

23030630053  
Hamemayu Hayuningrat

**EMT untuk Statistika**

---

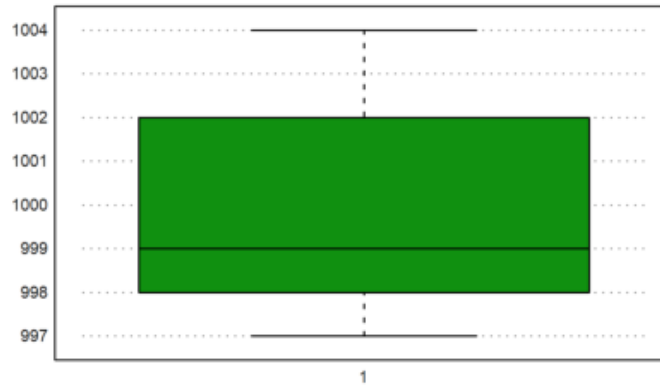
Dalam notebook ini, kami akan mendemonstrasikan plot utama statistika, uji dan distribusi dalam Euler. Mari kita mulai dengan beberapa deskriptif statistik. Ini bukanlah pengantar untuk statistik. Jadi kamu mungkin membutuhkan beberapa latar belakang untuk memahami secara detail. Asumsikan ukurn mengikuti berikut. Kita berharap untuk menghitung nilai rata-rata dan menghitung standar deviasi.

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

```
999  
999.9  
2.72641400622
```

Kami dapat membuat plot box-and-whiskers untuk data. Dalam kasus kami dimana tidak ada data yang keluar.

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



Kami menghitung kemungkinan bahwa sebuah nilai lebih dari 1005, asumsikan hitungan nilai dari sebuah distribusi normal.

Semua fungsi untuk distribusi-distribusi dalam Euler dan dengan ...dis dan menghitung cumulative probability distribution (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 hasil dalam % dengan akurasi 2 digit menggunakan fungsi print.

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

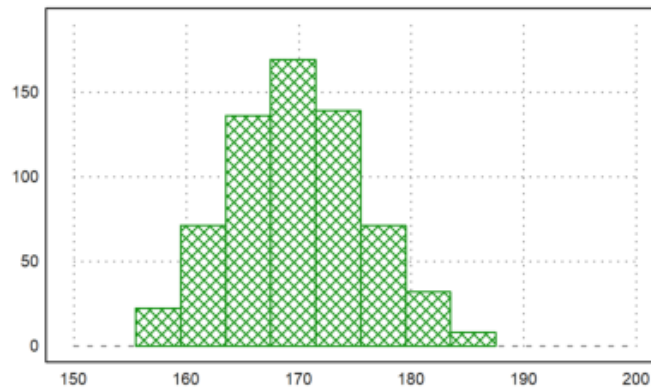
3.07 %

Untuk contoh selanjutnya, kami asumsi angka-angka berikut dari pria dengan diberikan sebuah rentang ukuran.

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

Ini merupakan sebuah plot distribusi.

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



Kami dapat mengambil sebarang data mentah ke sebuah tabel.

Tables merupakan sebuah method untuk menyimpan data statistik. Tabel kami seharusnya mengandung tiga kolom: rentang mulai, rentang akhir, banyaknya pria dalam rentang.

Tables dapat juga dicetak dengan headers. Kami menggunakan vektor string untuk mengatur headers.

```
>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 kami menggunakan nilai rata-rata dan statistik lain dari ukuran, kami membutuhkan untuk menghitung titik tengah dari rentang. Kami dapat menggunakan kolom kedua pertama dari tabel kami untuk ini.

Simbol "|" digunakan untuk memisahkan kolom, fungsi "writetable" digunakan untuk menulis tabel, dengan pilihan "labc" adalah secara spesifik kolom headers.

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

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

Tetapi ini 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 menghitung rata-rata dan deviasi dari contoh dengan frekuensi yang diberikan.

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

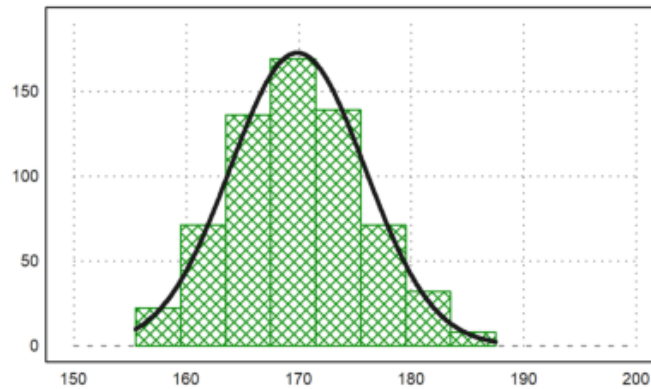
```
169.901234568  
5.98912964449
```

Mari kita tambahkan distribusi normal dari nilai nilai menjadi plot batang diatas. Formula untuk distribusi normal dengan rata-rata  $m$  dan standar deviasi  $d$  adalah:

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

Karena ini nilai-nilai diantara 0 dan 1, untuk membuat plot itu dalam plot batang kita harus mengalikan dengan 4 kali jumlahan angka dari data.

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



Dalam direktori dari notebook ini kamu menemukan sebuah file dengan sebuah tabel. Data tersebut menunjukkan hasil dari sebuah survey. Disini merupakan empat baris pertama dari file. Data berasal dari sebuah 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 dari angka-angka atau token (string). Kami ingin membaca tabel dari file. Pertama, kami menggunakan translasi sendiri untuk tokens.

Untuk ini, kami mendefinisikan himpunan dari token. Fungsi `strtokens()` mengambil sebuah vektor string dari token dari sebuah string yang diberikan.

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



Sekarang kita membaca tabel dengan translasi berikut.

Argumen-argumen tok2, tok4, dll merupakan translasi dari kolom tabel. Argumen-argumen ini tidak dalam daftar parameter dari readtable(), jadi kamu butuh untuk menyediakan mereka dengan ":=".

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

Untuk mencetak, kami membutuhkan secara spesifik himpunan token yang sama. Kami mencetak empat baris pertama saja.

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

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

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

Jika kita tidak ingin untuk secara spesifik token untuk translasi lebih lanjut, kami hanya butuh untuk secara spesifik, yang mana kolom berisi token dan bukan angka-angka.

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

Sekarang fungsi readtable() mengembalikan sebuah himpunan dari token.

```
>tok
```

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

Tabel berisi isian dari file dengan tokens yang diubah ke angka.

String spesial NA="." diinterpretasikan sebagai "Not Available" dan menghasilkan NAN (Not A Number) dalam tabel. Translasi ini dapat diubah dengan parameter-parameter NA dan NAval.

```
>MT[1]
```

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

Disini merupakan isi tabel dengan angka-angka yang tidak diubah.

```
>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 lebih jelas, kamu dapat mengambil keluaran dari readtable() ke sebuah list.

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

Menggunakan kolom-kolom token yang sama dan token token membaca dari file, kami dapat mencetak tabel. Kami dapat juga secara spesifik ctok, tok, dll. atau menggunakan list Table.

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

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

19	f	29	n	.	3.8	n
20	f	28	y	vg	1.8	n
21	f	28	y	m	2.8	y
22	f	28	y	vg	1.8	y
23	f	38	y	g	2.8	n
24	f	27	y	m	1.8	y
25	m	27	n	.	2.8	y

Fungsi `tablecol()` mengembalikan nilai-nilai dari kolom tabel. Melewati sebarang baris dengan nilai-nilai("." dalam file) NAN, dan indeks-indeks dari kolom yang mana berisi nilai-nilai ini.

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

Kami dapat menggunakan ini untuk mengekstraksi kolom-kolom dari tabel untuk sebuah 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, kami butuh untuk mengekstraksi tabel itu sendiri dari Table list dalam kasus ini.

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

Of course, we can also use it to determine the mean value of a column or any other statistical value.

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

2.175

Fungsi `getstatistics()` mengembalikan elemen-elemen dalam sebuah vektor dan hitungannya. Kami menerapkan ke nilai "m" dan "f" dalam kolom kedua dari tabel kami

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

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

Kami dapat mencetak hasil dalam sebuah tabel baru.

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

```
      m      12  
      f      13
```

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

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

```
[5, 6]
```

Sekarang kami dapat memilih baris-baris dari tabel, yang mana memiliki sebarang nilai dari v dalam baris ke-5 mereka.

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

Sekarang kami dapat mencetak tabel, dengan nilai-nilai yang diekstraksi dan diurutkan dalam kolom ke-5.

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

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



Untuk statistik selanjutnya, kami ingin untuk menghubungkan dua kolom dari tabel. Jadi kami mengekstrak kolom 2 dan 4 dan mengurutkan tabel.

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

[illegible]

Dengan `getstatistics()`, kami dapat juga menghubungkan hitungan dalam dua kolom dari tabel pada tiap-tiap yang 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

Sebuah tabel dapat juga ditulis ke sebuah file.

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

Lalu kami dapat membaca tabel dari file.

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

	n	y
m	7	5
f	1	12

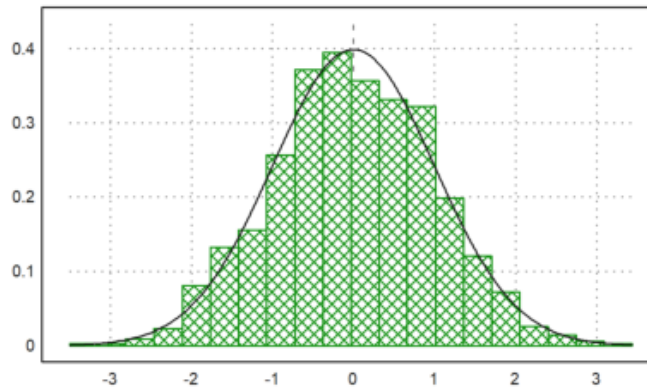
Dan menghapus file.

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

## Distribusi-Distribusi

Dengan plot2d, ini merupakan method paling mudah untuk membuat plot sebuah distribusi dari data eksperimental.

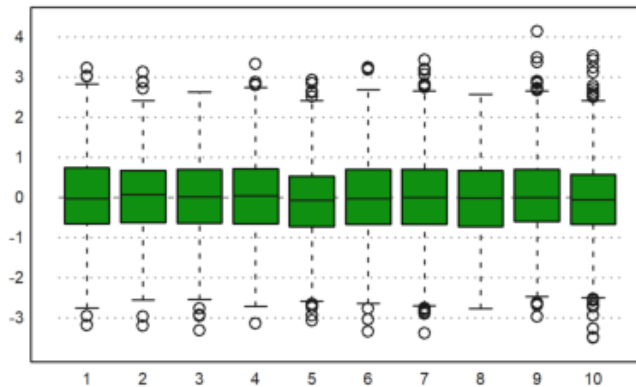
```
>p=normal(1,1000); //1000 random normal-distributed sample p  
>plot2d(p,distribution=20,style="/"); // plot the random sample p  
>plot2d("qnormal(x,0,1)",add=1): // add the standard normal distribution plot
```



Tolong catat perbedaan antara plot batang (contoh) dan kurva normal (distribusi real). Enter kembali tiga perintah untuk melihat hasil lainnya.

