Practica nº 5: Aproximación

Ejemplo 1

Uso del comando interno de *Mathematica*

En esta práctica estudiaremos cómo aproximar mediante polinomios un conjunto de datos aplicando aproximación minimo cuadrática (caso discreto). *Mathematica* posee un comando muy flexible y versátil,

Fit[*nube,base,var***]**,

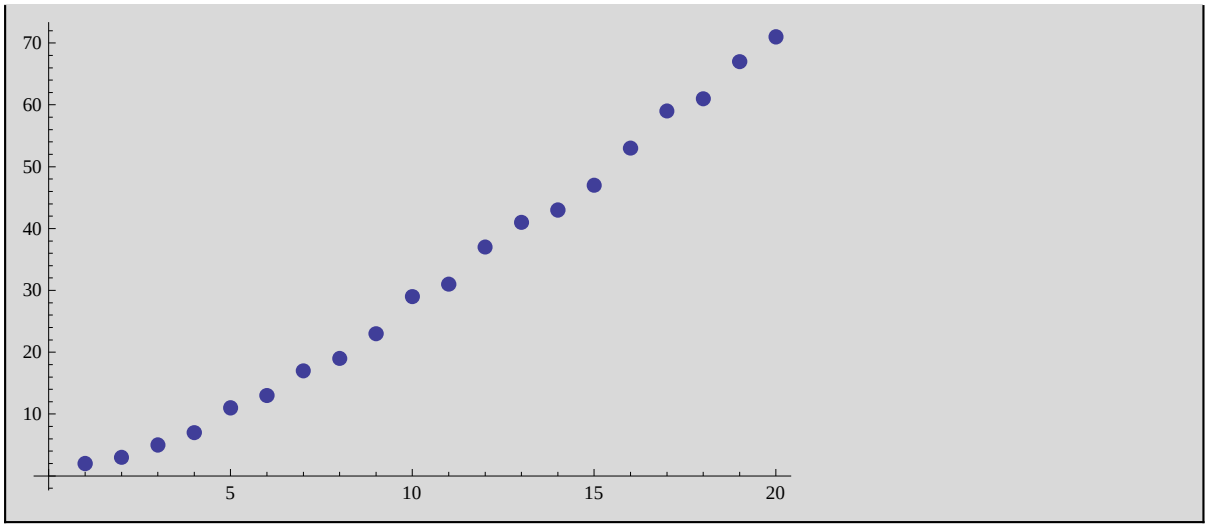
que permite construir el polinomio correspondiente:

a) Construir una tabla con los primeros 20 números primos y dibujar los puntos de esta tabla

```
a = Table[Prime[x], {x, 20}]
```

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71}
```

```
g1 = ListPlot[a, PlotStyle → PointSize[0.02]]
```



b) Obtener una aproximación lineal a los puntos de la tabla y dibujar la función de aproximación

c) Dibujar simultaneamente los puntos de la tabla y la función de aproximación

d) Obtener una aproximación mediante un polinomio de grado 2 de los puntos de la tabla

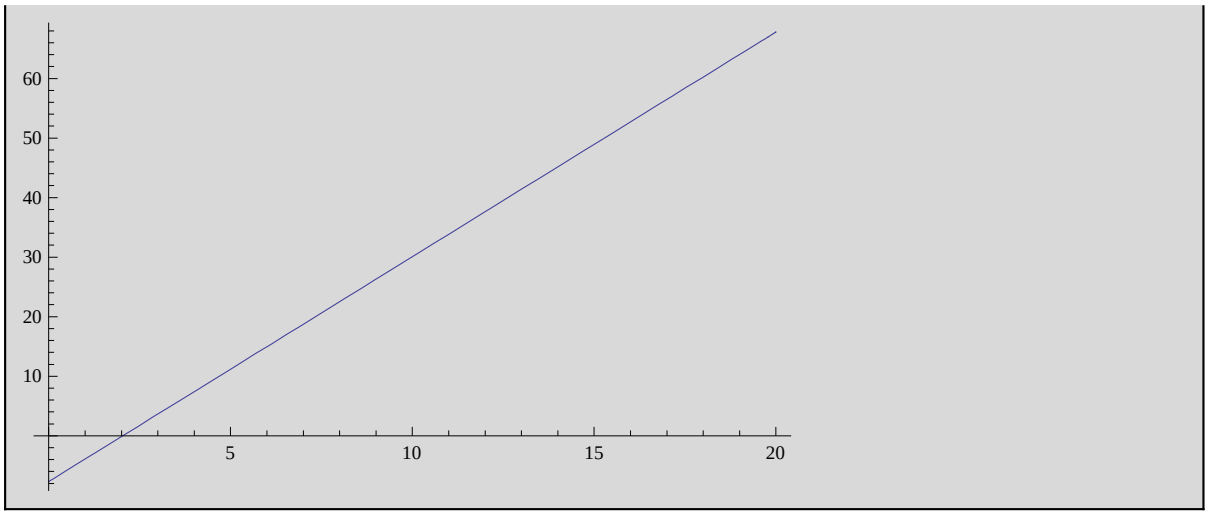
e) Dibujar simultaneamente los puntos de la tabla y la función de aproximación

Nota: Utilizar el comando Fit

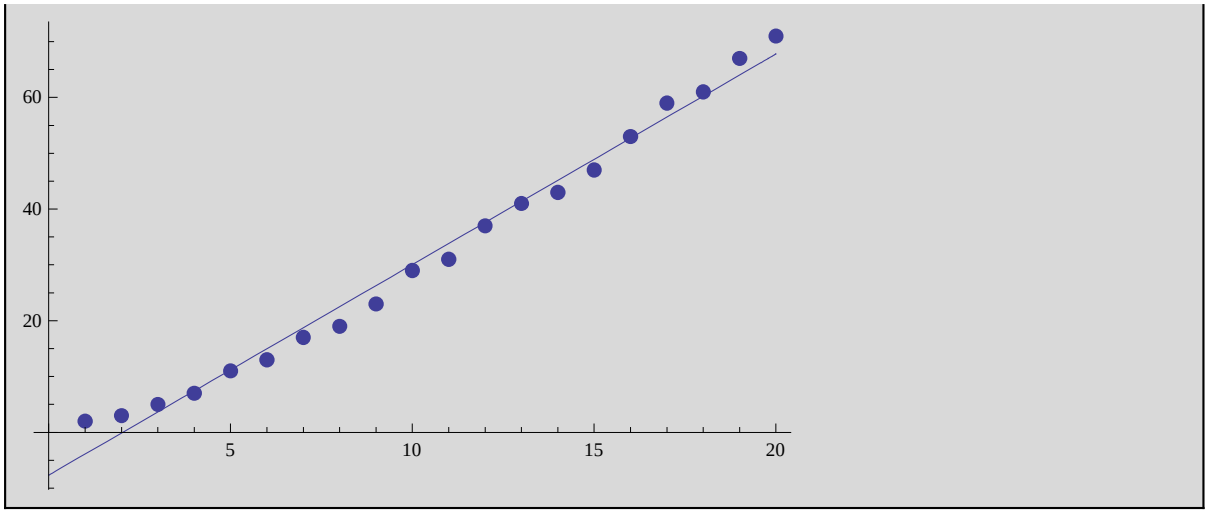
```
p1[x_] = Fit[a, {1, x}, x]
```

