Tema 10 - Introducción a la regresión lineal

Juan Gabriel Gomila & María Santos

Introducción

Seguramente, en algún momento de vuestra vida ya sea hojeando un libro de matemáticas, curioseando artículos científicos... habréis visto una línea recta o algún otro tipo de curva en un gráfico que se ajusta a las observaciones representadas por medio de puntos en el plano.

En general, la situación es la siguiente: supongamos que tenemos una serie de puntos en el plano cartesiano $\mathbb{R}^2$, de la forma

$$(x_1, y_1),\ \ldots,\ (x_n, y_n)$$

que representan las observaciones de dos variables numéricas. Digamos que x es la edad e y el peso de n estudiantes.

Introducción

Nuestro objetivo: describir la relación entre la variable independiente, x, y la variable dependiente, y, a partir de estas observaciones.

Para ello, lo que haremos será buscar una función $y = f(x)$ cuya gráfica se aproxime lo máximo posible a nuestros pares ordenados $(x_i, y_i)_{i=1,\ldots,n}$.

Esta función nos dará un modelo matemático de cómo se comportan estas observaciones, lo cual nos permitirá entender mejor los mecanismos que relacionan las variables estudiadas o incluso, nos dará la oportunidad de hacer predicccciones sobre futuras observaciones.

Introducción

La primera opción es la más fácil. Consiste en estudiar si los puntos $(x_i, y_i)_{i=1,\dots,n}$ satisfacen una relación lineal de la forma

$$y = ax + b$$

con $a, b \in \mathbb{R}$.

En este caso, se busca la recta $y = ax + b$ que mejor aproxime los puntos dados imponiendo que la suma de los cuadrados de las diferencias entre los valores y_i y sus aproximaciones $\tilde{y}_i = ax_i + b$ sea mínima. Es decir, que

$$\sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

sea mínima

Introducción

El objetivo de este tema no es otro más que enseñaros como hacer uso de R para obtener esta recta de regresión.

Veremos también cómo se puede evaluar numéricamente si esta recta se ajusta bien a las observaciones dadas.

Para ello, introduciremos algunas funciones de R y haremos uso de transformaciones logarítmicas para tratar casos en los que los puntos dados se aproximen mejor mediante una función exponencial o potencial.

Calculando rectas de regresión

Planteamiento del problema

Como ya hemos dicho, el objetivo de este tema es estudiar si existe relación lineal entre las variables dependiente e independiente.

Por lo general, cuando tenemos una serie de observaciones emparejadas, $(x_i, y_i)_{i=1,\ldots,n}$, la forma natural de almacenarlas en R es mediante una tabla de datos. Y la que más conocemos nostros es el data frame.

Como recordaréis de temas anteriores, la ventaja de trabajar con este tipo de organización de datos es que luego se pueden hacer muchas cosas.

Ejemplo 1

Ejemplo 1

En este ejemplo, nosotros haremos uso del siguiente data frame:

```
body = read.table("../data/bodyfat.txt", header = TRUE)
head(body,3)
```

```
  Density  Fat Age Weight Height Neck Chest Abdomen  Hip Th
1  1.0708 12.3  23 154.25  67.75 36.2  93.1    85.2 94.5  5
2  1.0853  6.1  22 173.25  72.25 38.5  93.6    83.0 98.7  5
3  1.0414 25.3  22 154.00  66.25 34.0  95.8    87.9 99.2  5
  Biceps Forearm Wrist
1   32.0    27.4  17.1
2   30.5    28.9  18.2
3   28.8    25.2  16.6
```

Ejemplo 1

Más concretamente, trabajaremos con las variables `fat` y `weight`.

```
body2 = body[,c(2,4)]
names(body2) = c("Grasa","Peso")
str(body2)
```

```
'data.frame':   252 obs. of  2 variables:
 $ Grasa: num  12.3 6.1 25.3 10.4 28.7 20.9 19.2 12.4 4.1 1
 $ Peso : num  154 173 154 185 184 ...
```

```
head(body2,3)
```

```
  Grasa   Peso
1  12.3 154.25
2   6.1 173.25
3  25.3 154.00
```

