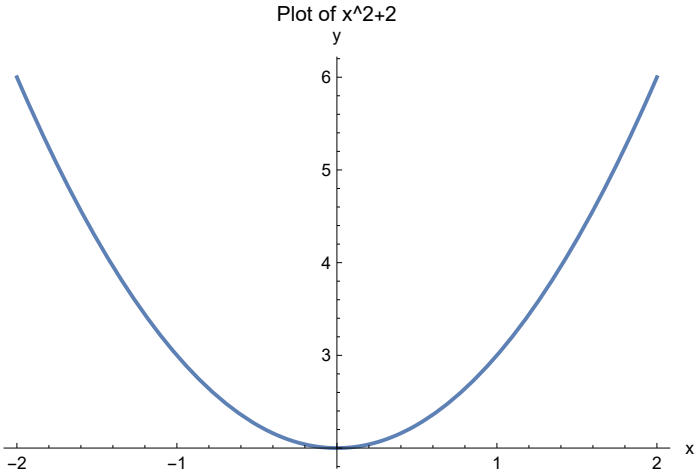
```
(*Graphics*)

(*In this section,we introduce  basic plotting commands and show how to use them.*)
```

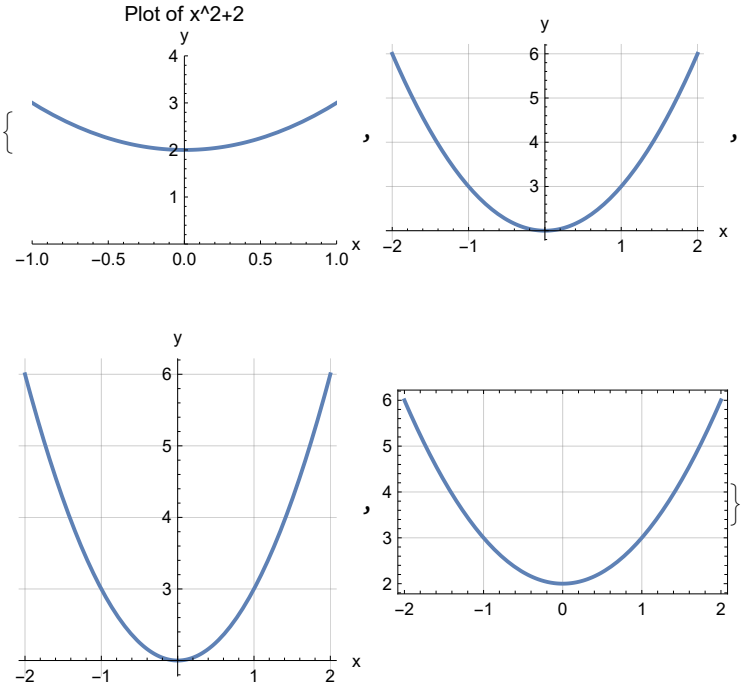
In[1]:=
```
Plot[x^2 + 2, {x, -2, 2}, AxesLabel → {"x", "y"}, PlotLabel → "Plot of x^2+2"]
```

Out[1]=




In[2]:=
```
{Plot[x^2 + 2, {x, -2, 2}, AxesLabel → {"x", "y"},
  PlotLabel → "Plot of x^2+2", GridLines → None, PlotRange → {{-1, 1}, {0, 4}}],
 Plot[x^2 + 2, {x, -2, 2}, AxesLabel → {"x", "y"}, GridLines → Automatic],
 Plot[x^2 + 2, {x, -2, 2}, AxesLabel → {"x", "y"}, GridLines → Automatic,
  AspectRatio → 1],
 Plot[x^2 + 2, {x, -2, 2}, AxesLabel → {"x", "y"}, GridLines → Automatic, Axes → None,
  Frame → True]}
```

Out[2]=



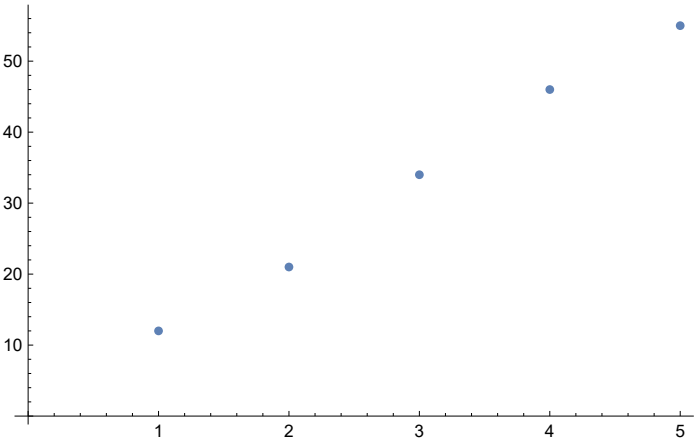
In[3]:=
```
(*Plotting a graph using ListPlot command for discrete data:

The plot command works on "vectors" (lists) of numerical data.For example,
supposed that you have experimental data set listed separately for the X
 and Y components, such as Xexp={1,2,3,4,5} and the corresponding response,
Yexp={12,21,34,46,55},you can type:*)
```

In[4]:=
```
Xexp = {1, 2, 3, 4, 5}; Yexp = {12, 21, 34, 46, 55};
```

In[5]:=
```
ListPlot[Yexp](*since Xexp is a natural list*)
```

Out[5]=




In[6]:=
```
(*or, create a list of {X,Y}pairs, which is the outcome of most experiments*)
```

In[7]:=
```
(*let's have a look*){Xexp, Yexp} // MatrixForm
```

Out[7]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 12 & 21 & 34 & 46 & 55 \end{pmatrix}$$

