

# Práctica 3. Regresión no lineal. Soluciones.

Universidad de Alcalá. Curso 2023-24.

Estadística (650008). Grado en biología sanitaria.

Actualizado: 2023-10-05

## Indicaciones:

- Puedes leer los datos con el botón `Import dataset` que hay en el marco superior derecho.
- Probablemente necesites tener a mano la **diapositiva con las transformaciones** (cambios de variable), es la última de la clase de regresión no lineal.

## Procedimiento general:

1. Leer el fichero de datos.
2. Visualizar la nube de puntos. Esto implica elegir las variables explicativa y respuesta.
3. Selección de modelo: nos quedamos con el que tenga mayor  $r^2$  (mayor poder explicativo).
4. Trabajar con las variables transformadas de acuerdo con el modelo elegido.

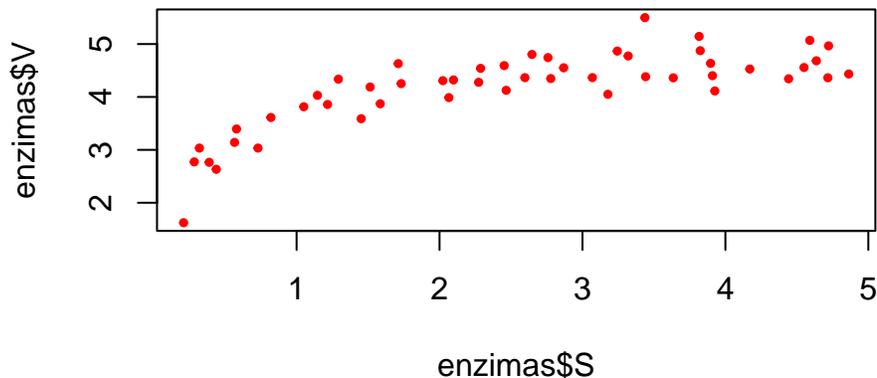
**Haz al menos el ejercicio 1, uno de entre el 2 y el 3, y el ejercicio 4.**

**Ejercicio 1** Este fichero contiene los datos de la velocidad  $V$  a la que tiene lugar cierta reacción enzimática para distintas concentraciones  $S$ .

Guarda el contenido del fichero de datos en la variable `enzimas`. Determina los coeficientes de la curva que mejor ajusta esta nube de puntos.

En primer lugar, hay que visualizar la nube de puntos. La segunda pregunta sugiere considerar  $S$  como variable explicativa y  $V$  como la variable respuesta

```
plot(enzimas$S, enzimas$V, cex = 0.5, pch = 19, col = "red")
```



La forma del gráfico sugiere o bien un modelo logarítmico, o bien una hipérbola rectangular (puede que ya sepas que las reacciones enzimáticas de tipo Michaelis-Menten siguen una hipérbola). En cualquier caso, calcula las correlaciones de los datos “linealizados” (es decir, de las variables una vez que has hecho el cambio de variable que linealizaría la nube de puntos; puede que quieras consultar la tabla de la última diapositiva de la clase en la que vimos la regresión no lineal):

Para el cambio logarítmico

```
cor(log(enzimas$S), enzymas$V)^2
```

```
## [1] 0.7950565
```

Para el cambio a una hipérbola rectangular

```
cor(1/enzimas$S, 1/enzimas$V)^2
```

```
## [1] 0.8581265
```

La hipérbola explica en mayor proporción el comportamiento de los datos. Recuerda que la ecuación de una hipérbola rectangular es

$$V = \frac{b_0 S}{1 + b_1 S}$$

El cambio de variable que transforma la expresión anterior en una recta es  $S = 1/S_{inv}$ ,  $V = 1/V_{inv}$ . Si haces ese cambio, tienes

$$\frac{1}{V_{inv}} = \frac{b_0 \frac{1}{S_{inv}}}{1 + b_1 \frac{1}{S_{inv}}}$$

multiplica el numerador de la fracción de la derecha por  $S_{inv}$  para obtener

$$\frac{1}{V_{inv}} = \frac{b_0}{S_{inv} + b_1}$$

lo que equivale a

$$S_{inv} + b_1 = b_0 V_{inv}$$

es decir

$$V_{inv} = \frac{S_{inv}}{b_0} + \frac{b_1}{b_0}$$

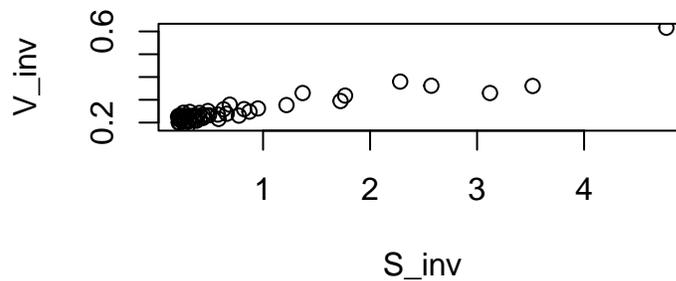
que es la ecuación de una recta.

