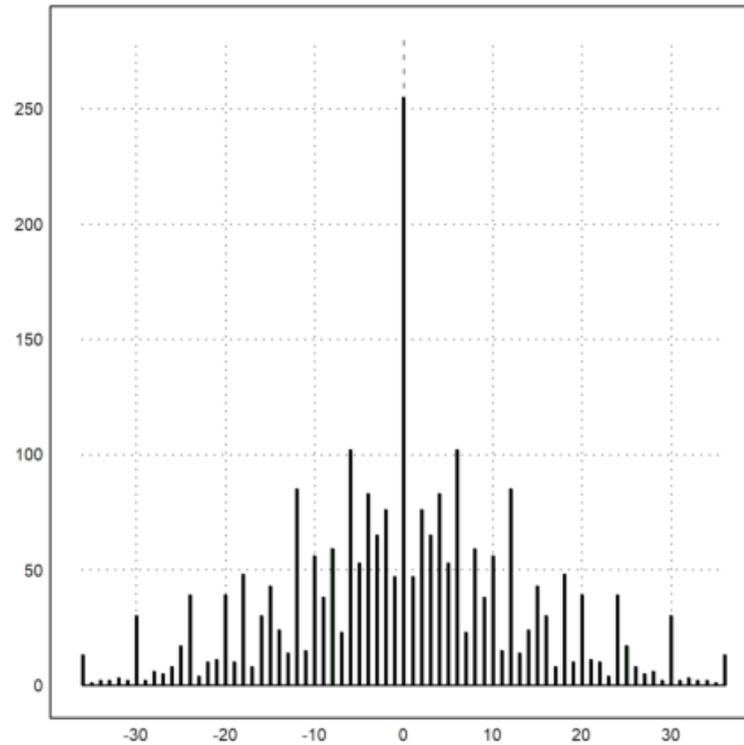
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

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

Tentu saja, ini bisa dicapai dengan satu putaran. Tapi loop tidak efektif di EMT atau R. Di EMT, kita bisa menulis loop di C dan itu akan menjadi solusi tercepat.

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

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



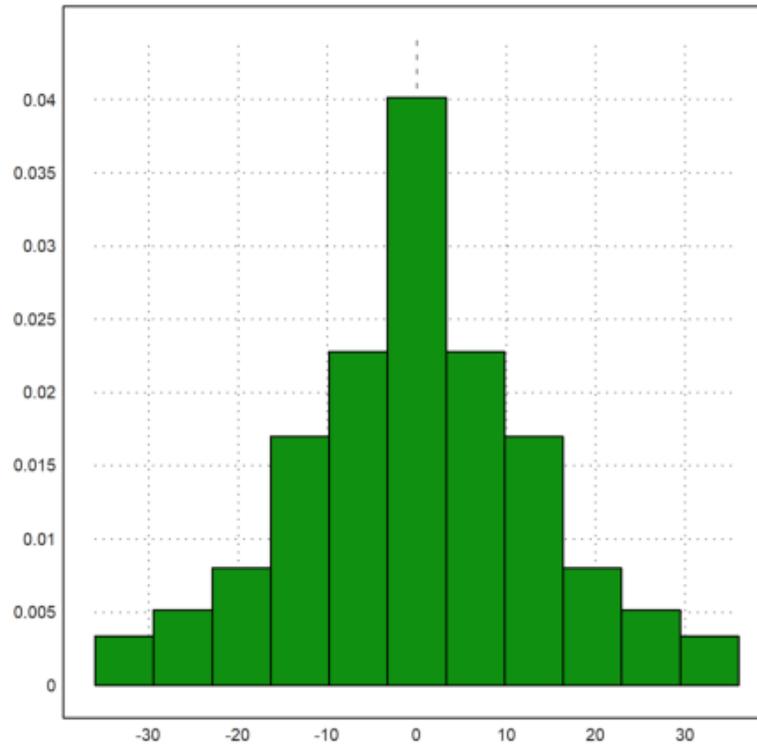
Selain perkalian yang tepat, EMT dapat menghitung frekuensi dalam vektor.

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

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

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

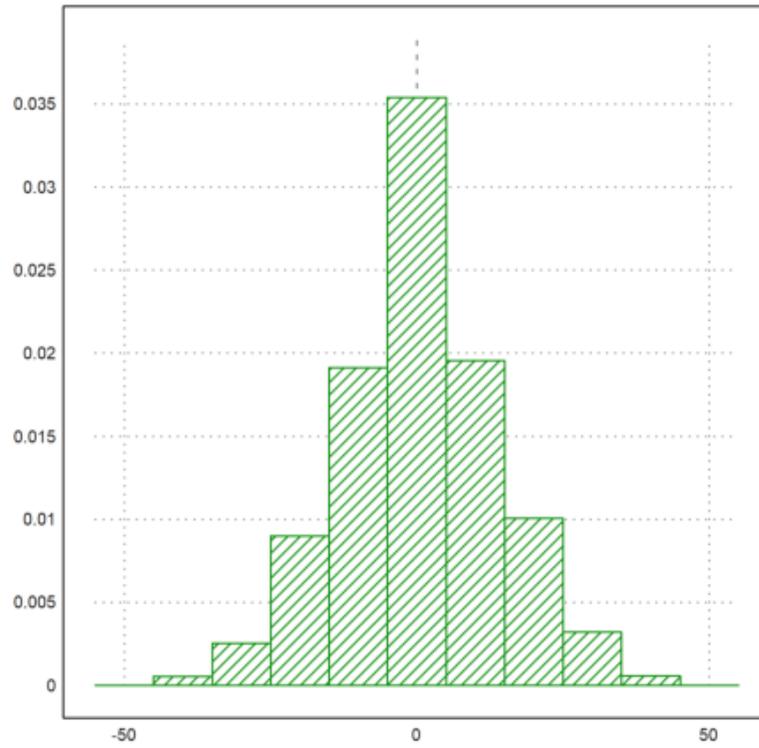
```
>plot2d(q,distribution=11):
```



Tetapi juga memungkinkan untuk melakukan pra-perhitungan hitungan 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. Salah satunya adalah daftar global yang bisa berubah, dan yang lainnya adalah tipe daftar yang tidak bisa diubah. Kami tidak peduli dengan daftar global di sini.

Jenis daftar yang tidak dapat diubah disebut koleksi di EMT. Ini berperilaku seperti struktur di C, tetapi elemennya hanya diberi nomor dan tidak diberi nama.

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

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

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

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

```
1992
```

File Input dan Output (Membaca dan Menulis Data)

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

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

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

```
0.47112  
0.27409
```

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

Karena pengantar ini kemungkinan besar ada di direktori, di mana pengguna tidak memiliki akses tulis, kami menulis data ke direktori home pengguna. Untuk buku catatan 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. Ini menghasilkan satu nomor di setiap baris file.

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

Untuk membaca data, kami menggunakan `readmatrix()`.

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

Dan hapus file tersebut.

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

0.47112

0.27409

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

Misalnya, jika Anda memiliki sistem bahasa Indonesia (titik desimal dengan koma), Excel Anda memerlukan nilai dengan koma desimal yang dipisahkan oleh titik koma dalam file csv (defaultnya adalah nilai yang dipisahkan koma). File berikut "test.csv" akan muncul di folder current Anda.

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

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

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

Terkadang kami memiliki string dengan token seperti berikut ini.

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

Untuk menandai ini, kami 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 header 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();
```

The file looks like this.

```
>printfile(file)
```

```
A,B,C  
0.7458474088446375,0.7065756581998108,0.1762109836460713  
0.157478018377736,0.6970501068277559,0.08533127338306896  
0.6792495519557092,0.5852681243549057,0.7464205252219664
```

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

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

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

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

	A	B	C
	0.74585	0.70658	0.17621
	0.15748	0.69705	0.08533
	0.67925	0.58527	0.74642

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

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

```
0.54288  
0.31329  
0.67031
```

File CSV

Pertama, mari kita menulis matriks ke dalam file. Untuk hasilnya, kami membuat file di direktori kerja saat ini.

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

Berikut adalah isi dari file ini.

```
>printfile(file)
```

```
0.3967151836123274,0.5187168657271813,0.9551641634613992  
0.7245413076998718,0.7031583566071746,0.8186483465065758  
0.4646315902907798,0.8738902114707362,0.9789905684134915
```

CVS ini dapat dibuka pada sistem bahasa Inggris ke dalam Excel dengan klik dua kali. Jika Anda mendapatkan file seperti itu di sistem Jerman, Anda perlu mengimpor data ke Excel dengan memperhatikan titik desimal.

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

```
>readmatrix(file)
```

```
0.39672  0.51872  0.95516
0.72454  0.70316  0.81865
0.46463  0.87389  0.97899
```

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();
```

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

Di Excel atau spreadsheet serupa, Anda dapat mengekspor matriks sebagai CSV (nilai yang dipisahkan koma). Di Excel 2007, gunakan "simpan sebagai" dan "format lain", lalu pilih "CSV". Pastikan, tabel saat ini hanya berisi data yang ingin Anda ekspor.