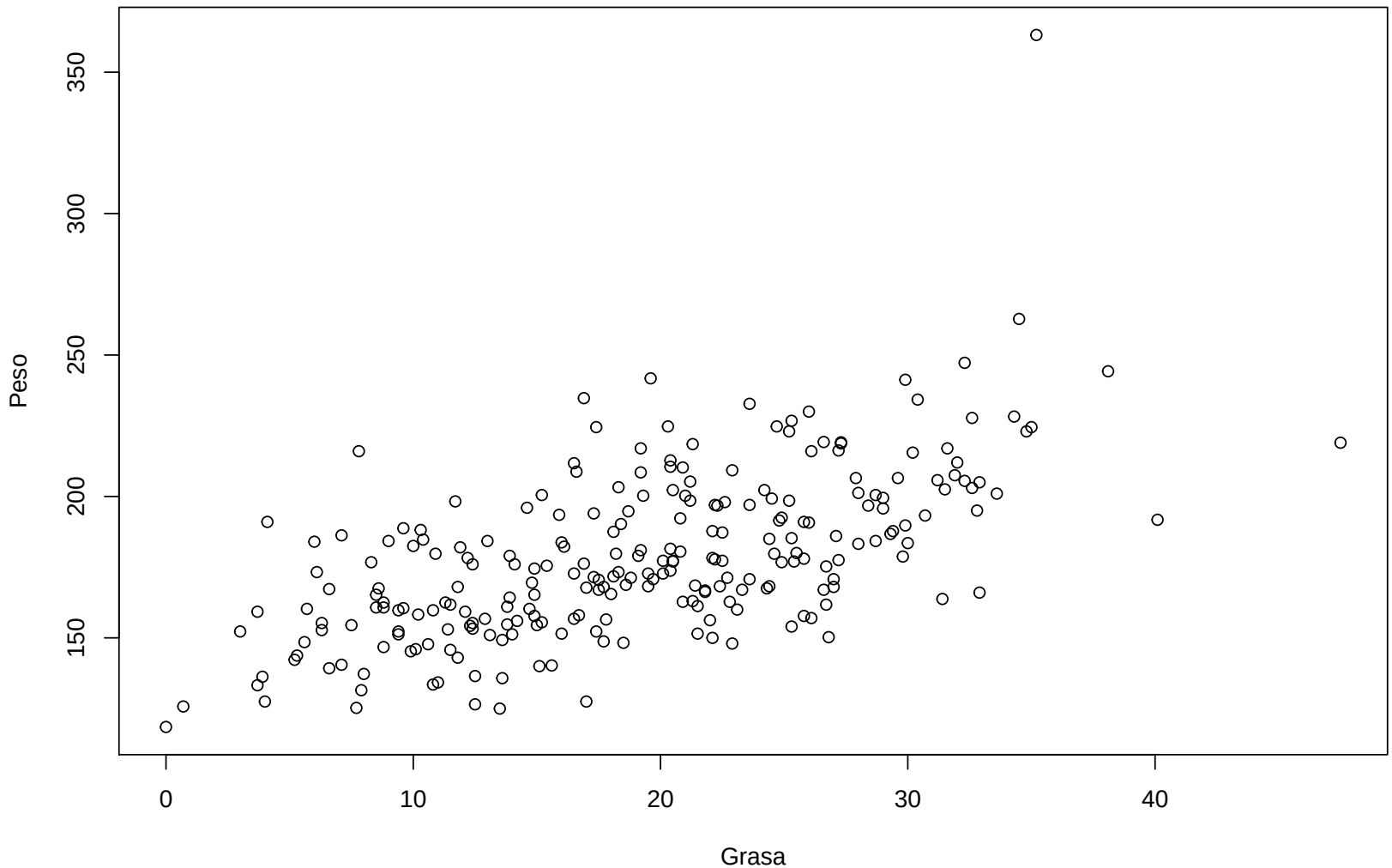
Representación gráfica

Al analizar datos, siempre es recomendable empezar con una representación gráfica que nos permita hacernos a la idea de lo que tenemos.

Esto se consigue haciendo uso de la función `plot`, que ya hemos estudiado en detalle en lecciones anteriores. No obstante, para lo que necesitamos en este tema nos conformamos con un gráfico básico de estos puntos que nos muestre su distribución.

Ejemplo 1

```
plot(body2)
```



Calculando la recta de regresión

Para calcular la recta de regresión con R de la familia de puntos $(x_i, y_i)_{i=1,\ldots,n}$, si `x` es el vector $(x_i)_{i=1,\ldots,n}$ e `y` es el vector $(y_i)_{i=1,\ldots,n}$, entonces, su recta de regresión se calcula mediante la instrucción

```
lm(y~x)
```

Cuidado con la sintaxis: primero va el vector de las variables dependientes y, seguidamente después de una tilde `~`, va el vector de las variables independientes.

Esto se debe a que R toma el significado de la tilde como "en función de". Es decir, la interpretación de `lm(y~x)` en R es "la recta de regresión de y en función de x".

Calculando la recta de regresión

Si los vectores `y` y `x` son, en este orden, la primera y la segunda columna de un data frame de dos variables, entonces es suficiente aplicar la función `lm` al data frame.