```
-7.67368 + 3.77368 x
```

```
g2 = Plot[p1[x], {x, 0, 20}]
```




```
Show[g1, g2]
```



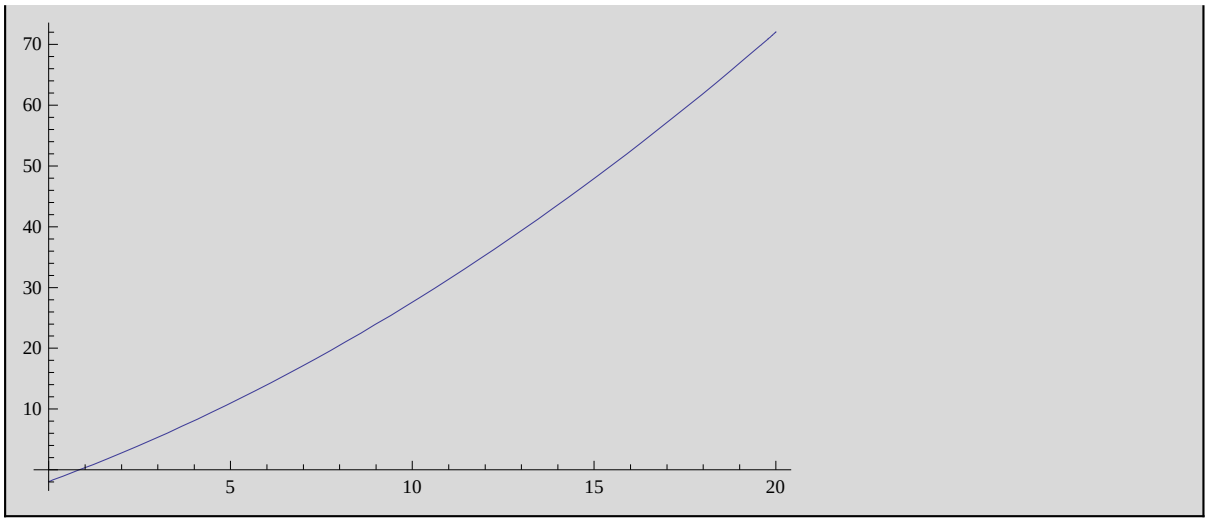
Plinomio de grado 2

d) Obtener una aproximación mediante un polinomio de grado 2 de los puntos de la tabla
e) Dibujar simultaneamente los puntos de la tabla y la función de aproximación

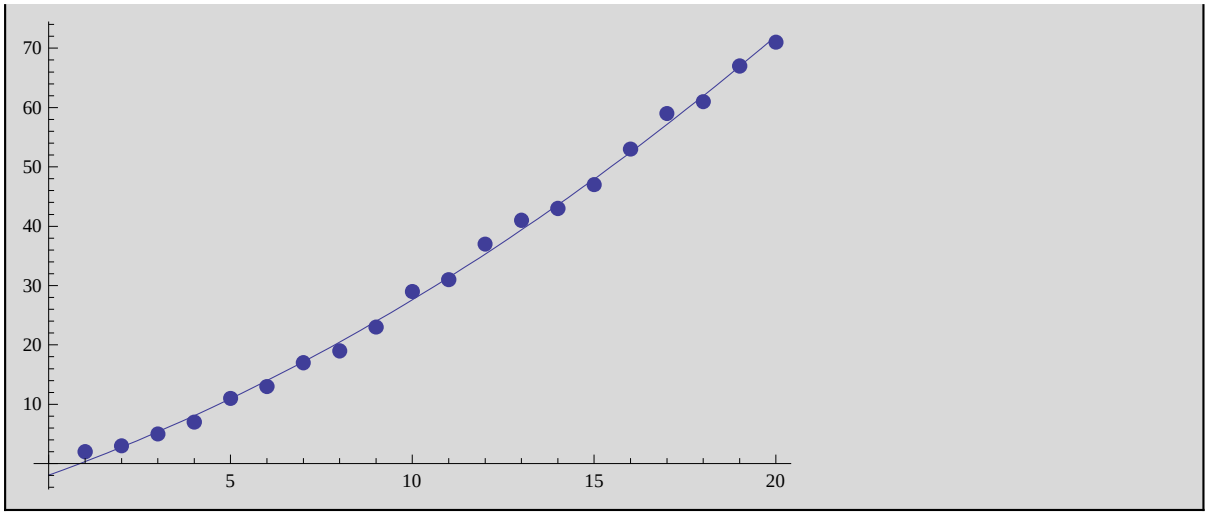
```
p2[x_] = Fit[a, {1, x, x^2}, x]
```

$-1.92368 + 2.2055\,x + 0.0746753\,x^2$