In[8]:=
```
data = Transpose[{Xexp, Yexp}]
```

Out[8]=
```
{{1, 12}, {2, 21}, {3, 34}, {4, 46}, {5, 55}}
```

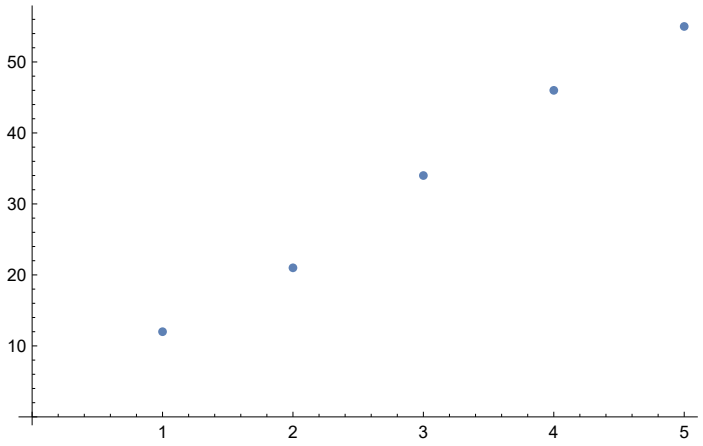
In[9]:=
```
MatrixForm[data]
```

Out[9]//MatrixForm=

$$\begin{pmatrix} 1 & 12 \\ 2 & 21 \\ 3 & 34 \\ 4 & 46 \\ 5 & 55 \end{pmatrix}$$

In[10]:=
```
ListPlot[data, AxesOrigin → {0, 0}] (*gives the same*)
```

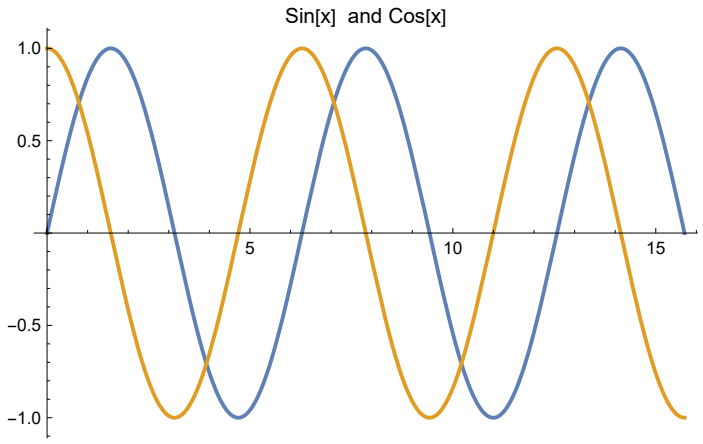
Out[10]=




In[11]:=
```
(*Plotting Multiple Curves in a plot
If you want to plot two or more graphs in a single plot,
you can list the functios as a list (if they are in the same domain)
 or use the "Show" command to combine separate plots. For example,
if you want to plot sin(x) and cos(x) curves in a single plot:*)
```

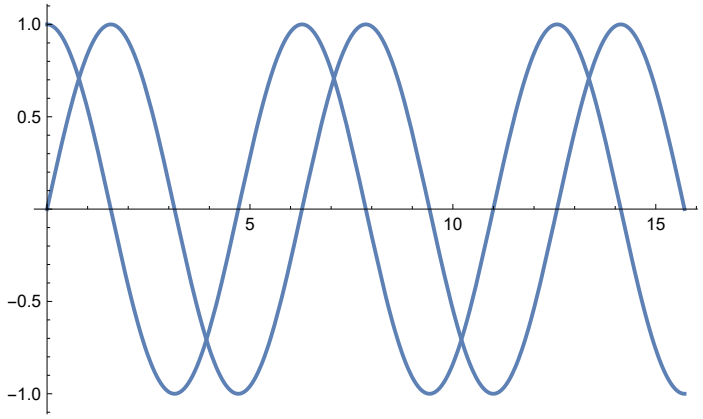
In[12]:=
```
Plot[{Sin[x], Cos[x]}, {x, 0, 5 Pi}, PlotLabel → "Sin[x]  and Cos[x]"]
```

Out[12]=



In[13]:=
```
(*or*)Show[Plot[Sin[x], {x, 0, 5 Pi}], Plot[Cos[x], {x, 0, 5 Pi}]]
(*gives the same; you can adjust color/style of curves*)
```

Out[13]=




In[14]:=
```
(*similarly, you can define fig1=Plot[Sin[x], {x,0,5Pi}],
fig2=Plot[Cos[x], {x,0,5Pi}] and then use Show[fig1,fig2 – check!]*)
```

In[15]:=
```
(*Exporting your graphics*)
```

In[16]:=
```
myfig = Show[Plot[Sin[x], {x, 0, 5 Pi}], Plot[Cos[x], {x, 0, 5 Pi}]];
```

In[17]:=
```
Export["mySinCosfig.JPEG", myfig, "JPEG"]
(*the file "...JPEG is created outside Mathwmatica"*)
```

Out[17]=
```
mySinCosfig.JPEG
```

In[18]:=
```
(*The following example shows you how to show
 together a discrete and a continous plots.In this example,
supposed that you conducted a simple harmonic motion experiment for spring system,
we have experimental data set of mass (mexp) vs.period (Texp) as shown below.*)
mexp = {0.06, 0.08, 0.1, 0.12, 0.16};
Texp = {4.5, 5.2, 5.8, 6.3, 7.2};
```