En general, si `x` e `y` son dos variables de un data frame, para calcular la recta de regresión de `y` en función de `x` podemos usar la instrucción

```
lm(y~x, data = data fame)
```

Ejemplo 1

```r
lm(body2$Peso~body2$Grasa) #Opción 1
```

```
Call:
lm(formula = body2$Peso ~ body2$Grasa)

Coefficients:
(Intercept)  body2$Grasa
    137.738         2.151
```

```r
lm(Peso~Grasa, data = body2) #Opción 2
```

```
Call:
lm(formula = Peso ~ Grasa, data = body2)

Coefficients:
(Intercept)        Grasa
    137.738         2.151
```

Ejemplo 1

Como podéis observar, las dos formas de llamar a la función dan exactamente lo mismo. Ninguna es mejor que la otra.

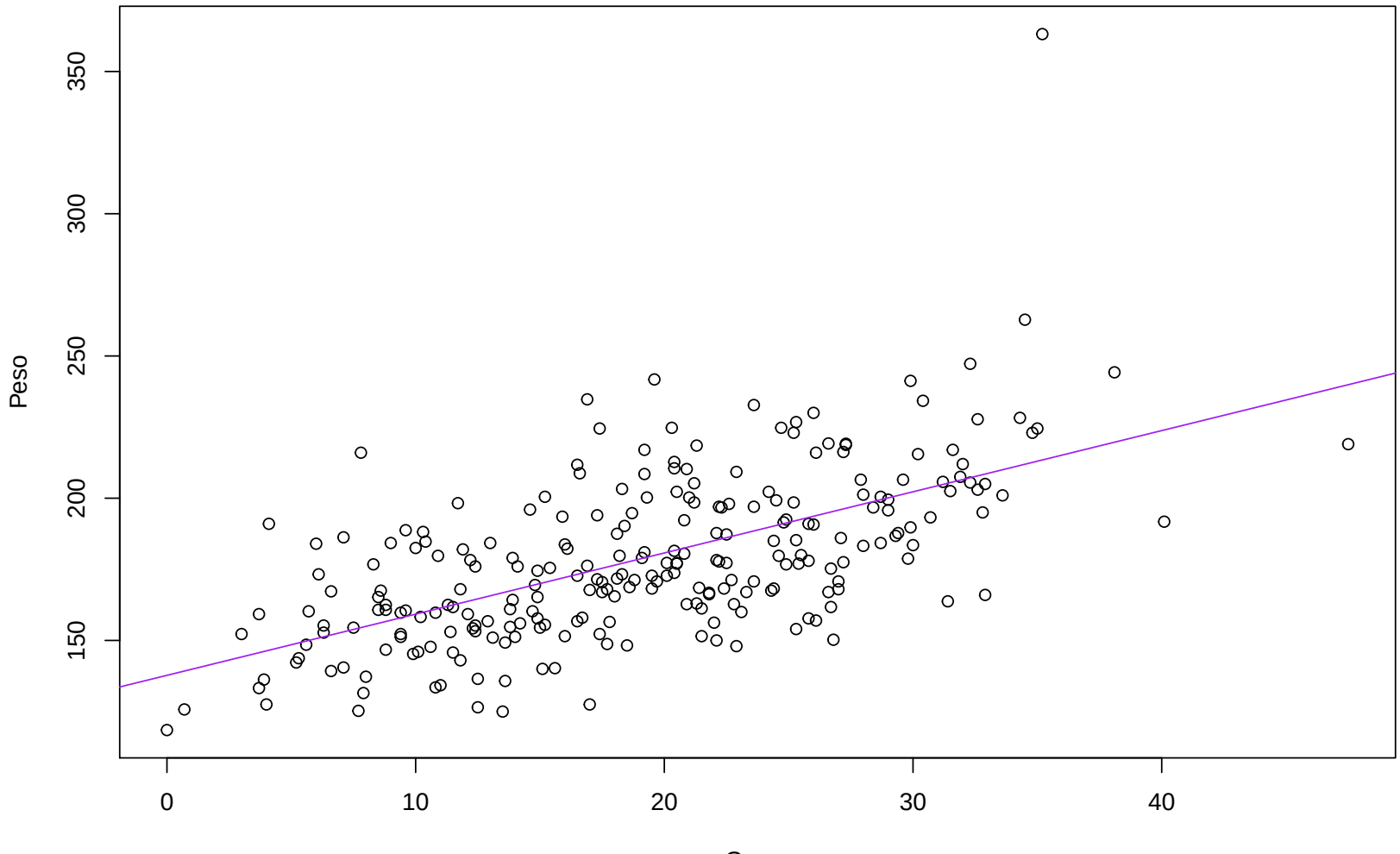
El resultado obtenido en ambos casos significa que la recta de regresión para nuestros datos es

$$y = 2.151x + 137.738$$

Ahora, podemos superponer esta recta a nuestro gráfico anterior haciendo uso de la función `abline()`.