Ini merupakan sebuah perbandingan dari 10 simulasi dari 1000 nilai berdistribusi normal menggunakan sebuah yang dipanggil box plot. Plot ini menunjukkan median, quartil 25% dan 75%, nilai minimal dan maksimal dan percilan.

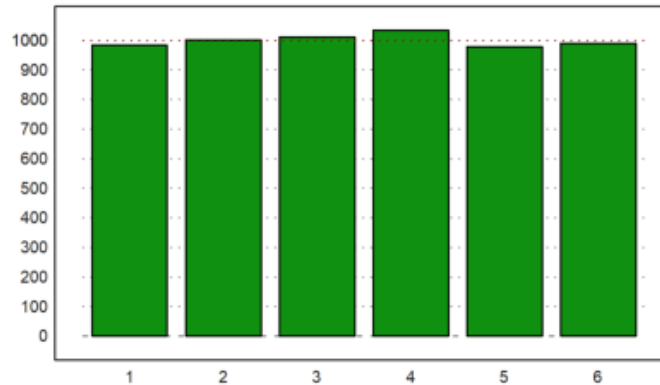
```
>p=normal(10,1000); boxplot(p):
```



Untuk menghasilkan bilangan acak, Euler memiliki intranomial. Mari kita simulasikan lemparan dadu dan plot distribusinya.

Kami menggunakan fungsi `getmultiplicities(v,x)`, yang mana menghitung seberapa sering elemen dari `v` muncul dalam `x`. Lantas kami membuat plot dari hasilnya menggunakan `columnplot()`.

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



Ketika `inrandom(n,m,k)` mengembalikan bilangan bulat secara seragam berdistribusi dari 1 ke k, ini memungkinkan untuk menggunakan sebarang lainnya diberikan distribusi dari bilangan bulat dengan `randpint()`.

Contoh berikut, peluang untuk 1,2,3 adalah 0.4,0.1,0.5 secara berurutan.

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

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

Euler dapat menghasilkan nilai-nilai acak dari distribusi-distribusi lainnya. Lihatlah ke referensi berikut.

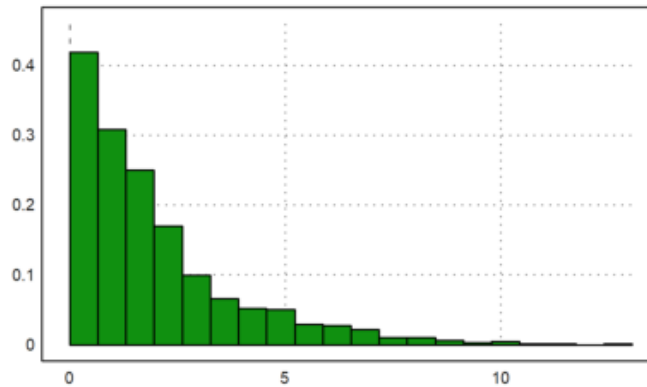
Contoh, kami mencoba distribusi eksponensial. Sebuah variabel kontinu acak X dikatakan memiliki distribusi eksponensial, jika PDF diberikan sebagai

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

dengan parameter

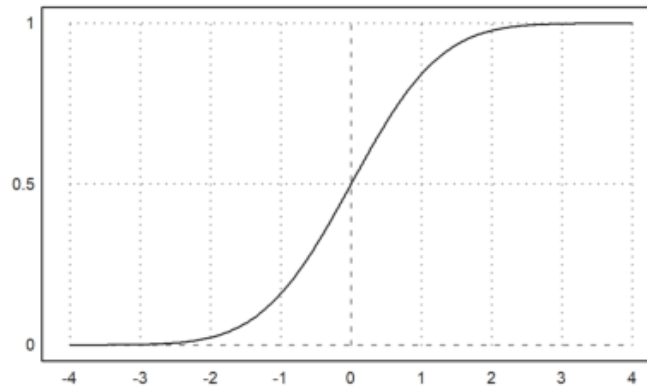
$\lambda = \frac{1}{\mu}$ ,  $\mu$  is the mean, and denoted by  $X \sim \text{Exponential}(\lambda)$ .

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



Untuk distribusi lainnya, Euler dapat menghitung fungsi distribusi dan sebaliknya.

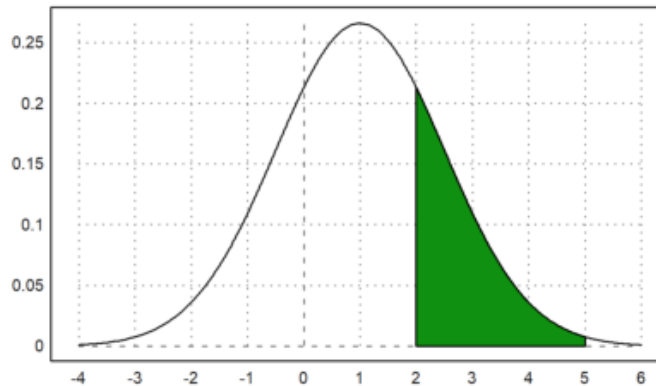
```
>plot2d("normaldis",-4,4):
```



Berikut merupakan sebuah cara untuk membuat plot sebuah quantile.

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





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

Peluang dalam luasan hijau seperti 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 dari rata-rata yang sama dan deviasi. Fungsi `invbindis()` menyelesaikan sebuah interpolasi linear antara nilai-nilai bilangan bulat.

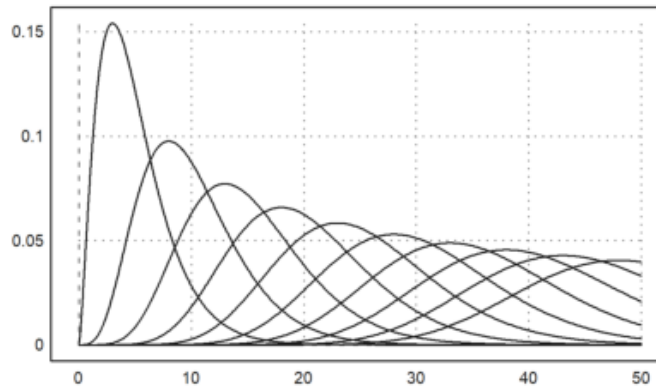
```
>invbindis(0.95,1000,0.5), invnormaldis(0.95,500,0.5*sqrt(1000))
```

```
525.516721219
```

```
526.007419394
```

Fungsi `qdis()` adalah sebuah kepadatan dari distribusi chi-square. Seperti biasa, Euler memetakan vektor-vektor ke fungsi ini. Lantas kami mendapatkan sebuah plot dari semua distribusi chi-square dengan derajat 5 ke 30 secara mudah mengikuti langkah berikut.

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



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

Penamaan mencoba untuk lebih konsisten. Contoh,

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

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

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

```
0.527633447259
```

```
0.527633447259
```

## Distribusi-Distribusi Diskret

---

Untuk mendefinisikan distribusi diskret milik anda, kamu dapat menggunakan method berikut. Pertama kami 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]
```

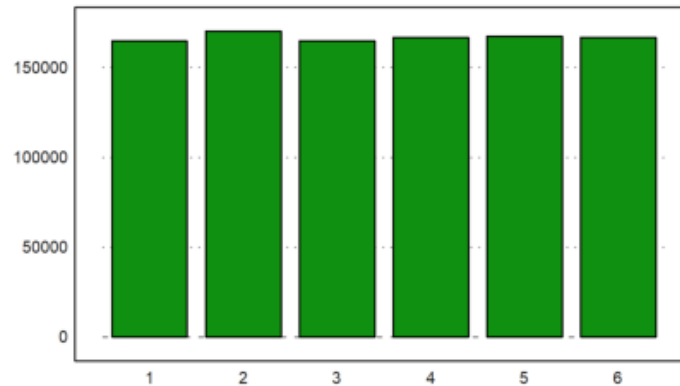
Arti ini bahwa dengan peluang  $wd[i+1]-wd[i]$  kami menghasilkan nilai  $i$  acak.

Ini hampir sebuah distribusi seragam. Mari kita definisikan sebuah pencetak angka acak untuk ini. Fungsi `find(v,x)` menemukan nilai  $x$  dalam vektor  $v$ . Ini bekerja untuk vektor  $x$  juga.

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

Error merupakan jelas bahwa kami melihatnya hanya dengan iterasi banyak sekali.

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



Ini merupakan fungsi sederhana untuk memeriksa untuk distribusi seragam dari nilai-nilai 1...K dalam v. Kami menerima hasil ini, 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
```

Tentusaja fungsi akan menolak distribusi seragam.

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

0

Dan ini menerima pencetak acak built-in.

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

1

Kami dapat menghitung distribusi binomial. Pertama ini adalah `binomialsun()`, yang mana mengembalikan peluang dari  $i$  kurang dari  $n$  percobaan.

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

0.751401349654

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

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

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

```
[0.37932, 0.441212]
```

Perintah-perintah berikut merupakan cara langsung untuk mendapatkan hasil diatas. Tetapi untuk  $n$  besar, penjumlahan langsung tidak akurat dan lambat.

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

```
0.751401349655
```

Selain itu, `invbinsum()` menghitung invers dari `binomsum()`.

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

```
409.932733047
```

Dalam Bridge, kami asumsikan 5 kartu yang keluar (dari 52) dalam dua tangan (26 kartu), kami mengasumsikan peluang dari sebuah distribusi yang buruk dari 3:2 (contohnya 0:5, 1:4, 4:1 atau 5:0).

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

```
0.321739130435
```

Ini juga sebuah simulasi dari 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



## Membuat Plot Data

---

Untuk memplot data, kami mencoba hasil dari pemilihan Jerman sejak 1990, dihitung dalam tempat duduk.

```
>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 partisi-partisi, kami gunakan sebuah string dari nama-nama.

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

Mari kita cetak presentase secara bagus.

Pertama kami ekstrak kolom-kolom yang dibutuhkan. Kolom 3 ke 7 merupakan tempat duduk dari setiap partai, dan kolom 2 merupakan banyaknya tempat duduk. Kolom merupakan tahun pemilihan.

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

Lalu kami mencetak statistik dalam bentuk tabel. Kami menggunakan nama-nama seperti kolom headers dan tahun sebagai headers untuk baris-baris. Lebar bawaan untuk kolom adalah  $wc=10$ , tapi kami menyarankan output yang lebih padat. Kolom-kolom akan di ekspansi untuk label-label dari kolom-kolom, jika dibutuhkan.

```
>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

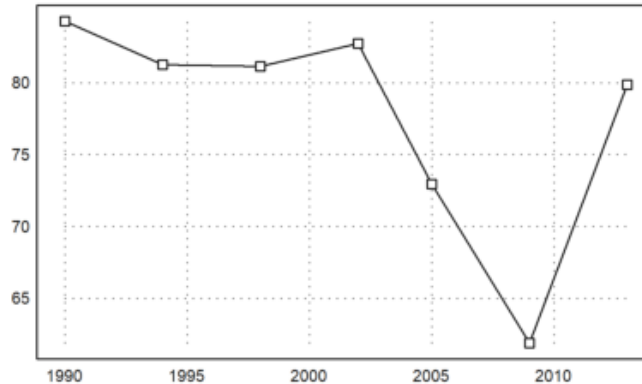
Perkalian matriks berikut ekstrak penjumlahan dari persentase dari dua partai besar menunjukkan partai kecil mendapatkan sisa dari parlemen hingga 2009.