In[20]:=
```
(*we first create pairs of data as you expect from experiment*)
```

In[21]:=
```
mydata = Transpose[{mexp, Texp}]
```

Out[21]=
```
{{0.06, 4.5}, {0.08, 5.2}, {0.1, 5.8}, {0.12, 6.3}, {0.16, 7.2}}
```

In[22]:=
```
mydata // MatrixForm
```

Out[22]//MatrixForm=

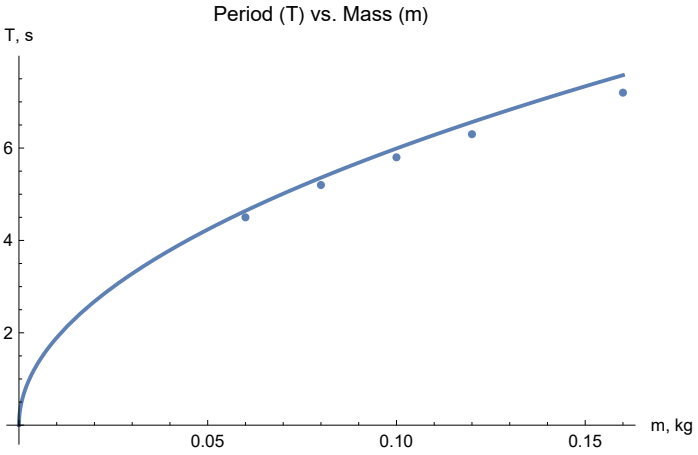
$$\begin{pmatrix} 0.06 & 4.5 \\ 0.08 & 5.2 \\ 0.1 & 5.8 \\ 0.12 & 6.3 \\ 0.16 & 7.2 \end{pmatrix}$$

In[23]:=
```
(*The spring constant (k) is 0.11.The equation of simple harmonic motion is given by T=
 2Pi Sqrt[m/k].The  code will be as following*)
```

In[24]:=
```
Clear[T, m, k]; k = .11; T[m_] := 2 Pi Sqrt[m / k]
```

In[25]:=
```
Show[Plot[T[m], {m, 0, 0.16}], ListPlot[mydata],
 AxesLabel → {"m, kg", "T, s"}, PlotLabel → "Period (T) vs. Mass (m)"]
```

Out[25]=




In[26]:=
```
(*Note: if possible, for a simple known or assumed theoretical dependence it is better
 to use variables with a linear dependence; let us repeat the previous example: *)
```

In[27]:=

In[28]:=
```
mexp = {0.06, 0.08, 0.1, 0.12, 0.16};
Texp = {4.5, 5.2, 5.8, 6.3, 7.2};
```

In[30]:=
```
Texp2 = Texp^2
```

Out[30]=
```
{20.25, 27.04, 33.64, 39.69, 51.84}
```

In[31]:=
```
mydata2 = Transpose[{mexp, Texp2}]
```

Out[31]=
```
{{0.06, 20.25}, {0.08, 27.04}, {0.1, 33.64}, {0.12, 39.69}, {0.16, 51.84}}
```

In[32]:=
```
Clear[T2, m, k]; k = .11; T2[m_] := (2 Pi Sqrt[m / k])^2
```

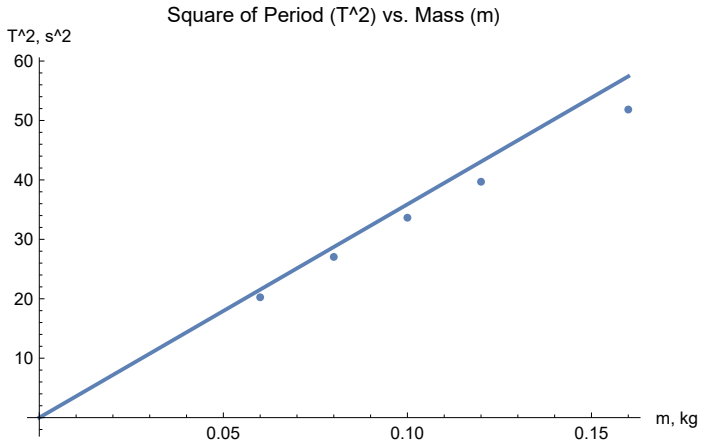
In[33]:=
```
T2[m]
```

Out[33]=
```
358.895 m
```

In[34]:=
```
figTheo = Plot[T2[m], {m, 0, 0.16}]; figExp = ListPlot[mydata2];
```

In[35]:=
```
Show[figTheo, figExp, AxesLabel → {"m, kg", "T^2, s^2"},
 PlotLabel → "Square of Period (T^2) vs. Mass (m)"]
```

Out[35]=




In[36]:=
```
(*Linear fitting of data*)
```

In[37]:=
```
Fexp = {1.9, 3.9, 5.9, 6.9, 7.8};
Dexp = {0.03, 0.1, 0.16, 0.20, 0.23};
```

In[39]:=
```
data = Transpose[{Dexp, Fexp}]
```

Out[39]=
```
{{0.03, 1.9}, {0.1, 3.9}, {0.16, 5.9}, {0.2, 6.9}, {0.23, 7.8}}
```

In[40]:=
```
data // MatrixForm
```

Out[40]//MatrixForm=

$$\begin{pmatrix} 0.03 & 1.9 \\ 0.1 & 3.9 \\ 0.16 & 5.9 \\ 0.2 & 6.9 \\ 0.23 & 7.8 \end{pmatrix}$$