Ini sebuah contoh.

```
>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 membaca matriks ke dalam EMT.

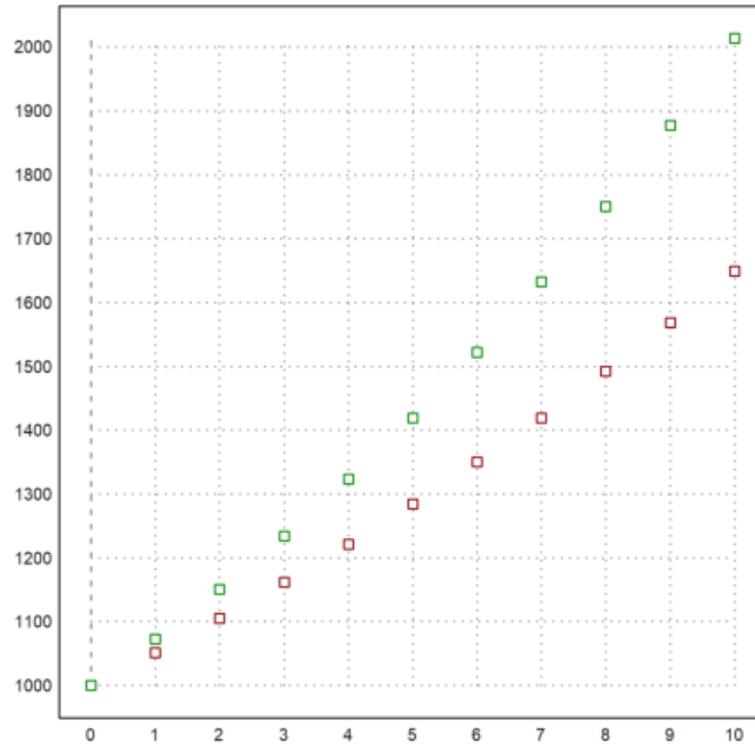
Cara termudah untuk membaca ini ke Euler adalah `readmatrix()`. Semua koma diganti dengan titik dengan parameter `>koma`. Untuk CSV bahasa Inggris, hilangkan saja parameter ini.

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

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

Let us plot this.

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



Ada 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 baris data. Secara default, ini mengharapkan titik desimal. Tapi itu juga bisa menggunakan koma desimal, jika Anda memanggil `setdecimaldot(",")` sebelum Anda menggunakan fungsi ini.

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

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

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

```
>myload(file)
```

```
0.39672  0.51872  0.95516  
0.72454  0.70316  0.81865  
0.46463  0.87389  0.97899
```

Dimungkinkan juga untuk membaca semua angka dalam file itu dengan `getvector()`.

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

```
0.39672  0.51872  0.95516  
0.72454  0.70316  0.81865  
0.46463  0.87389  0.97899
```

Thus it is very easy to save a vector of values, one value in each line and read back this vector.

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

0.50102

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

0.50102

Menggunakan Tabel

Tabel dapat digunakan untuk membaca atau menulis data numerik. Sebagai contoh, kami menulis tabel dengan tajuk 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
  0.2,    0.26,    0.5
  0.58,   0.14,    0.5
  0.12,   0.99,    0.09
```

Ini dapat diimpor ke Excel.

Untuk membaca file di EMT, kami menggunakan `readtable()`.

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

```
one    two    three
  0.2   0.26   0.5
  0.58  0.14   0.5
  0.12  0.99   0.09
```

Menganalisis Garis

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

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

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

Pertama kita dapat menandai garis.

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

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

Kemudian kita dapat mengevaluasi setiap elemen garis 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
```

Menggunakan ekspresi reguler, dimungkinkan untuk mengekstraksi hampir semua informasi dari sebaris data.

Asumsikan kita memiliki baris berikut sebuah dokumen HTML.

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

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

Untuk mengekstrak ini, kami menggunakan ekspresi reguler, yang mencari

- tanda kurung tutup > ,
- string apa pun yang tidak mengandung tanda kurung dengan

sub-pertandingan "(...)",

- braket pembuka dan penutup menggunakan solusi terpendek,
- sekali lagi string apa pun yang tidak mengandung tanda kurung,
- dan tanda kurung buka <.

Ekspresi reguler agak sulit dipelajari tetapi sangat kuat.

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

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

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

1145.5

5.6

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 terkini dari situs EMT. Kami menggunakan ekspresi reguler untuk memindai "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 = [ ..  
0.8167786572250472, 0.3865432556980181;  
0.4867362693755808, 0.8897016232947995];
```

We can now load the file. It will define the matrix M.

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

```
M =  
  0.81678    0.38654  
  0.48674    0.8897
```

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

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

```
M = [ ..  
  0.8167786572250472, 0.3865432556980181;  
  0.4867362693755808, 0.8897016232947995];  
inch$ = 0.0254;
```

Kami juga dapat membuka file baru atau menambahkan file yang sudah ada. Dalam contoh kami menambahkan file yang dihasilkan sebelumnya.

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

```
M1 =  
  0.54537  0.19342  
  0.47573  0.9986  
M2 =  
  0.070764  
    0.746  
    0.36595
```

Untuk menghapus file apa pun gunakan `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 = [  
0.744800648795  
0.714418072591  
0.469973724311  
0.439332202237  
0.885478301097  
];
```

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

```
[0.7448, 0.71442, 0.46997, 0.43933, 0.88548]
```

1. Menyimpan Data dalam Bentuk Matriks

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

dan diketahui matriks pengeluaran sebagai berikut

```
>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

2. Data acak dengan fungsi distribusi

Soal 1

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)
```

0.102951605366

```
>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)
```

0.0973831642309

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

```
548.347804004
```

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

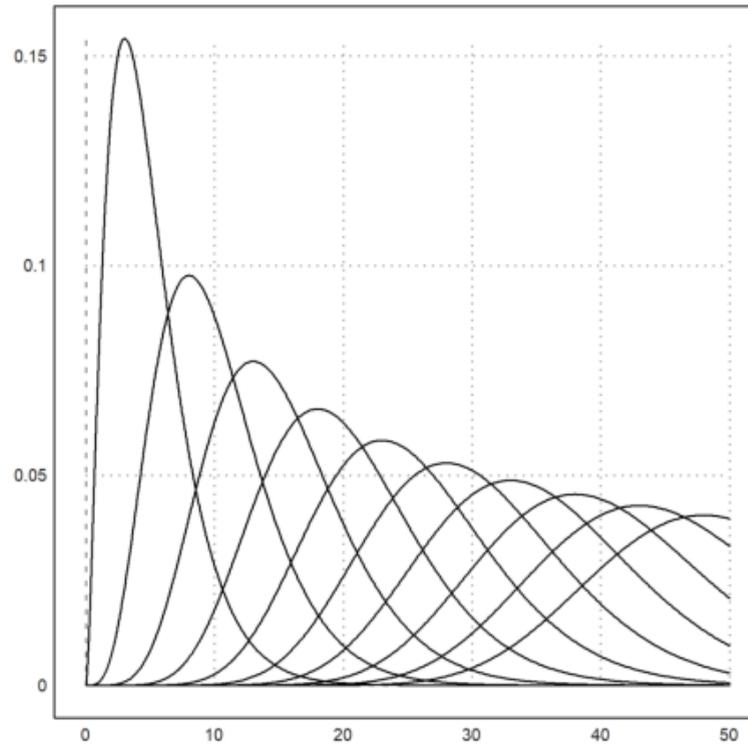
Mari kita bandingkan distribusi binomial dengan distribusi normal mean dan deviasi yang sama. Fungsi `invbindis()` memecahkan interpolasi linier antara nilai bilangan bulat.

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

```
525.516721219  
526.007419394
```

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)
```

```
0.527633447259
```

```
0.527633447259
```

3. Input dan Output File (Membaca dan Menulis Data)

Soal 1

Cara membaca file csv yang bernama "GOOG"

```
>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
```

Soal 2

Cara 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

4. Perhitungan Analisis Data Statistika Deskriptif

Soal 1

Seorang pelatih tembak ingin mengevaluasi nilai ketangkasan delapan anak buahnya jenis senapan yang dipakai M-16 dengan jarak 300 meter dan masing-masing mendapat nilai 76,85,70,65,40,70,50,dan 80. Berapakah rata-rata nilai ketangkasan delapan anak tersebut?

Penyelesain:

```
>x=[76,85,70,65,40,70,50,80]; mean(x),
```

67

Diketahui:

$$\sum x_i = 76 + 85 + 70 + 65 + 40 + 70 + 80 = 536$$

$$n = 8$$

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

$$\bar{X} = \frac{536}{8}$$

$$\bar{X} = 67$$

Sehingga rata-rata nilai ketangkasan delapan anak tersebut adalah 67

Soal 2

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

Penyelesaian:

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

4.8

Diketahui:

$$\sum x_i = 3 + 5 + 6 + 4 + 6 = 24$$

$$N = 5$$

$$\mu = \frac{\sum x_i}{N}$$

$$\mu = \frac{24}{5}$$

$$\mu = 4.8$$

Sehingga nilai rata-rata banyaknya pegawai di 5 apotik tersebut adalah 4.8

Soal 3

Data berikut menunjukkan nilai yang diperoleh 50 siswa SMP Negeri 1 Gabus pada Ujian Nasional mata pelajaran matematika.

