Assegnazioni e sostituzioni

```
In[1]:=
 x = Pi/4                              (* assegnazione *)
Out[1]=
 Pi
 --
 4
In[2]:=
 f = Sin
Out[2]=
 Sin
In[3]:=
 f[x]
Out[3]=
     1
 -------
 Sqrt[2]
In[4]:=
 x = Random[Integer,10]
Out[4]=
 6
In[5]:=
 X = {x,x,x,x,x,x,x,x,x}
Out[5]=
 {6, 6, 6, 6, 6, 6, 6, 6, 6}
In[6]:=
 y = x
Out[6]=
 6
```

```
In[7]:=
 z := x    (* assegnazione non valutata *)
In[8]:=
 z
Out[8]=
 6
In[9]:=
 x = {x,x}
Out[9]=
 {6, 6}
In[10]:=
 {y,z}
Out[10]=
 {6, {6, 6}}
In[11]:=
 ? y
 Global`y
 y = 6
In[12]:=
 ? z
 Global`z
 z := x
In[13]:=
 x := Random[Integer,10]
In[14]:=
 Y := {{x,x,x},{y,y,y},{z,z,z}}
In[15]:=
 Y
Out[15]=
 {{9, 7, 4}, {6, 6, 6}, {5, 2, 4}}
```

```
In[16]:=
 Y
Out[16]=
 {{0, 5, 7}, {6, 6, 6}, {4, 1, 1}}
In[17]:=
 Z = Y
Out[17]=
 {{3, 9, 10}, {6, 6, 6}, {4, 5, 3}}
In[18]:=
 Z
Out[18]=
 {{3, 9, 10}, {6, 6, 6}, {4, 5, 3}}
In[19]:=
 ?? =
 lhs = rhs evaluates rhs and assigns the
    result to be the value of lhs. From
    then on, lhs is replaced by rhs
    whenever it appears. {l1, l2, ... } =
    {r1, r2, ... } evaluates the ri, and
    assigns the results to be the values
    of the corresponding li.
 Attributes[Set] =
   {HoldFirst, Protected, SequenceHold}
```

In[20]:=

```
?? :=
```

```
lhs := rhs assigns rhs to be the delayed
   value of lhs. rhs is maintained in an
   unevaluated form. When lhs appears,
   it is replaced by rhs, evaluated
   afresh each time.
Attributes[SetDelayed] =
  {HoldAll, Protected, SequenceHold}
```

In[21]:=

```
Clear[f,x,y,z,X,Y,Z]
```

In[22]:=

```
a x^2 + b x + c
```

Out[22]=

$c + b\,x + a\,x^2$

In[23]:=

```
% /. x -> y^2          (* sostituzione *)
```

Out[23]=

$c + b\,y^2 + a\,y^4$

In[24]:=

```
% /. {a -> 1,b -> 2,c -> 3}
```

Out[24]=

$3 + 2\,y^2 + y^4$

In[25]:=

```
X = {x,{x,x},{{x,x},{x,x}}}
```

Out[25]=

```
{x, {x, x}, {{x, x}, {x, x}}}
```

```
In[26]:=
 X /. {x,x} -> x
Out[26]=
 {x, x, {x, x}}
In[27]:=
 X /. {{x,x} -> x,x -> y}
Out[27]=
 {y, x, {x, x}}
In[28]:=
 % /. List -> Plus
Out[28]=
 3 x + y
In[29]:=
 X //. {x,x} -> x     (* sost. ripetuta *)
Out[29]=
 {x, x, x}
In[30]:=
 X //. {{x,x} -> x,x -> y}
Out[30]=
 {y, y, y}
In[31]:=
 f[f[f[1]]] + f[1 + f[1 + f[1 + f[1]]]]
Out[31]=
 f[f[f[1]]] + f[1 + f[1 + f[1 + f[1]]]]
In[32]:=
 % /. f[1] -> 1
Out[32]=
 f[f[1]] + f[1 + f[1 + f[2]]]
```

In[33]:=
```
%% //. f[1] -> 1
```
Out[33]=
```
1 + f[1 + f[1 + f[2]]]
```
In[34]:=
```
%%% //. {f[1] -> 1,f[2] -> 0}
```
Out[34]=
```
1
```
In[35]:=
```
X /. x -> Random[Integer,10]
```
Out[35]=
```
{4, {4, 4}, {{4, 4}, {4, 4}}}
```
In[36]:=
```
X /. x :> Random[Integer,10]
```
Out[36]=
```
{4, {5, 10}, {{8, 8}, {9, 0}}}
```
In[37]:=
```
? ->
```
```
lhs -> rhs represents a rule that
   transforms lhs to rhs.
```
In[38]:=
```
? :>
```
```
lhs :> rhs represents a rule that
   transforms lhs to rhs, evaluating rhs
   only when the rule is used.
```

