```
(*
Instructor：Hiroyuki Akama
```

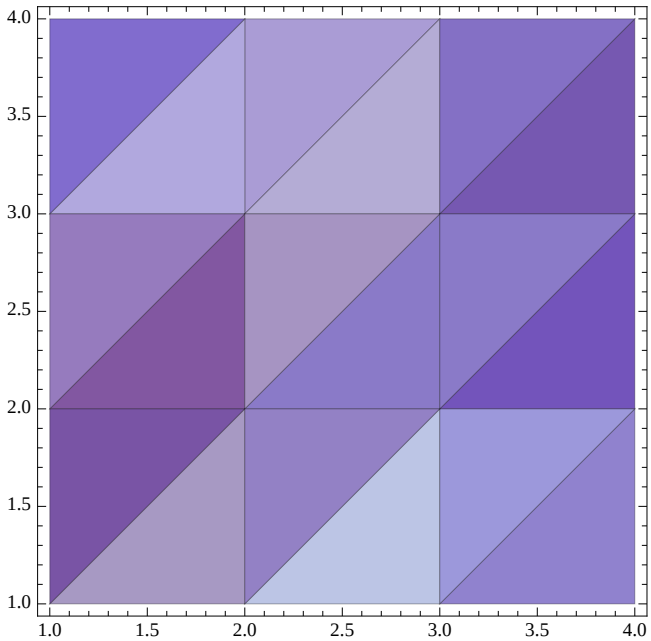
Basic Built–in Functions of Mathematica – Uploading and Downloading Data – 2D and 3D Graphics*)

```
t = {{1, 4, 2, 1}, {3, 10, 6, 0}, {0, 0, 4, 2}, {8, 8, 7, 2}};
Reverse[t]
? Reverse
```

```
{{8, 8, 7, 2}, {0, 0, 4, 2}, {3, 10, 6, 0}, {1, 4, 2, 1}}
```

Reverse[*expr*] reverses the order of the elements in *expr*.
Reverse[*expr*, *n*] reverses elements at level *n* in *expr*.
Reverse[*expr*, {n_1, n_2, …}] reverses elements at levels n_1, n_2, … in *expr*. ≫

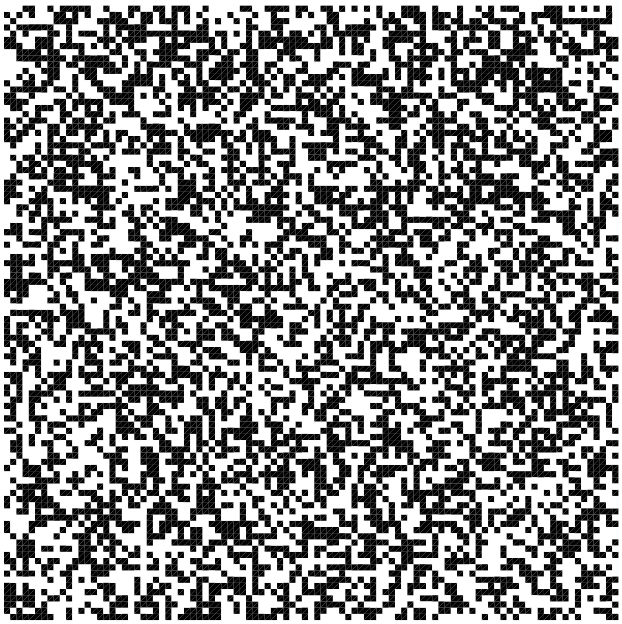
```
ListDensityPlot[Reverse[t], Mesh → All, Frame → True]
? ListDensityPlot
```



ListDensityPlot[*array*] generates a smooth density plot from an array of values.
ListDensityPlot[{{x_1, y_1, f_1}, {x_2, y_2, f_2}, …}]
generates a density plot with values defined at specified points. ≫

```
Table[RandomInteger[], {i, 1, 10}]
ListDensityPlot[Table[Table[RandomInteger[], {i, 1, 100}], {i, 1, 100}],
 ColorFunction → GrayLevel, InterpolationOrder → 0, Frame → False]
? RandomInteger
? Table
```