Siswa yang mendapat nilai dalam rentang 61-65 sebanyak 2 orang, dalam rentang 66-70 sebanyak 5 orang, dalam rentang 71-75 sebanyak 8 orang, dalam rentang 76-80 sebanyak 10 orang, dalam rentang 81-85 sebanyak 12 orang, dalam rentang 86-90 sebanyak 9 orang, dan dalam rentang 91-95 sebanyak 4 orang. Tentukan rata-rata nilai yang diperoleh 50 siswa tersebut!

Penyelesaian:

Menentukan tepi bawah kelas yang terkecil

$$>61-0.5$$

$$60.5$$

Menentukan panjang kelas

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

Menentukan tepi atas kelas yang terbesar

```
>95+0.5
```

95.5

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

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

Menentukan titik tengah

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

63
68
73

78
83
88
93

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

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

Menentukan mean(rata-rata)

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

79.8

Diketahui:

$$\sum t_i f_i = (2)(63) + (5)(68) + (8)(73) + (10)(78) + (12)(83) + (9)(88) + (4)(93) = 3.990$$

$$\sum f_i = 2 + 5 + 8 + 10 + 12 + 9 + 4 = 50$$

$$\bar{X} = \frac{\sum t_i f_i}{\sum f_i}$$

$$\bar{X} = \frac{3.990}{50}$$

$$\bar{X} = 79,8$$

Jadi rata-rata nilai yang diperoleh 50 siswa tersebut adalah 79,8

Soal 4

Data upah dari 8 karyawan yang dinyatakan dalam rupiah adalah sebagai berikut:

20,80,75,60,50,85,45,90.

Tentukan nilai median dari data tersebut!

Penyelesaian:

```
>data=[20,80,75,60,50,85,45,90];  
>urut=sort(data)
```

```
[20, 45, 50, 60, 75, 80, 85, 90]
```

```
>median(data)
```

```
67.5
```

Diketahui bahwa kasus ini merupakan data yang berjumlah genap, sehingga nilai median untuk kasus ini adalah terletak pada data ke- k dan data ke- $(k+1)$.

$$k = \frac{n}{2}$$

$$k = \frac{8}{2}$$

$$k = 4$$

$$Me = \frac{1}{2}(X_k + X_{k+1})$$

$$Me = \frac{1}{2}(X_4 + X_5)$$

$$Me = \frac{1}{2}(60 + 75)$$

$$Me = 67.5$$

Jadi nilai median pada data upah 8 karyawan adalah 67.5

Soal 5

Berikut adalah data hasil dari pengukuran berat badan 50 siswa SD Negeri Tambakrejo. Dari ke 50 siswa, mayoritas siswa memiliki berat badan yang ideal. Siswa yang mempunyai berat badan dalam rentang 21-26 kg sebanyak 5 orang, yang mempunyai berat badan dalam rentang 27-32 kg sebanyak 10 orang, yang mempunyai berat badan dalam rentang 33-38 kg sebanyak 12 orang, yang mempunyai berat badan dalam rentang 39-44 kg sebanyak 14 orang, yang mempunyai berat badan dalam rentang 45-50 kg sebanyak 7 orang, dan yang mempunyai berat badan 51-56 kg sebanyak 2 orang. Tentukan median dari data hasil pengukuran berat badan 50 siswa di SD tersebut!

Penyelesaian:

Menentukan tepi bawah kelas yang terkecil

>21-0.5

20.5

Menentukan panjang kelas

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

6

Menentukan tepi atas kelas yang terbesar

```
>56+0.5
```

56.5

```
>r=20.5:6:56.5; v=[5,10,12,14,7,2];  
>T=:r[1:6]' | r[2:7]' | v'; writetable(T,labc=["TB","TA","frek"])
```

TB	TA	frek
20.5	26.5	5
26.5	32.5	10
32.5	38.5	12
38.5	44.5	14
44.5	50.5	7
50.5	56.5	2

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

$Tb=32.5$, $p=6$, $n=50$, $Fks=15$, $fm=12$

32.5
6
50
15
12

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

37.5

Diketahui bahwa median berada di data ke 25

$$Me = Tb + \frac{\frac{1}{2}n - f_{ks}}{f_m} \cdot p$$
$$Me = 32.5 + \frac{\frac{1}{2}(50) - 15}{12} \cdot 6$$
$$Me = 32.5 + 5$$
$$Me = 37.5$$

Jadi median dari data hasil pengukuran berat badan 50 siswa SD Tambakrejo adalah 37.5

Soal 6

Berikut adalah data hasil dari pengukuran berat badan 30 siswa SD Negeri Tambakrejo. Dari ke 30 siswa, mayoritas siswa memiliki berat badan yang ideal. Siswa yang mempunyai berat badan dalam rentang 21-25 kg sebanyak 2 orang, yang mempunyai berat badan dalam rentang 26-30 kg sebanyak 8 orang, yang mempunyai berat badan dalam rentang 31-35 kg sebanyak 9 orang, yang mempunyai berat badan dalam rentang 36-40 kg sebanyak 6 orang, yang mempunyai berat badan dalam rentang 41-45 kg sebanyak 3 orang, dan yang mempunyai berat badan 46-50 kg sebanyak 2 orang. Tentukan modus dari data hasil pengukuran berat badan 30 siswa di SD tersebut!

Penyelesaian:

Menentukan tepi bawah kelas yang terkecil

$$>21-0.5$$

$$20.5$$

Menentukan panjang kelas

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

$$5$$

Menentukan tepi atas yang terbesar

```
>50+0.5
```

```
50.5
```

```
>r=20.5:5:50.5; v=[2,8,9,6,3,2];  
>T:=r[1:6]' | r[2:7]' | v'; writetable(T,labc=["TB","TA","frek"])
```

TB	TA	frek
20.5	25.5	2
25.5	30.5	8
30.5	35.5	9
35.5	40.5	6
40.5	45.5	3
45.5	50.5	2

Berdasarkan data, modus berada pada kelas 30.5-35.5.

```
>Tb=30.5, p=5, d1=1, d2=3
```

```
30.5  
5  
1  
3
```

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

31.75

5. Menggambar Grafik Statistika

Soal 1

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)
```

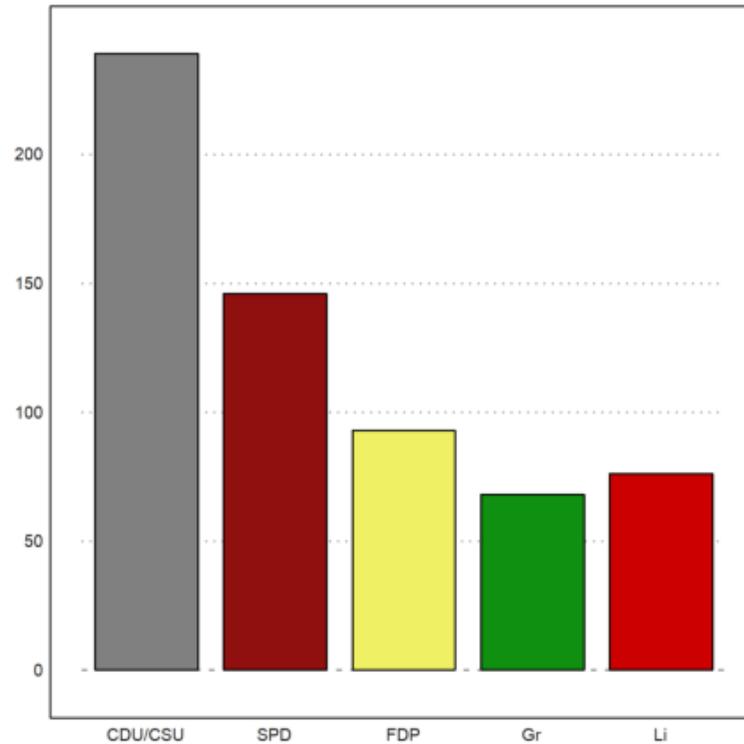
	CDU/CSU	SPD	FDP	Gr	Li
1990	48	36	12	1	3
1994	44	38	7	7	4
1998	37	45	6	7	5
2002	41	42	8	9	0
2005	37	36	10	8	9
2009	38	23	15	11	12
2013	49	31	0	10	10

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)];
```

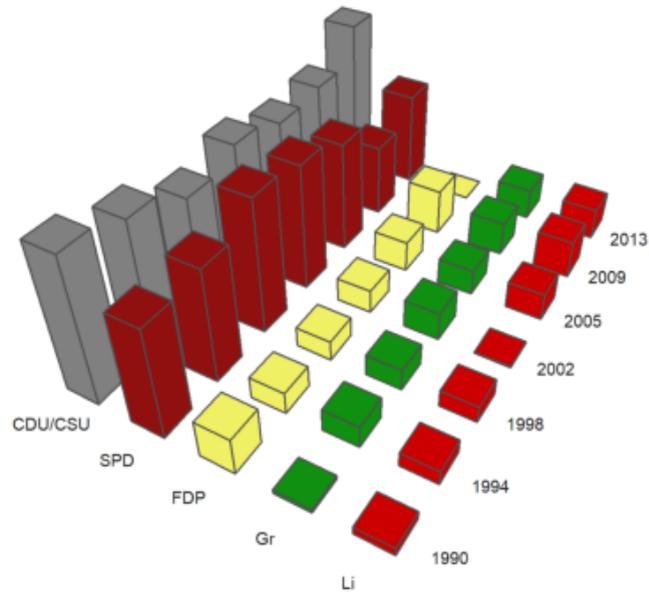
Fungsi CP disini digunakan nantinya untuk membuat perbedaan warna pada setiap batang yang menyesuaikan nama partai

```
>columnspot(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):
```