Equazioni e sistemi

In[39]:=

```
x == 2x + 1                    (* equazione *)
```

Out[39]=

```
x == 1 + 2 x
```

In[40]:=

```
x == x + 1
```

Out[40]=

```
x == 1 + x
```

In[41]:=

```
x == x
```

Out[41]=

```
True
```

In[42]:=

```
{x,1} == x
```

Out[42]=

```
{x, 1} == x
```

In[43]:=

```
{x,1} == {x,2}
```

Out[43]=

```
False
```

In[44]:=

```
Sin[Pi] == 0
```

Out[44]=

```
True
```

In[45]:=

```
Sin[x] == 0
```

Out[45]=

```
Sin[x] == 0
```

```
In[46]:=
 Sin[x]^2 + Cos[x]^2 == 1
Out[46]=
          2         2
 Cos[x]  + Sin[x]  == 1
In[47]:=
 Simplify[%]
Out[47]=
 True
In[48]:=
 x === 2x + 1                  (* identità *)
Out[48]=
 False
In[49]:=
 x === x + 1
Out[49]=
 False
In[50]:=
 x === x
Out[50]=
 True
In[51]:=
 Sin[Pi] === 0
Out[51]=
 True
In[52]:=
 Sin[x] === 0
Out[52]=
 False
```

In[53]:=

```
?? ==
lhs == rhs returns True if lhs and rhs
   are identical.
Attributes[Equal] = {Protected}
```

In[54]:=

```
?? ===
lhs === rhs yields True if the
   expression lhs is identical to rhs,
   and yields False otherwise.
Attributes[SameQ] = {Protected}
```

In[55]:=

```
Solve[a x^2 + b x + c == 0,x]
```

Out[55]=

```
                         2
{{x -> -b - Sqrt[b  - 4 a c]},
       ---------------------
               2 a

                          2
  {x -> -b + Sqrt[b  - 4 a c]}}
        ---------------------
                2 a
```

In[56]:=

```
sol1 = %[[1]]
```

Out[56]=

```
                        2
{x -> -b - Sqrt[b  - 4 a c]}
      ---------------------
              2 a
```

In[57]:=

```
sol2 = %%[[2]]
```

Out[57]=

$$\left\{x \to \frac{-b + \text{Sqrt}[b^2 - 4\ a\ c]}{2\ a}\right\}$$

In[58]:=

```
a x^2 + b x + c == 0 /. sol1
```

Out[58]=

$$c + \frac{b\ (-b - \text{Sqrt}[b^2 - 4\ a\ c])}{2\ a} + \frac{(-b - \text{Sqrt}[b^2 - 4\ a\ c])^2}{4\ a} == 0$$

In[59]:=

```
Simplify[%]
```

Out[59]=

True

In[60]:=

```
x1 = x /. sol1
```

Out[60]=

$$\frac{-b - \text{Sqrt}[b^2 - 4\ a\ c]}{2\ a}$$

```
In[61]:=
 Reduce[a x^2 + b x + c == 0,x]
Out[61]=
                              2
                   -b - Sqrt[b  - 4 a c]
 a != 0 && x == --------------------- ||
                         2 a

   a != 0 && x ==

                 2
      -b + Sqrt[b  - 4 a c]
      --------------------- ||
              2 a
   c == 0 && b == 0 && a == 0 ||

                     c
   b != 0 && x == -(-) && a == 0
                     b
In[62]:=
 Solve[Sin[x] == 1/2,x]
 Solve::ifun:
    Inverse functions are being used by
     Solve, so some solutions may not be
      found.
Out[62]=
          Pi
 {{x -> --}}
          6
```

In[63]:=

```
Solve[E^x == 2,x]
```

```
Solve::ifun:
   Inverse functions are being used by
    Solve, so some solutions may not be
     found.
```