Haz el cambio de variable en los datos

```
V_inv = 1/enzimas$V
S_inv = 1/enzimas$S
```

visualiza la nube de puntos

```
plot(S_inv, V_inv)
```



y, como los puntos aparecen razonablemente alineados, calcula el modelo lineal y obtén los coeficientes

```
modelo = lm(V_inv ~ S_inv)
modelo$coefficients
```

```
## (Intercept)      S_inv
## 0.19891571 0.06735554
```

Esto proporciona la fórmula

$$V_{inv} = 0.0673555 * S_{inv} + 0.1989157$$

Para obtener los valores de  $b_0$  y  $b_1$ , recuerda que

$$V_{inv} = \frac{S_{inv}}{b_0} + \frac{b_1}{b_0}$$

es decir,

$$\frac{1}{b_0} = 0.0673555 = \text{pendiente de la recta } S_{inv}, V_{inv}$$

$$\frac{b_1}{b_0} = 0.1989157 = \text{término independiente de la recta } S_{inv}, V_{inv}$$

Puedes despejar los coeficientes

$$b_0 = \frac{1}{0.0673555} = 14.8465879$$

y

$$b_1 = 0.1989157b_0 = 2.9532196$$

de donde la fórmula de la hipérbola es

$$V = \frac{14.8465879 * S}{1 + 2.9532196 * S}$$

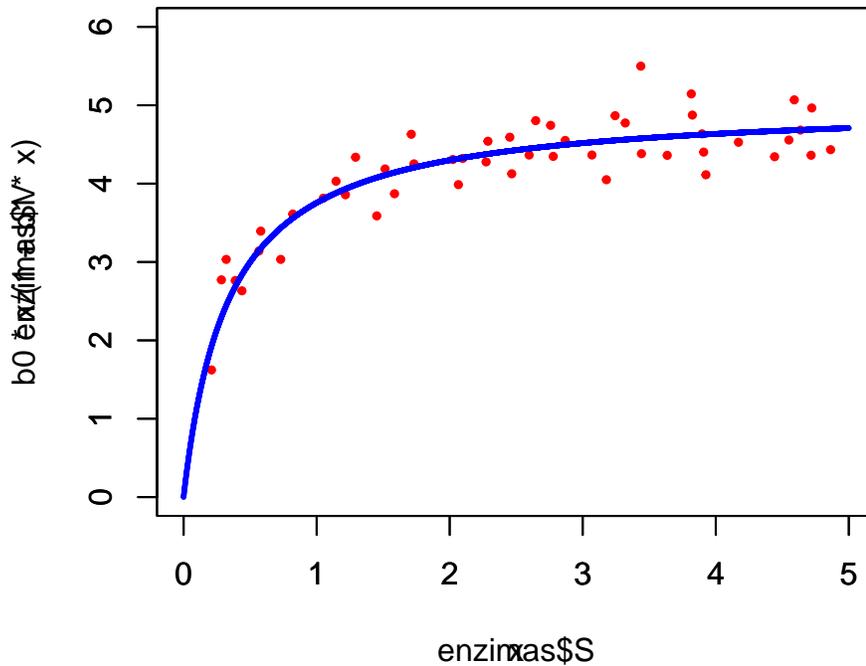
Si defines

```
termIndpte = unname(modelo$coefficients[1])
pendiente = unname(modelo$coefficients[2])
```

```
b0 = 1/pendiente
b1 = b0*termIndpte
```

puedes representar la nube de puntos original, y la curva que la aproxima

```
plot(enzimas$S, enzimas$V, cex = 0.5, pch = 19, col = "red",
     xlim = c(0,5), ylim = c(0, 6))
par(new = TRUE)
x = seq(from = 0, to = 5, by = 0.001)
plot(x, b0*x/(1+b1*x), cex = 0.25, pch = 19, col = "blue",
     xlim = c(0,5), ylim = c(0, 6))
```