{1, 0, 1, 0, 1, 1, 1, 0, 1, 1}

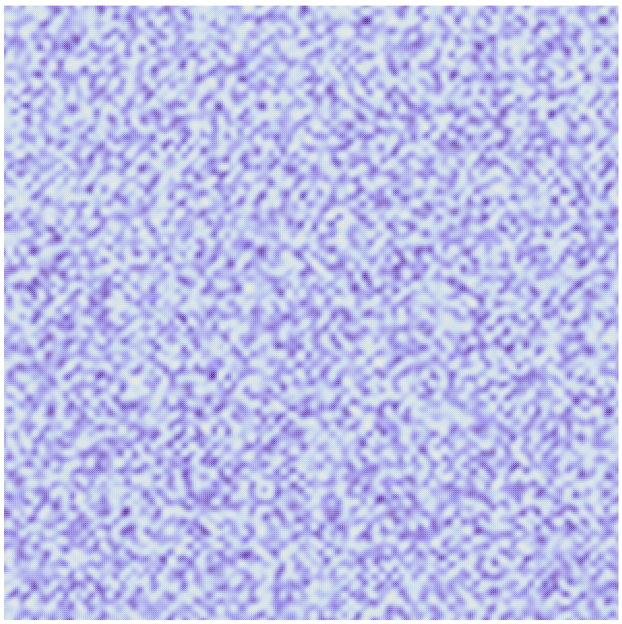


RandomInteger[{i_{min}, i_{max}}] gives a pseudorandom integer in the range {i_{min}, … , i_{max}}.
RandomInteger[i_{max}] gives a pseudorandom integer in the range {0, … , i_{max}}.
RandomInteger[] pseudorandomly gives 0 or 1.
RandomInteger[*range*, *n*] gives a list of *n* pseudorandom integers.
RandomInteger[*range*, {n_1, n_2, …}] gives an $n_1 \times n_2 \times \ldots$ array of pseudorandom integers.
RandomInteger[*dist*, …] samples from the symbolic discrete distribution *dist*. ≫

Table[*expr*, {i_{max}}] generates a list of i_{max} copies of *expr*.
Table[*expr*, {*i*, i_{max}}] generates a list of the values of *expr* when *i* runs from 1 to i_{max}.
Table[*expr*, {*i*, i_{min}, i_{max}}] starts with $i = i_{min}$.
Table[*expr*, {*i*, i_{min}, i_{max}, *di*}] uses steps *di*.
Table[*expr*, {*i*, {i_1, i_2, …}}] uses the successive values i_1, i_2, ….
Table[*expr*, {*i*, i_{min}, i_{max}}, {*j*, j_{min}, j_{max}}, …] gives a nested list. The list associated with *i* is outermost. ≫

```
Table[RandomReal[], {i, 1, 10}]
pict1 = ListDensityPlot[Table[Table[RandomReal[], {i, 1, 100}], {i, 1, 100}],
  InterpolationOrder → 5, Frame → False]
? RandomReal
```

```
{0.190833, 0.223826, 0.509133, 0.580357,
 0.246975, 0.415105, 0.495018, 0.798085, 0.182085, 0.000348291}
```



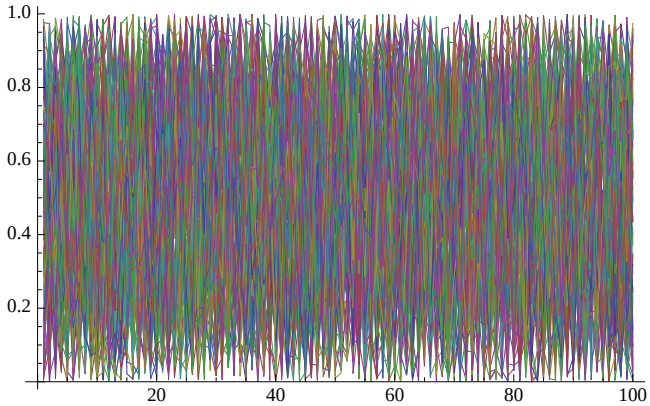
RandomReal[] gives a pseudorandom real number in the range 0 to 1.
RandomReal[{x_{min}, x_{max}}] gives a pseudorandom real number in the range x_{min} to x_{max}.
RandomReal[x_{max}] gives a pseudorandom real number in the range 0 to x_{max}.
RandomReal[*range*, *n*] gives a list of *n* pseudorandom reals.
RandomReal[*range*, {n_1, n_2, …}] gives an $n_1 \times n_2 \times \ldots$ array of pseudorandom reals.
RandomReal[*dist*, …] samples from the symbolic continuous distribution *dist*. ≫

```
Export["pict1.gif", pict1]
? Export
```

```
pict1.gif
```

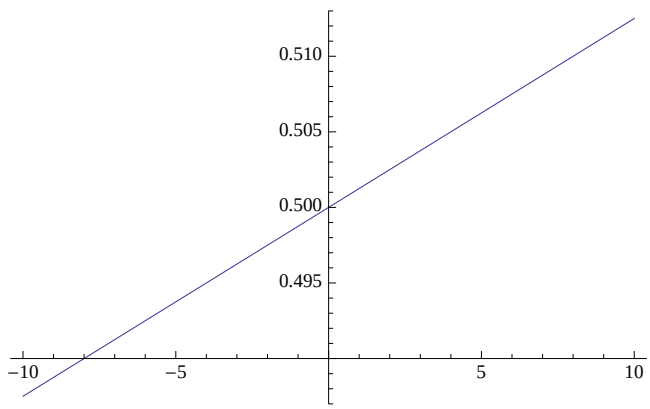
Export["*file.ext*", *expr*] exports data to a file, converting it to the format corresponding to the file extension *ext*.
Export[*file*, *expr*, "*format*"] exports data in the specified format.
Export[*file*, *exprs*, *elems*] exports data by treating *exprs* as elements specified by *elems*. ≫

```
ListLinePlot[Table[Table[RandomReal[], {i, 1, 100}], {j, 1, 100}]]
? ListLinePlot
```