Out[63]=

```
{{x -> Log[2]}}
```

In[64]:=

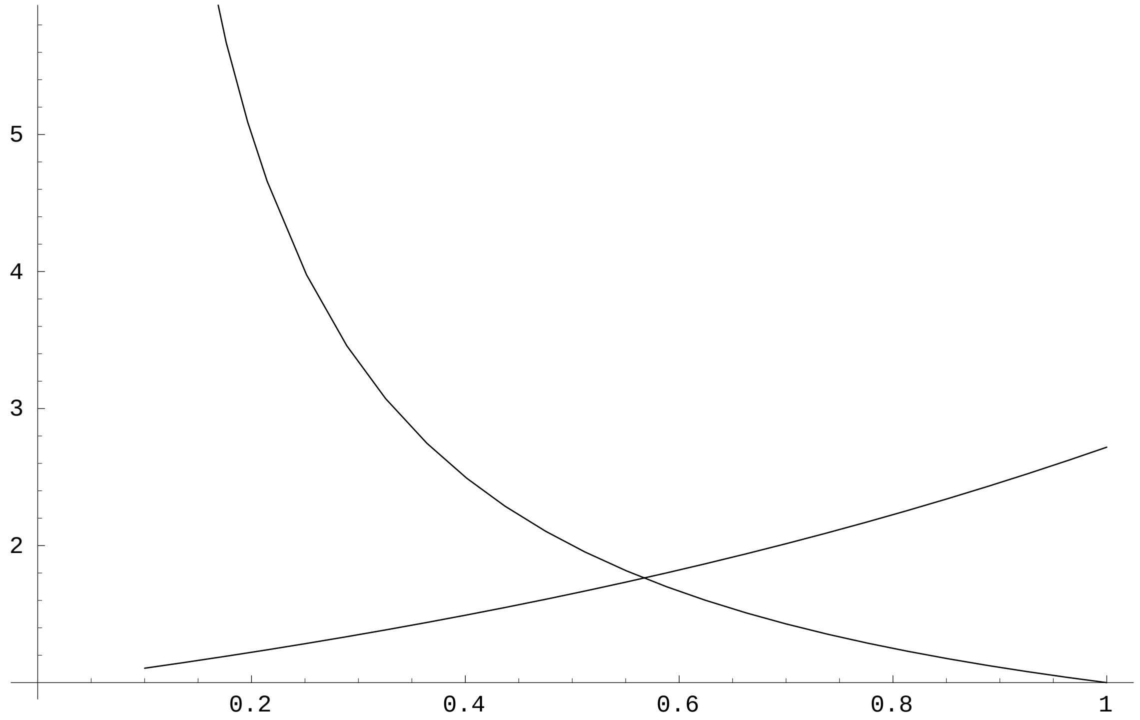
```
FindRoot[E^x == 1/x,{x,1}]
```

Out[64]=

```
{x -> 0.567143}
```

In[65]:=

```
Plot[{E^x,1/x},{x,0.1,1}]
```



Out[65]=

```
-Graphics-
```

In[66]:=

```
Solve[{a1 x + b1 y == c1,
       a2 x + b2 y == c2},{x,y}]
```

Out[66]=

```
          -(b2 c1) + b1 c2
{{x -> -(----------------),
          -(a2 b1) + a1 b2

             a2 c1 - a1 c2
    y -> -(----------------)}}
           -(a2 b1) + a1 b2
```

In[67]:=

```
Solve[{Log[x] + Log[y] == 0,
       x + y == 5/2},{x,y}]
```

Out[67]=

```
       1                               1
{{x -> -, y -> 2}, {x -> 2, y -> -}}
       2                               2
```

In[68]:=

```
Solve[2 x + 1 > 0,x]
Solve::eqf:
   1 + 2 x > 0 is not a well-formed
     equation.
```

Out[68]=

```
Solve[1 + 2 x > 0, x]
```

Espressioni analitiche

In[69]:=

```
Limit[Sin[a x]/x,x -> 0]
```

Out[69]=

```
a
```

In[70]:=

```
Limit[x E^(-1/x),x -> 0,Direction -> +1]
```

Out[70]=

```
-∞
```

In[71]:=

```
Limit[x E^(-1/x),x -> 0,Direction -> -1]
```

Out[71]=

```
0
```

In[72]:=

```
Limit[Sin[1/x],x -> 0]
```

Out[72]=

```
Interval[{-1, 1}]
```

In[73]:=

```
Limit[ArcTan[x],x -> +Infinity]
```

Out[73]=

```
Pi
--
2
```

In[74]:=

```
D[ArcTan[x],x]
```

Out[74]=

```
  1
------
     2
1 + x
```

```
In[75]:=
 D[ArcTan[x],x,x]
Out[75]=
   -2 x
 ---------
       2 2
 (1 + x )
In[76]:=
 D[ArcTan[x],{x,4}] // Simplify

Out[76]=
                 2
 -24 x (-1 + x )
 ---------------
          2 4
    (1 + x )
In[77]:=
 D[Sin[x] f[x,g[y]],x,y,y,y] // Simplify
Out[77]=
  (3)               (0,1)
 g   [y] (Cos[x] f     [x, g[y]] +

                 (1,1)
      Sin[x] f      [x, g[y]]) +
   g'[y] (3 g''[y]
                  (0,2)
       (Cos[x] f      [x, g[y]] +
                   (1,2)
          Sin[x] f      [x, g[y]]) +
            2             (0,3)
      g'[y]  (Cos[x] f      [x, g[y]] +
                   (1,3)
          Sin[x] f      [x, g[y]]))
```