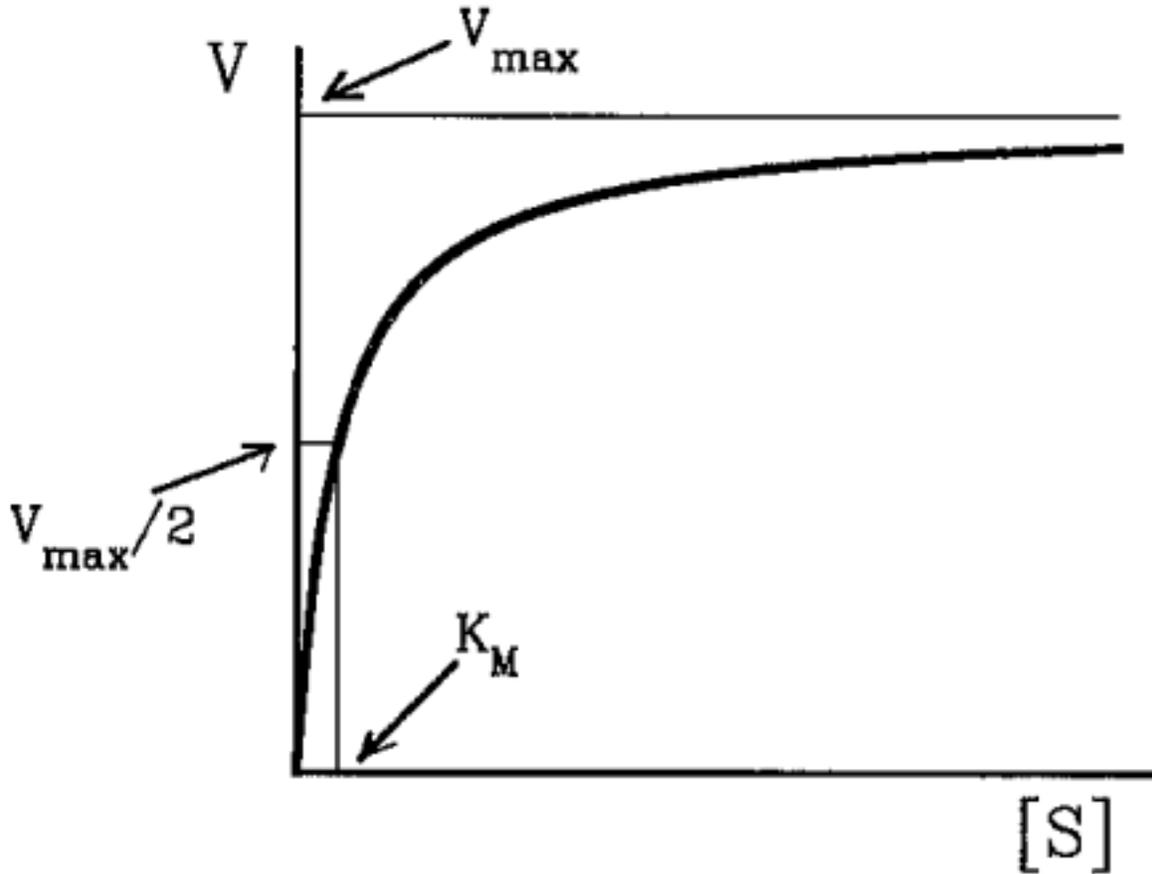


En el contexto de la cinética enzimática hay dos cantidades (parámetros) importantes que caracterizan la reacción:

- La velocidad máxima de reacción  $V_{max}$
- La constante de Michaelis  $K_M$ , que es una combinación de varias constantes cinéticas propia de cada reacción.

En el leguaje de las funciones:

- $V_{max}$  es la asíntota de la hipérbola
- $K_M$  es la concentración a la que se alcanza la mitad de la velocidad máxima



Calcula  $V_{max}$  y  $K_M$  para los datos que tienes

Por un lado, la asíntota de la hipérbola se calcula como un límite:

$$\lim_{S \rightarrow \infty} \frac{14.8465879 * S}{1 + 2.9532196 * S} = \frac{14.8465879}{2.9532196} = 5.027255$$

Por otro lado,  $K_M$  cumple la condición:

$$V_{max}/2 = \frac{b_0 * K_M}{1 + b_1 * K_M}$$

Despejando  $K_M$  se tiene

$$V_{max} * (1 + b_1 * K_M) = 2 * b_0 * K_M$$

que es equivalente a

$$KM = \frac{Vmax}{2b_0 - b_1 Vmax} = 0.3386135$$

```
KM = Vmax/(2*b0-b1*Vmax)
```

Puedes comprobar que

$$\frac{b_0 * KM}{1 + b_1 * KM} = \frac{14.8465879 * 0.3386135}{1 + 2.95321960 * 0.3386135} = 2.5136275$$

coincide con

$$\frac{Vmax}{2} = \frac{5.027255}{2} = 2.5136275$$

**Calcula la velocidad que predice el modelo para una concentración S = 3.**

Hay dos alternativas:

O bien sustituyes en la ecuación de la hipérbola

```
b0*3/(1+b1*3)
```

```
## [1] 4.517374
```

O bien, puedes sustituir directamente en la recta de regresión obtenida

$$V_{inv} = \frac{S_{inv}}{b_0} + \frac{b_1}{b_0}$$

es decir,

$$\frac{1}{V} = 0.0673555 \frac{1}{3} + 0.1989157 = 0.2213676$$

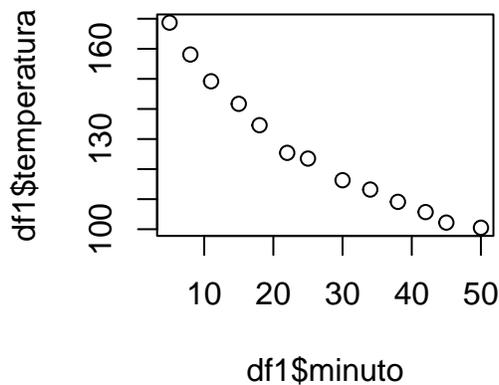
y despejar

$$V = \frac{1}{0.2213676} = 4.5173738$$

**Ejercicio 2** Ejercicio basado en este material. Se ha calentado un preparado hasta aproximadamente 179.5F (Fahrenheit) y se ha dejado enfriar durante 50 minutos. Este fichero contiene los valores de la temperatura medida en distintos instantes.

- **Elige las variables explicativa y respuesta.** A la vista del último apartado, la variable explicativa es el tiempo (minuto) y la variable respuesta es la temperatura.
- **Representa el diagrama de dispersión.** Suponer que el fichero de datos se llama `df1`

```
plot(df1$minuto, df1$temperatura)
```



- **Establece el/los modelo(s) candidato(s) a ajustar esos datos.** El diagrama anterior sugiere los modelos exponencial o potencial. También podría tratarse de un logaritmo.
- **Selecciona el mejor modelo y calcula los coeficientes del mismo.** La selección del modelo se hace con el coeficiente de determinación calculado sobre los datos linealizados (transformados de forma que puedan ser ajustados por una recta). En el caso de la exponencial se cambia temperatura por su logaritmo

```
cor(df1$minuto, log(df1$temperatura))^2
```

```
[1] 0.967841
```

En el caso de la potencia se cambian ambas variables por su logaritmo

```
cor(log(df1$minuto), log(df1$temperatura))^2
```

```
[1] 0.9803893
```

Para el logaritmo se cambian la variable independiente por su logaritmo

```
cor(log(df1$minuto), df1$temperatura)^2
```

```
[1] 0.9935502
```

