Modelos de Regresión Linealizables con wxMaxima

Ejemplo 1: *En cierta especie animal se estudia la relación entre el peso X (en Kgs) y el volumen pulmonar Y (en litros), obteniéndose los datos*

X =peso (kgs)	60	85	100	150	250
Y =vol pulmonar (l)	$2'3$	4	5	9	$19'5$

Ajustar los datos a un modelo $Y = aX^b$.

Solución: si fuera necesario, cargar el paquete *stats*, y empezar con *numer:true*, etc...

```
(%i1)   x:[60,85,100,150,250];y:[2.3,4,5,9,19.5]
(%i2)   xy:transpose(matrix(x,y))
(%i3)   u:log(x);v:log(y)
(%i4)   uv:transpose(matrix(u,v))
(%i5)   simple_linear_regression(uv)
```

$$\begin{array}{ll} \text{model} = 1{,}487\,x - 5{,}2416 & \text{correlation} = 0{,}9998 \\ v_{\text{est}} = 0{,}00027 & b_{\text{conf int}} = [1{,}4393, 1{,}5348] \\ \ldots & \ldots \end{array}$$

Es decir, la recta de regresión es $v = -5{,}2416 + 1{,}487u$ y $r = 0'999$. Conviene dibujarla para asegurarse que el ajuste es correcto (ver dibujo abajo izda)

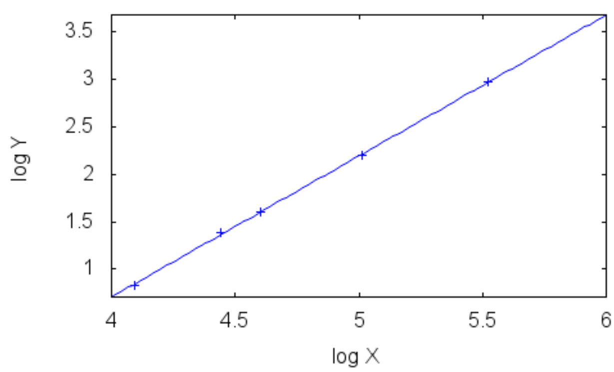
```
(%i6)   wxdraw2d(points(uv), explicit(-5.2416+1.487*'u, 'u, 4,6),
        xlabel="log X",ylabel="log Y" )
```

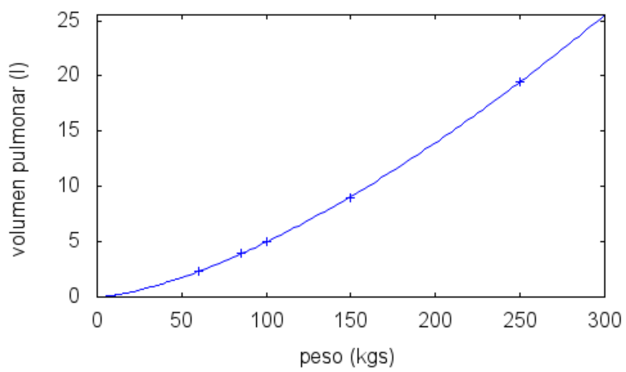
Por último, deshacemos el cambio de variables a mano

$$\log y = -5{,}2416 + 1{,}487 \log x \;\Rightarrow\; y = \exp\left(-5{,}2416 + 1{,}487 \log \mathrm{x}\right) = \mathrm{e}^{-5{,}2416}\mathrm{x}^{1{,}487} = 0{,}0053\,\mathrm{x}^{1{,}487}$$

y visualizamos el ajuste dibujando los puntos originales y la curva obtenida (abajo dcha)

```
(%i7)   wxdraw2d(points(xy), explicit(0.0053*'x^1.487, 'x, 0,300),
        xlabel="peso (kgs)",ylabel="volumen pulmonar (l)" )
```





Nota: si nos pidieran calcular $\sqrt{ECM_{x,y}}$ podemos escribir

```
(%i8)   sqrt(mean((y-0.0053*x^1.487)^2));
(%o8)   0.06743
```

Regresión No Lineal: Método de Mínimos Cuadrados

Ejemplo 2: *Ajustar los datos del ejemplo anterior a una curva* $Y = aX^b$ *usando el método de mínimos cuadrados, y comparar visualmente los dos ajustes*