In[78]:=

```
Series[Sin[x],{x,0,10}]
```

Out[78]=

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880} + O[x]^{11}$$

In[79]:=

```
% + O[x]^8
```

Out[79]=

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + O[x]^{8}$$

In[80]:=

```
Normal[%]
```

Out[80]=

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040}$$

```
In[81]:=
Table[Series[Sin[x],{x,0,i}] // Normal,
      {i,1,10,2}] // TableForm
Out[81]//TableForm=
```

$$x$$

$$x - \frac{x^3}{6}$$

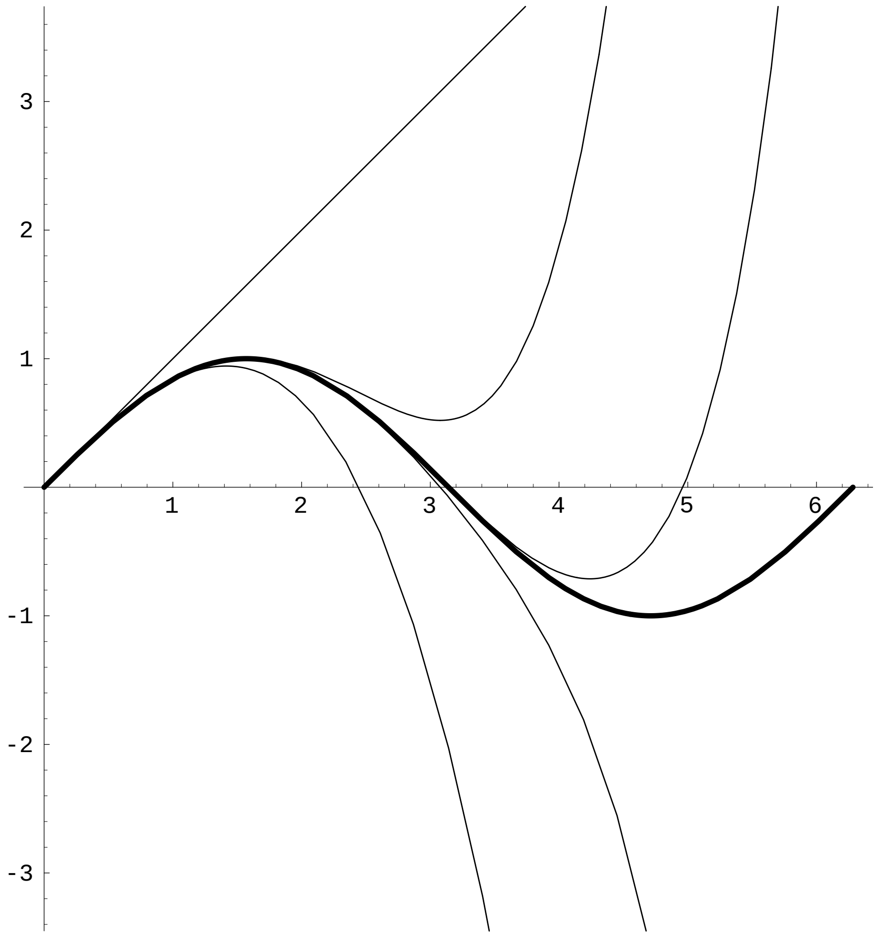
$$x - \frac{x^3}{6} + \frac{x^5}{120}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040}$$

$$x - \frac{x^3}{6} + \frac{x^5}{120} - \frac{x^7}{5040} + \frac{x^9}{362880}$$

In[82]:=

```
Plot[Append[%,Sin[x]] // Evaluate,
     {x,0,2 Pi},AspectRatio -> Automatic,
     PlotStyle -> {{},{},{},{},{},
                          AbsoluteThickness[2]}]
```



Out[82]=

-Graphics-

In[83]:=

```
Integrate[Sqrt[1 - x^2],x]
```

Out[83]=

$$\frac{x\ \mathrm{Sqrt}[1 - x^2]}{2} + \frac{\mathrm{ArcSin}[x]}{2}$$

```
In[84]:=
 Integrate[Sqrt[1 - x^2],{x,-1,1}]
Out[84]=
 Pi
 --
 2
In[85]:=
 Integrate[1/Sqrt[x],{x,0,1}]
Out[85]=
 2
In[86]:=
 Integrate[1/Sqrt[x],{x,1,Infinity}]
 Integrate::idiv:
                         1
    Integral of -------
                    Sqrt[x]
      does not converge on {1, ∞}.
Out[86]=
                  1
 Integrate[-------, {x, 1, ∞}]
              Sqrt[x]
In[87]:=
 Integrate[1/x^2,{x,1,Infinity}]
Out[87]=
 1
In[88]:=
 Integrate[Sin[Sin[x]],x]
Out[88]=
 Integrate[Sin[Sin[x]], x]
```