In[41]:=
```
LinearModelFit[data, x, x]
```

Out[41]=
```
FittedModel[ 1.01 + 29.7 x ]
```

In[42]:=
```
nfit = Normal[%]
```

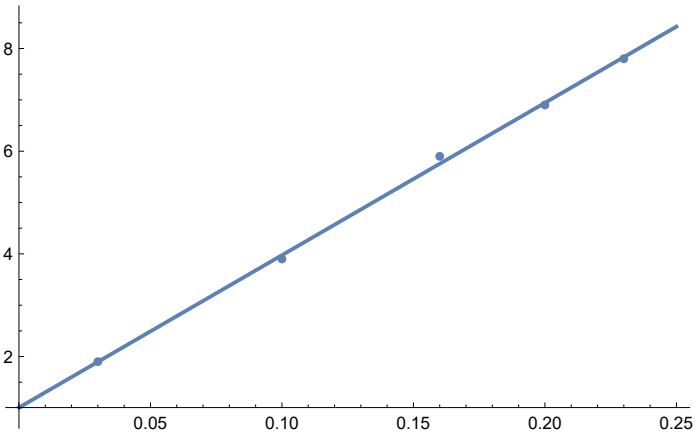
Out[42]=
```
1.00591 + 29.6812 x
```

In[43]:=

In[44]:=

In[45]:=
```
Show[Plot[nfit, {x, 0, 0.25}], ListPlot[data]]
```

Out[45]=




› System`

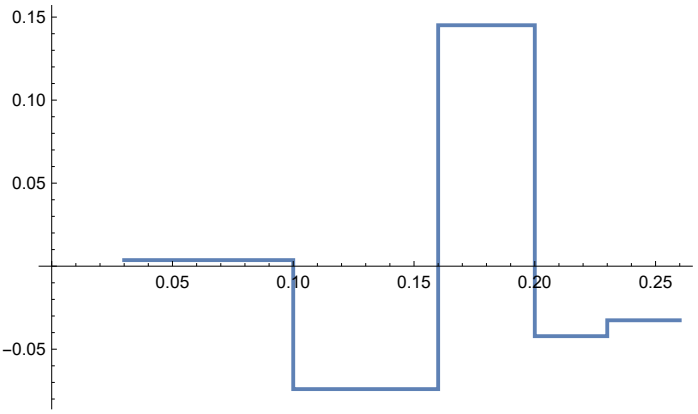
In[47]:=
```
dev = Fexp - (nfit /. x → Dexp)
```

Out[47]=

{0.00365474, −0.074028, 0.145101, −0.0421462, −0.0325816}

In[48]:=
```
ListStepPlot[Transpose[{Dexp, dev}]]
```

Out[48]=




In[49]:=
```
list = Fexp - Mean[Fexp] * Table[1, {Length[Fexp]}]
```

Out[49]=

{−3.38, −1.38, 0.62, 1.62, 2.52}

In[50]:=
```
devMean = Norm[list]^2
```

Out[50]=

22.688

In[51]:=

In[52]:=
```
totDev = Norm[dev]^2
```
Out[52]=
```
0.0293857
```

In[53]:=
```
r2 = 1 - totDev / devMean
```
Out[53]=
```
0.998705
```

In[54]:=
```
(*good!*)
```

In[55]:=
```
(*Graphics  of vectors and other objects*)
```

In[56]:=
```
? Arrow
```
Out[56]=

Symbol

Arrow[{$pt_1$, $pt_2$}] is a graphics primitive that represents an arrow from $pt_1$ to $pt_2$.

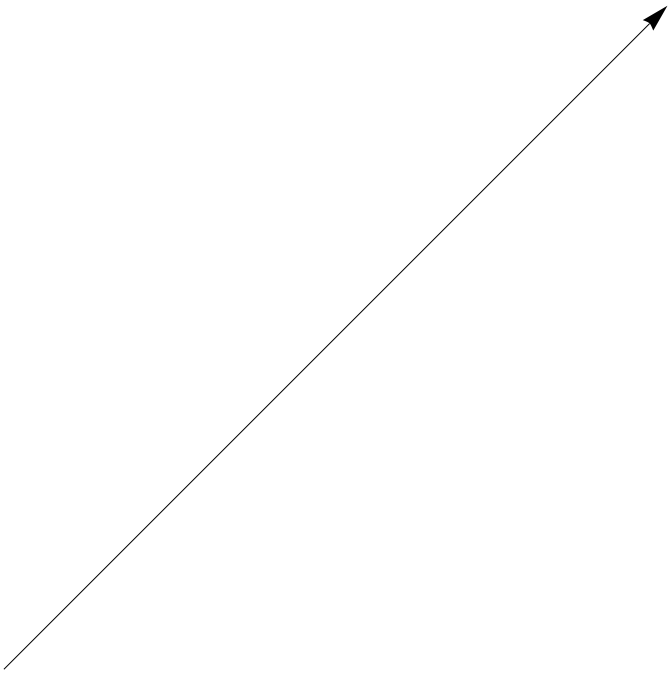
Arrow[{$pt_1$, $pt_2$}, $s$] represents an arrow with its ends set back from $pt_1$ and $pt_2$ by a distance $s$.

Arrow[{$pt_1$, $pt_2$}, {$s_1$, $s_2$}] sets back by $s_1$ from $pt_1$ and $s_2$ from $pt_2$.

Arrow[*curve*, …] represents an arrow following the specified *curve*.