```
g3 = Plot[p2[x], {x, 0, 20}]
```




```
Show[g1, g3]
```



Problema 1

Resolver el problema anterior mediante las ecuaciones normales del ajuste mínimo cuadrático

a) Construir una tabla con los primeros 20 números primos y dibujar los puntos de esta tabla

b) Obtener una aproximación lineal a los puntos de la tabla y dibujar la función de aproximación

c) Dibujar simultaneamente los puntos de la tabla y la función de aproximación

d) Obtener una aproximación mediante un polinomio de grado 2 de los puntos de la tabla

e) Dibujar simultaneamente los puntos de la tabla y la función de aproximación

```
exis = Table[i, {i, 1, 20}]
```

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

```
uno = Table[1, {i, 1, 20}]
```

```
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

```
y = Table[Prime[i], {i, 1, 20}]
```

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71}
```

$$\text{sol} = \text{Solve}\left[\begin{pmatrix} \text{exis.exis} & \text{exis.uno} \\ \text{exis.uno} & \text{uno.uno} \end{pmatrix}.\{A, B\} == \{\text{exis.y}, \text{uno.y}\}, \{A, B\}\right] \text{ // N}$$

```
{{A → 3.77368, B → -7.67368}}
```

```
p[x_] = A * x + B /. sol[[1]]
```

$-7.67368 + 3.77368\, x$

Plinomio de grado 2

```
xcuad = Table[i^2, {i, 1, 20}]
```

```
{1, 4, 9, 16, 25, 36, 49, 64, 81, 100,
 121, 144, 169, 196, 225, 256, 289, 324, 361, 400}
```

$$\text{solucion} = \text{Solve}\left[\begin{pmatrix} \text{uno.uno} & \text{uno.exis} & \text{uno.xcuad} \\ \text{exis.uno} & \text{exis.exis} & \text{exis.xcuad} \\ \text{xcuad.uno} & \text{xcuad.exis} & \text{xcuad.xcuad} \end{pmatrix}.\{A, B, C\} == \{\text{uno.y}, \text{exis.y}, \text{xcuad.y}\}, \{A, B, C\}\right] \text{ // N}$$