In[89]:=

```
Integrate[Sin[Sin[x]],{x,0,Pi}]
```

Out[89]=

$$2\ \text{HypergeometricPFQ}\left[\{1\},\ \left\{\frac{3}{2},\ \frac{3}{2}\right\},\ -\left(\frac{1}{4}\right)\right]$$

In[90]:=

```
N[%]
```

Out[90]=

```
1.78649
```

In[91]:=

```
Integrate[Sin[Sin[Sin[x]]],{x,0,Pi}]
```

Out[91]=

```
Integrate[Sin[Sin[Sin[x]]], {x, 0, Pi}]
```

In[92]:=

```
NIntegrate[Sin[Sin[Sin[x]]],{x,0,Pi}]
```

Out[92]=

```
1.64259
```

Il lancio più lungo

In[93]:=

```
DSolve[{x''[t] == a,
        x[0] == x0,
        x'[0] == v0},x[t],t]
```

Out[93]=

$$\left\{\left\{x[t] \to \frac{a\ t^2}{2} + t\ v0 + x0\right\}\right\}$$

In[94]:=

```
x[t] /. %[[1]]
```

Out[94]=

$$\frac{a\ t^2}{2} + t\ v0 + x0$$

In[95]:=

```
p = % /. {a -> {0,-g},
          x0 -> {0,0},
          v0 -> v {Cos[c],Sin[c]}}
```

Out[95]=

$$\left\{t\ v\ \mathrm{Cos}[c],\ \frac{-(g\ t^2)}{2} + t\ v\ \mathrm{Sin}[c]\right\}$$

In[96]:=

```
Solve[p[[2]] == 0,t]
```

Out[96]=

$$\left\{\{t \to 0\},\ \left\{t \to \frac{2\ v\ \mathrm{Sin}[c]}{g}\right\}\right\}$$

In[97]:=

```
T = t /. %[[2]]
```

Out[97]=

$$\frac{2\ v\ Sin[c]}{g}$$

In[98]:=

```
p /. t -> T
```

Out[98]=

$$\left\{\frac{2\ v^2\ Cos[c]\ Sin[c]}{g},\ 0\right\}$$

In[99]:=

```
d = Simplify[%[[1]]]
```

Out[99]=

$$\frac{v^2\ Sin[2\ c]}{g}$$

In[100]:=

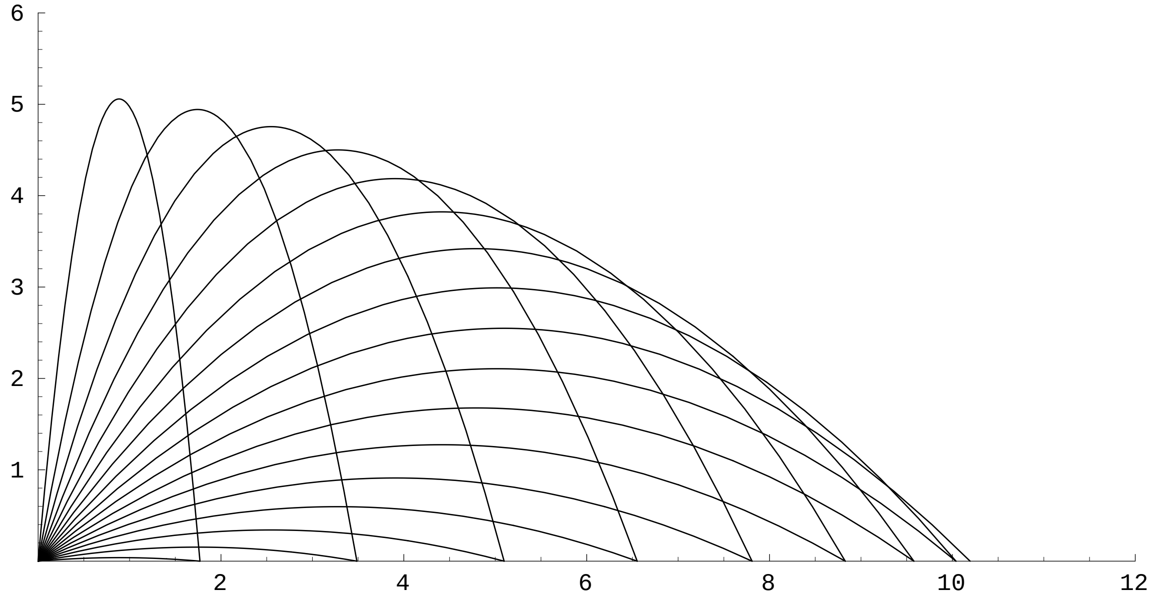
```
Table[
  Block[{g = 9.81,v = 10,c = n Degree},
    ParametricPlot[p//Evaluate,{t,0,T},
      AspectRatio->Automatic,
      PlotRange->{{0,12},{0,6}},
      DisplayFunction -> Identity]],
  {n,5,85,5}]
```