In[57]:=
```
myarr[xx_] := Arrow[{{0, 0}, xx}](*always from origin*)
```

In[58]:=
```
Graphics[myarr[{1, 1}]]
```
Out[58]=



In[59]:=

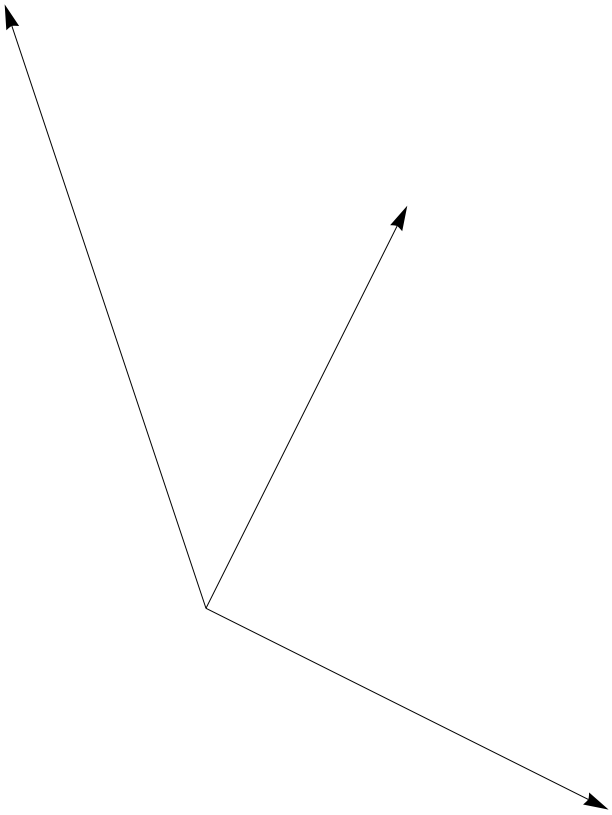
In[60]:=

In[61]:=
```
Clear[A, B, c];
```

In[62]:=
```
A = {1, 2}; B = {-1, 3}; c = A - B;
```

In[63]:=
```
Graphics[{myarr[A], myarr[B], myarr[c]}]
```

Out[63]=



In[64]:=
```
? Text
```

Out[64]=

Symbol

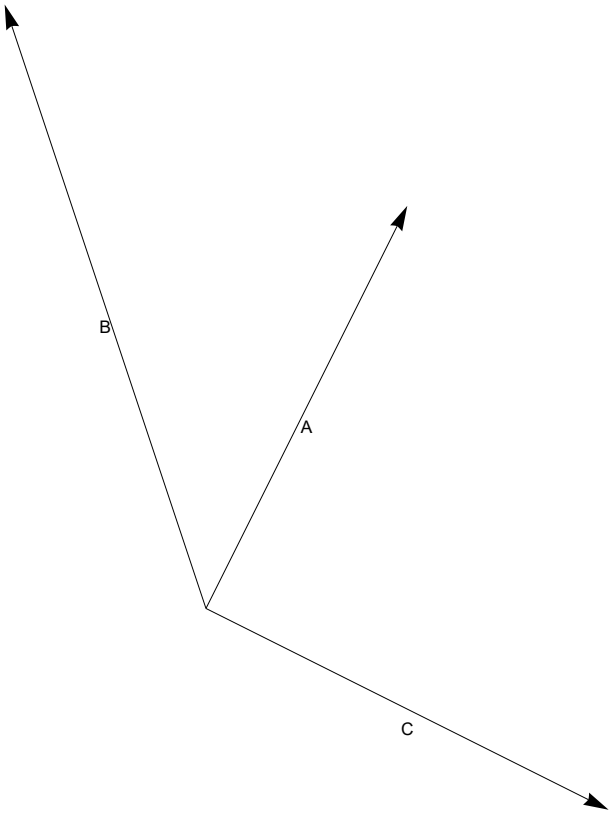
Text[*expr*] displays with *expr* in plain text format.

Text[*expr*, *coords*] is a graphics primitive that

displays the textual form of *expr* centered at the point specified by *coords*.

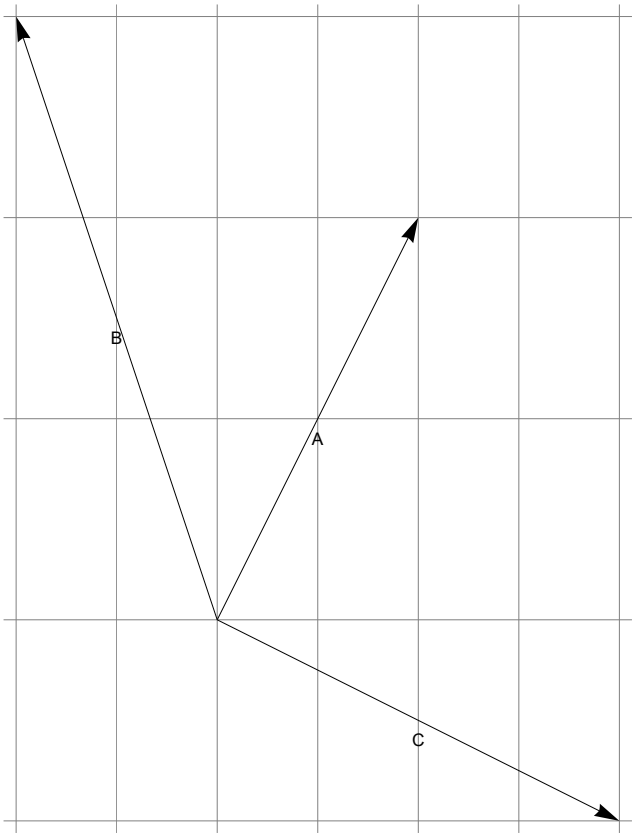
In[65]:=
```
Graphics[{myarr[A], myarr[B], myarr[c], Text["A", A / 2 - {0, .1}],
  Text["B", B / 2 - {0, .1}], Text["C", c / 2 - {0, .1}]}]
```