ListLinePlot[{y_1, y_2, …}] plots a line through a list of values, assumed to correspond to x coordinates 1, 2, ….
ListLinePlot[{{x_1, y_1}, {x_2, y_2}, …}] plots a line through specific x and y positions.
ListLinePlot[{$list_1$, $list_2$, …}] plots several lines. ≫

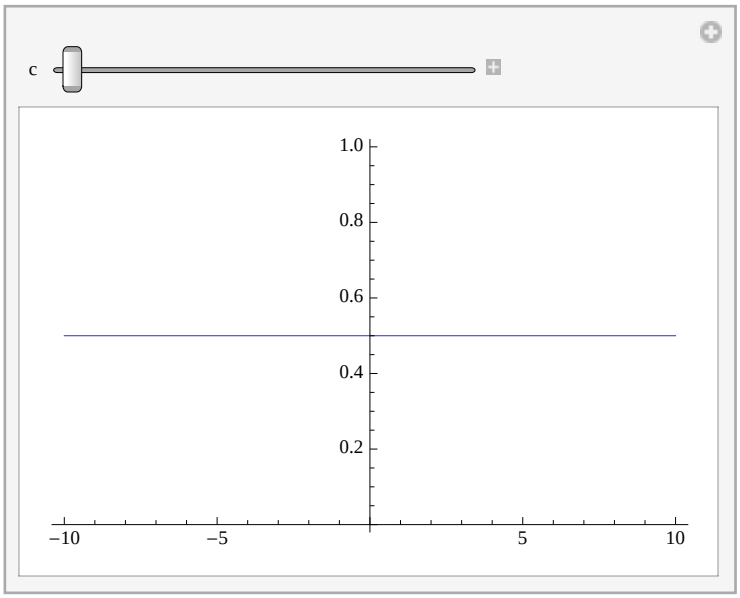
```
f[x_] := (1 + Exp[(-0.005) * x])^(-1);
Plot[f[x], {x, -10, 10}]
? Exp
? Plot
```



Exp[z] gives the exponential of z. ≫

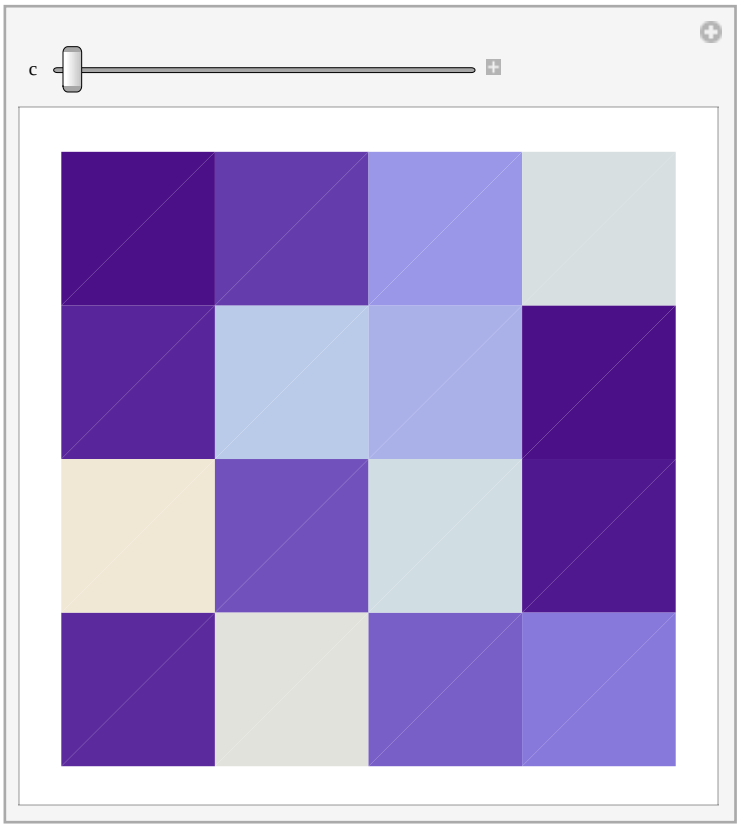
Plot[f, {x, x_{min}, x_{max}}] generates a plot of f as a function of x from x_{min} to x_{max}.
Plot[{f_1, f_2, …}, {x, x_{min}, x_{max}}] plots several functions f_i. ≫

```
Manipulate[Plot[(1 + Exp[(-c) * x])^(-1), {x, -10, 10}], {c, 0, 1}]
? Manipulate
```