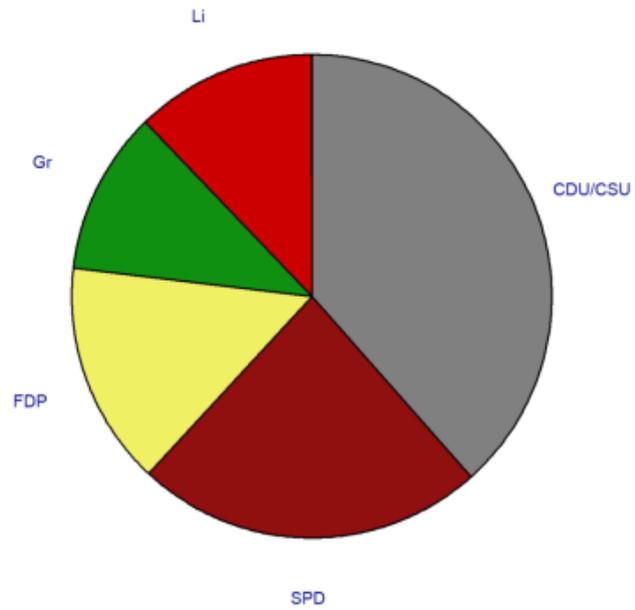


Ini adalah perintah untuk membuat diagram batang(kolom) dalam 3 Dimensi . Plot ini akan menggunakan data dari BT, dengan kolom-kolom yang dipilih pada data BW, yaitu nama-nama partai (scols=p). Selain itu, parameter angle=30° mengatur sudut pandang plot dalam ruang 3D, ccols=CP mungkin mengatur warna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

Soal 2

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="0#", color=CP[i],lab=P[i], textcolor=blue, r=1.5):
```



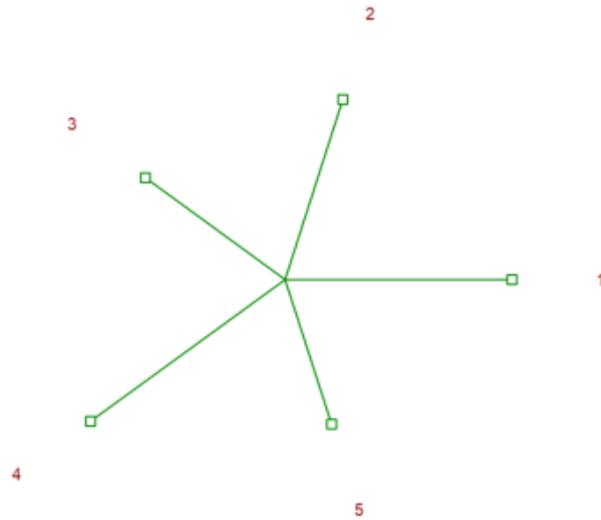
Soal 3

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):
```



Soal 4

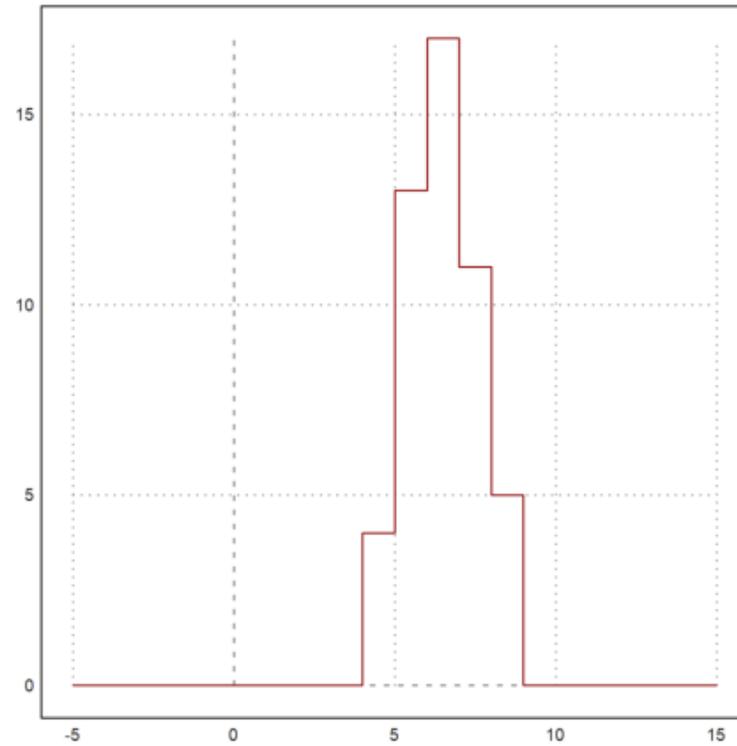
Sebuah universitas melakukan penelitian untuk mengidentifikasi distribusi waktu studi mahasiswa di suatu jurusan. Data waktu studi (dalam tahun) dari 100 mahasiswa lulusan jurusan tersebut telah dikumpulkan. Berikut adalah data tersebut:

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):
```



6. Menampilkan Tabel Frame Data

Soal 1

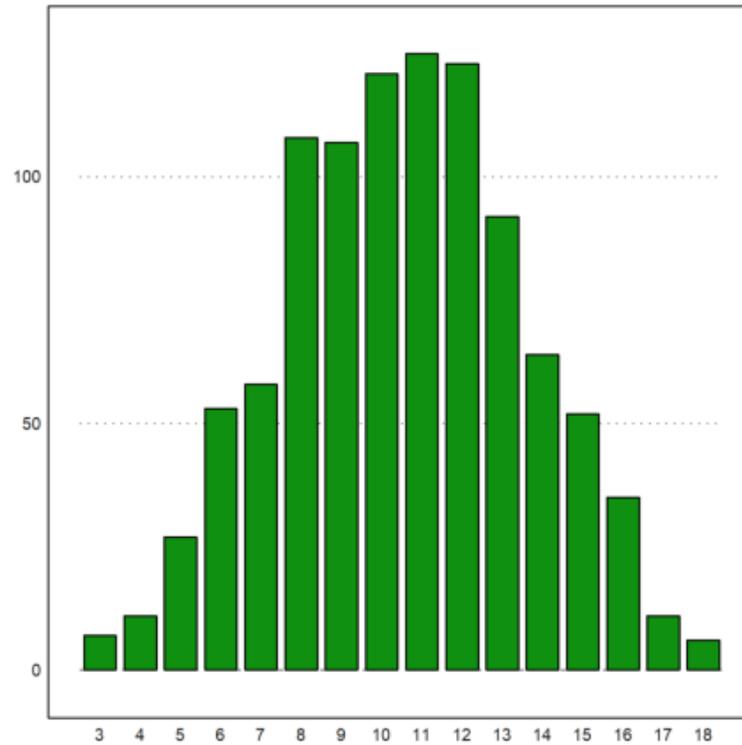
Berapa distribusi jumlah dari 1000 kali lemparan 3 dadu?

Penyelesaian

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

```
[7, 11, 27, 53, 58, 108, 107, 121, 125, 123, 92, 64, 52,  
35, 11, 6]
```

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



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

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