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

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

Ini juga merupakan sebuah plot statistik sederhana. Kami menggunakan ini untuk menampilkan garis dan titik secara simultan. Cara lainnya adalah memanggil plot2d dua kali dengan >add.

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

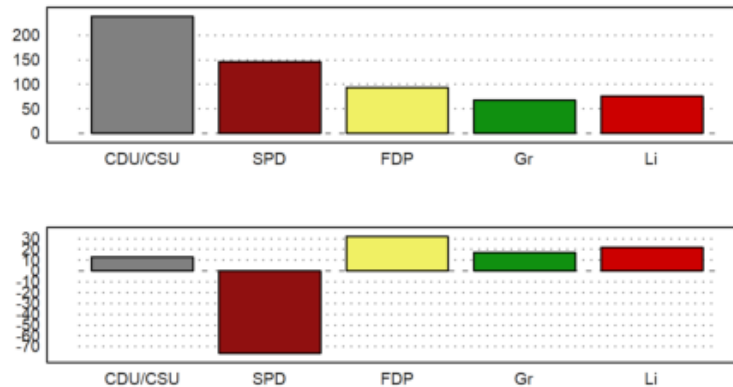


Definisi beberapa warna untuk setiap partai.

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

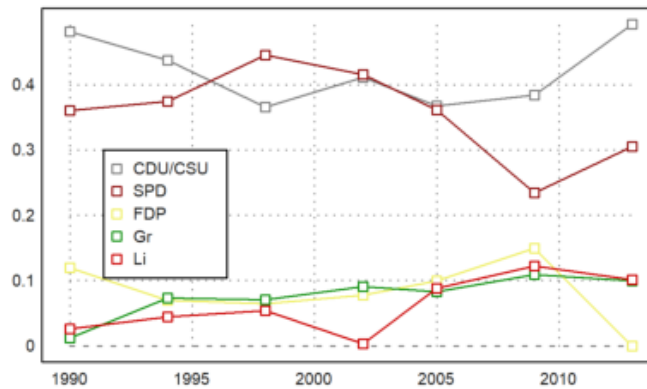
Sekarang kami dapat membuat plot hasil dari pemilihan 2009 dan mengubah menjadi satu plot menggunakan gambar. Kami dapat menambahkan sebuah vektor dari kolom ke setiap plot.

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



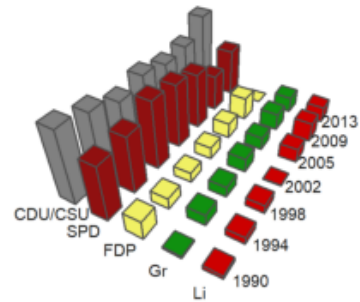
Plot data menggabungkan baris-baris dari data statistik dalam satu plot.

```
>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):
```



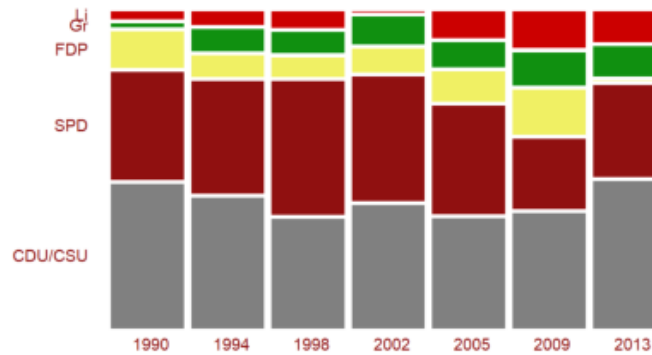
Sebuah plot 3D kolom menunjukkan baris baris dari data statistik dalam bentuk kolom. Kami menyediakan label-label untuk baris-baris dan kolom-kolom. Sudut adalah sudut pandangan.

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



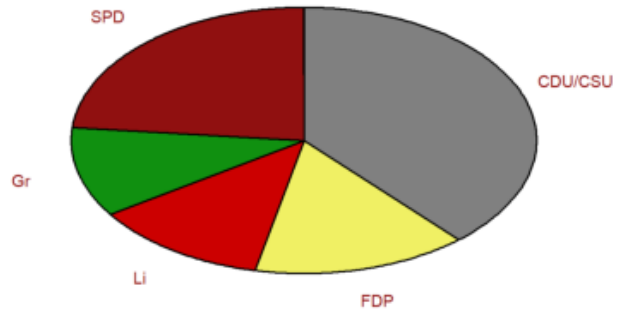
Representasi lainnya adalah plot mosaik. Catatan bahwa kolom-kolom dari plot mewakili kolom-kolom dari matrik ini. Karena panjang dari label CDU/CSU, kami mengambil jendela terkecil daripada biasanya.

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



Kami dapat juga melakukan sebuah chat pie. Karena hitam dan kuning membentuk sebuah koalisi, kami mengurutkan kembali elemen-elemen.

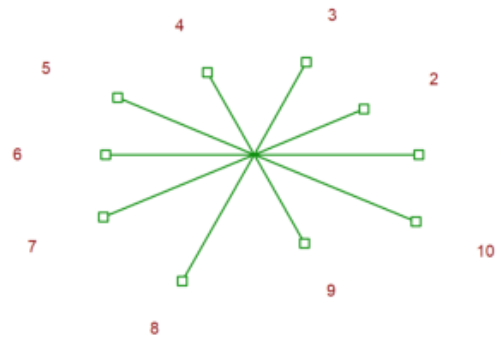
```
>i=[1,3,5,4,2]; piechart(BW[6,3:7][i],color=CP[i],lab=P[i]):
```



Ini merupakan jenis plot lainnya.

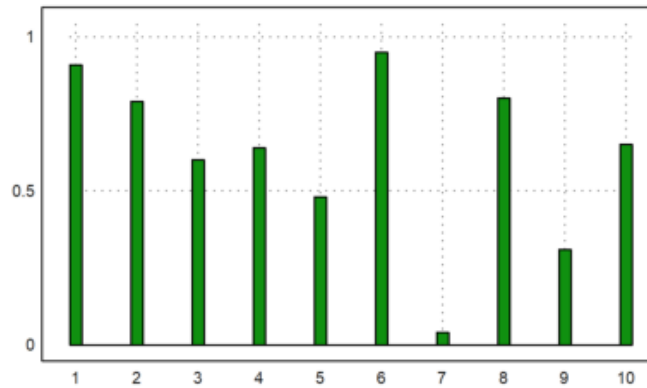
```
>starplot(normal(1,10)+4,lab=1:10,>rays):
```





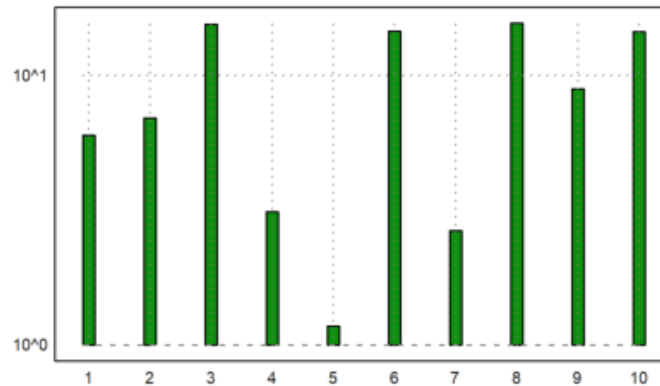
Beberapa plot dalam plot2d bagus dalam statistik. Ini merupakan plot implus dari data acak secara seragam berdistribusi dalam  $[0,1]$ .

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



Tetapi secara eksponensial data berdistribusi, kami membutuhkan sebuah plot logaritmik

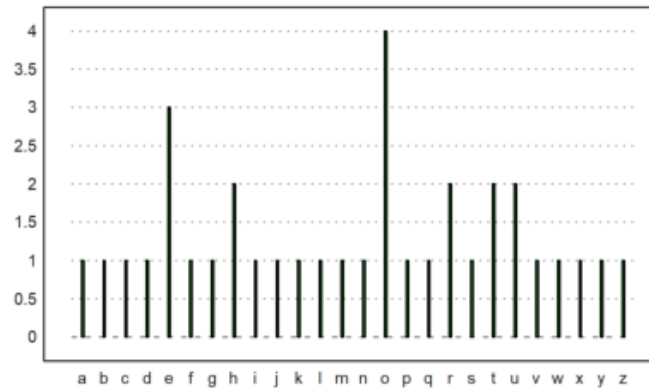
```
>logimpulseplot(1:10,-log(random(1,10))*10):
```



Fungsi `columnplot()` mudah untuk digunakan, karena ini hanya membutuhkan sebuah vektor dari nilai-nilai. Selebihnya, ini dapat diatur label-labelnya untuk sebarang yang kami inginkan, kami mendemonstrasikan ini setelah tutorial ini.

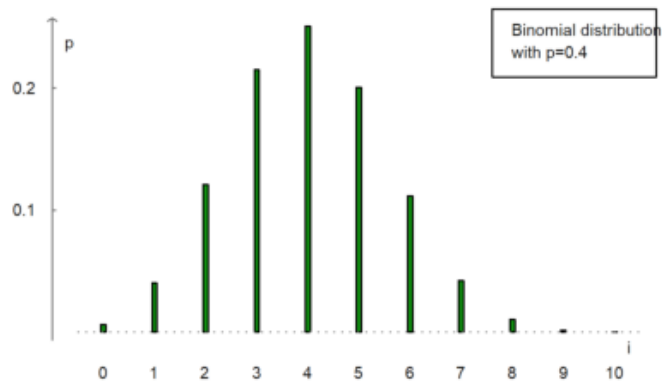
Ini merupakan aplikasi lainnya, dimana kami menghitung karakter dalam sebuah kalimat dan plot sebuah statistik.

```
>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):
```



Ini juga memungkinkan secara manual mengatur sumbu-sumbu.

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



Berikut merupakan sebuah cara untuk membuat plot frekuensi dari angka-angka dalam sebuah vektor. Kami membuat sebuah vektor dari bilangan bulat acak 1 ke 6.

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

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

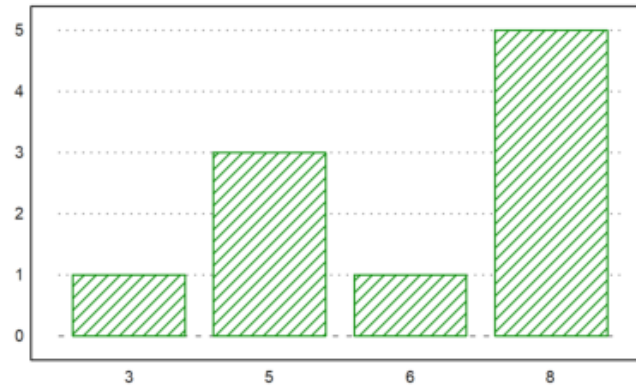
Lalu mengekstrasi angka-angka unique dalam v.

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

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

Dan plot frekuensi-frekuensi dalam sebuah plot kolom.