Manipulate[*expr*, {u, u_{min}, u_{max}}] generates a version of
expr with controls added to allow interactive manipulation of the value of u.
Manipulate[*expr*, {u, u_{min}, u_{max}, du}] allows the value of u to vary between u_{min} and u_{max} in steps du.
Manipulate[*expr*, {{u, u_{init}}, u_{min}, u_{max}, …}] takes the initial value of u to be u_{init}.
Manipulate[*expr*, {{u, u_{init}, u_{lbl}}, …}] labels the controls for u with u_{lbl}.
Manipulate[*expr*, {u, {u_1, u_2, …}}] allows u to take on discrete values u_1, u_2, ….
Manipulate[*expr*, {u, …}, {v, …}, …] provides controls to manipulate each of the u, v, ….
Manipulate[*expr*, c_u −> {u, …}, c_v −> {v, …}, …]
links the controls to the specified controllers on an external device. ≫

```
Manipulate[ListDensityPlot[Table[Table[RandomReal[], {i, 1, 5}], {i, 1, 5}],
  InterpolationOrder → c, Frame → False], {c, 0, 3}]
```




```
data1 = {{1, 0.0648412}, {1, 0.0803853}, {1, 0.0820752}, {1, 0.0864502}, {1, 0.0938933},
   {1, 0.0944069}, {1, 0.0951256}, {1, 0.0985412}, {1, 0.0990143}, {1, 0.10053},
   {1, 0.101813}, {1, 0.102028}, {1, 0.102532}, {1, 0.102562}, {1, 0.103437}, {1, 0.104444},
   {1, 0.104766}, {1, 0.106194}, {1, 0.110988}, {1, 0.112586}, {1, 0.114251}, {1, 0.114443},
   {1, 0.114963}, {1, 0.116433}, {1, 0.116433}, {1, 0.117504}, {1, 0.121111}, {1, 0.123744},
   {1, 0.124699}, {1, 0.130678}, {1, 0.13277}, {1, 0.132849}, {1, 0.135731}, {1, 0.136159},
   {1, 0.136774}, {1, 0.137725}, {1, 0.137725}, {1, 0.140201}, {1, 0.145363}, {1, 0.157755},
   {1, 0.159125}, {1, 0.160376}, {1, 0.16077}, {1, 0.161959}, {1, 0.16289}, {1, 0.163187},
   {1, 0.164903}, {1, 0.165592}, {1, 0.172321}, {1, 0.172524}, {1, 0.172534}, {1, 0.172534},
   {1, 0.172534}, {1, 0.172534}, {1, 0.172534}, {1, 0.172827}, {1, 0.172827}, {1, 0.172988},
   {1, 0.172988}, {1, 0.17532}, {1, 0.17532}, {1, 0.17532}, {1, 0.17532}, {1, 0.17532},
   {1, 0.183077}, {1, 0.183077}, {1, 0.186067}, {1, 0.191489}, {1, 0.193131}, {1, 0.201935},
   {1, 0.226004}, {1, 0.226004}, {1, 0.230835}, {1, 0.29222}, {2, 0.0106212}, {2, 0.0111648},
   {2, 0.0112381}, {2, 0.0114639}, {2, 0.0115644}, {2, 0.0115644}, {2, 0.0115644},
   {2, 0.0115644}, {2, 0.0115644}, {2, 0.0116063}, {2, 0.0116748}, {2, 0.0116748},
   {2, 0.0116748}, {2, 0.0116748}, {2, 0.0116748}, {2, 0.0118687}, {2, 0.0118736},
```