```
{{A → -1.92368, B → 2.2055, C → 0.0746753}}
```

```
q[x_] = A + B * x + C * x^2 /. solucion[[1]]
```

$-1.92368 + 2.2055\, x + 0.0746753\, x^2$

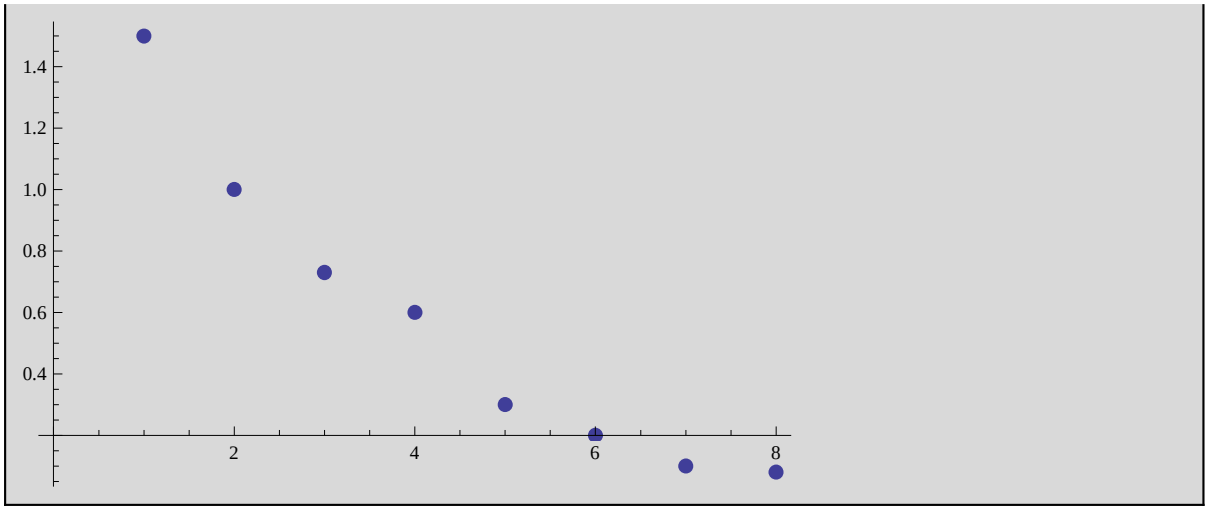
Problema 2

Dada la tabla b = { (1,1.5), (2,1),(3, 0.73),(4, 0.6), (5,0.3),(6, 0.2),(7, 0.1),(8, 0.08)}
a) Dibujar los puntos de la tabla
b) Calcular una tabla "c" con los logaritmos neperianos de los puntos de la tabla "b"
c) Realizar un ajuste de la tabla "b" calculando una aproximación lineal de la tabla "c"
d) Dibujar simultaneamente la función de aproximación y los puntos de la tabla "b".

Cuando se intuye que los datos se ajustan bien a una exponencial. Para confirmarlo se pueden representar los logaritmos neperianos de los datos y ver si se pueden ajustar a una función lineal.
Se puede realizar el ajuste lineal y luego deshacer el cambio o bien realizar el ajuste utilizando la base exponencial.

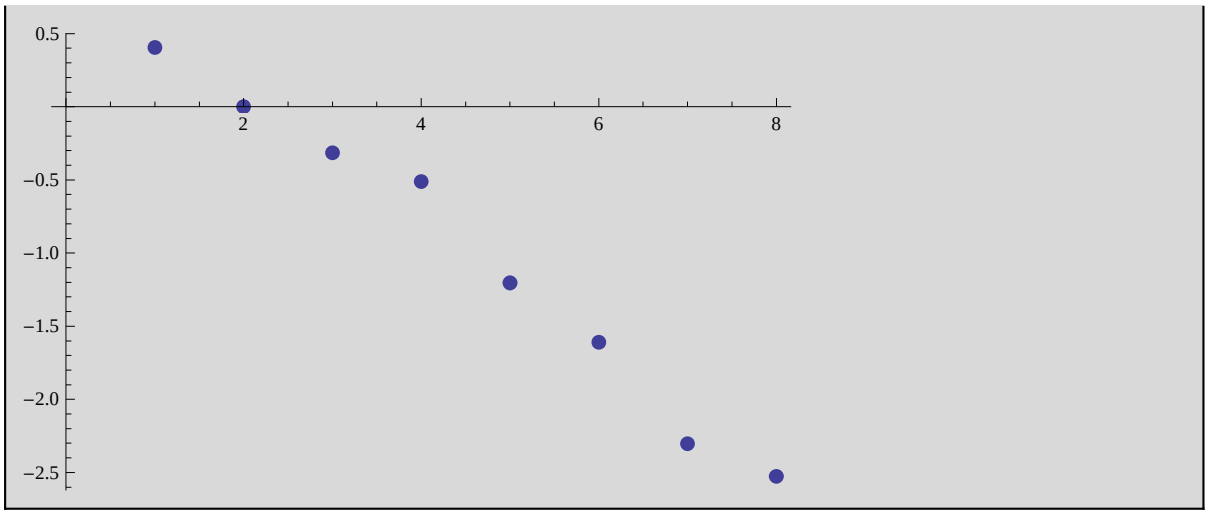
```
b = {1.5, 1, 0.73, 0.6, 0.3, 0.2, 0.1, 0.08};
```

```
g1 = ListPlot[b, PlotStyle → PointSize[0.02]]
```




```
c = Log[b];
```

```
g2 = ListPlot[c, PlotStyle → PointSize[0.02]]
```



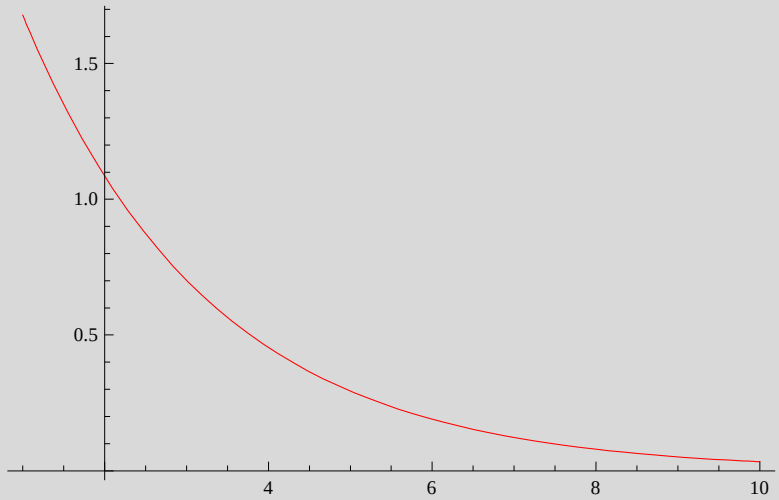

```
d[x_] = Fit[c, {1, x}, x]
```

```
0.953451 - 0.435817 x
```

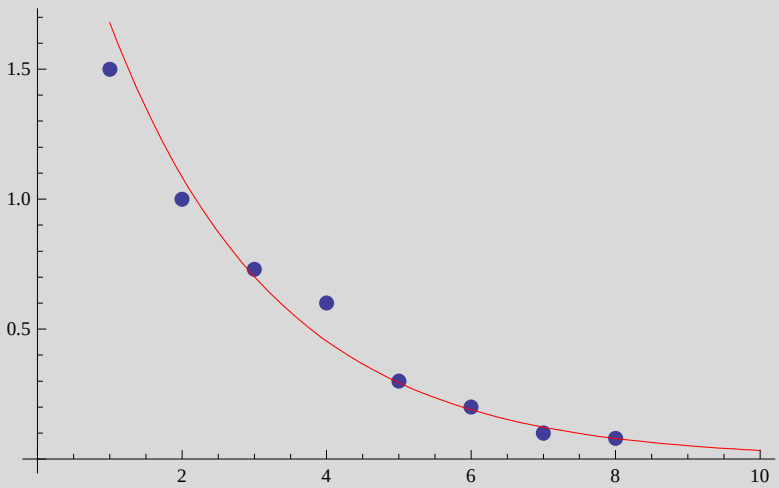
```
ed[x_] = Exp[d[x]]
```

$e^{0.953451-0.435817\,x}$