Out[65]=



In[66]:=
```
Show[%, GridLines → Automatic]
```

Out[66]=




In[67]:=

In[68]:=
```
(*Plotting vector fields*)

(*we consider the simplest graphical example
  of the gravitational field around a planet: g = -G*M /r^2,
with G - gravitational constant and M - the mass of the planet. For simplicity,
assue G*M=1, and we use the vector "field" = - r⃗ /r^3 .
    The same structure of field is observed for a negative
  electric charge*)

Clear[x, y]; vecr = {x, y}; r = Norm[vecr];
```

In[71]:=
```
field = -vecr / r^3
```

Out[71]=

$$\left\{-\frac{x}{\left(\text{Abs}[x]^2+\text{Abs}[y]^2\right)^{3/2}},\ -\frac{y}{\left(\text{Abs}[x]^2+\text{Abs}[y]^2\right)^{3/2}}\right\}$$

In[72]:=
```
FullSimplify[%, Assumptions → x > 0 && y > 0]
```

Out[72]=

$$\left\{-\frac{x}{\left(x^2+y^2\right)^{3/2}},\ -\frac{y}{\left(x^2+y^2\right)^{3/2}}\right\}$$

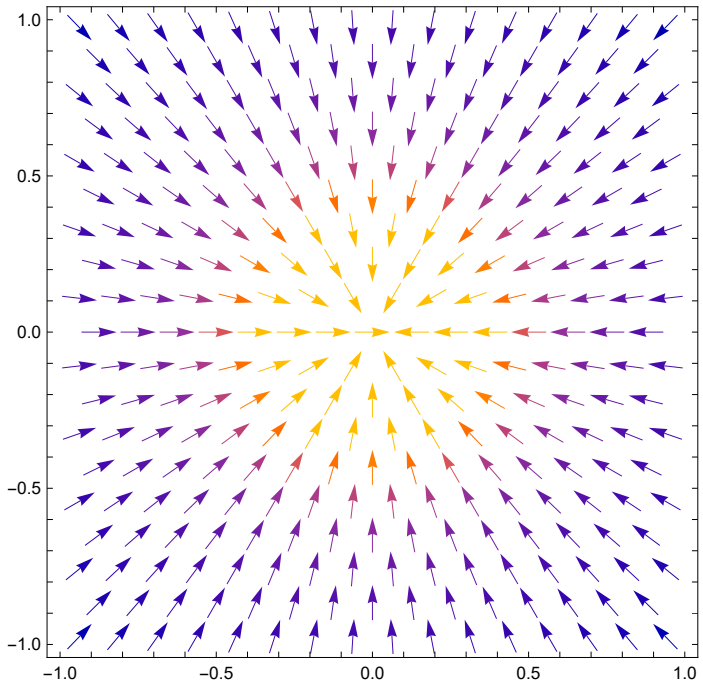
In[73]:=
```
field = %
```

Out[73]=

$$\left\{-\frac{x}{\left(x^2+y^2\right)^{3/2}}, -\frac{y}{\left(x^2+y^2\right)^{3/2}}\right\}$$

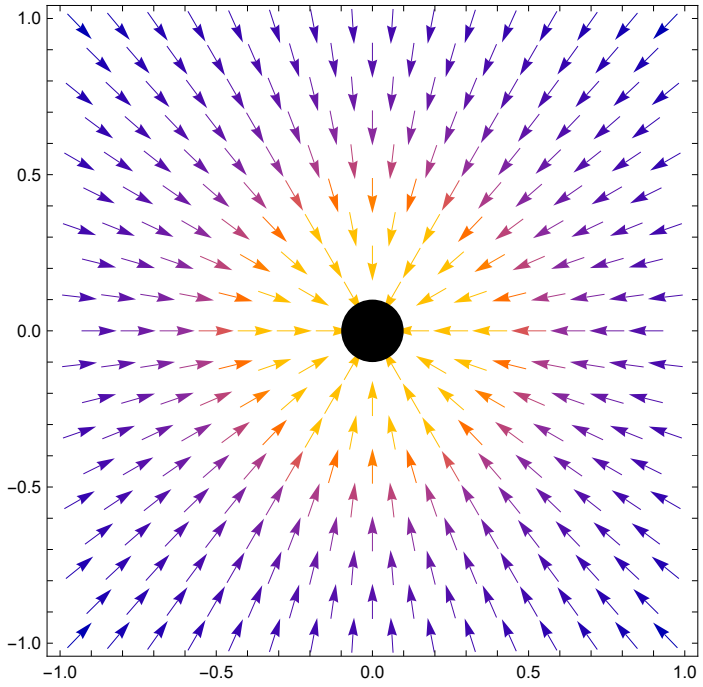
In[74]:=
```
VectorPlot[field, {x, -1, 1}, {y, -1, 1}]
```