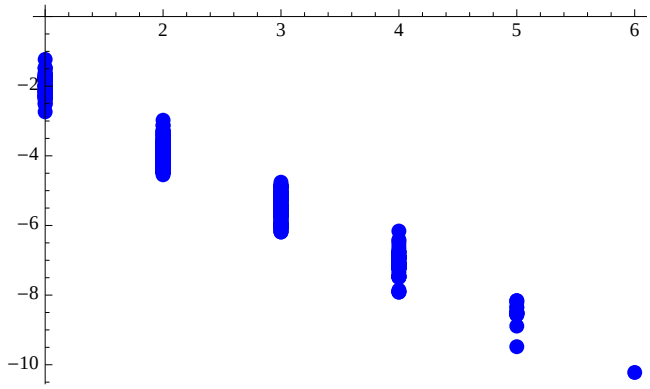
```
{2, 0.0119276}, {2, 0.0120884}, {2, 0.0120916}, {2, 0.0123302}, {2, 0.0124946},
{2, 0.0125849}, {2, 0.0128111}, {2, 0.0128209}, {2, 0.0129409}, {2, 0.0129885},
{2, 0.0130572}, {2, 0.0131763}, {2, 0.0131763}, {2, 0.013425}, {2, 0.0135381},
{2, 0.0136616}, {2, 0.0137489}, {2, 0.0137489}, {2, 0.01377}, {2, 0.01377}, {2, 0.0140361},
{2, 0.0141789}, {2, 0.0142828}, {2, 0.0147516}, {2, 0.0147896}, {2, 0.0147896},
{2, 0.0148011}, {2, 0.0148011}, {2, 0.0148011}, {2, 0.0148011}, {2, 0.0148011},
{2, 0.0148384}, {2, 0.0148384}, {2, 0.0148384}, {2, 0.0148384}, {2, 0.0148384},
{2, 0.0148736}, {2, 0.0152229}, {2, 0.0153301}, {2, 0.0153301}, {2, 0.0154376},
{2, 0.0154675}, {2, 0.0156443}, {2, 0.0156967}, {2, 0.0158035}, {2, 0.0158099},
{2, 0.0158411}, {2, 0.0158411}, {2, 0.0158411}, {2, 0.0158411}, {2, 0.0158411},
{2, 0.0160669}, {2, 0.0160669}, {2, 0.0161741}, {2, 0.016188}, {2, 0.016198},
{2, 0.0163461}, {2, 0.0164367}, {2, 0.0164561}, {2, 0.0164969}, {2, 0.0167484},
{2, 0.0167484}, {2, 0.016992}, {2, 0.017056}, {2, 0.0173831}, {2, 0.0173831},
{2, 0.0174425}, {2, 0.0174452}, {2, 0.0174452}, {2, 0.0175075}, {2, 0.0175622},
{2, 0.0176784}, {2, 0.0178541}, {2, 0.0179854}, {2, 0.0179854}, {2, 0.0179854},
{2, 0.0179854}, {2, 0.0180185}, {2, 0.0180359}, {2, 0.0180359}, {2, 0.0182435},
{2, 0.0182435}, {2, 0.0184526}, {2, 0.0184598}, {2, 0.0184598}, {2, 0.0184729},
{2, 0.0184729}, {2, 0.0184729}, {2, 0.0184729}, {2, 0.0185446}, {2, 0.0186784},
{2, 0.0186784}, {2, 0.0186784}, {2, 0.0186784}, {2, 0.0186784}, {2, 0.0187208},
{2, 0.0187277}, {2, 0.0187277}, {2, 0.0187561}, {2, 0.0187812}, {2, 0.018842},
{2, 0.0188529}, {2, 0.0188529}, {2, 0.0189167}, {2, 0.0189167}, {2, 0.018933},
{2, 0.0189349}, {2, 0.0189349}, {2, 0.0189918}, {2, 0.0192258}, {2, 0.0192408},
{2, 0.0192612}, {2, 0.0193488}, {2, 0.0194833}, {2, 0.0194833}, {2, 0.0195637},
{2, 0.0196842}, {2, 0.0199989}, {2, 0.0201272}, {2, 0.0201614}, {2, 0.0201614},
{2, 0.020246}, {2, 0.020246}, {2, 0.0204498}, {2, 0.0206093}, {2, 0.0206093},
{2, 0.0206489}, {2, 0.0207042}, {2, 0.0207524}, {2, 0.0208038}, {2, 0.0208261},
{2, 0.0210092}, {2, 0.0210092}, {2, 0.0211889}, {2, 0.021194}, {2, 0.0215231},
{2, 0.0217795}, {2, 0.0218524}, {2, 0.0220906}, {2, 0.0221608}, {2, 0.0223371},
{2, 0.0223371}, {2, 0.0223371}, {2, 0.0223371}, {2, 0.0223371}, {2, 0.0223598},
{2, 0.0228957}, {2, 0.0229074}, {2, 0.0229074}, {2, 0.0229074}, {2, 0.0229074},
{2, 0.0229074}, {2, 0.0230004}, {2, 0.0231991}, {2, 0.0233399}, {2, 0.0233417},
{2, 0.0233417}, {2, 0.0234537}, {2, 0.0236894}, {2, 0.0238375}, {2, 0.0240061},
{2, 0.0240244}, {2, 0.0240244}, {2, 0.0240244}, {2, 0.0240244}, {2, 0.0240244},
{2, 0.0242711}, {2, 0.0246887}, {2, 0.024697}, {2, 0.0248199}, {2, 0.0248199},
{2, 0.0248199}, {2, 0.0248199}, {2, 0.0248199}, {2, 0.0248199}, {2, 0.0248199},
{2, 0.0248199}, {2, 0.0248199}, {2, 0.0248199}, {2, 0.0248335}, {2, 0.0251611},
{2, 0.0251611}, {2, 0.0251611}, {2, 0.0251611}, {2, 0.0251611}, {2, 0.0252968},
{2, 0.0253052}, {2, 0.0253052}, {2, 0.0253052}, {2, 0.0253052}, {2, 0.0253052},
{2, 0.0256556}, {2, 0.0257211}, {2, 0.0257211}, {2, 0.0267578}, {2, 0.0270918},
{2, 0.0276236}, {2, 0.0276236}, {2, 0.0279984}, {2, 0.0279984}, {2, 0.0285625},
{2, 0.0286091}, {2, 0.0287881}, {2, 0.0290461}, {2, 0.0297613}, {2, 0.0297613},
{2, 0.0311322}, {2, 0.0316668}, {2, 0.0321355}, {2, 0.0333149}, {2, 0.0333149},
{2, 0.0337232}, {2, 0.0338792}, {2, 0.0349256}, {2, 0.0363101}, {2, 0.0364543},
{2, 0.0372224}, {2, 0.0372704}, {2, 0.0386304}, {2, 0.0439246}, {2, 0.0439246},
{2, 0.0509282}, {3, 0.00205521}, {3, 0.00205521}, {3, 0.00211804}, {3, 0.00211804},
{3, 0.00227498}, {3, 0.0023128}, {3, 0.00232445}, {3, 0.00232445}, {3, 0.00234521},
{3, 0.00234521}, {3, 0.00239435}, {3, 0.00239435}, {3, 0.00241297}, {3, 0.0025161},
{3, 0.0025161}, {3, 0.0025161}, {3, 0.0025161}, {3, 0.0025161}, {3, 0.0025238},