Out[100]=

```
{-Graphics-, -Graphics-, -Graphics-,
  -Graphics-, -Graphics-, -Graphics-,
  -Graphics-, -Graphics-, -Graphics-,
  -Graphics-, -Graphics-, -Graphics-,
  -Graphics-, -Graphics-, -Graphics-,
  -Graphics-, -Graphics-}
```

In[101]:=

```
Show[%,DisplayFunction ->
        $DisplayFunction]
```



Out[101]=

-Graphics-

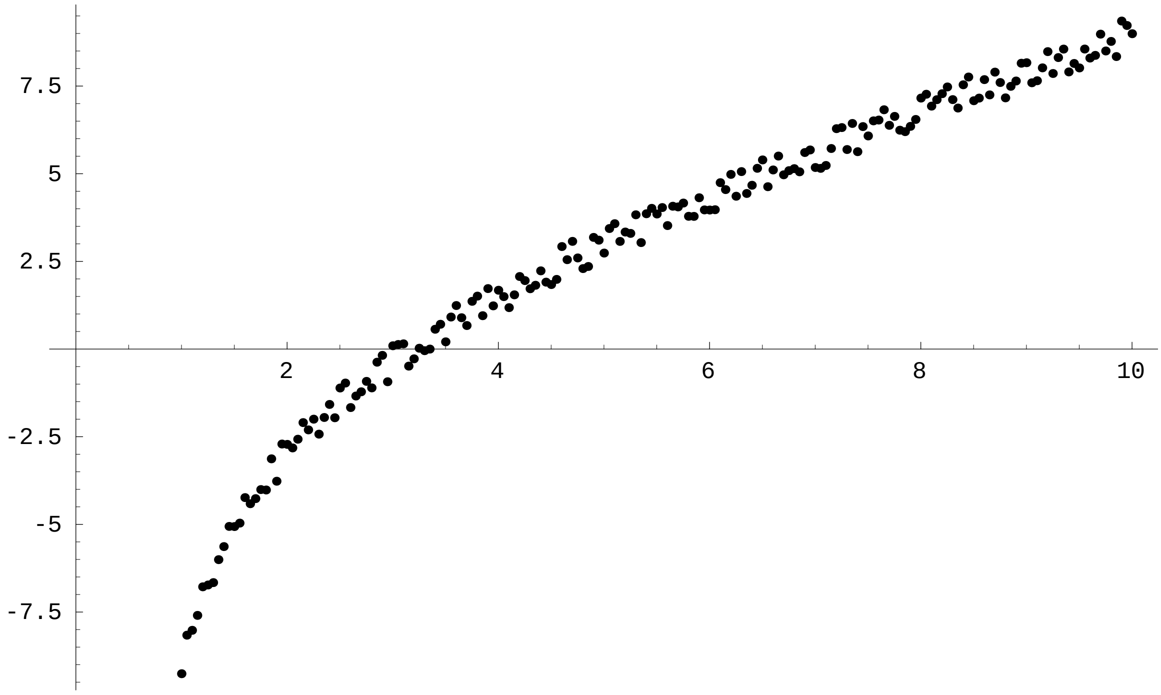
Analisi di dati numerici

In[102]:=

```
dati =
  Table[{x,x - 10/x
            + Random[Real,{-.5,.5}]},
        {x,1,10,.05}];
```

In[103]:=

```
punti = ListPlot[dati]
```



Out[103]=

-Graphics-

In[104]:=

```
? Fit
```

Fit[data, funs, vars] finds a least-squares fit to a list of data as a linear combination of the functions funs of variables vars. The data can have the form {{x1, y1, ... , f1}, {x2, y2, ... , f2}, ... }, where the number of coordinates x, y, ... is equal to the number of variables in the list vars. The data can also be of the form {f1, f2, ... }, with a single coordinate assumed to take values 1, 2, ... The argument funs can be any list of functions that depend only on the objects vars.