Ejemplo 1

```
plot(body2)
abline(lm(Peso~Grasa, data = body2), col = "purple")
```



Observación

Hay que tener en cuenta que el análisis llevado a cabo hasta el momento de los pares de valores $(x_i, y_i)_{i=1,\ldots,n}$ ha sido puramente descriptivo.

Es decir, hemos mostrado que estos datos son consistentes con una función lineal, pero no hemos demostrado que la variable dependiente sea función aproximadamente lineal de la variable dependiente. Esto último necesitaría una demostración matemática, o bien un argumento biológico, pero no basta con una simple comprobación numérica.

Haciendo predicciones

Eso sí, podemos utilizar todo lo hecho hasta ahora para predecir valores $\tilde{y}_i$ en función de los x_i resolviendo una simple ecuación lineal

Coeficiente de determinación

El coeficiente de determinación, R^2, nos es útil para evaluar numéricamente si la relación lineal obtenida es significativa o no.

No explicaremos de momento como se define. Eso lo dejamos para curiosidad del usuario. Por el momento, es suficiente con saber que este coeficiente se encuentra en el intervalo $[0, 1]$. Si R^2 es mayor a 0.9, consideraremos que el ajuste es bueno. De lo contrario, no.

La función summary

La función `summary` aplicada a `lm` nos muestra los contenidos de este objeto. Entre ellos encontramos `Multiple R-squared`, que no es ni más ni menos que el coeficiente de determinación, R^2.

Para facilitarnos las cosas y ahorrarnos información que, de momento, no nos resulta de interés, podemos aplicar `summary(lm(...))$r.squared`

Ejemplo 1

```
summary(lm(Peso~Grasa, data = body2))
```

```
Call:
lm(formula = Peso ~ Grasa, data = body2)

Residuals:
    Min      1Q  Median      3Q     Max
-46.799 -14.999  -3.469  11.860 149.709

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 137.7375     3.6684   37.55   <2e-16 ***
Grasa         2.1507     0.1756   12.25   <2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’

Residual standard error: 23.28 on 250 degrees of freedom
Multiple R-squared:  0.3751,    Adjusted R-squared:  0.3726
```

Ejemplo 1

```
summary(lm(Peso~Grasa, data = body2))$r.squared
```

```
[1] 0.3750509
```

En este caso, hemos obtenido un coeficiente de determinación de 0.3751, cosa que confirma que la recta de regresión no aproxima nada bien nuestros datos.

Rectas de regresión y transformaciones logarítmicas

Transformaciones logarítmicas

No siempre encontraremos dependencias lineales. A veces nos encontraremos otro tipo de dependencias, como por ejemplo pontencias o exponenciales.

Estas se pueden transformar a lineales mediante un cambio de escala

Escalas logarítmicas

Por lo general, es habitual encontrarnos gráficos con sus ejes en escala lineal. Es decir, las marcas en los ejes están igualmente espaciadas.

A veces, es conveniente dibujar alguno de los ejes en escala logarítmica, de modo que la misma distancia entre las marcas significa el mismo cociente entre sus valores. En otras palabras, un eje en escala logarítmica representa el logaritmo de sus valores en escala lineal.

Diremos que un gráfico está en escala semilogarítmica cuando su eje de abcisas está en escala lineal y, el de ordenadas, en escala logarítmica.

Diremos que un gráfico está en escala doble logarítmica cuando ambos ejes están en escala logarítmica.

Interpretación gráfica

Si al representar unos puntos $(x_i, y_i)_{i=1,\ldots,n}$ en escala semilogarítmica observamos que siguen aproximadamente una recta, esto querrá decir que los valores $\log(y)$ siguen una ley aproximadamente lineal en los valores x, y, por lo tanto, que y sigue una ley aproximadamente exponencial en x.

En efecto, si $\log(y) = ax + b$, entonces,

$$y = 10^{\log(y)} = 10^{ax+b} = 10^{ax} \cdot 10^{b} = \alpha^x \beta$$

con $\alpha = 10^a$ y $\beta = 10^b$

Interpretación gráfica

Si al representar unos puntos $(x_i, y_i)_{i=1,\ldots,n}$ en escala doble logarítmica observamos que siguen aproximadamente una recta, esto querrá decir que los valores $\log(y)$ siguen una ley aproximadamente lineal en los valores $\log(x)$, y, por lo tanto, que y sigue una ley aproximadamente potencial en x.