```

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

```

Ini adalah hasil dari tiga lemparan dadu.

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

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

Atau kita bisa menemukan secara manual yaitu :

Jumlah minimal dari 3 dadu tersebut adalah 3 dan jumlah maksimal dari 3 dadu tersebut adalah 18, diperoleh nilai terkecil dan nilai terbesar dari dadu. Sedangkan jumlah seluruh sampelnya yaitu $6^3 = 216$, sehingga diperoleh frekuensi dari jumlah 3 hingga jumlah 18 yaitu :

Jumlah 3 = 1

Jumlah 4 = 3

Jumlah 5 = 6

Jumlah 6 = 10

Jumlah 7 = 15

Jumlah 8 = 21

Jumlah 9 = 25

Jumlah 10 = 27

Jumlah 11 = 27

Jumlah 12 = 25

Jumlah 13 = 21

$$\text{Jumlah } 14 = 15$$

$$\text{Jumlah } 15 = 10$$

$$\text{Jumlah } 16 = 6$$

$$\text{Jumlah } 17 = 3$$

$$\text{Jumlah } 18 = 1$$

sehingga peluang dari 3 dadu yang dilempar sebanyak 1000 kali adalah :

Peluang muncul dadu dengan jumlah 3

$$\text{Peluang muncul dadu dengan jumlah } 3 = \frac{1}{216} 1000$$

$$\text{Peluang muncul dadu dengan jumlah } 3 = \frac{1000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah } 3 = 4.629269$$

Peluang muncul dadu dengan jumlah 4

$$\text{Peluang muncul dadu dengan jumlah } 4 = \frac{3}{216} 1000$$

$$\text{Peluang muncul dadu dengan jumlah } 4 = \frac{3000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah } 4 = 13.8888$$

Peluang muncul dadu dengan jumlah 5

$$\text{Peluang muncul dadu dengan jumlah 5} = \frac{6}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 5} = \frac{6000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 5} = 27.7777$$

Peluang muncul dadu dengan jumlah 6

$$\text{Peluang muncul dadu dengan jumlah 6} = \frac{10}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 6} = \frac{10000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 6} = 46.296296$$

Peluang muncul dadu dengan jumlah 7

$$\text{Peluang muncul dadu dengan jumlah 7} = \frac{15}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 7} = \frac{15000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 7} = 69.44444$$

Peluang muncul dadu dengan jumlah 8

$$\text{Peluang muncul dadu dengan jumlah 8} = \frac{21}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 8} = \frac{21000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 8} = 97.22222$$

Peluang muncul dadu dengan jumlah 9

$$\text{Peluang muncul dadu dengan jumlah 9} = \frac{25}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 9} = \frac{25000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 9} = 115.740740$$

Peluang muncul dadu dengan jumlah 10

$$\text{Peluang muncul dadu dengan jumlah 10} = \frac{27}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 10} = \frac{27000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 10} = 125$$

Peluang muncul dadu dengan jumlah 11

$$\text{Peluang muncul dadu dengan jumlah 11} = \frac{27}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 11} = \frac{27000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 11} = 125$$

Peluang muncul dadu dengan jumlah 12

$$\text{Peluang muncul dadu dengan jumlah 12} = \frac{25}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 12} = \frac{25000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 12} = 115.740740$$

Peluang muncul dadu dengan jumlah 13

$$\text{Peluang muncul dadu dengan jumlah 13} = \frac{21}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 13} = \frac{21000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 13} = 97.22222$$

Peluang muncul dadu dengan jumlah 14

$$\text{Peluang muncul dadu dengan jumlah 14} = \frac{15}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 14} = \frac{15000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 14} = 69.44444$$

Peluang muncul dadu dengan jumlah 15

$$\text{Peluang muncul dadu dengan jumlah 15} = \frac{10}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 15} = \frac{10000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 15} = 46.296296$$

Peluang muncul dadu dengan jumlah 16

$$\text{Peluang muncul dadu dengan jumlah 16} = \frac{6}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 16} = \frac{6000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 16} = 27.7777$$

Peluang muncul dadu dengan jumlah 17

$$\text{Peluang muncul dadu dengan jumlah 17} = \frac{3}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 17} = \frac{3000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 17} = 13.8888$$

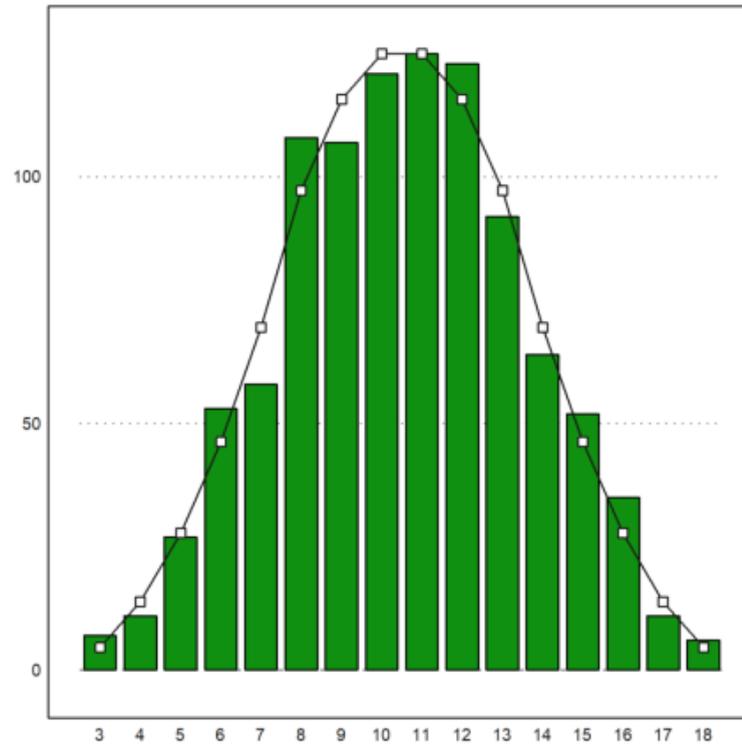
Peluang muncul dadu dengan jumlah 18

$$\text{Peluang muncul dadu dengan jumlah 18} = \frac{1}{216} \cdot 1000$$

$$\text{Peluang muncul dadu dengan jumlah 18} = \frac{1000}{216}$$

$$\text{Peluang muncul dadu dengan jumlah 18} = 4.6296296$$

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



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

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

0.316227766017

Pada grafik tersebut, dipilih $n=10$ karena Jika dihasilkan 10 variabel acak berdistribusi normal, maka rata-rata dari 10 nilai acak tersebut akan memiliki deviasi yang lebih kecil dibandingkan deviasi dari setiap variabel acak secara individu. Sehingga

$$\text{Deviasi rata-rata} = \frac{1}{\sqrt{(n)}}$$

$$\text{Deviasi rata-rata} = \frac{1}{\sqrt{(10)}}$$

$$\text{Deviasi rata-rata} = 0.3162277660$$

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

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

```
0.314862821476
```

```
>M=normal(10000,10)
```

```
Real 10000 x 10 matrix
```

```
1.32742839864    1.21132031654    ...  
-0.404532672668  0.767145747382    ...  
1.09371812982   -1.06910776866    ...  
0.0509302162481 -0.193362683084    ...  
-0.561902121172  0.176931299604    ...  
0.782300019941   -1.4122036835     ...  
1.14872590338    -0.7391417643     ...
```

```

1.837168604    -0.411688140845    ...
1.73612072661    -0.628118073745    ...
0.564376087522    -0.625521612541    ...
0.12938739382    -0.935180043208    ...
0.0172630557352    0.254296740951    ...
0.525869917034    0.530927163119    ...
1.3337703066    -0.854339040039    ...
-1.16826263521    1.34504590489    ...
-0.357287937302    1.14294177345    ...
0.219470682243    2.01630973862    ...
-0.251312432096    -0.0695968354654    ...
-1.46614216276    1.08555936851    ...
0.214424145088    -0.341176862123    ...
...

```

Fungsi normal di sini kemungkinan besar menghasilkan sampel acak dari distribusi normal (atau distribusi Gauss) dengan dua parameter: rata-rata (mean) dan simpangan baku (standard deviation).

0000 adalah jumlah sampel yang dihasilkan.

10 adalah simpangan baku distribusi normal yang digunakan (mungkin nilai default untuk rata-rata adalah 0, kecuali jika ada parameter lain yang diberikan). M berisi 10000 nilai acak dari distribusi normal dengan rata-rata 0 dan simpangan baku 10.

```
>mean(M)'
```

```

[0.146937300333, -0.283796789353, -0.218307060411, -0.235150844401,
0.308148683058, 0.176563901802, 0.079338613721, 0.246721984418,
0.467285287011, -0.0695808128704, -0.219224624115, 0.635909728169,
0.624264884023, -0.326072267857, 0.571748566856, 0.0878721321231,
0.100692833338, -0.144070285575, -0.581562668776, 0.112168375458,
-0.0914128775585, -0.520199730673, -0.0525956291678,
-0.272539318296, 0.122696563831, 0.385344820629, -0.282827830331,

```

```
0.240892331264, 0.0969554900205, -0.105297227854, 0.279055672038,  
0.228953678388, 0.109252212767, -0.151384154317, -0.0492706068543,  
-0.0958549961946, -0.327594489641, 0.0360451094902,  
-0.0348249013377, 0.165495309624, -0.280482831662, -0.33458219643,  
-0.207136881409, 0.0528837515673, 0.389534207486, -0.399121400828,  
0.0927698944254, -0.248841484744, -0.466261067023, -0.240763907313,  
-0.119435110695, -0.159157043435, -0.0464993535525,  
-0.165742881731, 0.298236141453, 0.00975580834287, 0.289486770546,  
0.237137099953, -0.365138692433, 0.301021837393, 0.0778876901934,  
0.0163522128566, 0.0198144205596, 0.352972701444, -0.0235229329728,  
-0.128942606092, 0.350400737651, -0.0642052451372, 0.250949938853,  
-0.0434507065796, 0.42943664133, 0.234624527728, 0.365777671707,  
0.265412054349, 0.327537938914, 0.429715898415, -0.388345129931,  
... ]
```

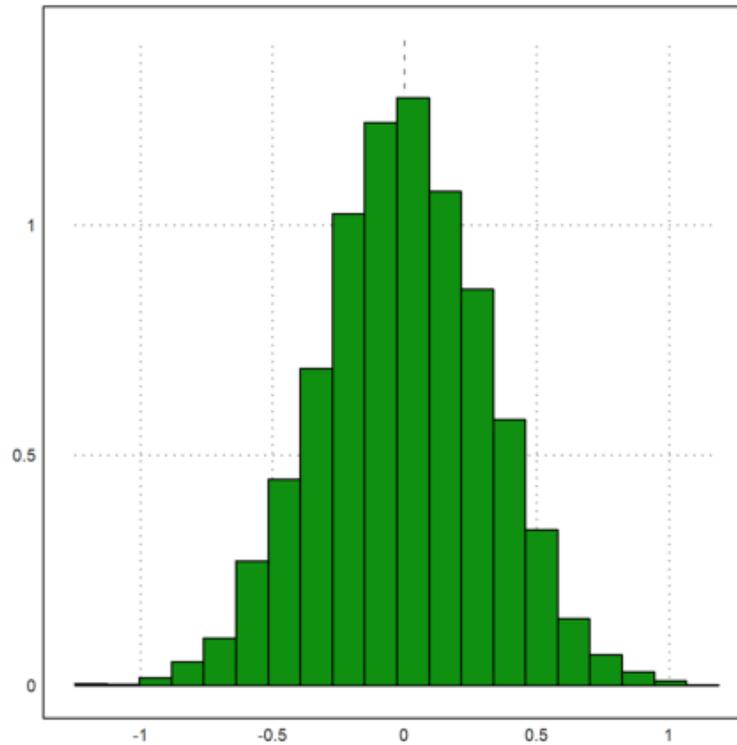
Fungsi `mean(M)` menghitung rata-rata (mean) dari data yang ada dalam matriks `M`.
anda ' setelah `mean(M)` menandakan operasi transpose (mengubah vektor baris menjadi vektor kolom
atau sebaliknya). Jadi, ini menghasilkan rata-rata dari data dalam `M` dalam bentuk vektor kolom.