```
g3 = Plot[ed[x], {x, 1, 10}, PlotStyle → RGBColor[1, 0, 0]]
```




```
Show[g1, g3]
```



Error cuadrático

$$\sum_{i=1}^{8} (c[[i]] - d[i])^2$$

0.14404

Error del ajuste

$$\sum_{i=1}^{8} (b[[i]] - ed[i])^2$$

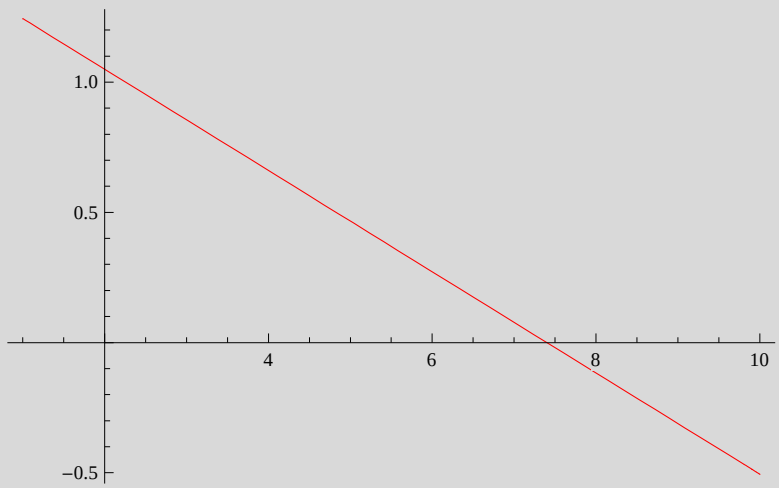
```
0.0617636
```

Ajuste mediante una recta de la tabla inicial

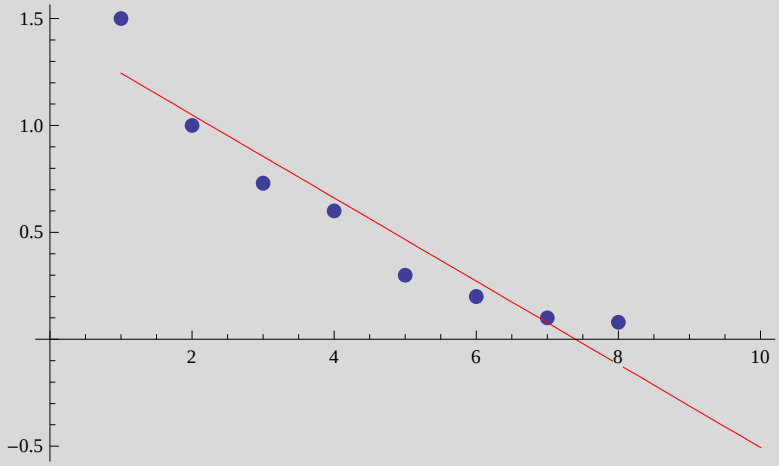
```
rec[x_] = Fit[b, {1, x}, x]
```

```
1.43857 - 0.194405 x
```

```
g4 = Plot[rec[x], {x, 1, 10}, PlotStyle → RGBColor[1, 0, 0]]
```




```
Show[g1, g4]
```



Error cuadrático = error del ajuste

$$\sum_{i=1}^{8} (b[[i]] - rec[i])^2$$

```
0.159473
```

Problema 3

Dada la siguiente tabla
d={(1,2), (2,1.5), (3,1.33), (4,1.25),(5, 1.2),(6, 1.17),(7,1.14), (8,1.13),(9,1.11),(10, 1.1),(11, 1.09),(12, 1.08),(13, 1.07),(14, 1.06),(15, 1.05)}

a) Estudiar de las siguientes funciones a cual de ellas se podría ajustar:

y = b.Exp(cx)

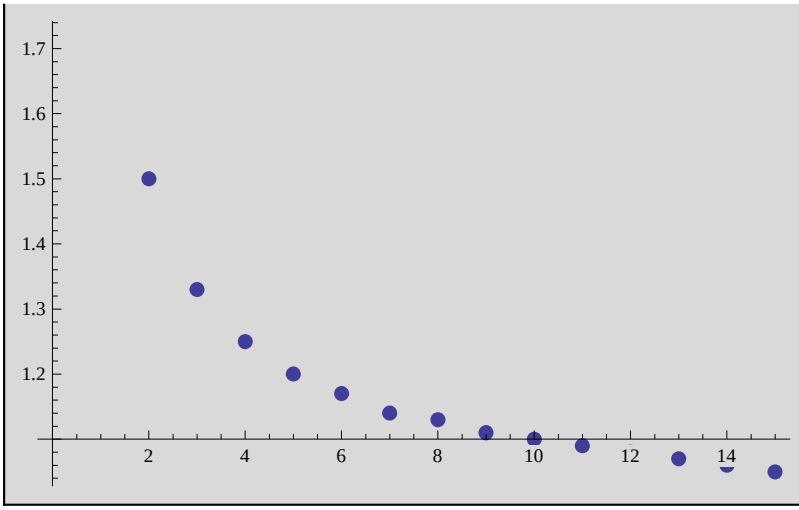
y = 1 / (a x + b)

y = a/x+b

b) Realizar el cambio de variable correspondiente para linealizar la función y obtener la función de aproximación a los puntos de la Tabla "d".