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



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

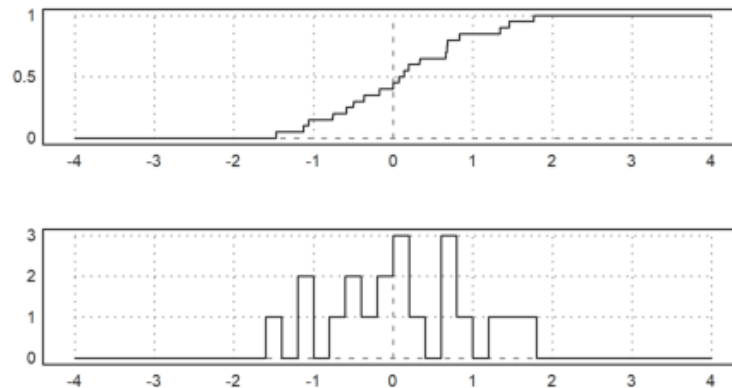
```
>x=normal(1,20);
```

Fungsi `empdist(x,vs)` membutuhkan sebuah nilai yang terurut dari nilai-nilai. Jadi kami harus mengurutkan `x` sebelum kami digunakan.

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

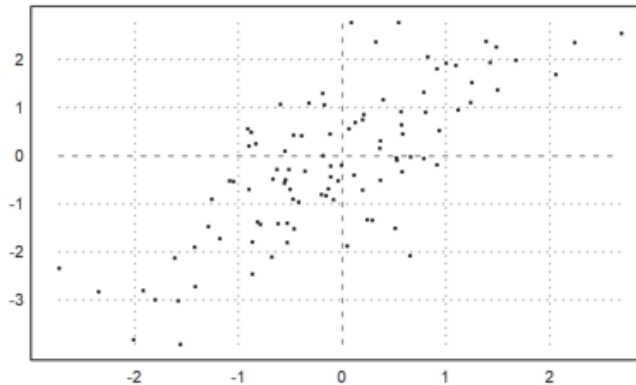
Lalu kami plot distribusi empirik dan beberapa kotak kepadatan kedalam satu plot. Daripada membuat plot batang untuk distribusi kami dengan sebuah sawtooth plot pada waktu ini.

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



Sebuah plot pencar mudah untuk dilakukan pada Euler dengan titik plot umum. Graph berikut menunjukkan bahwa  $X$  dan  $X+Y$  terlihat memiliki korelasi positif.

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

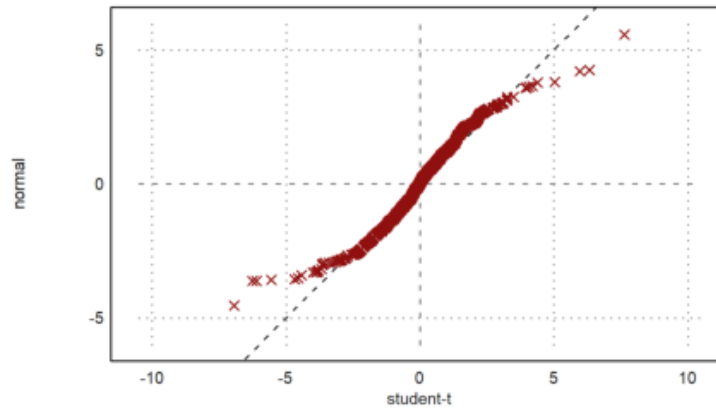


Terkadang, kami berharap untuk membandingkan dua contoh dari distribusi yang berbeda. Ini dapat dilakukan dengan sebuah quantile-quantile-plot.

Untuk menguji, 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):
```





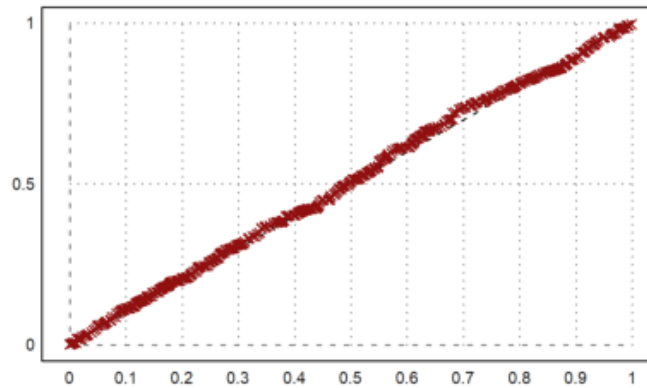
Dengan jelas plot menunjukkan bahwa nilai nilai normal terdistribusi cenderung lebih kecil pada ekstrim terakhirnya.

Jika kita memiliki dua distribusi dari ukuran yang berbeda, kami dapat memperluas yang satu lebih kecil atau menyusut yang terbesar. Fungsi berikut bagus dikeduanya. Ini mengambil nilai median dengan persentase diantara 0 dan 1.

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

Mari kita membandingkan dua distribusi yang seimbang.

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



## Regresi dan Korelasi

---

Regresi linear dapat dilakukan dengan fungsi `polyfit()` atau fungsi fit lainnya.

Untuk memulainya 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 tidak memiliki beban dan memiliki beban yang cocok. Pertama koefisien dari kecocokan linear.

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

```
[0.733333, 0.812121]
```

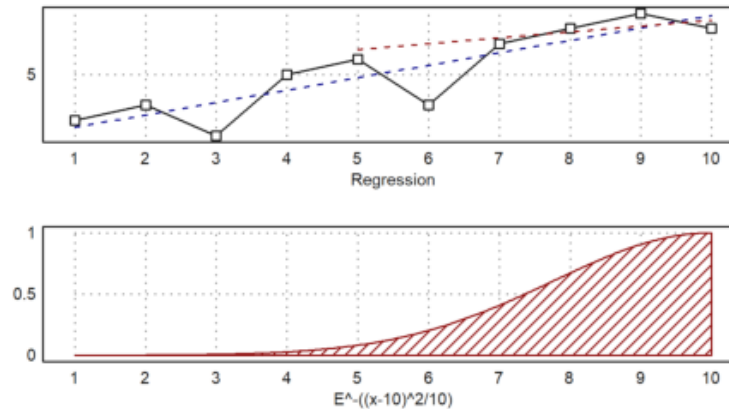
Sekarang koefisien dengan bobot yang menekankan nilai terakhir.

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

```
[4.71566, 0.38319]
```

Kita ambil semuanya ke dalam sebuah plot untuk titik titik dan garis regresi, dan bobot yang digunakan.

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



Untuk contoh lainnya kami membaca sebuah survey dari murid-murid, umur mereka dan umur orang tua mereka dan banyaknya saudara dari sebuah file.

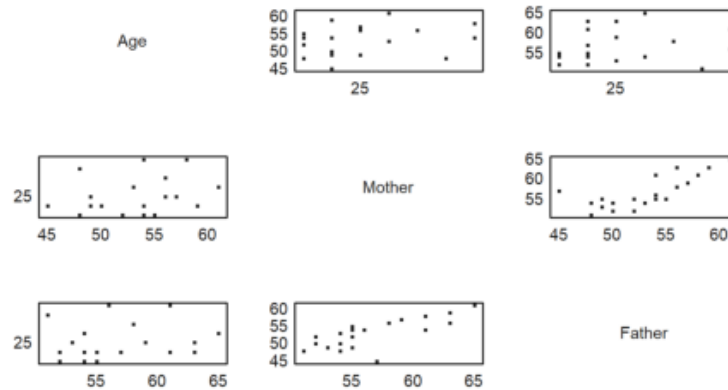
Tabel ini berisi "m" dan "f" di kolom kedua. Kami gunakan variabel tok2 untuk mengatur translasi yang pas daripada membiarkan readtable() mengumpulkan translasi.

```
>{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 umur bergantung dari yang lainnya? Kesan pertama datang dari plot pencar berpasangan.

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



Jelas bahwa umur dari ayah dan ibu tergantung pada setiapnya. Mari kita tentukan dan plot garis regresi.

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

```
[17.3789, 0.740964]
```

Ini secara jelas merupakan model yang salah. Garis regresi seharusnya  $=17+0.74t$ , dimana  $t$  adalah umur dari ibu dan  $s$  adalah umur dari ayah. Perbedaan umur mungkin tergantung sedikit dari umur, tetapi tidak banyak.

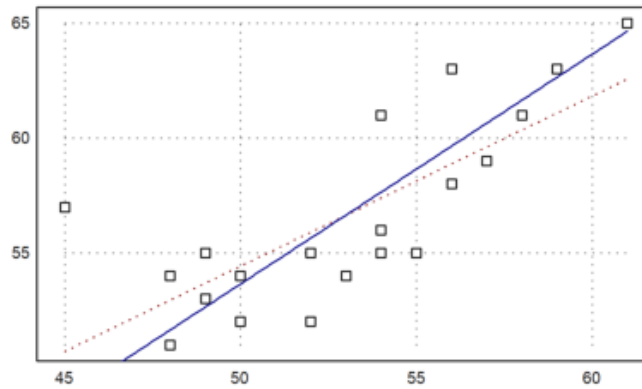
Selain itu, kami mengira sebuah fungsi seperti  $s=a+t$ . Lalu  $a$  adalah rata rata dari  $s-t$ . Ini merupakan usia rata-rata perbedaan antara ayah dan ibu.

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

3.65

Mari kita plot ini ke satu plot pencar.

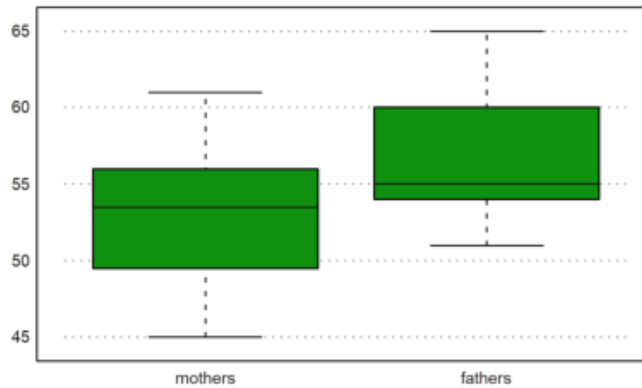
```
>plot2d(cs[1],cs[2],>points); ...  
>plot2d("evalpoly(x,ps)",color=red,style=".",>add); ...  
>plot2d("x+da",color=blue,>add):
```



Ini merupakan plot kotak dari dua umur. Ini hanya menunjukkan, bahwa umur berbeda.

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





Ini menarik bahwa perbedaan dalam nilai tengah tidak sebesar seperti perbedaan dalam nilai rata-rata.

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

1.5

Koefisien korelasi menyarankan korelasi positif.

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

0.7588307236

Korelasi dari peringkat merupakan sebuah ukuran untuk urutan yang sama dalam vektor keduanya. Ini juga sedikit positif.

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

```
0.758925292358
```

## Membuat fungsi-fungsi baru

---

Tentu saja, bahasa EMT dapat digunakan untuk membuat program fungsi baru. Sebagai contoh kami mendefinisikan fungsi skewness.

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

dimana  $m$  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 kamu lihat, kami dengan mudah menggunakan bahasa matriks untuk mendapatkan sebuah implementasi yang sangat singkat dan efisien. Mari kita coba fungsi ini.

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

```
-0.198710316203
```

Ini merupakan fungsi lainnya, sebut saja koefisien skewness Pearson.