{3, 0.00254319}, {3, 0.00254319}, {3, 0.00256853}, {3, 0.00256853}, {3, 0.00258207},
{3, 0.0025849}, {3, 0.00260874}, {3, 0.00260874}, {3, 0.00262093}, {3, 0.00262093},
{3, 0.00280113}, {3, 0.00282348}, {3, 0.00289776}, {3, 0.00299236}, {3, 0.00303471},
```

```
{3, 0.00320357}, {3, 0.00320357}, {3, 0.00321524}, {3, 0.00327168}, {3, 0.00327509},
{3, 0.00328655}, {3, 0.0033382}, {3, 0.0033382}, {3, 0.00342768}, {3, 0.00342768},
{3, 0.00342768}, {3, 0.00342768}, {3, 0.00342768}, {3, 0.00359503}, {3, 0.00368399},
{3, 0.00369446}, {3, 0.00369446}, {3, 0.00372408}, {3, 0.0038734}, {3, 0.00394364},
{3, 0.0039877}, {3, 0.00402789}, {3, 0.00405519}, {3, 0.00406992}, {3, 0.00407206},
{3, 0.00407206}, {3, 0.00412778}, {3, 0.00412778}, {3, 0.00414501}, {3, 0.00414501},
{3, 0.00414882}, {3, 0.0041743}, {3, 0.00423424}, {3, 0.00423424}, {3, 0.00423424},
{3, 0.00423424}, {3, 0.00423424}, {3, 0.00427685}, {3, 0.00431036}, {3, 0.00431036},
{3, 0.00437734}, {3, 0.00437734}, {3, 0.00439167}, {3, 0.00456935}, {3, 0.00459786},
{3, 0.00470815}, {3, 0.00478114}, {3, 0.00484513}, {3, 0.00490962}, {3, 0.00513291},
{3, 0.00517821}, {3, 0.00520335}, {3, 0.00521253}, {3, 0.00534471}, {3, 0.00534471},
{3, 0.00534471}, {3, 0.00534471}, {3, 0.00534471}, {3, 0.00540098}, {3, 0.0054117},
{3, 0.00546937}, {3, 0.0054907}, {3, 0.00554164}, {3, 0.00563268}, {3, 0.00566627},
{3, 0.00581168}, {3, 0.00584888}, {3, 0.00585368}, {3, 0.00585368}, {3, 0.00585368},
{3, 0.00585368}, {3, 0.00585368}, {3, 0.00599675}, {3, 0.00599675}, {3, 0.00602349},
{3, 0.00602415}, {3, 0.00602415}, {3, 0.00623683}, {3, 0.00627862}, {3, 0.00631636},
{3, 0.00640368}, {3, 0.0064052}, {3, 0.0064052}, {3, 0.0064052}, {3, 0.0064052},
{3, 0.0064052}, {3, 0.00641928}, {3, 0.00642579}, {3, 0.00698884}, {3, 0.00708291},
{3, 0.00722696}, {3, 0.00729516}, {3, 0.00736695}, {3, 0.00737641}, {3, 0.00757965},
{3, 0.00762139}, {3, 0.00762542}, {3, 0.00762542}, {3, 0.00762542}, {3, 0.00762542},
{3, 0.00762542}, {3, 0.00778245}, {3, 0.00806451}, {3, 0.00858332}, {4, 0.000365721},
{4, 0.000365721}, {4, 0.000371711}, {4, 0.000371711}, {4, 0.000387225}, {4, 0.000560984},
{4, 0.000560984}, {4, 0.000562981}, {4, 0.000572769}, {4, 0.000572769}, {4, 0.000573609},
{4, 0.000573609}, {4, 0.000610372}, {4, 0.000664759}, {4, 0.000714982}, {4, 0.000734336},
{4, 0.000734336}, {4, 0.000736481}, {4, 0.000736481}, {4, 0.000736481}, {4, 0.000736481},
{4, 0.000736481}, {4, 0.000736481}, {4, 0.000736481}, {4, 0.000736481}, {4, 0.000736481},
{4, 0.000736481}, {4, 0.000739631}, {4, 0.000748577}, {4, 0.00075453}, {4, 0.00075453},
{4, 0.000755613}, {4, 0.000755613}, {4, 0.000755613}, {4, 0.000755613}, {4, 0.000755613},
{4, 0.000755613}, {4, 0.000755613}, {4, 0.000755613}, {4, 0.000755613}, {4, 0.000755613},
{4, 0.00076622}, {4, 0.000794919}, {4, 0.000794919}, {4, 0.000821943}, {4, 0.000821943},
{4, 0.00082774}, {4, 0.00082774}, {4, 0.00082774}, {4, 0.00082774}, {4, 0.00082774},
{4, 0.00083202}, {4, 0.00083202}, {4, 0.000836596}, {4, 0.00084607}, {4, 0.00084607},
{4, 0.000856549}, {4, 0.000856549}, {4, 0.000947791}, {4, 0.00094987}, {4, 0.000952986},
{4, 0.000952986}, {4, 0.000952986}, {4, 0.000952986}, {4, 0.000952986}, {4, 0.000961241},
{4, 0.00100795}, {4, 0.00100795}, {4, 0.00100942}, {4, 0.00100942}, {4, 0.00100942},
{4, 0.00100942}, {4, 0.00100942}, {4, 0.00100942}, {4, 0.00100942}, {4, 0.00100942},
{4, 0.00100942}, {4, 0.00100942}, {4, 0.00104956}, {4, 0.00110901}, {4, 0.00111822},
{4, 0.00115146}, {4, 0.00115146}, {4, 0.00116503}, {4, 0.00116503}, {4, 0.00122613},
{4, 0.0013354}, {4, 0.00147707}, {4, 0.00158587}, {4, 0.00164844}, {4, 0.00211145},
{5, 0.0000762661}, {5, 0.000137823}, {5, 0.00019081}, {5, 0.00019081}, {5, 0.000194625},
{5, 0.000194625}, {5, 0.000201037}, {5, 0.000201037}, {5, 0.000201037}, {5, 0.000201037},
{5, 0.000201037}, {5, 0.000202701}, {5, 0.000208913}, {5, 0.000234414}, {5, 0.000237548},
{5, 0.000270872}, {5, 0.000285862}, {5, 0.000285862}, {6, 0.0000363664}};
```