```
d = {2, 1.5, 1.33, 1.25, 1.2, 1.17,
   1.14, 1.13, 1.11, 1.1, 1.09, 1.08, 1.07, 1.06, 1.05};
```

```
g1 = ListPlot[d, PlotStyle → PointSize[0.02]]
```

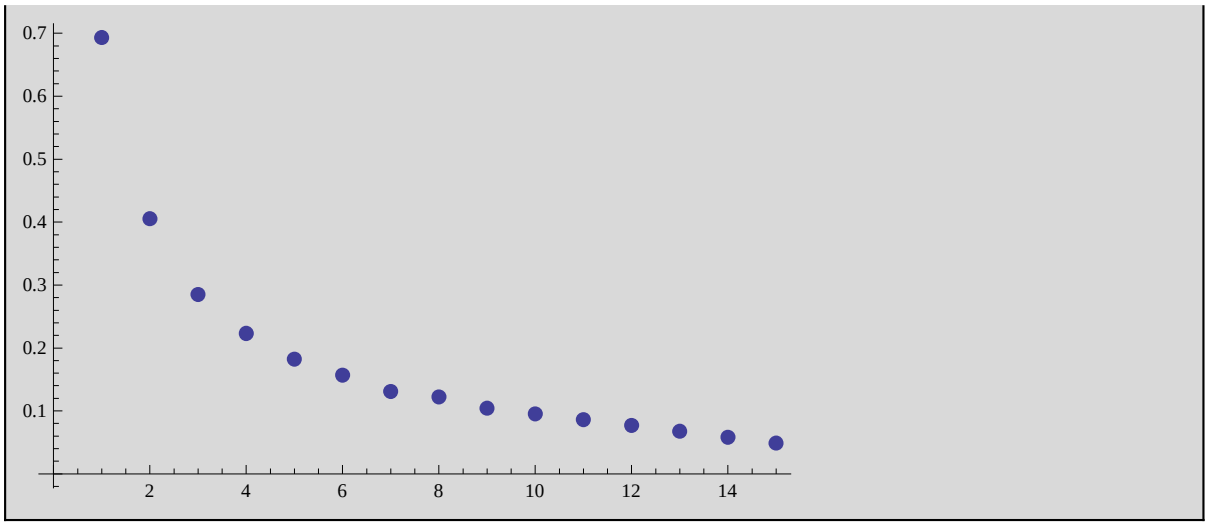


Al representar gráficamente los puntos se ve que no se ajustan a una recta. A continuación se toman logaritmos y se representan los nuevos puntos.

```
d1 = Table[Log[d[[i]]], {i, 1, 15}]
```

```
{Log[2], 0.405465, 0.285179, 0.223144, 0.182322, 0.157004, 0.131028, 0.122218,
 0.10436, 0.0953102, 0.0861777, 0.076961, 0.0676586, 0.0582689, 0.0487902}
```

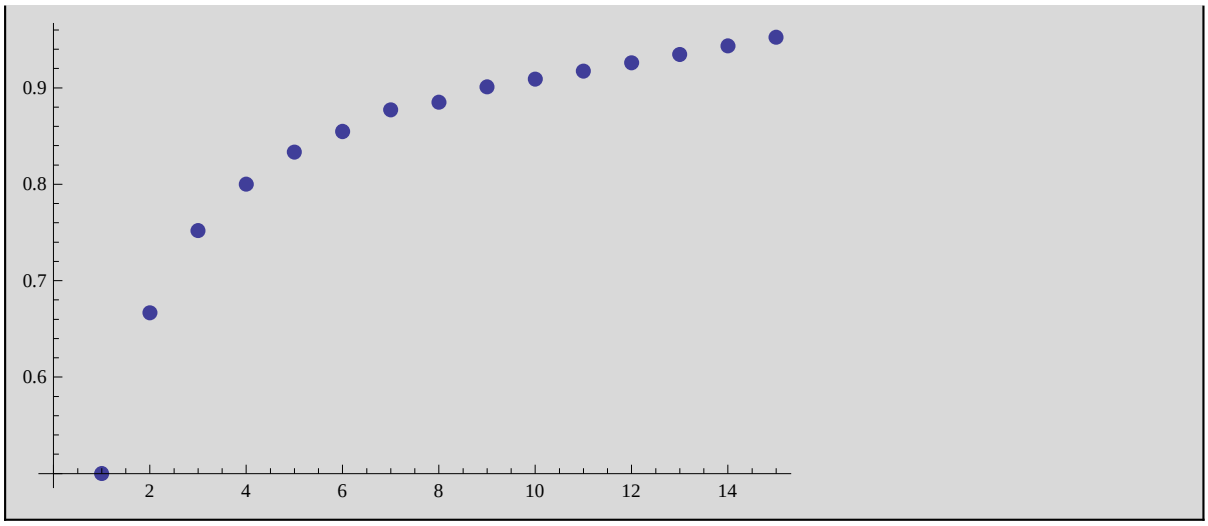
```
ListPlot[d1, PlotStyle → PointSize[0.02]]
```



Al tomar logaritmos tampoco se obtiene una recta. A continuación se representan los inversos de la tabla inicial.