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

-0.0801873249135

## Simulasi Monte Carlo

---

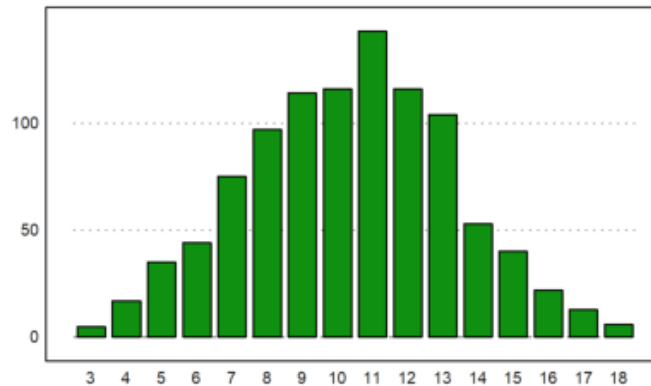
Euler dapat digunakan untuk mensimulasikan kejadian acak. Kami telah melihatnya contoh sederhana di atas. Ini merupakan yang lainnya, yang mana mensimulasikan 1000 kali 3 dadu dilempar, dan menanyakan distribusi dari jumlahnya.

```
>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]
```

Kami dapat plot ini sekarang.

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



Untuk menentukan distribusi yang diinginkan tidak mudah. Kami menggunakan sebuah rekursi lanjut untuk ini.

Berdasarkan perhitungan fungsi jumlah cara angka k dapat merepresentasikan sebagai jumlahan dari angka n dalam rentang 1 ke m. ini bekerja secara rekursif dalam 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 merupakan hasil dari lemparan tiga dadu.

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

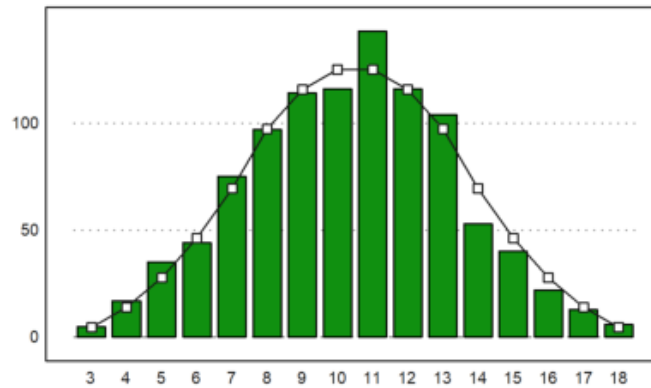
```
[1, 5, 15, 35, 70, 121, 185, 255, 320, 365, 381, 365, 320,  
255, 185, 121, 70, 35, 15, 5, 1]
```

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

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

Kami tambahkan nilai harapan ke plot.

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



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

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

0.316227766017

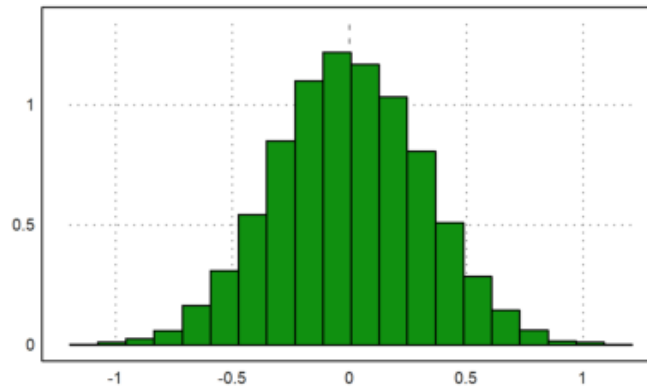


Mari kita periksa ini dengan sebuah simulasi. Kami menghasilkan 10000 kali 10 vektor acak.

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

0.319493614817

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



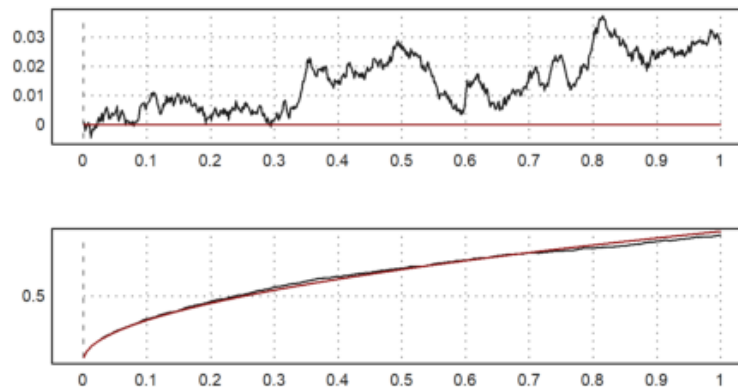
Nilai median dari 10 0-1-normal distribusi angka acak memiliki deviasi yang lebih besar.

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

0.374460271535

Karena kami dapat dengan mudah membuat langkah acak, kami dapat mensimulasikan proses Wiener. Kami mengambil 1000 langkah dari 1000 proses. Kami lalu plot standar deviasi dan rata-rata dari langkah ke-n dari proses bersama ini dengan nilai harapan dalam merah.

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



## Pengujian

---

Pengujian merupakan alat yang penting dalam statistika. Dalam Euler, banyak uji yang diimplementasikan. Semua uji ini mengembalikan error yang dapat kita terima apabila menolak hipotesis nol.

Sebagai contoh, kami menguji pelemparan dadu untuk distribusi seragam. Pada lemparan 600, kami mendapatkan nilai berikut, yang mana dapat kami masukkan kedalam uji chi-square.

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

```
0.498830517952
```

Uji chi-square juga memiliki sebuah mode, yang mana menggunakan simulasi Monte Carlo untuk menguji statistik. Hasilnya seharusnya hampir sama. Parameter >p menunjukkan vektor-y sebagai vektor peluang.

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

```
0.526
```

Error ini terlalu besar. Jadi kami dapat menolak distribusi seragam. Ini tidak menunjukkan bahwa dadu kami tidak adil. Tapi kami tidak dapat menolak hipotesis kami.

Selanjutnya kami membuat 1000 dadu dilemparkan menggunakan pembuat angka acak dan menggunakan uji yang sama.

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

0.528028118442

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

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

0.0218365848476

ttest() fungsi membutuhkan nilai rata-rata, deviasi, banyaknya data, dan nilai rata-rata untuk mengujinya.

Sekarang mari kita periksa perhitungan untuk nilai rata-rata yang sama. Kami menolak hipotesis bahwa mereka memiliki rata-rata yang sama, jika hasilnya adalah  $< 0.05$ .

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

0.38722000942

Jika kita menambahkan sebuah bias ke distribusi satu, kami mendapatkan lebih banyak penolakan. Mengulang simulasi ini beberapa kali untuk melihat pengaruhnya.

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

```
5.60009101758e-07
```

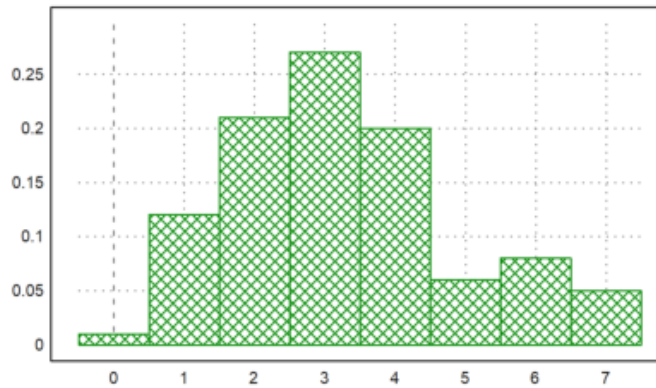
Contoh selanjutnya, kami membuat 20 dadu acak dilempar 100 kali dan menghitung satunya. Terdapat satu  $20/6=3.3$  dalam rata-rata.

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

```
3.28
```

Kami sekarang membandingkan angka satu dengan distribusi binomial. Pertama kami plot distribusi angka satu.

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



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

Lalu kami hitung nilai harapan.

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

Kami mengumpulkan beberapa angka untuk mendapatkan kategori, yang mana 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 kami merupakan sebuah distribusi binomial, jika hasilnya adalah  $< 0.05$ .

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

0.53921579764

Contoh berikut memiliki hasil dari dua grup dari orang-orang (laki-laki dan perempuan, 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 berharap menguji secara bebas dari suara untuk jenis kelamin. Tabel uji  $\chi^2$  melakukan ini. Hasilnya terlalu besar untuk menolak saling bebas. Jadi kita tidak dapat mengatakan, jika suara tergantung pada jenis kelamin untuk data ini.

```
>tabletest(A)
```

0.990701632326

Tabel nilai harapan berikut, jika kita berasumsi 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

Kami dapat menghitung koefisien kontigensi yang dikoreksi. Karena ini sangat dekat dengan 0, kami menyimpulkan bahwa pemungutan suara tidak tergantung pada jenis kelamin.

```
>contingency(A)
```

```
0.0427225484717
```



## Beberapa Pengujian Lainnya

---

Selanjutnya kami menggunakan analisis variansi (uji-F) untuk menguji tiga sample dari data distribusi secara normal untuk nilai rata-rata yang sama. Metode ini dinamakan ANOVA (variansi analisis). Dalam Euluer, 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 dari nilai rata-rata yang sama. Kami melakukan ini dengan peluang error dari 1.3%.

Ini juga pengujian median, yang mana menolak data sampel dengan perbedaan rata-rata distribusi pengujian median dari sampel gabungan.

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

0.0241724220052

Uji lainnya dalam kesamaan adalah uji rank. Ini lebih tajam daripada uji median.

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

0.00199969612469

Contoh berikut, distribusi keduanya memiliki nilai rata-rata yang sama.

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

0.129608141484

Mari kita coba untuk mensimulasikan perlakuan a dan b diaplikasikan ke orang-orang yang berbeda.

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

Uji signum memutuskan, jika a lebih baik daripada b.

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

0.0546875

Ini memiliki banyak error. Kami tidak bisa menolak bahwa a sebaik b.

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

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

0.0296680599405

Mari kita coba uji dua tes lagi menggunakan deret yang dibuat.

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

0.0068706451766

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

0.275145971064

## Angka-Angka Acak

---

Uji berikut untuk pembuat angka acak. Euler menggunakan sebuah pengacak angka yang bagus, jadi kita tidak mengharapkan masalah lainnya.

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

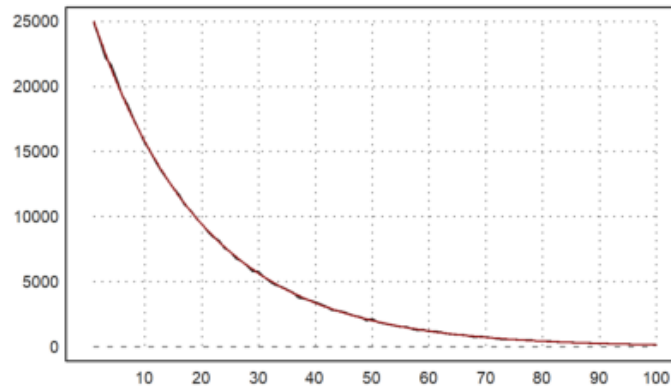
```
>n:=10000000; r:=random(1,n);
```

Sekarang kita hitung jarak antara dua angka kurang dari 0.05.

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

Terakhir, kami plot angka beberapa kali, setiap jarak yang terjadi dan bandingkan dengan nilai harapan.