Solución: si fuera necesario, cargar el paquete *lsquares*.

```
(%i1)  x:[60,85,100,150,250];y:[2.3,4,5,9,19.5]
(%i2)  xy:transpose(matrix(x,y))
(%i3)  lsquares_estimates(xy, [X, Y], Y = a * X^b, [a, b])
```

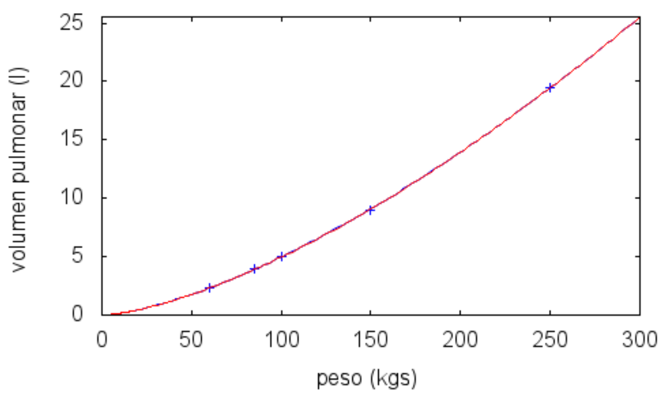
Recordar: $xy \to$ matriz de datos, $[X, Y] \to$ nombre de las variables en la matriz de datos, $Y = a * X^b \to$ curva a la que se quieren ajustar los datos, $[a, b] \to$ parámetros buscados

El output devuelve, tras varias líneas con iteraciones del método numerico (que no usamos), la estimación buscada de $[a, b]$

```
(%o3)  [[a=0.0052,b=1.4901]]
```

Conclusión: la curva que mejor se ajusta con Mínimos Cuadrados es $Y = 0'0052 * X^{1'4901}$. Es muy parecida a la anterior, y por tanto las gráficas son casi indistinguibles

```
(%i4)  wxdraw2d(points(xy), explicit(0.0053*'x^1.487, 'x, 0,300),
       color=red, explicit(0.0052*'x^1.4901, 'x, 0,300),
       xlabel="peso (kgs)",ylabel="volumen pulmonar (l)" )
```



Nota: `lsquares_estimates` devuelve un mensaje de error cuando el procedimiento numérico para estimar $[a, b]$ es demasiado complejo. En tal caso hay que introducir información adicional: una estimación inicial de a, b (que se da a ojo, mirando la gráfica), y un nivel de tolerancia del error. Para ello se procede en 2 pasos, con los comandos

```
mse : lsquares_mse(xy, [X, Y], Y = a * X^b);
lsquares_estimates_approximate(mse, [a, b], initial =[0,005, 1,5], tol =0,001);
```

donde `initial` es la estimación inicial de [a,b] (la damos a ojo), y `tol` es la tolerancia del error (10^{-3} es razonable). Esta manera de proceder se hace necesaria cuando las curvas tienen expresiones más complicadas, como en los modelos de Michaelis-Menten $Y = a * X/(b + X)$.

Ejemplo 3: *Usando mínimos cuadrados, ajustar los datos siguientes a un modelo de Michaelis-Menten* $V = a * S/(k + S)$

S =concentr sustrato (mol/ml)	$0'10$	$0'25$	$0'50$	1	2	4	8
V =veloc reacción (mol/min)	21	$34'6$	$44'1$	$61'1$	$73'8$	$73'9$	$76'3$

Solución:

(%i1) `s: [0.10,0.25,0.50,1,2,4,8];v:[21,34.6,44.1,61.1,73.8,73.9,76.3]`

(%i2) `sv:transpose(matrix(s,v))`

(%i3) `mse : lsquares_mse(`$sv, [S, V], V = a * S/(k + S)$`);`

(%i4) `lsquares_estimates_approximate(`$mse, [a, k]$`, initial =`$[75, 0'25]$`, tol =`$0{,}001$`);`

Los datos aproximados $a = 75$, $k = 0'25$ los introducimos a ojo. El output devuelve (tras varias líneas con iteraciones del método numerico) la estimación buscada de $[a, k]$

(%o4) `[[a=81.159,k=0.3398]]`

(%i5) `wxdraw2d(points(sv),color=red, explicit(`$81{,}159{*}S/(0{,}3398{+}S)$`), S, 0,12), xlabel="concentr S (mol/l)",ylabel="veloc reacc (mol/min)" )`