```
d2 = Table[1/d[[i]], {i, 1, 15}];
```

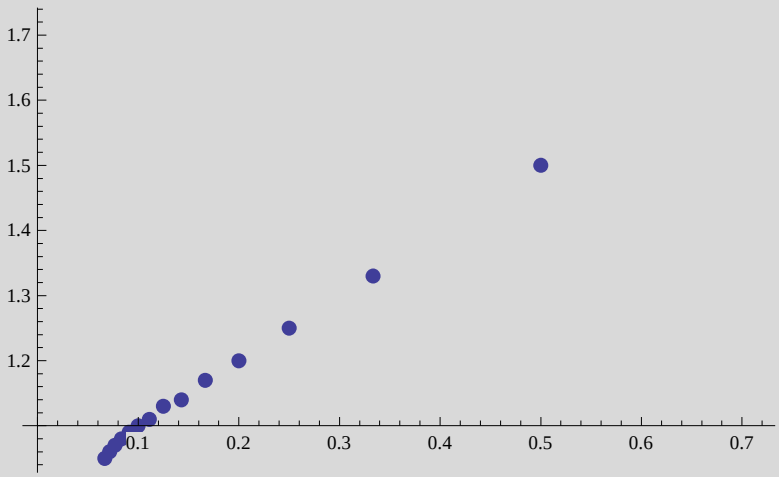
```
ListPlot[d2, PlotStyle → PointSize[0.02]]
```



Al tomar los inversos tampoco se obtiene una recta.

```
d3 = Table[{1/i, d[[i]]}, {i, 1, 15}];
```

```
ListPlot[d3, PlotStyle → PointSize[0.02]]
```



Al sustituir x por 1/x sí se obtiene una recta, luego parece que la función a la que mejor se ajustaría es : y= (a/x) + b.

```
yl[x_] = Fit[d3, {1, x}, x]
```

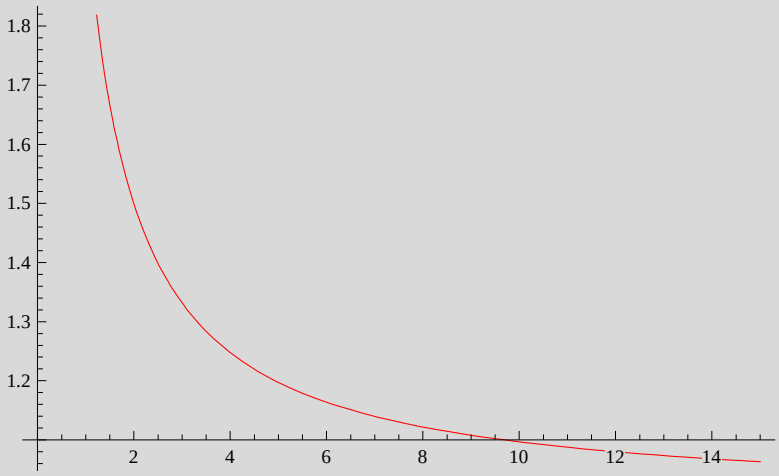
$$0.996099 + 1.00611\,x$$

```
Clear[y]
```

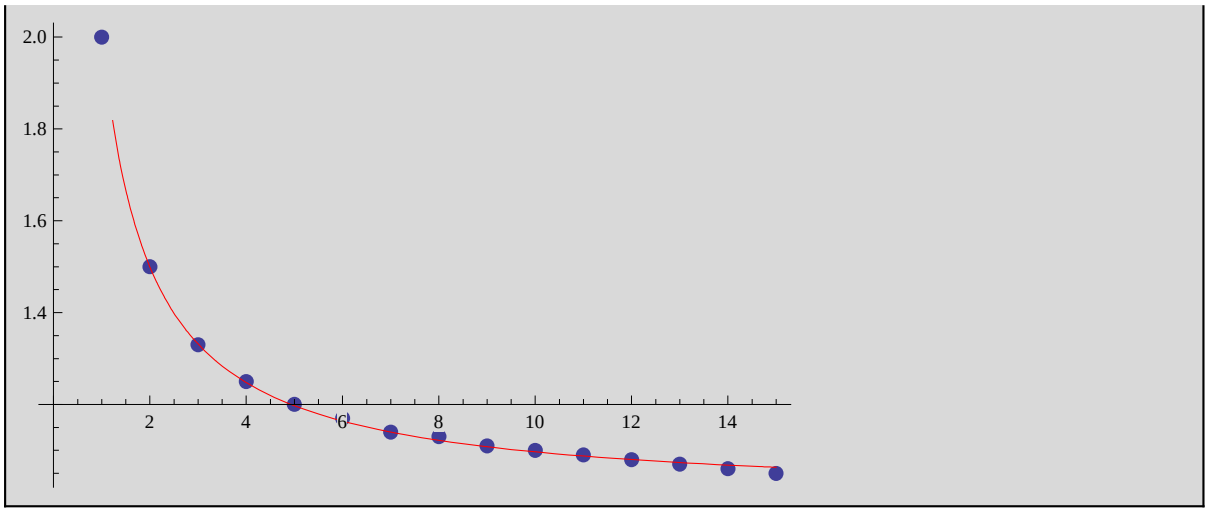
```
y[x_] = yl[1/x]
```

$$0.996099 + \frac{1.00611}{x}$$

```
g2 = Plot[y[x], {x, 0, 15}, PlotStyle → RGBColor[1, 0, 0]]
```



```
Show[g1, g2]
```



A este mismo resultado se puede llegar utilizando como funciones de aproximación 1 y 1/x.