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



Hapus data.

```
>remvalue n;
```

## Pengantar untuk pengguna dalam Projek R

---

Secara jelas, EMT tidak bersaing dengan R sebagai package statistik. Namun, ini banyak prosedur statistik dan fungsi yang tersedia dalam EMT juga. Jadi EMT mungkin memenuhi kebutuhan dasar. Kemudian, EMT datang dengan numerical package dan sistem komputer aljabar.

Notebook ini untuk kamu jika umum dengan R, tetapi ingin tahu perbedaan sintaks dari EMT dan R. Kami mencoba untuk memberikan gambaran jelas dan kurang jelas yang kamu ketahui.

Selebihnya, kami lihat pada cara untuk bertukar data pada kedua sistem.

Catatan ini dalam proses pengerjaan.

**Sintaks Dasar**

---

Hal pertama yang dipelajari dalam R adalah membuat sebuah vektor. Dalam EMT, perbedaan utamanya pada operator : yang dapat mengambil jarak langkah. Lebih dari itu memiliki daya ikat 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. Ini memungkinkan untuk membuat vektor menjadi benda-benda yang bergabung. Contoh berikut, seperti lainnya, dari "Introduction to R" datang dari proyek R. Jika kamu membaca PDF ini, kamu akan menemukan bahwa I mengikuti path 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 colon dengan ukuran langkah dari EMT menggantikan fungsi seq() dalam R. Kami dapat menulis fungsi ini dalam EMT.

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

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

Fungsi rep() dari R tidak menunjukkan dalam EMT. Untuk masukan vektor, ini dapat ditulis seperti berikut.

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

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

Catatan bahwa "=" atau ":=" digunakan untuk penugasan. Operator "->" digunakan untuk ukuran dalam EMT.

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

```
77.6713990297 miles
```

Operator "<-" untuk penugasan adalah cara yang menyesatkan, dan ide buruk dari R. Berikut akan dibandingkan a dan -4 dalam EMT.

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

```
0
```

Dalam R, "a < -4 < 3" bekerja, tetapi "a < -4 < -3" tidak. Saya memiliki ambiguitas yang sama dalam EMT juga, tetapi mencoba untuk menghapuskan mereka sedikit demi sedikit.

EMT dan R memiliki vektor dari tipe boolean. Tetapi dalam EMT, angka 0 dan 1 digunakan untuk mewakili salah dan benar. Dalam R, nilai true dan false dapat demikian digunakan dalam aritmetik biasa seperti dalam EMT.

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

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

EMT mengeluarkan error atau menghasilkan NAN tergantung pada "error".

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

```
NAN  
1
```

String merupakan hal yang sama dalam R dan EMT. Keduanya memiliki locale sekarang, bukan dalam Unicode.

Dalam R terdapat packages dalam Unicode. Dalam EMT, sebuah string dapat berupa string Unicode. Sebuah string unicode dapat diubah kedalam encode local dan sebaliknya. Selanjutnya, u"...” dapat berisi entitas HTML.

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

© René Grothmann

Berikut mungkin atau mungkin tidak menampilkan secara benar dalam sitem kamu sebagai A dengan dot dan dash diatas. Tergantung pada font yang digunakan.

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

Gabungan string dapat dilakukan dengan "+" atau "|". Ini berisi angka, yang mana akan mencetak dalam format sekarang.

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

```
pi = 3.14159265359
```

## Pengindeks-an

---

Hampir setiap kali, ini bekerja pada R.

Tetapi EMT mengintrepetasikan indeks negatif dari belakang vektor sementara R mengintrepetasikan `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 dari R dapat di dapatkan dalam EMT dengan `drop()`.

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

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

Vektor logika tidak jauh berbeda seperti indeks dalam EMT, dalam beda ke R. Kamu butuh untuk mengekstrak elemen bukan nol pertama dalam EMT.

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

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

Seperti dalam R, indeks vektor dapat berisi repetisi.

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

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

Tetapi nama untuk indeks tidak mungkin dalam EMT. Untuk sebuah paket statistika, ini mungkin dibutuhkan untuk memudahkan akses elemen dari vektor.

Untuk meniru perilaku ini, kami dapat mendefinisikan sebuah fungsi seperti berikut.

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

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

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

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

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

[10.4, 3.1]

## Tipe Data

---

EMT memiliki tipe data tetap dari pada R. Secara jelas, dalam R memiliki vektor bertumbuh. KAmu dapat mengatur sebuah vektor numerik kosong `v` dan menugaskan sebuah nilai ke elemen `v[17]`. Ini tidak mungkin dalam EMT.

Berikut sedikit tidak efisien.

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

EMT sekarang akan membuat sebuah vektor dengan `v` dan `i` diatambahkan pada stek dan menyalin vektor itu kembali ke variabel global `v`.

Lebih efisien pra-definisi vektor.

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

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

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

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

Pengubahan ke string memungkinkan untuk tipe data dasar saja. Format sekarang digunakan untuk penggabungan string sederhana. Tetapi terdapat fungsi seperti `print()` atau `frac()`.

Untuk vektor, kamu dapat dengan mudah menulis fungsi kamu 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 sebuah fungsi `convertmxm()`, yang mana juga dapat digunakan mengubah sebuah vektor untuk keluaran.

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

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



Untuk Latex perintah tex dapat digunakan untuk mendapatkan perintah Latex.

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

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

## Faktor Faktor dan Tabel Tabel

---

Dalam pengantar R terdapat sebuah contoh bernama faktor-faktor.

Berikut merupakan sebuah daftar dari wilayah 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 berkorespondensi dalam 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 dalam wilayah. Menjadi sebuah program statistik, R memiliki `factor` dan `tapply()` untuk ini.

EMT dapat membuat ini dengan menemukan indeks dari wilayah dalam daftar unik dari 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 titik ini, kami dapat menulis fungsi putaran untuk melakukan hal ini pada hanya faktor.

Atau kami dapat menyerupai fungsi `tapply()` dalam cara yang sama.

```
>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 ini menghitung wilayah unik untuk setiap `i`, tetapi ini bekerja.

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

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

Catatan bahwa ini bekerja untuk setiap vektor dalam wilayah.

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

```
[44.5, 57.3333333333]
```

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

Jadi kita dapat mencetak rata-rata pendapatan negara bagian dalam wilayah sebuah cara yang mudah.

```
>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

Sekarang dapat juga mencoba untuk meniru cara kerja dari R secara utuh.

Faktor seharusnya dengan jelas menyimpan dalam sebuah collection dengan tipe dan kategori (negara bagian dan wilayah dalam contoh kita). Untuk EMT, kami menambahkan indeks pra-perhitungan.

```
>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 collection 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 kami dapat meniru `tapply()` dalam cara berikut. Ini akan mengembalikan sebuah tabel sebagai sebuah collection dari data tabel dan headings 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

```

Kita tidak menambahkan banyak pemeriksaan tipe disini. Hanya pendekatan pencegahan kategori (faktor-faktor) tanpa data. Tetapi yang seharusnya diperiksa untuk panjang yang benar dari t dan untuk kebenaran dari tf collection.

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

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

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

EMT hanya memiliki dua dimensi untuk array. Tipe data ini disebut matriks. Ini akan lebih mudah untuk menulis fungsi untuk dimensi yang lebih tinggi atau sebuah library C untuk ini, namun

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

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

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

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

Ekstrasi baris dan kolom, atau sub-matriks, seperti dalam R.

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

2	3
7	8
12	13
17	18

Namun, dalam R ini memungkinkan untuk mengatur sebuah baris dari indeks spesifik dari vektor ke sebuah nilai. Kemungkinan sama dalam EMT hanya dengan sebuah 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 dilalui dalam EMT. Jika kamu tidak ingin mengubah matriks asli M, kamu harus menyalinnya dalam fungsi.

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

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



Perkalian luar dalam EMT hanya dapat dilakukan antara vektor-vektor. Ini secara otomatis karena bahasa matriks. Satu vektor butuh menjadi sebuah kolom vektor dan yang lainnya sebuah vektor baris.

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

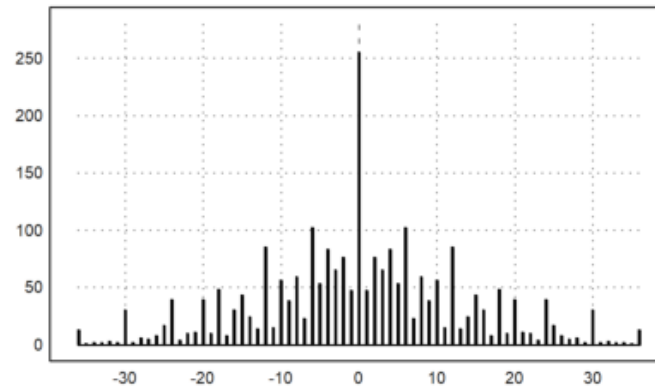
1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25

Dalam PDF pengantar untuk R disini terdapat sebuah contoh, yang mana menghitung distribusi dari ab-cd untuk a,b,c,d dipilih dari 0 ke n secara acak. Solusi dalam R adalah bentuk sebuah matriks 4-dimensi dan menjalankan table() setelah itu.

Tentu saja, ini dapat tercapai dengan sebuah putaran. Tetapi loop tidak efektif dalam EMT atau R. Dalam EMT, kami dapat menuliskan loop dalam C dan akan menjadi solusi tercepat.

Tetapi kami ingin menyerupai perilaku dari R. Untuk ini, kami butuh untuk meratakan perkalian ab dan membuat sebuah matriks dari 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"):
```



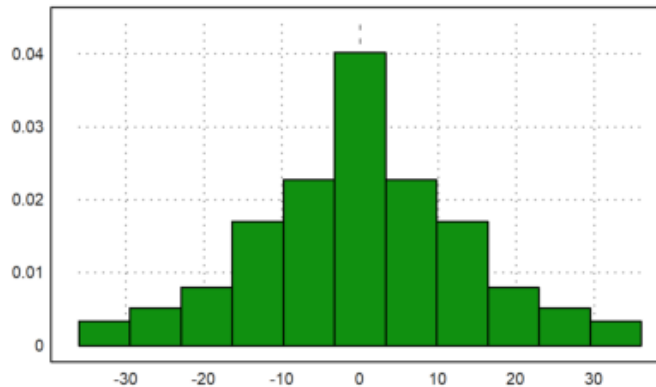
Disamping perkalian eksak, EMT dapat menghitung frekuensi dalam vektor-vektor.

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

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

Cara termudah untuk plot ini seperti sebuah distribusi berikut.

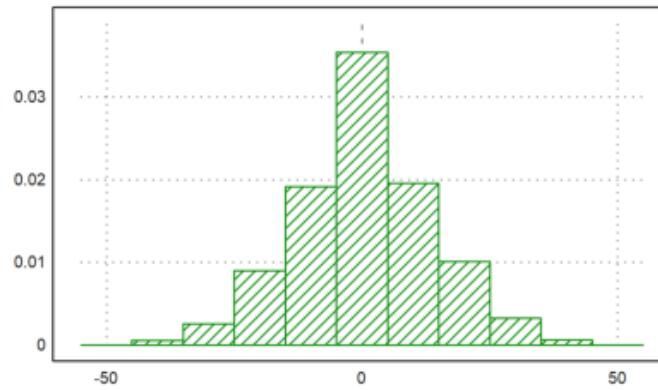
```
>plot2d(q,distribution=11):
```



Tetapi ini juga dapat untuk pra-komputasi menghitung dalam interval yang dipilih sebelumnya. Tentu saja, berikut menggunakan `getfrequencies()` secara internal.

Karena fungsi `histo()` mengembalikan frekuensi, kami butuh untuk men-skakan ini jadi bahwa integral dibawah grafik batang adalah 1.

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



EMT memiliki dua macam list. Satu dari list global yang mana mutable dan lainnya adalah sebuah tipe list yang mana immutable. Kami tidak peduli tentang list global disini.

Tipe list immutable dinamakan sebuah collection dalam EMT. Ini berperilaku seperti struktur dalam C, tetapi elemen-elemenya hanya diberi angka dan tidak dinamai.

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

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

Sekarang elemen-elemen tidak memiliki nama-nama, melalui nama dapat diatur untuk tujuan khusus. Mereka dapat diakses dengan angka.

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