In[105]:=

```
f1 = Fit[dati,{1},x]          (* media *)
```

Out[105]=

2.92914

In[106]:=

```
f2 = Fit[dati,{1,x},x]  (* regressione *)
```

Out[106]=

-5.89686 + 1.60473 x

In[107]:=

```
f3 = Fit[dati,{1/x,1,x},x]
```

Out[107]=

$$-0.0402012 - \frac{9.87817}{x} + 1.00234\, x$$

In[108]:=

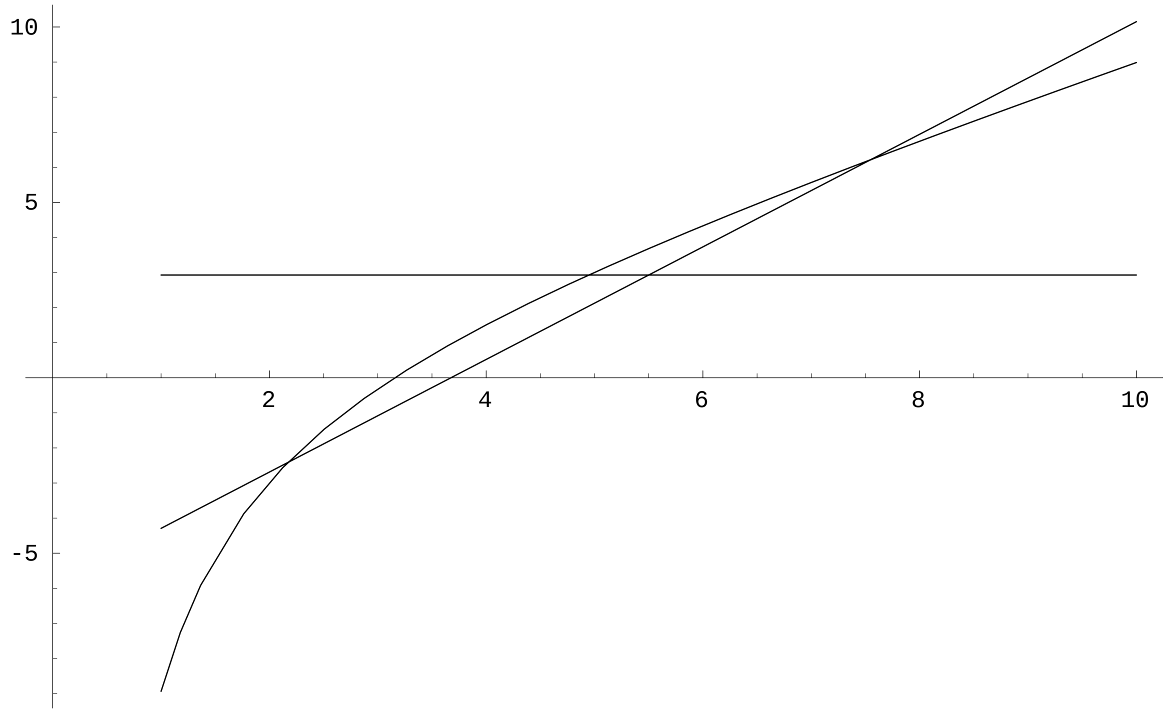
```
f4 = Fit[dati,{1/x,x},x]
```

Out[108]=

$$\frac{-9.93643}{x} + 0.997949\ x$$

In[109]:=

```
Plot[{f1,f2,f4},{x,1,10}]
```

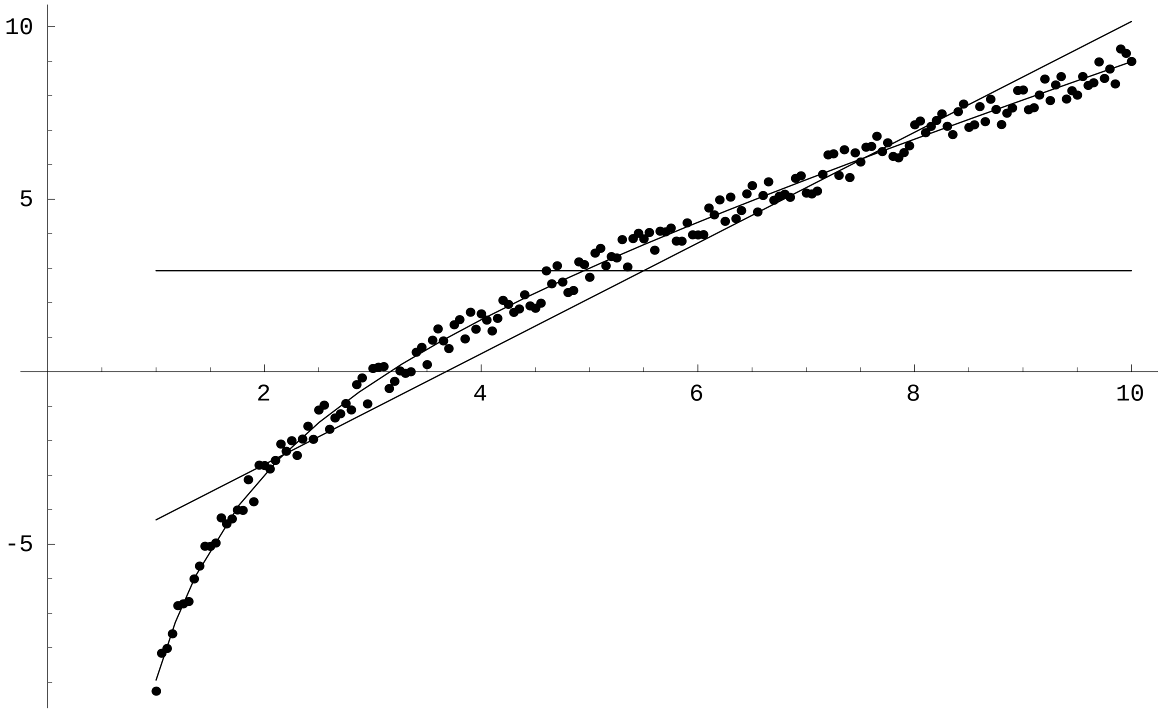


Out[109]=

-Graphics-

In[110]:=

```
Show[%,punti]
```



Out[110]=

-Graphics-

Valutazione delle espressioni

In[111]:=

```
Trace[2 + 3,TraceOriginal -> True]
```

Out[111]=

```
{2 + 3, {Plus}, {2}, {3}, 2 + 3, 5}
```

In[112]:=

```
Trace[f[x,y],TraceOriginal -> True]
```

Out[112]=

```
{f[x, y], {f}, {x}, {y}, f[x, y]}
```

In[113]:=

```
Trace[f[x,g[y]],TraceOriginal -> True]
```

Out[113]=

```
{f[x, g[y]], {f}, {x},
  {g[y], {g}, {y}, g[y]}, f[x, g[y]]}
```

In[114]:=

```
TracePrint[f[x,g[y]]]
```

```
 f[x, g[y]]

  f

  x

  g[y]

   g

   y
```

Out[114]=

```
f[x, g[y]]
```

In[115]:=

```
f[x,y] := g[x]
g[x] := h
```

In[117]:=

```
TracePrint[f[x,y]]
```

```
 f[x, y]

  f

  x

  y

 g[x]

  g

  x

 h
```

Out[117]=

```
h
```

In[118]:=

```
f[x] := x
```

In[119]:=

```
TracePrint[f[f[f[x]]]]
```

```
 f[f[f[x]]]
  f
  f[f[x]]
   f
   f[x]
    f
    x
   x
  f[x]
  x
 f[x]
 x
```