No está de más ver qué sucede con el modelo lineal

```
cor(df1$minuto, df1$temperatura)^2
```

```
[1] 0.9394671
```

y con el hiperbólico

```
cor(1/df1$minuto, 1/df1$temperatura)^2
```

```
[1] 0.7501289
```

Concluimos que el modelo logaritmo  $temperatura = b_0 + b_1 \ln(minuto)$  es el mejor de los cinco

Calculamos los coeficientes del modelo linealizado

```
(lm1 = lm(df1$temperatura ~ log(df1$minuto)))
```

```
Call:
lm(formula = df1$temperatura ~ log(df1$minuto))
```

```
Coefficients:
(Intercept)  log(df1$minuto)
      222.16          -30.98
```

Recuerda que en este caso tanto el término independiente como la pendiente obtenidas a partir de los datos transformados son directamente los coeficientes del modelo

```
(b0 = unname(lm1$coefficients[1]))
```

```
[1] 222.1606
```

```
(b1 = unname(lm1$coefficients[2]))
```

```
[1] -30.98433
```

de modo que el modelo que hemos obtenido es

$$temperatura = 222.1606069 - 30.9843278 * \ln(minuto)$$

A tener en cuenta:

–  $\ln$  se refiere al logaritmo neperiano, que en R se escribe como `log()`

- **Calcula la temperatura a la que según modelo estaba el preparado a los 20 minutos.** Hay que sustituir en la fórmula anterior

```
b0 + b1 * log(20)
```

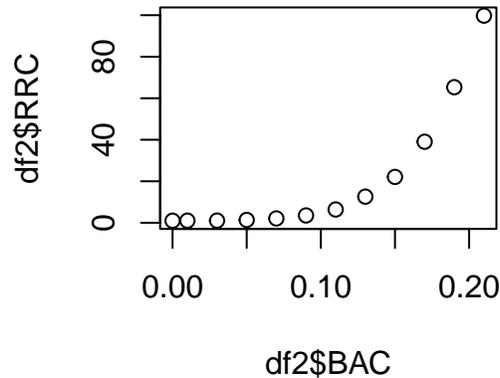
```
[1] 129.3399
```

No es necesario usar tantos decimales y se puede redondear a 129.3F.

**Ejercicio 3** Ejercicio basado en este material. Este fichero contiene datos de la concentración de alcohol en sangre (BAC, blood alcohol concentration) y el riesgo relativo de tener un accidente al volante (RRC, relative risk of crashing).

- **Elige las variables explicativa y respuesta.** A la vista del último apartado, la variable explicativa es la concentración de alcohol en sangre (BAC) y la variable respuesta es el riesgo relativo de chocar RRC.
- **Representa el diagrama de dispersión.** Suponer que el fichero de datos se llama `df1`

```
plot(df2$BAC, df2$RRC)
```



- **Establece el/los modelo(s) candidato(s) a ajustar esos datos.** El diagrama anterior sugiere los modelos exponencial o potencial. De todos modos, no está de más comprobarlos todos (también el lineal, logarítmico e hiperbólico).
- **Selecciona el mejor modelo y calcula los coeficientes del mismo.** La selección del modelo se hace con el coeficiente de determinación calculado sobre los datos linealizados (transformados de forma que puedan ser ajustados por una recta). En el caso de la exponencial se cambia temperatura por su logaritmo

```
cor(df2$BAC, log(df2$RRC))^2
```

[1] 0.9712205

En el caso de la potencia se cambian ambas variables por su logaritmo

```
cor(log(df2$BAC), log(df2$RRC))^2
```

[1] NaN

Se produce un error al calcular la correlación. Si observamos los datos de `log(df2$BAC)`, que es lo que hay de diferente con el último cálculo que funcionó vemos que

```
log(df2$BAC)
```

```
[1]      -Inf -4.605170 -3.506558 -2.995732 -2.659260 -2.407946 -2.207275
[8] -2.040221 -1.897120 -1.771957 -1.660731 -1.560648
```

Aparece un valor `-Inf` que tiene que ver con intentar calcular el logaritmo de 0

```
df2$BAC
```

```
[1] 0.00 0.01 0.03 0.05 0.07 0.09 0.11 0.13 0.15 0.17 0.19 0.21
```

Así es que debemos tomar una decisión: o bien usar directamente la exponencial (porque no es posible probar con los modelos que involucran el logaritmo de BAC), o bien prescindir de la primera observación (para poder calcular los logaritmos). Vamos a optar por esta última opción, ya que introduce algo nuevo (eliminar filas de una tabla). Como nos queremos quedar con todas las filas excepto la primera, y con todas las columnas, podemos escribir

```
df2bis = df2[2:nrow(df2), ]
```

Compara

```
head(df2)
```

```
  BAC  RRC
1 0.00 1.00
2 0.01 1.03
3 0.03 1.06
4 0.05 1.38
5 0.07 2.09
6 0.09 3.54
```

```
head(df2bis)
```

```
  BAC  RRC
2 0.01 1.03
3 0.03 1.06
4 0.05 1.38
5 0.07 2.09
6 0.09 3.54
7 0.11 6.41
```

Calculamos ahora los coeficientes de determinación:

Recta

```
cor(df2bis$BAC, df2bis$RRC)^2
```

```
[1] 0.7163863
```

Exponencial

```
cor(df2bis$BAC, log(df2bis$RRC))^2
```

```
[1] 0.9813868
```

Potencia

```
cor(log(df2bis$BAC), log(df2bis$RRC))^2
```

```
[1] 0.7454332
```

Logaritmo

```
cor(log(df2bis$BAC), df2bis$RRC)^2
```

```
[1] 0.4004605
```

Hiperbola

```
cor(1/df2bis$BAC, 1/df2bis$RRC)^2
```

```
[1] 0.5966757
```

Claramente el modelo exponencial  $RRC = b_0 e^{b_1 * BAC}$  es el mejor de los dos. **Volvemos a usar el conjunto de datos original, con la primera fila, porque el 0 no da problema en el cambio a la exponencial**

Calculamos los coeficientes del modelo linealizado.

```
(lm2 = lm(log(df2$RRC) ~ df2$BAC))
```

Call:

```
lm(formula = log(df2$RRC) ~ df2$BAC)
```

```

Coefficients:
(Intercept)    df2$BAC
   -0.5395      23.8176

```

Recuerda que el término independiente obtenido es el logaritmo de  $b_0$ , es decir, el valor de  $b_0$  es

```
(b0 = exp(unname(lm2$coefficients[1])))
```

```
[1] 0.5830483
```

```
(b1 = (unname(lm2$coefficients[2])))
```

```
[1] 23.81758
```

de modo que el modelo que hemos obtenido es

$$RCC = 0.5830483 * e^{23.8175766 * BAC}$$

**Si hubieras optado por continuar sin usar la primera fila de la tabla** el resultado sería el siguiente

Calculamos los coeficientes del modelo linealizado.

```
(lm2bis = lm(log(df2bis$RRC) ~ df2bis$BAC))
```

Call:

```
lm(formula = log(df2bis$RRC) ~ df2bis$BAC)
```

Coefficients:

```
(Intercept)  df2bis$BAC
   -0.7369      25.1663
```

Recuerda que el término independiente obtenido es el logaritmo de  $b_0$ , es decir, el valor de  $b_0$  es

```
(b0bis = exp(unname(lm2bis$coefficients[1])))
```

```
[1] 0.4786011
```

```
(b1bis = (unname(lm2bis$coefficients[2])))
```

```
[1] 25.16629
```

de modo que el modelo que hemos obtenido es

$$RCC = 0.4786011 * e^{25.1662898 * BAC}$$

- **Calcula el RRC que, según modelo, está asociado con un BAC de 0.16.** Basta sustituir en la fórmula anterior

```
b0 * exp(b1*0.16)
```

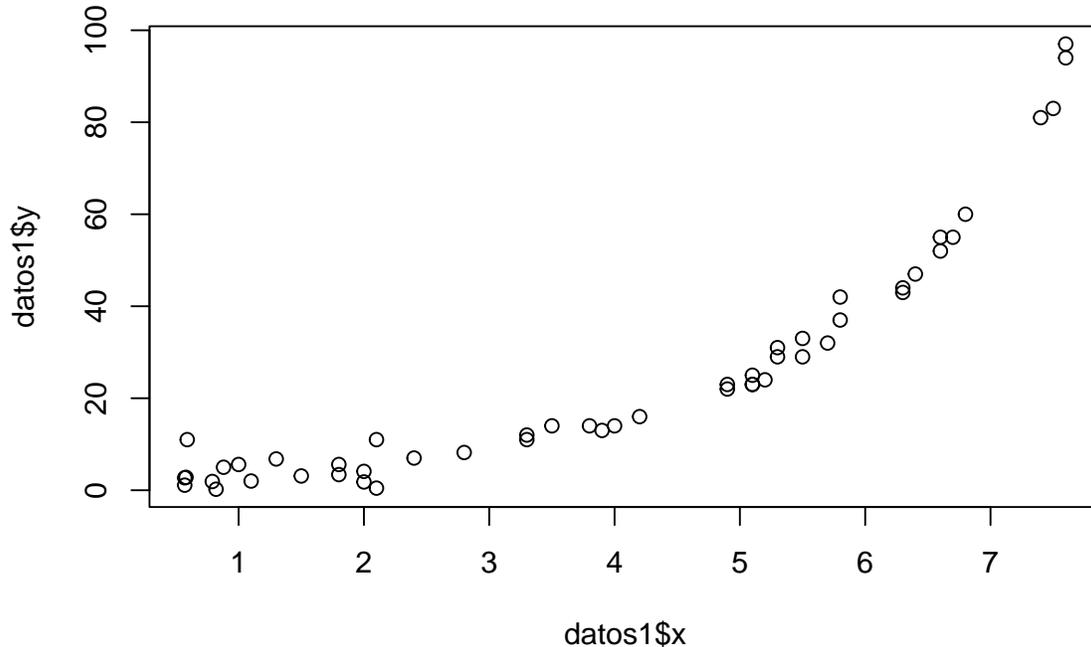
```
[1] 26.34628
```

Obviamente, **no es preciso (ni conveniente) usar tantos decimales para dar el resultado final**, que se puede redondear a 81.00 con dos cifras significativas. Es decir, según el modelo es 27 veces más probable tener un accidente con 0.16 de concentración de alcohol en sangre que con 0.