```
1992
```

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

---

Kamu akan sering ingin untuk import sebuah matriks data dari sumber lainnya ke EMT. Tutorial ini memberitahu kamu tentang banyak cara untuk meraihnya. Fungsi sederhana adalah `writematrix()` dan `readmatrix()`.

Mari kita demonstrasikan bagaimana membaca dan menulis sebuah vektor ke file yang sebenarnya.

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

```
0.49815
```

```
0.28037
```

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

Karena pengantar ini banyak seperti dalam direktori, dimana pengguna tidak memiliki akses, kami menulis data ke direktori pengguna home. Untuk notebook sendiri, ini tidak dibutuhkan, karena file data akan ditulis kedalam direktori yang sama.

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

Sekarang kami menulis vektor kolom a' kedalam file. Ini menghasilkan angka satu dalam setiap bari dari file.

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

Untuk membaca data, kami gunakan readmatrix().

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

Dan menghapus file.

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

```
0.49815  
0.28037
```

Fungsi writematrix() atau writetable() dapat diatur untuk bahasa lain.

Sebagai contoh, jika kamu menggunakan sistem Indonesia (titik desimal dengan koma), Excel kamu harusnya memiliki nilai dengan desimal koma dipisahkan dengan titik koma dalam sebuah csv (secara bawaan koma memisahkan nilai) file. File berikut "test.csv" harusnya tampak dalam folder kamu sekarang.

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

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

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

Terkadang kami memiliki strings dengan token seperti berikut.

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

Untuk membuat token ini, kami mendefinisikan sebuah vektor dari token-token.

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

```
f  
m
```

Lalu kami dapat menghitung angka dari banyaknya token yang muncul dalam string, dan mengambil hasilnya kedalam sebuah tabel.

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



Tulis tabel dengan header token.

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

	f	m
1	6	7
2	5	2

Untuk statistik, EMT dapat membaca dan menulis tabel.

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

File terlihat seperti ini.

```
>printfile(file)
```

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

fungsi `readtable()` dalam bentuk yang sederhana dapat membaca ini dan mengembalikan sebuah collection dari nilai nilai dan garis heading.

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

Collection ini dapat mencetak dengan `writetable()` untuk notebook atau ke dalam sebuah file.

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

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

Nilai-nilai matriks adalah elemen pertama dari L. Catatan bahwa `mean()` dalam EMT menghitung nilai mean dari baris-baris dari sebuah matriks.

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

```
0.40472  
0.37102  
0.75547
```

## File-File CSV

---

Pertama, mari kita tulis sebuah matriks kedalam sebuah file. Untuk keluaran, kami menghasilkan sebuah file dalam direktori sekarang yang dikerjakan.

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

Here is the content of this file.

```
>printfile(file)
```

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

CSV ini dapat dibuka dalam sistem English kedalam Excel dengan sebuah klik double. Jika kamu mendapatkan sebuah file dalam sistem German, perlu untuk mengimpor data ke Excel untuk berhati-hati dalam titik desimal.

Tetapi titik desimal adalah format bawaan untuk EMT juga. Kamu dapat membaca matriks dari sebuah file dengan `readmatrix()`.

```
>readmatrix(file)
```

0.82212	0.82153	0.77712
0.84829	0.32378	0.65014
0.14823	0.32975	0.62619

Ini memungkinkan untuk menulis beberapa matriks kedalam satu file. Perintah `open()` dapat membukan file untuk menulisnya dengan parameter "w". Secara bawaan adalah "r" untuk membaca.

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

Matriks-matriks dipisahkan dengan baris kosong. Untuk membaca matriks-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

Dalam Excel atau spreadsheet yang sama, kamu dapat mengexport sebuah matriks sebagai CSV (comma separated values). Dalam Excel 2007, gunakan "save as" dan "other formats", lalu pilih "CSV". Pastikan, tabel sekarang hanya memiliki data yang ingin kamu ekspor.

Ini sebagai contohnya.

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

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

Seperti yang dapat kamu lihat, sister German saya menggunakan titik koma sebagai pemisah dan sebuah koma desimal. Kamu dapat merubah ini dalam pengaturan sistem atau dalam Excel, tetapi ini tidak dibutuhkan untuk membaca matriks ke dalam EMT.

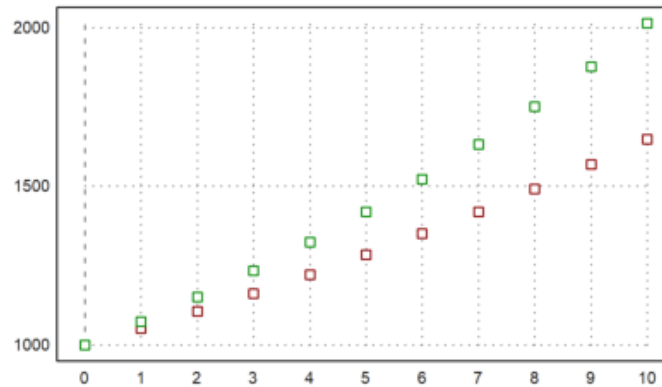
Cara termudah untuk membaca ini kedalam Euler adalah `readmatrix()`. Semua koma digantikan dengan titik dengan parameter `>comma`. Untuk CSV English, secara sederhana buang parameter ini.

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

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

Mari kita buat plot ini.

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



Banyak cara yang lebih mendasar untuk membaca data dari sebuah file. Kamu dapat membuka file dan membaca angka per baris. Fungsi `getvectorline()` akan membaca angka-angka dari sebuah baris dari data. Secara bawaan, ini mempunyai harapan sebuah titik desimal. Tetapi ini dapat juga menggunakan sebuah koma desimal, jika kamu memanggil `setdecimaldot(",")` sebelum kamu menggunakan fungsi ini.

Fungsi berikut merupakan sebuah contoh untuk ini. Ini akan berhenti pada akhir file atau baris kosong.

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

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

```
>myload(file)
```

0.82212	0.82153	0.77712
0.84829	0.32378	0.65014
0.14823	0.32975	0.62619

Ini juga akan memungkinkan membaca semua angka dalam file tersebut dengan `getvector()`.

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

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

Lalu ini sangat mudah untuk menyimpan sebuah vektor dari nilai-nilai, satu nilai setiap baris dan membaca kembali vektor ini.

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

```
0.50303
```

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

```
0.50303
```



## Menggunakan Tabel Tabel

---

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

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

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

Ini dapat diimpor ke Excel.

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

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

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

## Menganalisis Sebuah Garis

---

Kamu bahkan dapat mengevaluasi setiap baris dengan tangan. Andaikan, kami memiliki baris dengan format berikut.

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

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

Pertama kami dapat membuat token untuk baris.

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

```
2020-11-03
```

```
Tue
```

```
1'114.05
```

Lalu kami dapat mengevaluasi setiap elemen dari baris menggunakan evaluasi yang dibutuhkan.

```
>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, ini memungkinkan untuk mengekstrak hampir sebarang informasi dari sebuah baris data.

Asumsikan kita memiliki baris berikut dalam dokumen HTML.

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

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

Untuk mengekstrak ini, kami menggunakan regular expression, yang mana mencari

- sebuah braket tertutup > ,
- sebarang string yang tidak memiliki bracket dengan sebuah sub-cocok

”(...)”,

- sebuah pembuka dan penutup bracket menggunakan solusi terpendek,
- kembali sebarang string yang tidak memiliki bracket,
- dan sebuah bracket terbuka <.

Regular expression merupakan hal yang sulit untuk dipelajari tetapi sangat kuat.

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

Hasil merupakan posisi yang cocok, string yang cocok dan sebuah vektor string untuk sub-cocok.

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

1145.5

5.6

Ini merupakan sebuah fungsi, yang mana membaca bagian bagian numerik diantara <td> dan </td>.

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

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

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

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

## Membaca dari Web

---

Sebuah web site atau sebuah file dengan sebuah URL dapat dibuka dalam EMT dan dapat dibaca baris per baris.

Dalam contoh, kami membaca versi sekarang dari situs EMT. Kami menggunakan regular expression untuk mengamati "Version ..." dalam sebuah heading.

```
>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

## Variabel Input dan Output

---

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

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

```
    mypi = 3.141592653589793;
```

Untuk sebuah pengujian, kami membuat sebuah file Euler dalam direktori bekerja dari EMT.

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

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

Kami sekarang dapat memuat file. Ini dapat didefinisikan matriks M.

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

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

Omong-omong, jika `writevar()` digunakan dalam sebuah variabel, ini akan mencetak definisi variabel dengan nama dari variabel ini.

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

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



Kami dapat juga membukan file baru atau menambahkan file yang telah ada. Dalam contoh kami menambahkan file yang dibuat sebelumnya.

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

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

Untuk menghapus sebarang file gunakan `fileremove()`.

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

vektor baris dalam sebuah file tidak butuh koma, jika setiap angka dalam baris baru. Mari kita buat sedemikian hingga sebuah file, menulis setiap baris satu per satu dengan `writeln()`.

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

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

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

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

## Latihan

---

### Latihan 1

---

diberikan lemparan dadu acak sebanyak 1200 kali, tentukan nilai chi-kuadratnya apabila nilai harapan dadunya memiliki distribusi seragam

```
>throws=1200; sides=6;  
>randomDice = intrandom(throws, sides);  
>freqDice = multofsorted(sort(randomDice), 1:sides);  
>expectDice = dup(throws/sides, sides)';  
>chitest(freqDice, expectDice)
```

0.91075

Karena error memiliki nilai yang besar (lebih dari 0.05) maka dapat disimpulkan bahwa kita tidak bisa menolak distribusi seragam. Ini belum membuktikan bahwa dadu kita adil tetapi kita tidak bisa menolak hipotesis kita

### Latihan 2:

---

Diberikan hasil dari lemparan koin sebagai berikut, dengan G adalah gambar dan A adalah angka

G, A, A, G, A, A, A, G, A, A

Apakah koin tersebut adil?