En efecto, si $\log(y) = a\log(x) + b$, entonces, por propiedades de logaritmos

$$y = 10^{\log(y)} = 10^{a\log(x)+b} = (10^{\log(x)})^a \cdot 10^b = x^a\beta$$

con $\beta = 10^b$

Ejemplo 2

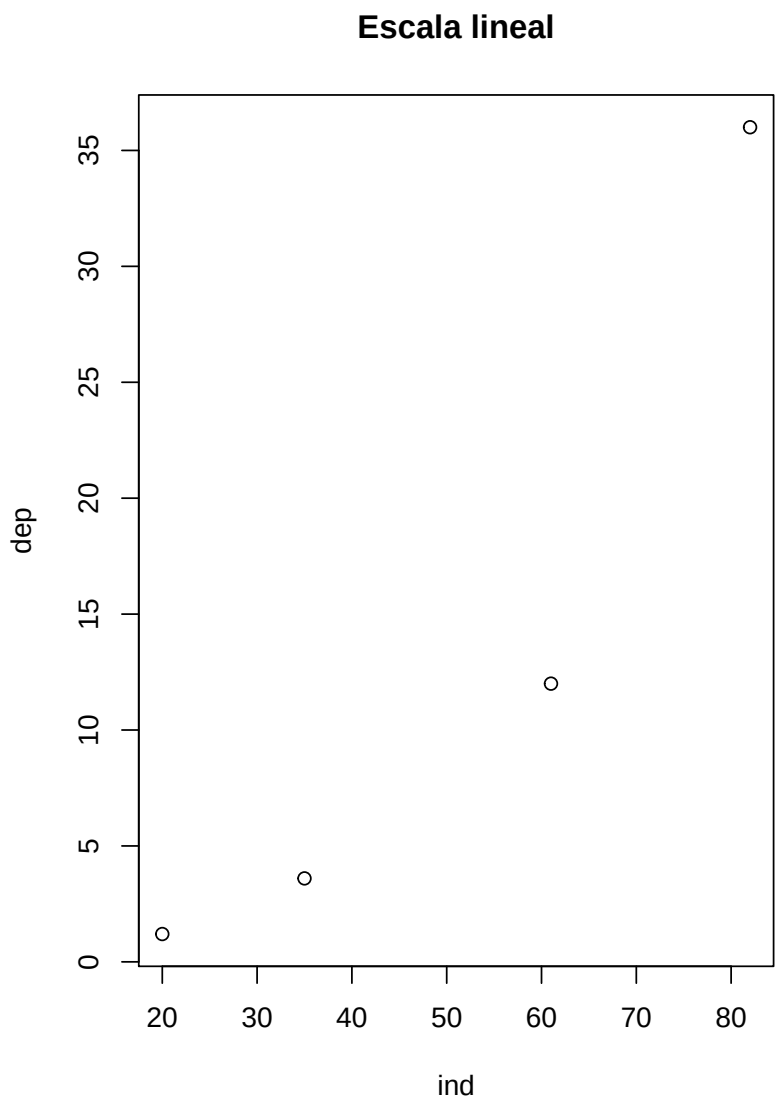
Ejemplo 2

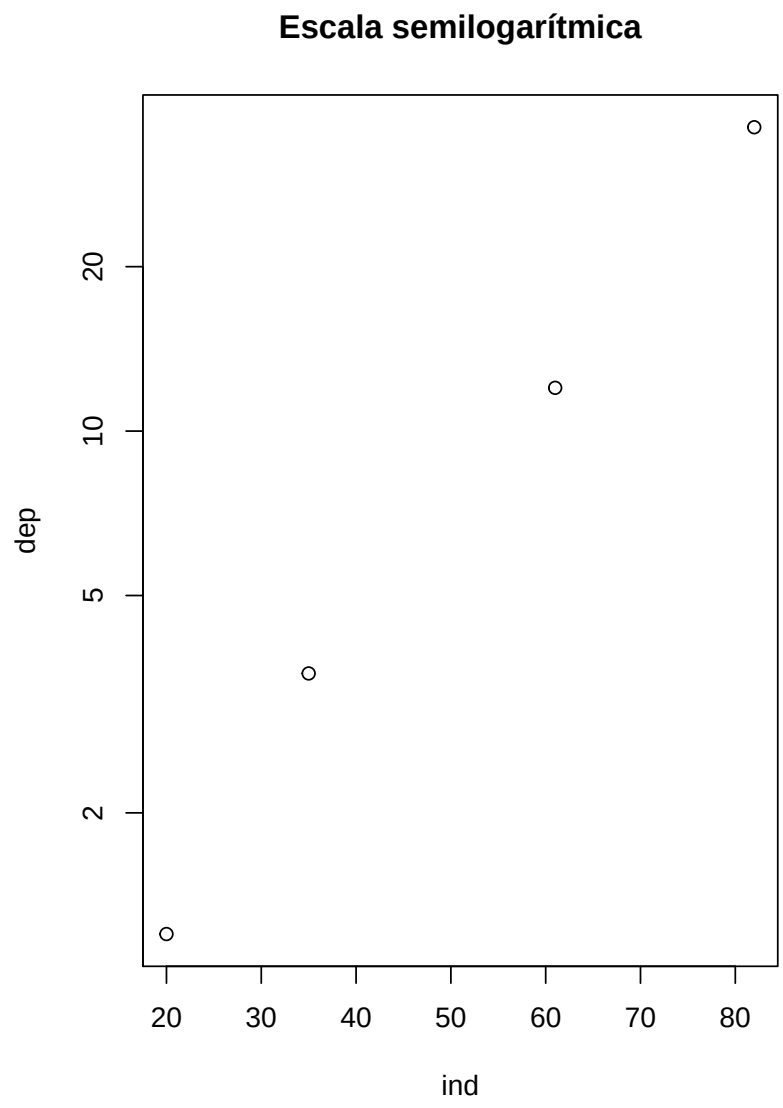
En este caso trabajaremos no con un data frame, sino directamente con los dos vectores siguientes:

```
dep = c(1.2,3.6,12,36)
ind = c(20,35,61,82)
```

```
plot(ind,dep, main = "Escala lineal")
plot(ind,dep, log = "y", main = "Escala semilogarítmica")