Out[119]=

x

```
In[120]:=
 x = 10
Out[120]=
 10
In[121]:=
 Trace[y = x, TraceOriginal -> True]
Out[121]=
 {y = x, {Set}, {x, 10}, y = 10, 10}
In[122]:=
 Trace[y := x, TraceOriginal -> True]
Out[122]=
 {y := x, {SetDelayed}, y := x, Null}
In[123]:=
 Trace[x := Evaluate[y],
              TraceOriginal -> True]
Out[123]=
 {x := Evaluate[y], {SetDelayed},

   {y, x, 10}, x := 10, Null}
In[124]:=
 ?? Evaluate
 Evaluate[expr] causes expr to be
    evaluated even if it appears as the
    argument of a function whose
    attributes specify that it should be
    held unevaluated.
 Attributes[Evaluate] = {Protected}
```

In[125]:=

```
Trace[Table[Random[],{n,3}]]
```

Out[125]=

```
{Table[Random[], {n, 3}],
  {Random[], 0.734472},
  {Random[], 0.596345},
  {Random[], 0.964057},
  {0.734472, 0.596345, 0.964057}}
```

In[126]:=

```
Trace[Table[Evaluate[Random[]],{n,3}]]
```

Out[126]=

```
{{Random[], 0.716486},
  Table[0.716486, {n, 3}],
  {0.716486, 0.716486, 0.716486}}
```

In[127]:=

```
M = Array[m,{3,3}]
```

Out[127]=

```
{{m[1, 1], m[1, 2], m[1, 3]},
  {m[2, 1], m[2, 2], m[2, 3]},
  {m[3, 1], m[3, 2], m[3, 3]}}
```

In[128]:=

```
Table[M[[i,i]],{i,3}]
```

Out[128]=

```
{m[1, 1], m[2, 2], m[3, 3]}
```

In[129]:=

```
Table[M[[i,i]]//Evaluate,{i,3}]
```

```
Part::pspec:
   Part specification i
     is neither an integer nor a list of
     integers.
```

Out[129]=

```
{m[1, 1], m[2, 2], m[3, 3]}
```