```
>coinRes = "G, A, A, G, A, A, A, G, A, A";
```

buat fungsi untuk memisahkan setiap token

```
>function replaceTokenToArr(strToken, separateToken, arrToken, arrSubsToken)...
```

```
tokens = strtokens(strToken, separateToken);
transformedArr = [];
for i = 1 to length(tokens) step 1
    token = tokens[i];
    indexToken = indexof(arrToken, token);
    subs = arrSubsToken[indexToken];
    transformedArr = [transformedArr, subs];
end;
return transformedArr;
endfunction
```

```
>coinDis = replaceTokenToArr(coinRes, ", ", ["G", "A"], [1, 2])
```

```
[1, 2, 2, 1, 2, 2, 2, 1, 2, 2]
```

Tambahkan fungsi untuk menghitung setiap frekuensi

```
>function getFrequencies(arr)...
```

```
    uniqueArr = unique(arr);  
    sortArr = sort(arr);  
    firstIndexes = [];  
    freqs = [];  
    for i = 1 to length(uniqueArr) step i  
        currUnique = uniqueArr[i];  
        currFirstIndex = indexof(sortArr, currUnique) - 1;  
        firstIndexes = [firstIndexes, currFirstIndex];  
    end;  
    firstIndexes = [firstIndexes, length(arr)];  
    for i = 1 to (length(firstIndexes) - 1) step i  
        currIndex = firstIndexes[i];  
        nextIndex = firstIndexes[i + 1];  
        currFreq = nextIndex - currIndex;  
        freqs = [freqs, currFreq];  
    end;  
    return freqs;  
endfunction
```

```
>coinFreq = getFrequencies(coinDis)
```

[3, 7]

Didapatkan frekuensi untuk A sebanyak 3 dan G sebanyak 7

```
>coinExpected = dup(sum(coinFreq)/length(coinFreq), length(coinFreq))'
```

```
[5, 5]
```

```
>chitest(coinFreq, coinExpected)
```

```
0.2059
```

Karena nilai error dari chi-kuadratnya besar (lebih dari 0.05), maka kita dapat menolak hipotesis bahwa koin tersebut adil (setimbang).

### Latihan 3

---

Diberikan tabel yang menyatakan hubungan kondisi keluarga dengan perilaku deliquensi anak pada sebuah SD

Perilaku Deliquensi | Nakal | Nurut

Harmonis | 25 | 85

Broken | 75 | 15

Apakah terdapat hubungan antara kondisi keluarga dengan perilaku deliquensi anak?

```
>delinquencyChilds = [25, 85; 75, 15];  
>writetable(delinquencyChilds, wc=10, labr=["Harmonis", "Broken"], labc=["Nakal", "Nurut"])
```

	Nakal	Nurut
Harmonis	25	85
Broken	75	15

Periksa terlebih dahulu dengan uji tabel chi-kuadrat

```
>tabletest(delinquencyChilds)
```

0

Dimana hasilnya kecil, kita tidak bisa menentukan apakah menolak saling asing antara data yang diberikan.

Lalu, dapat dibuat tabel harapan dari matriks tersebut, untuk melihat harapan yang ada dalam setiap barisnya.

```
>writetable(expectedtable(delinquencyChilds), wc=10, dc=1, labr=["Harmonis", "Broken"], labc=["Nakal", "Nurut"])
```

	Nakal	Nurut
Harmonis	55	55
Broken	45	45

Selanjutnya, menghitung nilai koefisien kontingensi dari tabel tersebut

```
>contingency(delinquencyChilds)
```

0.7303

## Latihan 4

---

Penelitian dilakukan untuk mengetahui partisipasi warga dalam suatu kepala desa yang dilihat dari jenis kelamin. Hasil penelitian tersebut diperoleh data sebagai berikut

Jenis Kelamin | Ikut | Tidak

Pria | 10 | 5

Wanita | 9 | 6

Apakah ada pengaruh jenis kelamin terhadap keikutsertaan dalam pemilihan kepala desa?



```
>absenceGenders = [10,5;9,6]
```

10	5
9	6

Buat tabel agar mudah melihatnya

```
>writetable(absenceGenders, wc = 10, labr=["Pria", "Wanita"], labc = ["Ikut", "Tidak"])
```

	Ikut	Tidak
Pria	10	5
Wanita	9	6

Memeriksa tabel untuk uji chi-kuadratnya

```
>tabletest(absenceGenders)
```

0.70479

Karena nilai terlalu besar, maka belum bisa menyimpulkan apakah terdapat hubungan antara keikutsertaan dengan gender

Lalu, dibuat tabel nilai harapan untuk melihat nilai harapan pada tabel tersebut

```
>writetable(expectedtable(absenceGenders), dc=1, wc=10, labr=["Pria", "Wanita"], labc=["Ikut", "Tidak Ikut"])
```

	Ikut	Tidak Ikut
Pria	9.5	5.5
Wanita	9.5	5.5

Terakhir, adalah menghitung nilai koefisien kontingensi

```
>contingency(absenceGenders)
```

```
0.09759
```

Karena nilai koefisien kontingensi mendekati 0, maka dapat disimpulkan bahwa hubungan antara keikutsertaan dengan jenis kelamin tidak ada (sangat lemah). Terlihat bahwa koefisien korelasi kontingensinya mendekati 1, artinya bahwa variabel kategorik ini saling bergantung dimana hubungan antara kondisi keluarga dengan perilaku deliquensi anak memiliki hubungan yang kuat.

Diperkirakan rata-rata berat badan siswa kelas 2 SMA adalah kurang dari 55 kg. Akan tetapi untuk melakukan pembuktian perkiraan tersebut diambil data dan didapat sebagai berikut

berat badan:

45, 60, 65, 55, 65, 60, 50, 70, 60, 60

Apakah perkiraan ini benar?

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah berat badan dari populasi siswa kelas 2 SMA

$$\begin{array}{ll}\text{Hipotesis nol} & H_0 : \mu \leq 55. \\ \text{Hipotesis alternatif} & H_1 : \mu > 55.\end{array}$$

```
>weights = [45, 60, 65, 55, 65, 60, 50, 70, 60, 60];  
>meanEst = 55;  
>dev(weights)
```

7.3786

```
>ttest(mean(weights), dev(weights), length(weights), meanEst)
```

0.06031

$$t_{\text{hitung}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{59 - 55}{7,37/\sqrt{9}} \approx 1,63.$$

Daerah kritis:

Uji yang dilakukan diatas merupakan uji satu arah (kanan). Dengan menggunakan tabel-t, nilai-t, dengan taraf signifikansi 0.05 dan derajat kebebasan  $n - 1 = 10 - 1 = 9$ , maka

$$t_{\text{tabel}} = t_{0,05; 9} \approx 1,833.$$

Dengan demikian, diperoleh bahwa daerah kritis terletak di

$$t > 1,83.$$

Keputusan:

Karena

$$t_{\text{hitung}} = 1,63 < 1,833 = t_{\text{tabel}},$$

disimpulkan bahwa statistik uji akan terdapat diluar daerah kritis. Dengan demikian hipotesis nol tidak ditolak

Kesimpulan:

Dugaan bahwa perkiraan rata-rata berat badan siswa kelas 2 SMA kurang dari 55 kg akan di terima

Berdasarkan 100 laporan kematian di AS yang diambil secara acak, diperoleh bahwa rata-rata usia hidup orang AS adalah 71.8 tahun dengan simpangan baku 8.9 tahun. Hal ini memberikan dugaan bahwa rata-rata usia hidup orang AS lebih dari 70 tahun. Dengan taraf signifikansi 5% ujliah kebenaran dugaan tersebut.

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah rata-rata hidup dari populasi orang AS

$$\begin{array}{ll}\text{Hipotesis nol} & H_0 : \mu \leq 70. \\ \text{Hipotesis alternatif} & H_1 : \mu > 70.\end{array}$$

```
>nReport = 100; meanLifeExpect = 71.8; devLife = 8.9; meanLifeEstimate = 70;  
>ttest(meanLifeEstimate, devLife, nReport, meanLifeExpect)
```

0.022912

$$t_{\text{hitung}} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} = \frac{71,8 - 70}{8,9/\sqrt{100}} \approx 2,02.$$

Daerah kritis:

Uji yang dilakukan diatas merupakan uji satu arah (kanan). Dengan menggunakan tabel-t, nilai-t, dengan taraf signifikansi 0.05 dan derajat kebebasan  $n - 1 = 100 - 1 = 99$ , maka

$$t_{\text{tabel}} = t_{0,05; 99} \approx 1,66.$$

Dengan demikian, diperoleh bahwa daerah kritis terletak di

$$t > 1,66.$$

Keputusan:  
Karena

$$t_{\text{hitung}} = 2,02 > 1,66 = t_{\text{tabel}},$$

disimpulkan bahwa statistik uji akan terdapat pada daerah kritis. Dengan demikian hipotesis nol akan ditolak

Kesimpulan:

Dugaan bahwa rata-rata usia hidup orang AS lebih dari 70 tahun diterima.

Kesimpulan dari fungsi:

Karena nilai  $t_{\text{test}} < 0.05$ , maka hipotesis nol akan ditolak

## Latihan 7

---

Seorang perawat ingin mengetahui apakah terdapat perbedaan antara lama hari dirawat dengan kelas tempat dirawat, yang mana terdapat kelas I, II, dan kelas III, sebagai berikut

Pasien | Kelas I | Kelas II | Kelas III

1 | 5 | 2 | 3

2 | 3 | 6 | 6

3 | 2 | 4 | 7

4 | 6 | 2 | 3

5 | 3 | | 8

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah rata-rata lama perawatan dari masing-masing kelas

$$\begin{array}{ll} \text{Hipotesis nol} & H_0 : k_i = 0, i = k_1, k_2, k_3 \\ \text{Hipotesis alternatif} & H_1 : \text{setidaknya ada } k_i \neq 0 \end{array}$$

```
>classOne = [5,3,2,6,3];  
>classTwo = [2,6,4,2];  
>classThree = [3,6,7,3,8];  
>varanalysis(classOne, classTwo, classThree)
```

0.32137

Karena nilai dari fungsi  $> 0.05$ , maka kita terima hipotesis nol, yang artinya tidak ada perbedaan antara lama hari dirawat dengan kelas tempat pasien dirawat

## Latihan 8

---

Terdapat tiga kelompok subjek penelitian untuk menguji metode pengajaran mana yang paling baik dalam meningkatkan pembelajaran. Metode pertama adalah ceramah, kedua adalah diskusi dan ketiga adalah praktek

Penelitian tersebut menghasilkan data sebagai berikut.

Ceramah | Diskusi | Praktek

25 | 17 | 26

11 | 16 | 20

16 | 18 | 17

26 | 20 | 26

32 | 10 | 43

25 | 14 | 46

30 | 19 | 35

17 | 34

Penyelesaian

Rumusan Hipotesis

Parameter populasi yang akan diuji disini adalah hasil peningkatan pembelajaran untuk metode-metode diatas

$$\begin{array}{ll} \text{Hipotesis nol} & H_0 : m_i = 0, i = m_1, m_2, m_3 \\ \text{Hipotesis alternatif} & H_1 : \text{setidaknya ada } m_i \neq 0 \end{array}$$

```
>lecture = [25,11,16,26,32,25,30,17];  
>discuss = [17,16,18,20,10,14,19];  
>practice = [26,20,17,26,43,46,35,34];  
>varanalysis(lecture, discuss, practice)
```

0.0060525

karena hasil fungsi  $< 0.05$ , maka kita tolak hipotesis nol, yang artinya bahwa hasil peningkatan pembelajaran setelah melalui metode-metode tersebut memiliki hasil yang berbeda.