**Ejercicio 4** Lee este fichero de datos. Se pide:

- **Representa el diagrama de dispersión.** Si llamamos a la tabla de datos `datos1`, obtenemos el diagrama de dispersión con:

```
plot(datos1$x, datos1$y)
```



- **Establece el/los modelo(s) candidato(s) a ajustar esos datos.** los candidatos son, o bien una función exponencial

$$y = b_0 e^{b_1 x}$$

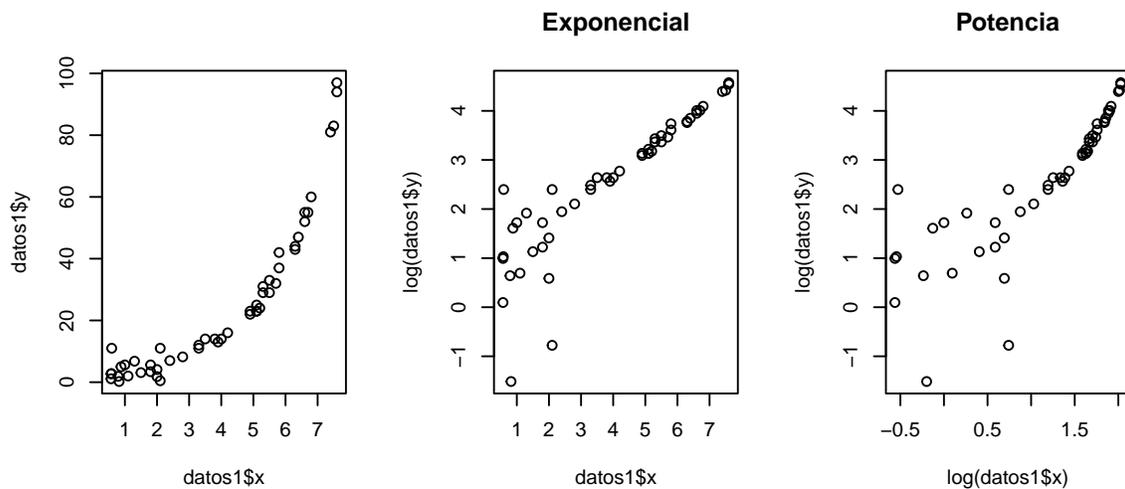
o bien una función potencial

$$y = b_0 x^{b_1}$$

- **Selecciona el mejor modelo y calcula los coeficientes del mismo.** Usaremos el modelo que tenga un coeficiente de determinación mayor, es decir, aquel que explique en mayor medida (proporción) la variabilidad conjunta de las variables involucradas. Para hacer el cálculo hay que “linealizar” los datos, es decir, aplicar una transformación (o cambio de variable). Para la exponencial hay que trabajar con  $x$ ,  $w = \log(y)$ , mientras que para una potencial es  $z = \log(x)$ ,  $w = \log(y)$ .

Fíjate en los diagramas de dispersión

```
par(mfrow = c(1,3))
plot(datos1$x, datos1$y)
plot(datos1$x, log(datos1$y), main = "Exponencial")
plot(log(datos1$x), log(datos1$y), main = "Potencia")
```



```
par(mfrow = c(1,1))
```

Salvo por unos pocos valores, la figura del centro parece ser la que mejor transforma los datos.

Los coeficientes de determinación (que son los de correlación al cuadrado) son

```
# exponencial
cor(datos1$x, log(datos1$y))^2
```

```
[1] 0.8051923
```

```
# potencia
cor(log(datos1$x), log(datos1$y))^2
```

```
[1] 0.7198342
```

Puedes comprobar el resto de modelos que conoces

```
# lineal
cor(datos1$x, datos1$y)^2
```

```
[1] 0.7831438
```

```
# logaritmico
cor(log(datos1$x), datos1$y)^2
```

```
[1] 0.549002
```

```
# hiperbólico
cor(1/datos1$x, 1/datos1$y)^2
```

```
[1] 0.1464847
```

Efectivamente, el de la exponencial es el mejor de todos: no tiene un valor muy próximo a uno, pero corrobora la inspección visual.

Calculamos los coeficientes de la recta de regresión de las variables linealizadas

```
# guardar el modelo lineal en una variable
lmExp = lm(log(datos1$y) ~ datos1$x)
# acceder a los coeficientes del modelo
lmExp$coefficients
```

```
(Intercept)  datos1$x
0.4120421    0.5444136
```

Recuerda que el valor de  $b_0$  que hemos obtenido es el logaritmo de  $b_0$ , de modo que

```
(b0 = exp(lmExp$coefficients[1]))
```

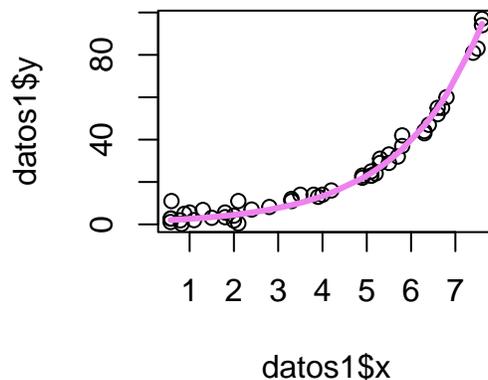
```
(Intercept)
1.509898
```

```
b1 = lmExp$coefficients[2]
```

y la expresión de la función que buscábamos es, redondeando a 3 cifras significativas,  $y = 1.509898 * e^{0.5444136*x}$

Si quieres representar la nube de puntos junto con la función que la ajusta:

```
plot(datos1$x, datos1$y)
par(new = TRUE)
lines(datos1$x, b0*exp(b1*datos1$x), col = "violet", lwd = 3)
```

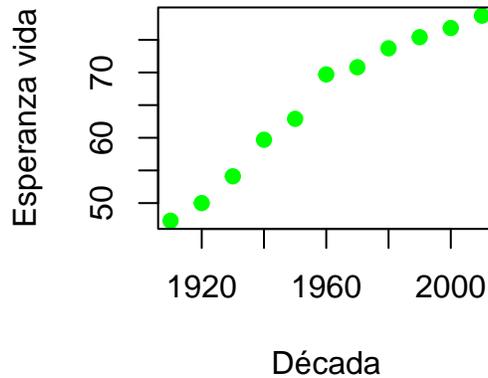


**Ejercicio 5** Ejercicio basado en este material. Este fichero contiene datos de la evolución de la esperanza de vida a lo largo de varias décadas.