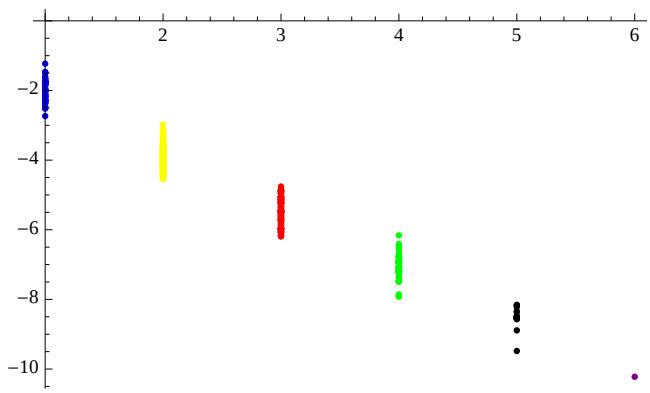
```
ListPlot[data1, PlotStyle → {Red}, AspectRatio → Automatic]
? ListPlot
```

ListPlot[{y_1, y_2, …}] plots points corresponding to a list of values, assumed to correspond to x coordinates 1, 2, … .
ListPlot[{{x_1, y_1}, {x_2, y_2}, …}] plots a list of points with specified x and y coordinates.
ListPlot[{$list_1$, $list_2$, …}] plots several lists of points. ≫

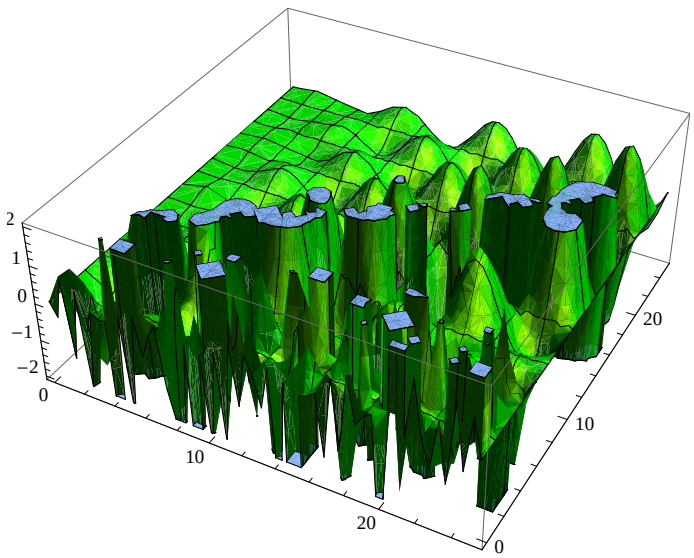
```
ListPlot[{#[[1]], Log[#[[2]]]} & /@ data1, PlotStyle → {Blue, PointSize[Large]}]
```




```
data1splitlog = Split[{#[[1]], Log[#[[2]]]} & /@ data1, First[#1] == First[#2] &];
ListPlot[data1splitlog, PlotStyle → {Blue, Yellow, Red, Green, Black, Purple}]
```



```
pict = Plot3D[Sin[x] Cos[y] Tan[x / y], {x, 0, 8 Pi}, {y, 0, 8 Pi},
  PlotStyle → Directive[Opacity[1.0], Green, Specularity[Yellow, 10]]]
? Plot3D
```



Plot3D[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] generates a three−dimensional plot of f as a function of x and y.
Plot3D[{f_1, f_2, …}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] plots several functions. ≫

```
Export["3dgreen.bmp", pict, ImageResolution -> Automatic, ImageSize -> {850, 850}]
```

```
3dgreen.bmp
```

```
pictimp = Import["3dgreen.bmp"];
? Import
```

Import["*file*"] imports data from a file, returning a complete *Mathematica* version of it.
Import["*file*", *elements*] imports the specified elements from a file.
Import["http://*url*", …] and Import["ftp://*url*", …] imports from any accessible URL. ≫

```
CharacterRange["a", "z"]
? CharacterRange
```

```
{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}
```