```
>dev(mean(M)')
```

```
0.316787851565
```

`dev(mean(M)')` akan menghasilkan deviasi standar dari rata-rata, yang biasanya akan sangat kecil karena kita hanya menghitung deviasi standar dari satu nilai (rata-rata tersebut)

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



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

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

```
0.375279710106
```

Fungsi dev() di EMT digunakan untuk menghitung deviasi standar (standard deviation) dari argumen yang diberikan. Dalam hal ini, dev(median(M)') akan menghitung deviasi standar dari vektor hasil transpos median(M).

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

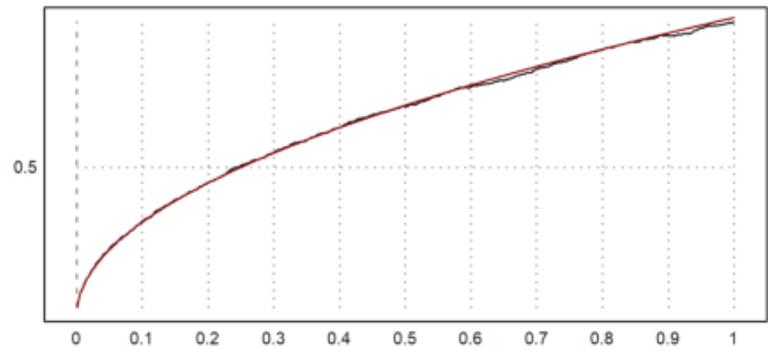
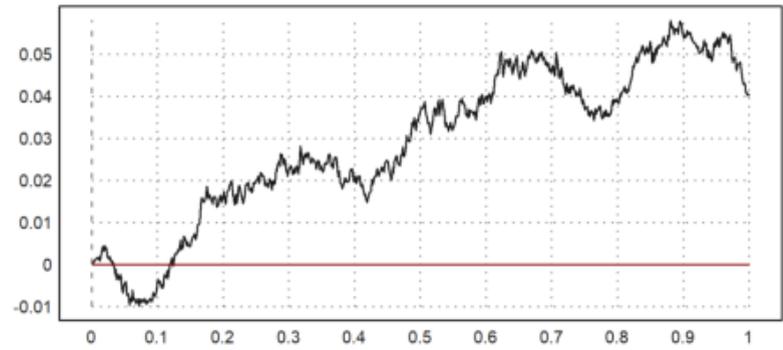
```
[0.435253441872, -0.176104899977, -0.261357250134, -0.300490662125,  
0.24735601816, 0.0325482814611, 0.134443809807, 0.215276338532,  
0.616508683891, 0.00182405931929, -0.102144906616, 0.488148985802,  
0.528398540076, -0.449171007092, 0.991667101405, 0.144211079323,  
-0.0535320522794, -0.256568161152, -0.879827440262, 0.156829348155,  
-0.3903055246, -0.7704010459, 0.0381372856622, -0.383159326089,  
0.109191765182, 0.388165202221, -0.750095512267, 0.382653811756,  
-0.00962159593649, -0.318261417586, 0.1566434701, 0.0417031284853,  
-0.105544375746, -0.0986565391085, -0.308098598401,  
-0.173042729299, -0.570212950124, 0.0654283409697, -0.224582299423,  
-0.0608214213517, -0.219873405411, -0.371499521187,  
-0.533088259816, -0.107424976374, 0.248576720805, -0.533868235423,  
0.0531034844935, -0.160323194089, -0.125487020166, -0.158491044432,  
0.0906761537006, -0.191710797259, -0.0558522371754, 0.184837248944,  
0.268383527011, 0.218998347285, 0.104357767512, 0.277997939119,  
-0.34503501907, 0.289852317409, 0.224763386537, 0.182353125513,
```

```
-0.0607878810754, 0.152837024488, -0.197329835927, -0.43566146825,  
0.272175590785, -0.00637497767505, -0.121998837954,  
-0.0614737876364, 0.668085161915, 0.22372847095, 0.805148406755,  
0.32314121608, 0.581489706016, 0.466375493432, -0.480865090272,  
... ]
```

`median(M)` akan mengembalikan vektor baris atau kolom yang berisi median dari setiap kolom (untuk matriks). Artinya, setiap elemen di vektor hasil adalah median dari nilai dalam satu kolom dari `M`. Tanda `(')` berarti transpos matriks atau vektor. Jadi, `median(M)'` akan melakukan transpos terhadap hasil dari `median(M)`. Jika hasilnya adalah vektor kolom, maka transpos akan mengubahnya menjadi vektor baris, dan sebaliknya.

Karena kita dapat membuat jalan acak, kita dapat mensimulasikan proses Wiener, yaitu Proses Wiener adalah istilah dalam teori probabilitas yang merujuk pada model matematis yang digunakan untuk menggambarkan gerak acak. Proses ini juga dikenal sebagai Brownian motion dalam konteks fisika dan merupakan contoh dari proses stokastik kontinu. Kita mengambil 1000 langkah dari 1000 proses. kemudian memplot deviasi standar dan mean dari langkah ke-n dari proses ini bersama dengan nilai yang diharapkan berwarna 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):
```



7. Uji Statistik

Soal 1

Kami menguji lemparan dadu untuk distribusi seragam. Pada 600 lemparan, kami mendapatkan nilai berikut, yang kami masukkan ke dalam uji chi-square.

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

0.498830517952

Data pengamatan atau observasi yang diperoleh adalah 90,103,114,101,103,89. Jika tipe pelemparan dadu tersebut sama, maka proporsi pelemparan berada dalam masing-masing katgori, yaitu

$$\frac{1}{k} = \frac{1}{6} = 0.16$$

Sehingga diperoleh frekuensi harapan (e_i) yaitu

$$e_i = \frac{600}{6} = 100$$

Hipotesis

$$H_0 : p_1 = 0.16, p_2 = 0.16, \dots, p_6 = 0.16$$

H_1 : terdapat p_i tidak sama dengan

$$p_{i0}, i = 1, 2, 3, 4, 5, 6$$

Taraf signifikansi:

$$\alpha = 0.05$$

Statistik uji

$$X^2 = \sum_{i=1}^k \frac{(n_i - e_i)^2}{e_i}$$

kriteria keputusan:

$$k = 4, X_{0.05(5)}^2 = 11.070$$

$$H_0 \text{ ditolak jika } X^2 > 11.070$$

Sekarang kita akan melakukan statistik uji :

$$X^2 = \sum_{i=1}^6 \frac{(n_i - e_i)^2}{e_i}$$

$$\frac{(90 - 100)^2}{100} + \frac{(103 - 100)^2}{100} + \frac{(114 - 100)^2}{100} \\ + \frac{(101 - 100)^2}{100} + \frac{(103 - 100)^2}{100} + \frac{(89 - 100)^2}{100}$$

$$X^2 = 1 + 0.09 + 1.96 + 0.01 + 0.09 + 1,21$$

$$X^2 = 4.36$$

oleh karena

$$X^2 = 4.36 < 11.070$$

dan

$$p\text{-value} = 0.499 > 0.05$$

maka H_0 tidak ditolak. pada taraf signifikansi $\alpha = 0.05$, tidak ada bukti untuk menyimpulkan bahwa ada proporsi sebenarnya yang berbeda dari 0.16.

Uji chi-square juga memiliki mode, yang menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya harusnya hampir sama. Parameter $> p$ mengartikan vektor y sebagai vektor probabilitas.

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

0.492

Selanjutnya kita menghasilkan 1000 lemparan dadu menggunakan generator nomor acak, dan melakukan tes yang sama.

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

0.498830517952

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

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

0.0907768397616

Fungsi ttest () membutuhkan nilai mean, deviasi, jumlah data, dan nilai mean untuk diuji.