- **Elige las variables explicativa y respuesta.** Parece razonable expresar la esperanza de vida en función del año y no al revés. De todos modos, como correlación no implica causalidad, se puede buscar la relación contraria (aunque no podremos responder la última pregunta).
- **Representa el diagrama de dispersión.**

Si llamamos `df4` a la tabla de datos, basta con hacer

```
plot(df4$decada, df4$esperanza_vida,
     pch = 19, col = "green",
     xlab = "Década", ylab = "Esperanza vida")
```



- **Establece el/los modelo(s) candidato(s) a ajustar esos datos.** El diagrama anterior sugiere los modelos logarítmico

$$y = b_0 + b_1 \log(x) \Rightarrow \text{cambio} \Rightarrow w = \log(x)$$

o hiperbólico

$$y = \frac{b_0}{1 + b_1 x} \Rightarrow \text{cambio} \Rightarrow w = 1/x, z = 1/x$$

- **Selecciona el mejor modelo y calcula los coeficientes del mismo.** Como en otras ocasiones, se calculan los correspondientes coeficientes de determinación (calculado sobre los datos linealizados, transformados de forma que puedan ser ajustados por una recta) para hacer la selección del modelo. En el caso logarítmico

```
cor(log(df4$decada), df4$esperanza_vida)^2
```

```
[1] 0.9624487
```

Para la hipérbola

```
cor(1/df4$decada, 1/df4$esperanza_vida)^2
```

```
[1] 0.9194304
```

No cuesta nada calcular  $r^2$  para el resto de modelos que conocemos

```
# lineal
cor(df4$decada, df4$esperanza_vida)^2
```

```
[1] 0.9600881
```

```
# exponencial
cor(df4$decada, log(df4$esperanza_vida))^2
```

```
[1] 0.938958
```

```
# potencia
cor(log(df4$decada), log(df4$esperanza_vida))^2
```

```
[1] 0.9420928
```

El logarítmico arroja un coeficiente algo mayor. Ten en cuenta que se trata del “fragmento” de curva que mejor aproxima los datos, aunque pueda no tener sentido biológico a largo plazo. Es decir, la

esperanza de vida, ¿está acotada superiormente (hipérbola) o no (logaritmo)? La regresión no responde esas preguntas :) Calculamos los coeficientes del modelo linealizado

```
lm4 = lm(df4$esperanza_vida ~ log(df4$decada))
(b0 = unname(lm4$coefficients[1]))
```

```
[1] -4856.402
```

```
(b1 = unname(lm4$coefficients[2]))
```

```
[1] 649.2619
```

En este caso (el logarítmico) se obtienen directamente los valores de  $b_0$  y  $b_1$ , de modo que el modelo que hemos obtenido es

$$RCC = -4856.4020573 + 649.2618715 * \log(decada)$$

- **Calcula la esperanza de vida predicha por el modelo para el año 1974.** Basta sustituir en la fórmula anterior

```
b0+ b1*log(1974)
```

```
[1] 70.07835
```

No tiene sentido usar todos esos decimales, es suficiente con decir, or ejemplo, casi 70 años y medio.

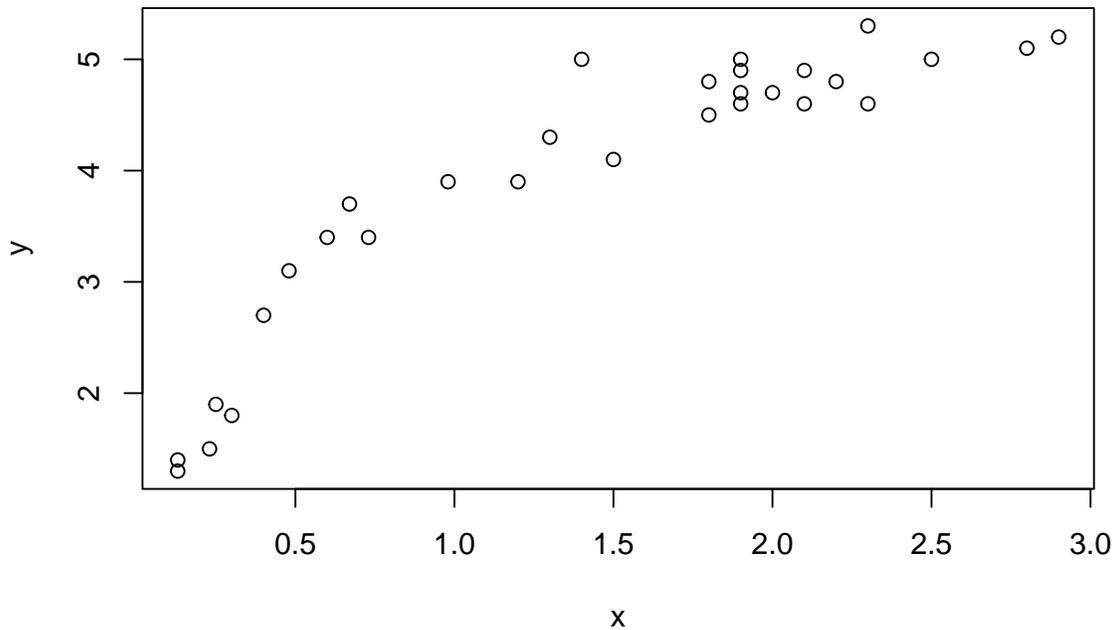
#### Ejercicio 6 Lee este fichero de datos. Se pide:

- **Representa el diagrama de dispersión.** En primer lugar, hay que leer los datos

```
datos2 = read.table(file = "Practica03-nolineal2.csv", sep = " ", header = T, dec = ".")
```

para poder hacer el diagrama de dispersión:

```
x = datos2$x
y = datos2$y
plot(x, y)
```



- **Establece el/los modelo(s) candidato(s) a ajustar esos datos.** los candidatos son, o bien una función logarítmica

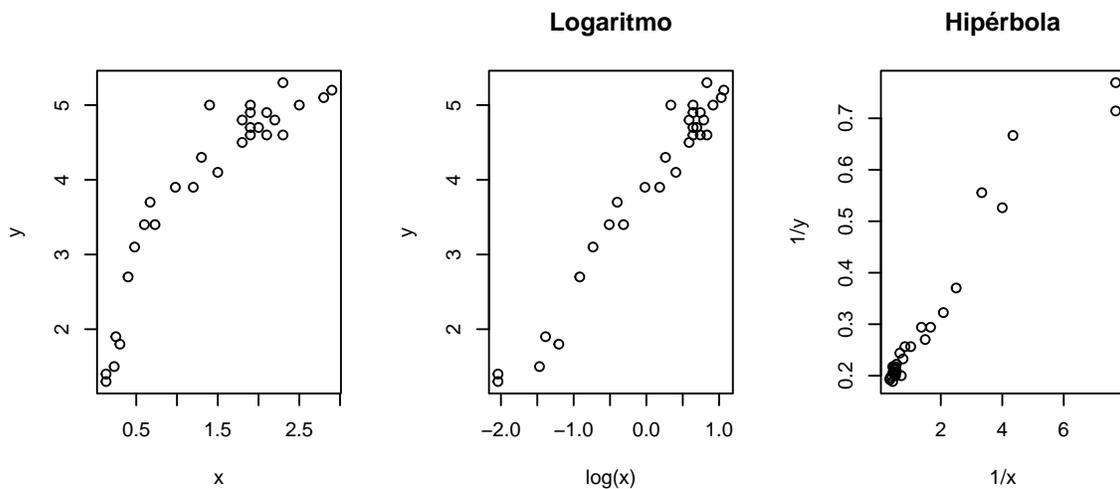
$$y = b_0 + b_1 \log(x)$$

o bien una hipérbola

$$y = \frac{b_0 x}{1 + b_1 x}$$

- **Selecciona el mejor modelo y calcula los coeficientes del mismo.** Usaremos el modelo que tenga un coeficiente de determinación mayor, es decir, aquel que explique en mayor medida (proporción) la variabilidad conjunta de las variables involucradas. Para hacer el cálculo hay que “linealizar” los datos, es decir, aplicar una transformación (o cambio de variable). Par la logarítmica hay que trabajar con  $w = \log(x)$ ,  $y$ , mientras que para una hipérbola es  $z = 1/x$ ,  $w = 1/y$ . Fíjate en los diagramas de dispersión

```
par(mfrow = c(1,3))
plot(x, y)
plot(log(x), y, main = "Logaritmo")
plot(1/x, 1/y, main = "Hipérbola")
```



```
par(mfrow = c(1,1))
```

Ahora los gráficos no parecen nada claros debido a los valores extremos a la derecha del gráfico de la derecha. Los coeficientes de determinación (que son los de correlación al cuadrado) son

```
# logaritmica
cor(y, log(x))^2
```

```
[1] 0.9589466
```

```
# hiperbolica
cor(1/x, 1/y)^2
```

```
[1] 0.9441914
```

De nuevo, calculamos el  $r^2$  del resto de modelos

```
# lineal
cor(y, x)^2
```

```
[1] 0.8270014
```

```
# exponencial
cor(y, log(x))^2
```

```
[1] 0.9589466
```

```
# potencia
cor(log(x), log(y))^2
```

```
[1] 0.9380373
```

Efectivamente, los coeficientes de correlación aclaran la duda en favor del modelo logarítmico. Calculamos los coeficientes de la recta de regresión de las variables linealizadas

```
z = y
w = log(x, base = exp(1))
lmLOG = lm(z ~ w)
lmLOG$coefficients
```

```
(Intercept)          w
  3.869179      1.319270
```

Recuerda que en este caso  $b_0$  y  $b_1$  no han sufrido transformación alguna, por lo que podemos recuperarlos directamente

```
(b0 = unname(lmLOG$coefficients[1]))
```

```
[1] 3.869179
```

```
(b1 = unname(lmLOG$coefficients[2]))
```

```
[1] 1.31927
```

y la expresión de la función que buscábamos es, redondeando a 3 cifras significativas,

$$y = 3.87 + 1.32 \ln(x)$$

donde  $\ln$  representa el logaritmo neperiano. Si quieres representar la nube de puntos junto con la función que la ajusta:

```
plot(x, y)
par(new = TRUE)
lines(x, b0 + b1*log(x, base = exp(1)), col = "violet", lwd = 3)
```