```

Ejemplo 2


Escala lineal
dep
ind
0
5
10
15
20
25
30
35
20
30
40
50
60
70
80


Escala semilogarítmica
dep
ind
2
5
10
20
20
30
40
50
60
70
80

Ejemplo 2

```
lm(log10(dep)~ind)
```

```
Call:
lm(formula = log10(dep) ~ ind)

Coefficients:
(Intercept)          ind
   -0.32951      0.02318
```

```
summary(lm(log10(dep)~ind))$r.squared
```

```
[1] 0.9928168
```

Ejemplo 2

Lo que acabamos de obtener es que

$$\log(dep) = 0.023 \cdot ind - 0.33$$

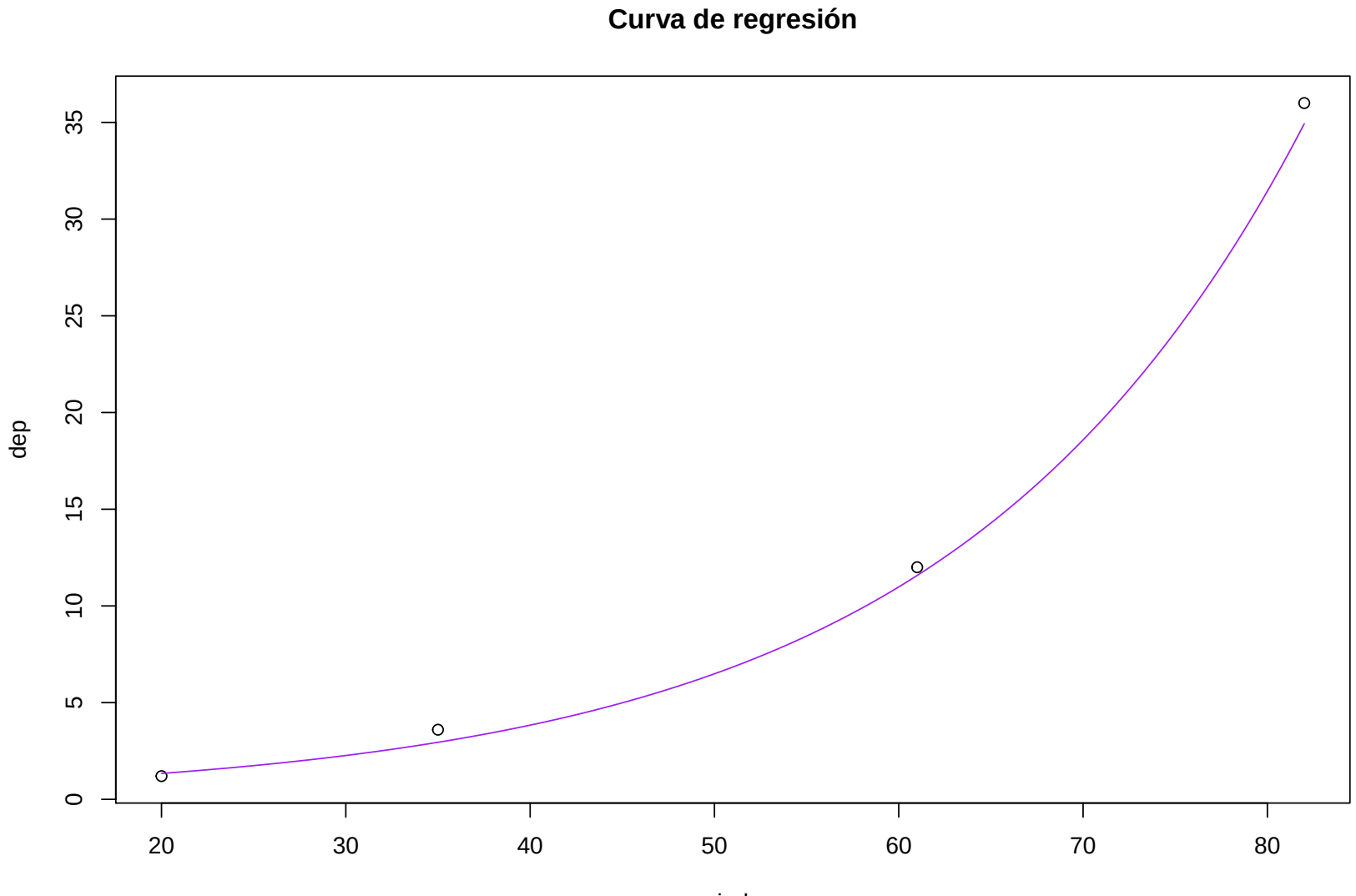
es una buena aproximación de nuestros datos.