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

```
>s=200+normal([1,100])*10
```

```
[205.134237952, 181.459472546, 189.883709832, 190.650150651,  
182.992896622, 199.716975332, 197.649230255, 191.275243109,  
204.124274924, 189.262548626, 201.288617094, 192.408414496,  
204.322957504, 207.982650102, 195.456674205, 207.562085016,  
196.768141551, 195.684808851, 204.835501168, 196.806104734,  
190.084758827, 212.17007018, 202.827337816, 195.380418919,  
188.858845943, 194.789067484, 210.306584426, 198.937250421,  
188.491347785, 199.421003144, 191.222978739, 196.641930052,  
207.048444856, 202.659606299, 191.878347269, 189.218514572,  
190.421352736, 188.240022843, 190.348998078, 198.82840212,  
200.548198201, 204.232359106, 210.506587654, 185.758446233,  
189.761186838, 196.884152235, 205.28902548, 198.231505731,  
195.526091309, 193.89540334, 188.82696811, 218.662975463,  
191.206664417, 191.567663312, 206.206577732, 193.475418822,  
197.950073245, 210.206639062, 192.61932733, 202.105327856,
```

```
194.328756145, 203.125388271, 195.958512822, 199.432721805,  
199.738321478, 201.349880092, 196.809772008, 201.085343084,  
202.620827665, 199.828795593, 203.107721193, 194.170224432,  
208.561554785, 192.424581684, 191.041761027, 199.855401062,  
197.476553751, 190.108875933, 196.531322576, 196.535434517,  
... ]
```

```
>mean(s)
```

```
198.144100912
```

```
>dev(s)
```

```
7.74285944682
```

Uji t adalah Uji t adalah uji statistik yang digunakan untuk membandingkan rata-rata sampel dengan nilai yang diharapkan (misalnya rata-rata populasi) atau untuk membandingkan rata-rata dua sampel.

Adapun cara menghitung uji-t adalah :

1. menentukan hipotesis

$$H_0 : \text{rata-rata sampel} = 200$$

$$H_1 : \text{rata-rata sampel tidak sama dengan } 200$$

2. menentukan signifikansi

$$\alpha = 0.05$$

3. menghitung statistik uji :

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

dengan

\bar{x} = rata-rata sampel

μ = rata-rata populasi atau yang diharapkan

s = deviasi standar sampel

n = jumlah sampel

sehingga

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$
$$t = \frac{199.225 - 200}{\frac{10.757}{\sqrt{100}}}$$
$$t = \frac{-0.775}{1.0757}$$
$$t = -0.720$$

Untuk tingkat signifikansi $\alpha=0.05$ dan uji dua sisi, kita menggunakan tabel distribusi t untuk menemukan nilai kritis t. Berdasarkan tabel distribusi t, untuk $df=99$ dan $\alpha=0.05$, nilai kritis t adalah sekitar ± 1.984 . Berdasarkan distribusi t, untuk $t=-0.720$ dan $df=99$, p-value yang didapat adalah sekitar 0.3414. Dengan $t = -0.720$ dan $p\text{-value} > 0.05$, kita gagal menolak hipotesis nol, yang berarti rata-rata sampel tidak berbeda signifikan dari nilai yang diuji, yaitu 200.

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

```
0.424330572317
```

Fungsi `tcomparedata()` digunakan untuk membandingkan dua set data yang diberikan. Fungsi ini mengasumsikan bahwa kedua set data yang dibandingkan berasal dari distribusi yang sama dan melakukan uji perbandingan, seperti uji t, untuk melihat apakah terdapat perbedaan yang signifikan antara kedua set data tersebut. `tcomparedata(normal(1,10), normal(1,10))` akan membandingkan dua sampel acak yang dihasilkan dari distribusi normal dengan parameter rata-rata 1 dan simpangan baku 10.

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

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

```
[0.599099171693, 0.710625525652, 0.801693167767, 0.18975652142,  
-0.376684471198, -1.0565243843, -0.0592447416325, 0.881040750385,  
0.223002740928, -2.20122187666]
```

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

```
[-1.5631635179, 0.474324292439, 0.763629818294, -0.237197611902,  
-0.147692671664, -0.173721808923, -0.0904542579489, -2.63455245895,  
0.804836131829, -0.737197224828]
```

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

3.29

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

```
>R=random(100,20)
```

Real 100 x 20 matrix

```
0.545443582052    0.623426478352    ...  
0.290613483033    0.438596926295    ...  
0.107069179463    0.293072621297    ...  
0.32384719957     0.354241092275    ...  
0.373353767551    0.745716822513    ...  
0.0927492719533    0.9391798027      ...  
0.596919074067    0.674429703656    ...  
0.150387363873    0.377238402639    ...  
0.077026109466    0.501311899448    ...  
0.0359313600018    0.546165720757    ...  
0.323359847295    0.406159059388    ...  
0.905401971638    0.135309763021    ...  
0.799028983301    0.0578931253035    ...
```

```

0.465905837529    0.862183393578    ...
0.420251528603    0.745195364739    ...
0.92985517254     0.404850288021    ...
0.884137105267    0.164770572449    ...
0.986710986993    0.71549398071     ...
0.73124179065     0.0489269686671   ...
0.878058770036    0.965069827172    ...
...

```

Fungsi `random(100, 20)`: Fungsi ini menghasilkan sebuah matriks acak dengan ukuran 100 x 20, yang berarti 100 baris dan 20 kolom. Setiap elemen dalam matriks R adalah angka acak dalam interval (0, 1).

```
>R=sum(R*6<=1)'
```

```

[3, 4, 5, 4, 4, 5, 1, 7, 3, 4, 1, 7, 7, 3, 2, 3, 4, 2,
5, 1, 6, 3, 2, 2, 5, 1, 5, 2, 5, 2, 1, 4, 5, 5, 0, 2,
6, 2, 3, 4, 3, 5, 6, 2, 1, 4, 3, 4, 5, 5, 4, 1, 3, 3,
7, 3, 4, 3, 3, 3, 3, 1, 1, 3, 4, 4, 1, 2, 3, 4, 3, 3,
7, 2, 1, 2, 4, 6, 4, 3, 4, 3, 3, 3, 7, 6, 3, 7, 5, 2,
3, 2, 1, 2, 5, 3, 6, 4, 3, 2]

```

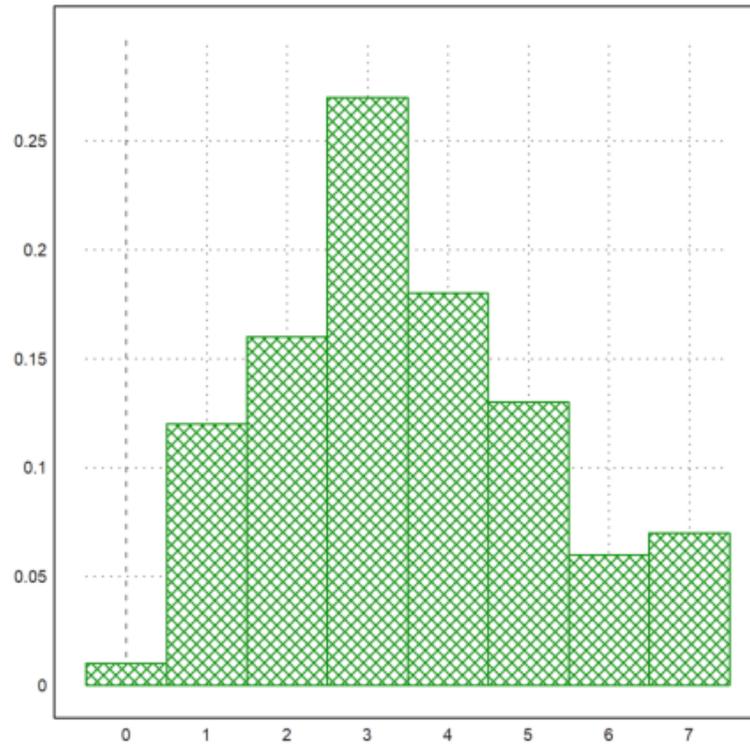
`sum(R*6<=1)`: Fungsi `sum` menghitung jumlah nilai True (atau 1) di setiap kolom dari matriks yang memenuhi kondisi $R \times 6 \leq 1$. Dengan kata lain, fungsi ini menghitung berapa banyak elemen dalam setiap kolom yang memenuhi kondisi tersebut. Tanda `'` (transpose): Setelah itu, hasil penjumlahan ini ditranspose (diubah dari vektor baris menjadi vektor kolom). Hasilnya adalah vektor kolom yang berisi jumlah elemen yang memenuhi kondisi $R \times 6 \leq 1$ untuk setiap kolom matriks R.

```
>mean(R)
```

3.47

mean(R): Fungsi mean menghitung nilai rata-rata dari elemen-elemen dalam vektor kolom R.

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



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

`count(R, 21)`: Fungsi `count` menghitung jumlah kemunculan nilai 21 dalam array `R`. Jadi, jika ada elemen-elemen di dalam array `R` yang bernilai 21, perintah ini akan mengembalikan jumlah berapa kali angka 21 muncul dalam `R`.

Kemudian kami menghitung nilai yang diharapkan.

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

Perintah ini menghasilkan sebuah array `b` yang berisi probabilitas dalam persen untuk setiap jumlah kemunculan angka "1" dari 0 hingga 20 kali dalam 20 lemparan dadu.