Out[74]=



In[75]:=
```
Show[%, Graphics[Disk[{0, 0}, .1]]]
```

Out[75]=




In[76]:=
```
? ListStreamPlot
```

Out[76]=

Symbol

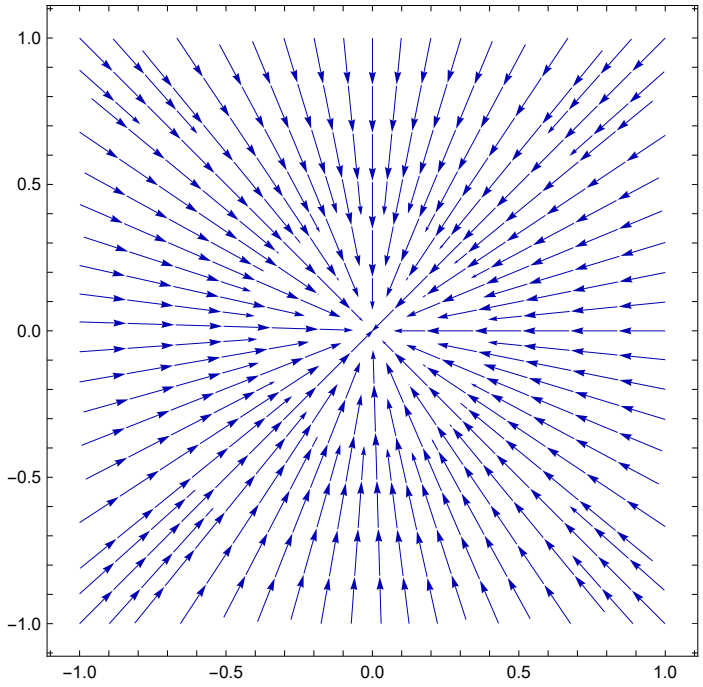
ListStreamPlot[*varr*] generates a stream plot from an array *varr* of vectors.

ListStreamPlot[{{{$x_1$, $y_1$}, {$vx_1$, $vy_1$}}, …}] generates a stream plot from vectors {$vx_i$, $vy_i$} given at points {$x_i$, $y_i$}.

ListStreamPlot[{$data_1$, $data_2$, …}] plots data for several vector fields.

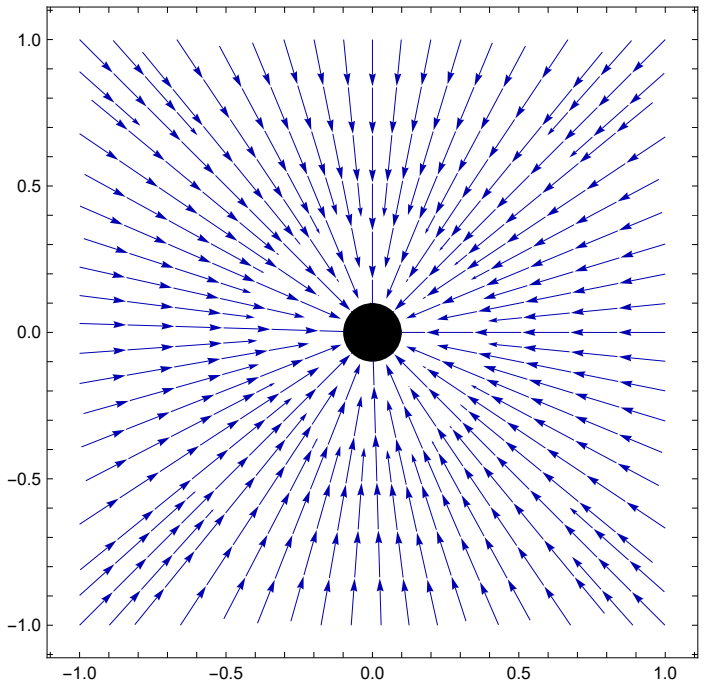
In[77]:=
```
StreamPlot[field, {x, -1, 1}, {y, -1, 1}]
```

Out[77]=




In[78]:=
```
Show[%, Graphics[Disk[{0, 0}, .1]]]
```

Out[78]=




In[79]:=

In[80]:=

```
(*Advanced: Mathematica modelling of Propagation of errors*)

myrand := 2 RandomReal[] - 1 (*random number between -1 and 1*)
```

In[105]:=
```
Table[myrand, {10}]
```

Out[105]=
```
{0.817509, -0.17352, -0.336672, 0.444658,
 -0.460805, -0.606929, 0.315531, -0.622823, 0.899948, 0.132814}
```

```
X := x + 0.5 myrand (*X is a random "measurement", scattered around the
    fixed value of x. The 0.5*myrand is "noise" (experimental error) *)
```

In[223]:=
```
x = 3; listX = Table[X, {50}] (*is the result of 50 measurements*)
```

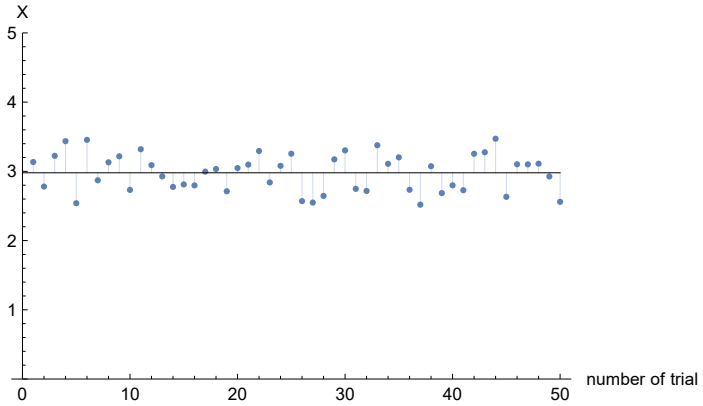
Out[223]=
```
{3.13497, 2.78048, 3.22463, 3.43589, 2.5394, 3.45472, 2.87116, 3.13051, 3.21753, 2.73164,
 3.31978, 3.08968, 2.92723, 2.77519, 2.81071, 2.79714, 2.99667, 3.03381, 2.71216, 3.04764,
 3.09765, 3.29402, 2.83988, 3.08005, 3.25484, 2.57037, 2.54881, 2.6455, 3.17365, 3.3035,
 2.74743, 2.71668, 3.37735, 3.10932, 3.20238, 2.73416, 2.51854, 3.07417, 2.68613, 2.79818,
 2.72743, 3.25413, 3.27587, 3.47177, 2.63199, 3.10277, 3.101, 3.11136, 2.92625, 2.56043}
```