Con lo cual

$$dep = 10^{0.023 \cdot ind} \cdot 10^{-0.33} = 1.054^{ind} \cdot 0.468$$

Ejemplo 2

```
plot(ind,dep, main = "Curva de regresión")
curve(1.054^x*0.468, add = TRUE, col = "purple")
```



Ejemplo 3

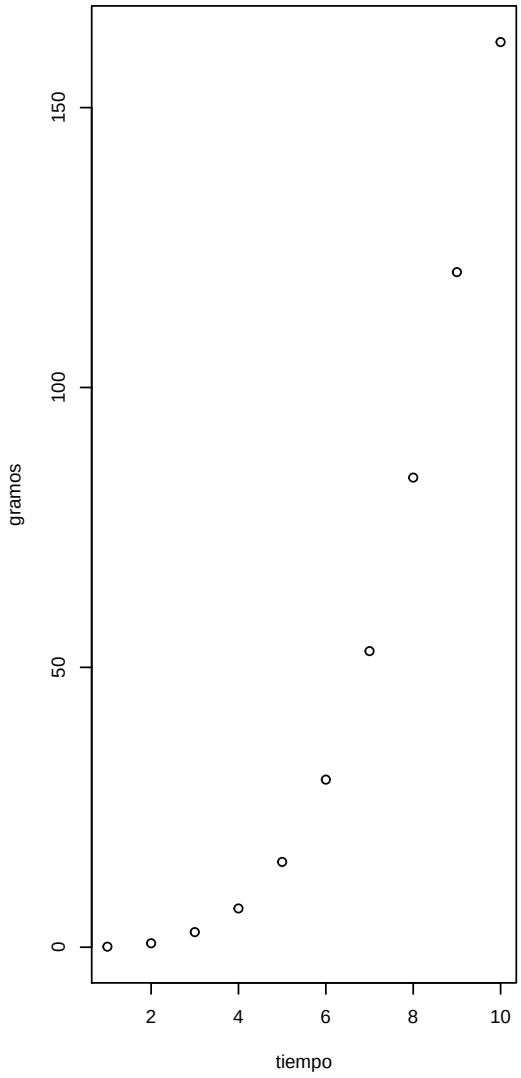
Ejemplo 3

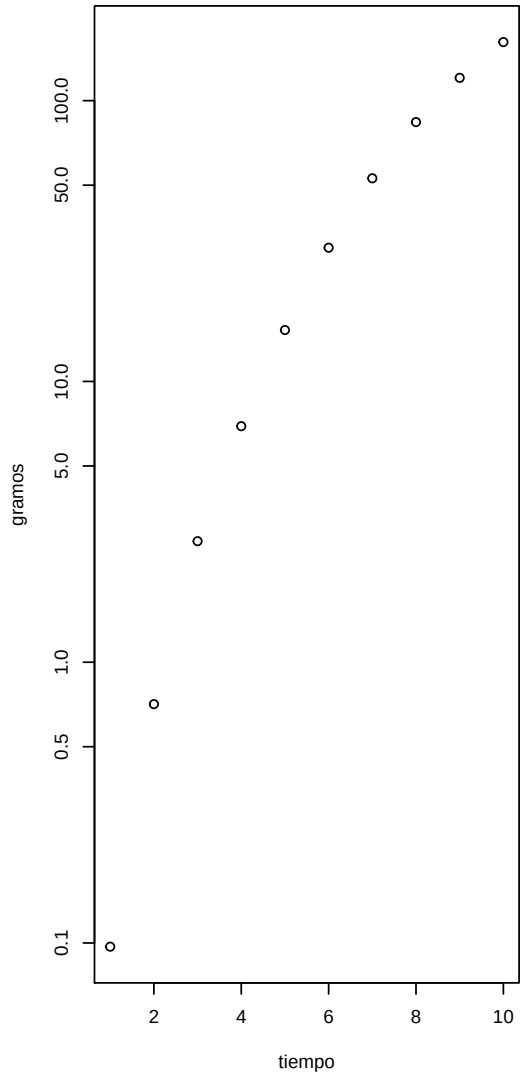
En este caso trabajaremos con el siguiente data frame:

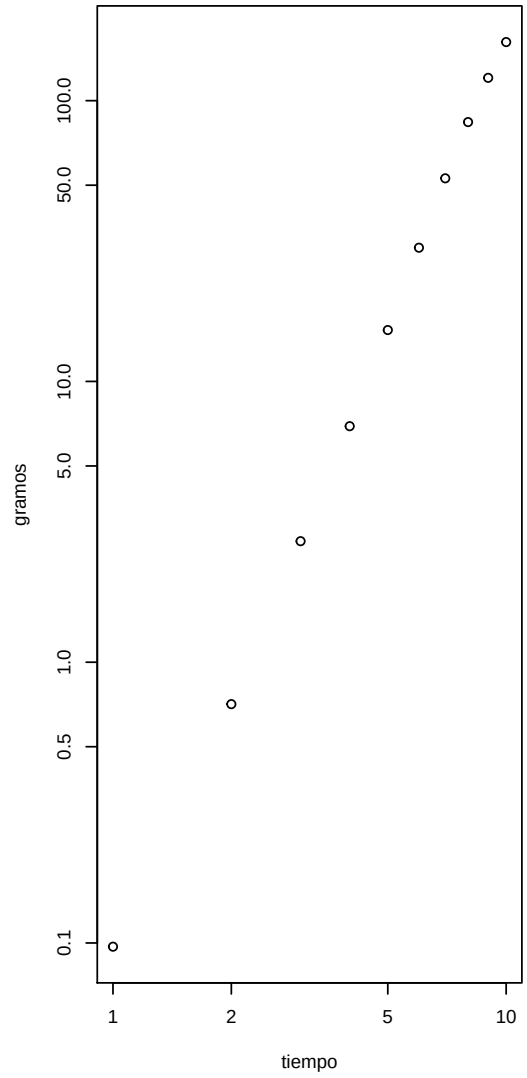
```
tiempo = 1:10
gramos = c(0.097,0.709,2.698,6.928,15.242,29.944,52.902,83.
d.f = data.frame(tiempo,gramos)
```

```
plot(d.f)
plot(d.f, log = "y")
plot(d.f, log = "xy")
```

Ejemplo 3


gramos
0
50
100
150
2
4
6
8
10
tiempo


gramos
0.1
0.5
1.0
5.0
10.0
50.0
100.0
2
4
6
8
10
tiempo


gramos
0.1
0.5
1.0
5.0
10.0
50.0
100.0
1
2
5
10
tiempo

Ejemplo 3

```
lm(log10(gramos)~log10(tiempo), data = d.f)
```

```
Call:
lm(formula = log10(gramos) ~ log10(tiempo), data = d.f)

Coefficients:
  (Intercept)  log10(tiempo)
       -1.093          3.298
```

```
summary(lm(log10(gramos)~log10(tiempo), data = d.f))$r.squa
```

```
[1] 0.9982009
```

Ejemplo 3

Lo que acabamos de obtener es que

$$\log(\mathit{gramos}) = 3.298 \cdot \log(\mathit{tiempo}) - 1.093$$

es una buena aproximación de nuestros datos.

Con lo cual

$$\mathit{gramos} = 10^{3.298 \cdot \log(\mathit{tiempo})} \cdot 10^{-1.093} = \mathit{tiempo}^{3.298} \cdot 0.081$$

Ejemplo 3

```
plot(d.f, main = "Curva de regresión")
curve(x^(3.298)*0.081, add=TRUE, col = "purple")
```