CharacterRange[c_1, c_2] yields a list of the characters in the range from "c_1" to "c_2". ≫

```
toIntRule = MapThread[#2 → #1 &, {CharacterRange["a", "z"], Insert[Range[25], 0, 1]}]
? MapThread
```

```
{0 → a, 1 → b, 2 → c, 3 → d, 4 → e, 5 → f, 6 → g, 7 → h, 8 → i, 9 → j, 10 → k, 11 → l, 12 → m, 13 → n,
 14 → o, 15 → p, 16 → q, 17 → r, 18 → s, 19 → t, 20 → u, 21 → v, 22 → w, 23 → x, 24 → y, 25 → z}
```

MapThread[f, {{a_1, a_2, …}, {b_1, b_2, …}, …}] gives {f[a_1, b_1, …], f[a_2, b_2, …], …}.
MapThread[f, {$expr_1$, $expr_2$, …}, n] applies f to the parts of the $expr_i$ at level n. ≫

```
randints = Table[RandomInteger[25], {i, 1, 50}]
randchars = (# /. toIntRule) & /@ Table[RandomInteger[25], {i, 1, 150}]
randstrings = StringJoin[#] & /@ Partition[randchars, 3]
randdata = Transpose[{randstrings, randints}]
```

```
{18, 9, 17, 16, 25, 21, 12, 8, 18, 4, 0, 10, 4, 18, 25, 24, 8, 12, 8, 19, 9, 13, 25, 14, 8,
 16, 6, 16, 15, 15, 22, 1, 17, 19, 14, 14, 10, 10, 22, 7, 3, 23, 9, 1, 0, 16, 0, 3, 7, 2}
```

```
{q, m, h, g, g, a, z, s, l, p, g, j, b, l, z, z, r, x, y, m, p, s, l, v, c, y, x, p, n, n,
 j, c, w, g, i, q, v, o, i, g, o, w, f, j, t, k, r, q, b, h, e, o, e, f, s, k, w, h, n, z,
 v, z, g, x, i, t, g, g, o, l, v, z, r, x, e, t, y, o, z, p, j, x, b, o, s, a, p, d, v, w,
 h, f, y, p, s, a, d, j, v, n, l, l, r, h, o, m, m, i, g, o, c, x, r, v, b, f, i, i, c, i,
 p, z, v, b, o, g, l, b, l, f, h, o, o, m, y, i, k, m, y, a, p, f, u, q, q, l, b, c, x, c}
```

```
{qmh, gga, zsl, pgj, blz, zrx, ymp, slv, cyx, pnn, jcw, giq, voi, gow, fjt, krq,
 bhe, oef, skw, hnz, vzg, xit, ggo, lvz, rxe, tyo, zpj, xbo, sap, dvw, hfy, psa, djv,
 nll, rho, mmi, goc, xrv, bfi, ici, pzv, bog, lbl, fho, omy, ikm, yap, fuq, qlb, cxc}
```

```
{{qmh, 18}, {gga, 9}, {zsl, 17}, {pgj, 16}, {blz, 25}, {zrx, 21}, {ymp, 12}, {slv, 8}, {cyx, 18},
 {pnn, 4}, {jcw, 0}, {giq, 10}, {voi, 4}, {gow, 18}, {fjt, 25}, {krq, 24}, {bhe, 8},
 {oef, 12}, {skw, 8}, {hnz, 19}, {vzg, 9}, {xit, 13}, {ggo, 25}, {lvz, 14}, {rxe, 8},
 {tyo, 16}, {zpj, 6}, {xbo, 16}, {sap, 15}, {dvw, 15}, {hfy, 22}, {psa, 1}, {djv, 17},
 {nll, 19}, {rho, 14}, {mmi, 14}, {goc, 10}, {xrv, 10}, {bfi, 22}, {ici, 7}, {pzv, 3},
 {bog, 23}, {lbl, 9}, {fho, 1}, {omy, 0}, {ikm, 16}, {yap, 0}, {fuq, 3}, {qlb, 7}, {cxc, 2}}
```

```
Export["randdata", randdata, "Table"]
```

```
randdata
```

```
Import["randdata"]
```


```
{{qmh, 18}, {gga, 9}, {zsl, 17}, {pgj, 16}, {blz, 25}, {zrx, 21}, {ymp, 12}, {slv, 8}, {cyx, 18},
 {pnn, 4}, {jcw, 0}, {giq, 10}, {voi, 4}, {gow, 18}, {fjt, 25}, {krq, 24}, {bhe, 8},
 {oef, 12}, {skw, 8}, {hnz, 19}, {vzg, 9}, {xit, 13}, {ggo, 25}, {lvz, 14}, {rxe, 8},
 {tyo, 16}, {zpj, 6}, {xbo, 16}, {sap, 15}, {dvw, 15}, {hfy, 22}, {psa, 1}, {djv, 17},
 {nll, 19}, {rho, 14}, {mmi, 14}, {goc, 10}, {xrv, 10}, {bfi, 22}, {ici, 7}, {pzv, 3},
 {bog, 23}, {lbl, 9}, {fho, 1}, {omy, 0}, {ikm, 16}, {yap, 0}, {fuq, 3}, {qlb, 7}, {cxc, 2}}
```