In[225]:=
```
xmean = Mean[listX] (*close to 3 since 50 is a "big" number*)
```

Out[225]=
```
2.97933
```

In[226]:=
```
Show[ListPlot[listX, Filling → xmean, PlotRange → {0, 5}],
 Graphics[Line[{{0, xmean}, {50, xmean}}]], AxesLabel → {"number of trial", "X"}]
```

Out[226]=




```
(*iin the above vertical lines indicate the distance of individual measurement from
 the mean value of X. The average of those distances, given by MeanDeviation,
is "simple error". The average of squares of the distances is "variance",
and the root of variance is StandardDeviation*)
```

In[137]:=
```
errx = MeanDeviation[listX]
```
Out[137]=
```
0.241687
```

In[138]:=
```
sigx = StandardDeviation[listX]
```
Out[138]=
```
0.280813
```

In[229]:=
```
Y := y + 0.3 myrand (*another "experimental" variable with smaller average and noise*)
```

In[230]:=
```
y = 1; listY = Table[Y, {50}];
```

In[231]:=
```
erry = MeanDeviation[listY]
```
Out[231]=
```
0.143192
```

In[232]:=
```
sigy = StandardDeviation[listY]
```
Out[232]=
```
0.16595
```

In[233]:=
```
listPlus = listX + listY; (* we measure X+Y*)
```

In[234]:=
```
listMinus = listX - listY;(* we measure X-Y*)

Mean[listPlus](*close to sum of means*)
```
Out[235]=
```
3.944
```

```
Mean[listMinus](*close to the difference*)
```
Out[236]=
```
2.01466
```

In[237]:=

In[153]:=
```
errPlus = MeanDeviation[listPlus]
```
Out[153]=
```
0.255255
```

In[154]:=
```
errMinus = MeanDeviation[listMinus]
```
Out[154]=
```
0.286256
```

```
errx + erry (*simple errors do not add up!*)
```

Out[155]=

```
0.393381
```

In[250]:=

```
sigPlus = StandardDeviation[listPlus]
```

Out[250]=

```
0.278877
```

In[251]:=

In[239]:=

```
sigx
```

Out[239]=

```
0.280813
```

In[240]:=

```
sigy
```

Out[240]=

```
0.16595
```

```
Sqrt[sigx^2 + sigy^2](*close to sigPlus, but not very since 50 is not too "big"*)
```

Out[241]=

```
0.326183
```

```
(*we now try a larger number of trials*)
```

In[242]:=

```
X := x + .2 * myrand
```

In[243]:=

```
x = 5; listX = Table[X, {100 000}];
```

In[244]:=

```
MeanDeviation[listX]
```

Out[244]=

```
0.100037
```

In[245]:=

```
StandardDeviation[listX]
```

Out[245]=

```
0.115464
```

In[246]:=

```
Y := y + 0.3 myrand
```

In[247]:=

```
y = 3; listY = Table[Y, {100 000}];
```

In[252]:=

```
listZp = listX + listY;
```

In[253]:=

```
listZm = listX – listY;
```

In[254]:=
```
Mean[listZp]
```
Out[254]=
```
7.99915
```

In[255]:=
```
Mean[listZm]
```
Out[255]=
```
2.00034
```

In[256]:=
```
MeanDeviation[listZp]
```
Out[256]=
```
0.172016
```

In[265]:=
```
sigP = StandardDeviation[listZp]
```
Out[265]=
```
0.208115
```

In[266]:=
```
sigM = StandardDeviation[listZm](*very close !!*)
```
Out[266]=
```
0.208231
```

In[262]:=
```
sigX = StandardDeviation[listX]
```
Out[262]=
```
0.115464
```

In[263]:=
```
sigY = StandardDeviation[listY]
```
Out[263]=
```
0.173217
```

```
Sqrt[sigX^2 + sigY^2](*should be close to sigP and/or to sigM*)
```
Out[264]=
```
0.208173
```

Out[99]=
```
{0.20001, 0.299989, 0.360813, 0.381893}
```

In[100]:=

In[267]:=

Out[200]=
0.0300131

(* PRODUCT  X*Y , RATIO X/Y, etc.*)

In[270]:=
```
listZprod = listX * listY;
```

In[271]:=
```
Length[listZprod]
```
Out[271]=
100 000

In[272]:=
```
listZrat = listX / listY;
```

In[273]:=
```
sd := StandardDeviation
```

In[274]:=
```
{sd[listX] / Mean[listX], sd[listY] / Mean[listY],
 sd[listZprod] / Mean[listZprod], sd[listZrat] / Mean[listZrat]}
```
Out[274]=
{0.0230941, 0.0577503, 0.0621907, 0.0624101}

In[211]:=
```
Sqrt[(%[[1]])^2 + (%[[2]]^2)]
```
Out[211]=
0.00608818

In[275]:=
```
listZpow = listX^3 * listY^2;
```

In[276]:=
```
{3 sd[listX] / Mean[listX], 2 sd[listY] / Mean[listY], sd[listZpow] / Mean[listZpow]}
```
Out[276]=
{0.0692822, 0.115501, 0.134518}

In[277]:=
```
Sqrt[(%[[1]])^2 + (%[[2]]^2)]
```
Out[277]=
```
0.134686
```

```
(*General function of X*)
```

In[278]:=
```
listZsin = Sin[listX];
```

In[284]:=
```
{sd[listX], sd[listZsin]}
```
Out[284]=
```
{0.115464, 0.0330943}
```

In[285]:=
```
%[[1]] Cos[Mean[listX]]
```
Out[285]=
```
0.0327248
```

› System`