Kami harus mengumpulkan beberapa nomor 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])
```

```
[13.0420266523, 19.8238805115, 23.7886566138, 20.2203581217,  
12.9410291979, 6.47051459895, 3.71353430387]
```

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

```
0.628342473673
```

Soal 2

Data berikut berisi hasil dari dua kelompok orang (misalnya laki-laki dan perempuan) 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. Tes tabel χ^2 melakukan ini. Hasilnya adalah cara yang besar untuk menolak kemerdekaan. Jadi kami tidak bisa mengatakan, apakah voting tergantung jenis kelamin dari data ini.

dari data tersebut, nilai n total i dan j adalah

$$n_{11} = 293$$

$$n_{21} = 295$$

$$n_{i1} = 50$$

$$n_{i2} = 76$$

$$n_{i3} = 84$$

$$n_{i4} = 101$$

$$n_{i5} = 127$$

$$n_{i6} = 150$$

n=11176

perhitungan :

$$e_{11} = \frac{50 \times 293}{1176}$$

$$e_{11} = 12.457$$

$$e_{21} = \frac{50 \times 295}{1176}$$

$$e_{21} = 12.543$$

$$e_{21} = \frac{76 \times 293}{1176}$$

$$e_{21} = 18.935$$

$$e_{22} = \frac{76 \times 295}{1176}$$

$$e_{22} = 19.065$$

$$e_{31} = \frac{84 \times 293}{1176}$$

$$e_{31} = 20.929$$

$$e_{32} = \frac{84 \times 295}{1176}$$

$$e_{32} = 21.071$$

$$e_{41} = \frac{101 \times 293}{1176}$$

$$e_{41} = 25.164$$

$$e_{42} = \frac{101 \times 295}{1176}$$

$$e_{42} = 25.336$$

$$e_{51} = \frac{127 \times 293}{1176}$$

$$e_{51} = 31.642$$

$$e_{52} = \frac{127 \times 295}{1176}$$

$$e_{52} = 31.858$$

$$e_{61} = \frac{150 \times 293}{1176}$$

$$e_{62} = 37.372$$

$$e_{62} = \frac{150 \times 295}{1176}$$

$$e_{62} = 37.628$$

Taraf signifikansi

$$\alpha = 0.01$$

Kriteria keputusan :

$$(I - 1)(J - 1) = (2 - 1)(6 - 1) = 5, X_{0.01,5}^2 = 15.086$$

Statistik Uji :

$$X^2 = \sum_{i=1}^I \sum_{j=1}^J \frac{(n_{ij} - e_{ij})^2}{e_{ij}}$$

$$X^2 = \frac{(23 - 12.457)^2}{12.457} + \frac{(27 - 12.543)^2}{12.543} + \frac{(37 - 18.935)^2}{18.935}$$

$$+ \frac{(39 - 19.065)^2}{19.065} + \frac{(43 - 20.929)^2}{20.929} + \frac{(41 - 21.071)^2}{21.071}$$

$$+ \frac{(52 - 25.164)^2}{25.164} + \frac{(49 - 25.336)^2}{25.336} + \frac{(64 - 31.642)^2}{31.642}$$

$$+ \frac{(63 - 31.858)^2}{31.858} + \frac{(74 - 37.372)^2}{37.372} + \frac{(76 - 37.628)^2}{37.628}$$

$$X^2 = 8.923 + 16.663 + 17.235 + 20.845 + 23.275 + 18.849$$

$$+ 28.619 + 22.102 + 33.090 + 30.442 + 35.899 + 39.130$$

$$X^2 = 295.072$$

karena nilai statistik uji $X^2=295.072 > 15.086$, maka H_0 ditolak, sehingga pada taraf signifikansi 0.05 dapat disimpulkan bahwa ada bukti bahwa suara laki-laki dan perempuan adalah dependen.

```
>tabletest(A)
```

```
0.990701632326
```

Berikut 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 yang dikoreksi. Karena sangat mendekati 0, kami menyimpulkan bahwa pemungutan suara tidak bergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```

Soal 3

Bagi peternak lebah yang ingin memanen lebih banyak madu, ada empat kemungkinan untuk mendapatkan lebih banyak lebah: package bees, nucs, colonies, swarms. Departemen ilmu pertanian dari universitas tertentu memperoleh sampel acak dari pembelian lebah, dan masing-masing diklasifikasikan ke dalam salah satu dari empat kategori tersebut. Gunakan tabel frekuensi satu arah berikut untuk menguji hipotesis bahwa empat kemungkinan pembelian lebah terjadi dengan frekuensi yang sama. Gunakan taraf signifikansi $\alpha = 0.05$

```
>chitest([31,36,26,20],dup(28.25,4)')
```

0.173097757603

Diperoleh

$$p\text{-value} = P(X^2 > 4.9823) = 0.1731$$

Oleh karena

$$X^2 = 4.9823 < 7.8147 \text{ atau } p\text{-value} = 0.1731 > 0.05$$

maka H_0 tidak ditolak. Pada taraf signifikansi $\alpha = 0.05$, tidak ada bukti untuk menyimpulkan bahwa ada proporsi sebenarnya yang berbeda dari 0.25. Proporsi tipe pembelian lebah itu sama.

Soal 4

National Advisory Committee bagian imunisasi di Canada menyediakan rekomendasi kesehatan tertentu dan laporan ringkasan. Suatu sampel acak pasien-pasien yang dites positif influenza selama musim flu tahun 2012-2013 dicatat. Masing-masing orang diklasifikasikan dalam grup umur dan jenis flu. Ringkasan data disajikan dalam tabel berikut:

```
>B=[207,849,1592,617;66,630,754,895;319,1207,1617,512;300,1196,1946,510;120,3599,5860,597]; ...  
> writetable(B,wc=6,labr=["KDL","LEP","DPEE","ELEE","ELP"],labc=1:4)
```

	1	2	3	4
KDL	207	849	1592	617
LEP	66	630	754	895
DPEE	319	1207	1617	512
ELEE	300	1196	1946	510
ELP	120	3599	5860	597

```
>contingency(B)
```

```
0.359159327147
```

```
>writetable(expectedtable(B),wc=6,dc=1,labr=["KDL","LEP","DPEE","ELEE","ELP"],labc=1:4)
```

	1	2	3	4
KDL	141.21044	.11642	.6	437
LEP	101.4	749.91179	.8	313.9
DPEE	158.11168	.91838	.8	489.2
ELEE	1711263	.81988	.2	528.9
ELP	440.23254	.25119	.5	1362

```
>contingency(expectedtable(B))
```

0

Oleh karena ($p\text{-value} < 0.01$) maka H_0 ditolak, sehingga pada taraf signifikansi 0.01 dapat disimpulkan bahwa ada bukti bahwa grup umur dan jenis influenza adalah dependen.

Soal 4

Suatu studi penelitian menunjukkan bahwa diet tinggi garam pada wanita tua meningkatkan risiko patah tulang. Meskipun mekanisme biologisnya masih tidak jelas, namun terlihat ada hubungan antara asupan natrium yang berlebihan dan kerapuhan tulang. Salah satu ukuran kesehatan tulang adalah kadar vitamin D dalam darah. Seperti yang ditentukan oleh kuesioner makanan, sampel acak independen dari wanita tua dalam empat kategori asupan garam diperoleh. Kadar vitamin D dalam darah (dalam nmol / L) diukur di masing-masing. Data diberikan dalam tabel berikut. Adakah bukti yang menunjukkan bahwa setidaknya dua dari rata-rata populasi kadar vitamin D dalam darah berbeda? Gunakan $\alpha = 0.05$.

```
>a1=[91.5,77.5,94.5,77.5,92.0]; mean(a1),
```

86.6

```
>a2=[89.0,92.0,98.2,80.0,86.7]; mean(a2),
```

89.18

```
>a3=[92.5,100.7,94.0,93.3,106.3]; mean(a3),
```

97.36

```
>a4=[100.1,98.0,99.1,103.9,97.6]; mean(a4),
```

99.74

```
>varanalysis(a1,a2,a3,a4)
```

0.0116566288294

Oleh karena $F = 5.08 > 3.24$ dan $p\text{-value} < 0.05$ maka H_0 ditolak. Jadi pada taraf signifikansi 0.05 dapat disimpulkan ada bukti bahwa minimal ada dua rata-rata populasi kadar vitamin D dalam darah yang berbeda.

Soal 5

Dengan taraf nyata 5% ujilah apakah (peringkat) pendapatan di departemen Q lebih kecil dibandingkan departemen Z?

```
>p1=[6,10,15,32];  
>p2=[12,13,15,15,20,31,38,40];  
>ranktest(p1,p2)
```

0.117241303547

Karena didapat z hitung = -1.19 ada di daerah penerimaan H_0 maka H_0 diterima, sehingga peringkat Pendapatan di kedua departemen sama.

Soal 6

Untuk melihat apakah ada perbedaan produksi per hektar tanaman jagung karena pengaruh dua metode penanaman yang digunakan, pertumbuhan tanaman jagung dipilih dari sejumlah plottanah yang berbeda secara random. Kemudian produksi per hektar dari masing-masing plot dihitung dan hasilnya adalah sebagai berikut: ($\alpha = 5\%$)

Metode 1 : 83 91 94 89 96 91 92 90 92 85

Metode 2 : 91 90 81 83 84 83 88 91 90 84 80 85

Apakah dua metode tersebut memiliki nilai median yang sama?

```
>b1=[83,91,94,89,96,91,92,90,92,85];  
>b2=[91,90,81,83,84,83,88,91,90,84,80,85];  
>mediantest(b1,b2)
```

0.985013438574

Harga Chi kuadrat tabel dk =1 dan alpha 5%= 3.841 karena Chi hitung < chi kuadrat tabel maka H₀ tidak ditolak. Sehingga disimpulkan dua metode mempunyai nilai median yang sama untuk produksi per hektar.