```
yy[x_] = Fit[d, {1, 1 / x}, x]
```

$$0.996099 + \frac{1.00611}{x}$$

Problema 4. Ajuste continuo

Dada la función Cos[x],
a) Realizar un ajuste continuo mediante un polinomio de grado dos
b) Dibujar simultaneamentre la funcion Cos[x] y el polinomio
c)Dibujar la funcion diferencia.
d) realizar el mismo proceso mediante un polinomio de grado cuatro

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

$$\{1, x, x^2\}$$

```
coef = Table[∫_0^1 base[[i]] * base[[j]] dx,
  {i, 1, Length[base]}, {j, 1, Length[base]}]
```

$$\left\{\left\{1, \frac{1}{2}, \frac{1}{3}\right\}, \left\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}\right\}, \left\{\frac{1}{3}, \frac{1}{4}, \frac{1}{5}\right\}\right\}$$

```
MatrixForm[coef]
```

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$$

```
f[x_] := Cos[x]
```

$$\text{ind} = \text{Table}\left[\int_0^1 f[x] * \text{base}[[i]]\, dx,\ \{i, 1, \text{Length}[\text{base}]\}\right]$$

```
{Sin[1], -1 + Cos[1] + Sin[1], 2 Cos[1] - Sin[1]}
```

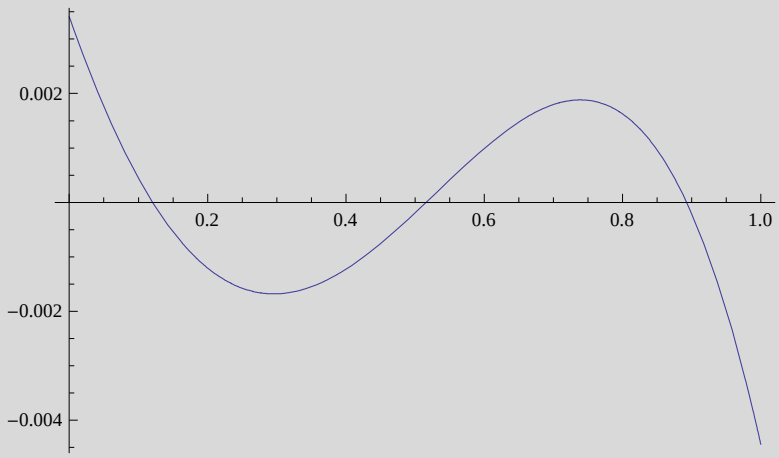
```
sol = N[LinearSolve[coef, ind]]
```

```
{1.00341, -0.0365365, -0.43101}
```

```
p[x_] := sol.base
```

```
pol = Plot[p[x], {x, 0, 1},
   DisplayFunction -> Identity, PlotStyle → {RGBColor[1, 0, 0]}];
fun = Plot[f[x], {x, 0, 1}, DisplayFunction -> Identity];
Show[{pol, fun}, DisplayFunction -> $DisplayFunction];
```

```
error1 = Plot[p[x] - Cos[x], {x, 0, 1}]
```



Polinomio de grado cuatro.

```
base = {1, x, x^2, x^3, x^4}
```

$$\{1, x, x^2, x^3, x^4\}$$

```
coef = Table[∫_0^1 base[[i]] * base[[j]] dx,
  {i, 1, Length[base]}, {j, 1, Length[base]}]
```

$$\left\{\left\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}\right\}, \left\{\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}\right\}, \left\{\frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}\right\}, \left\{\frac{1}{4}, \frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}\right\}, \left\{\frac{1}{5}, \frac{1}{6}, \frac{1}{7}, \frac{1}{8}, \frac{1}{9}\right\}\right\}$$

```
MatrixForm[coef]
```

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}$$

```
f[x_] := Cos[x]
```

```
ind = Table[∫_0^1 f[x] * base[[i]] dx, {i, 1, Length[base]}]
```

```
{Sin[1], -1 + Cos[1] + Sin[1], 2 Cos[1] - Sin[1],
 6 - 3 Cos[1] - 5 Sin[1], -20 Cos[1] + 13 Sin[1]}
```

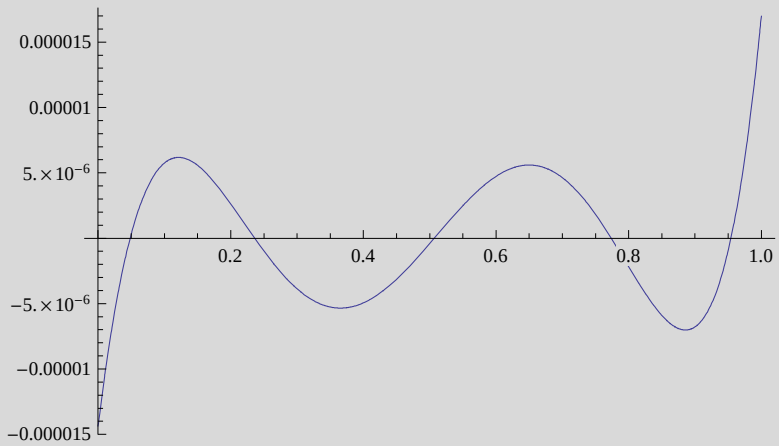
```
sol = N[LinearSolve[coef, ind]]
```

```
{0.999986, 0.000414706, -0.502729, 0.00649594, 0.0361524}
```

```
p[x_] := sol.base
```

```
pol = Plot[p[x], {x, 0, 1},
   DisplayFunction -> Identity, PlotStyle → {RGBColor[1, 0, 0]}];
fun = Plot[f[x], {x, 0, 1}, DisplayFunction -> Identity];
Show[{pol, fun}, DisplayFunction -> $DisplayFunction];
```

```
error2 = Plot[p[x] - Cos[x], {x, 0, 1}]
```




```
Show[error1, error2]
```