

Regresión Lineal. Ejemplo.

```
# En química analítica se usa el modelo de regresión para calibrar
# un método de medición. Para calibrar un fluorímetro se han
# examinado 7 soluciones estándar de fluoresceína (de las que se
# conoce la concentración medida con mucha precisión) en el
# fluorímetro. Los siguientes datos son las concentraciones
# y la intensidad de fluorescencia observada en el fluorímetro:
#
# Concentración, pg/ml:      0      2      4      6      8      10     12
# Intensidad de fluorescencia: 2.1   5.0   9.0   12.6  17.3  21.0   24.7

#####
# 1) Cargamos los datos a R
#####

> conc<-scan()
1: 0      2      4      6      8      10     12
8:
Read 7 items
> fluo<-scan()
1: 2.1   5.0   9.0   12.6  17.3  21.0   24.7
8:
Read 7 items
#####
# 2) Grafico de dispersion
#####

#scatter plot de los datos
> plot(conc,fluo,xlab="concentracion",ylab = "fluorescencia",
+       main="diagrama de dispersion
+       de fluorescencia vs. concentracion")
> # ponemos la concentracion en las x, como variable independiente,
> # la fluorescencia como variable dependiente, en las y

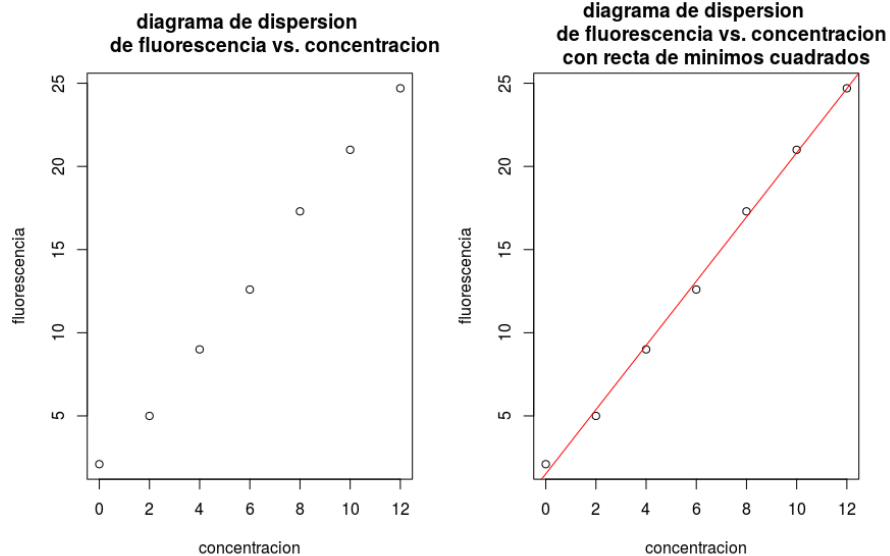
#####
# 3) Ajuste del modelo
#####
#la instruccion lm (linear model) ajusta la recta de minimos cuadrados a los datos

> salida<-lm(fluo~conc)
# el lm produce un objeto (una lista) que contiene muchas cosas
> names(salida)
[1] "coefficients" "residuals"    "effects"      "rank"         "fitted.values" "assign"
[7] "qr"           "df.residual"  "xlevels"      "call"         "terms"         "model"

# para obtener los coeficientes, hay dos formas
> salida$coefficients
(Intercept)      conc
  1.517857      1.930357
> coefficients(salida)
(Intercept)      conc
  1.517857      1.930357

#superponemos la recta de minimos cuadrados al grafico
> plot(conc,fluo,xlab="concentracion",ylab = "fluorescencia",
+       main="diagrama de dispersion
```

```
+ de fluorescencia vs. concentracion
+ con recta de minimos cuadrados")
> abline(salida,col="red")
```



```
> summary(salida)
```

```
Call:
lm(formula = fluo ~ conc)
```

```
Residuals:
    1      2      3      4      5      6      7
 0.58214 -0.37857 -0.23929 -0.50000  0.33929  0.17857  0.01786
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.5179     0.2949   5.146  0.00363 **
conc         1.9304     0.0409  47.197  8.07e-08 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.4328 on 5 degrees of freedom
Multiple R-squared:  0.9978, Adjusted R-squared:  0.9973
F-statistic: 2228 on 1 and 5 DF, p-value: 8.066e-08
```

```
#tambien es un objeto de R
```

```
> names(summary(salida))
 [1] "call"          "terms"         "residuals"     "coefficients"  "aliases"       "sigma"
 [7] "df"            "r.squared"     "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
#####
# 4) estimamos el valor esperado de la fluorescencia cuando conc = 6 y 10
#####
```

```
> # a mano
> coefficients(salida)[1]+coefficients(salida)[2]*6
(Intercept)
    13.1
> coefficients(salida)[1]+coefficients(salida)[2]*10
(Intercept)
    20.82143
```

```

> # con la instruccion predict que calcula directamente el R.
> predict(salida,newdata=data.frame(conc=6))
  1
13.1
> predict(salida,newdata=data.frame(conc=10))
  1
20.82143

> # que calcula la instruccion fitted.values?
> fitted.values(salida)
  1          2          3          4          5          6          7
1.517857  5.378571  9.239286 13.100000 16.960714 20.821429 24.682143

> # que calcula la instruccion residuals?
> residuals(salida)
  1          2          3          4          5          6          7
0.58214286 -0.37857143 -0.23928571 -0.50000000  0.33928571  0.17857143  0.01785714

> #####
> # 5) intervalos de confianza para coeficientes
> #####
> #el cuantil
> qt(0.975,df=5)
[1] 2.570582
>
> #los coeficientes estimados, y sus desvios estandares en las
> #primeras dos columnas
> summary(salida)$coef
      Estimate Std. Error  t value    Pr(>|t|)
(Intercept)  1.517857  0.29493600   5.146395 3.625829e-03
conc         1.930357  0.04090026  47.196691 8.066023e-08
>
> #los extremos superiores de los ic
> summary(salida)$coef[,1]+ summary(salida)$coef[,2]*qt(0.975,df=5)
(Intercept)      conc
  2.276014    2.035495
>
> #los extremos inferiores de los ic
> summary(salida)$coef[,1]- summary(salida)$coef[,2]*qt(0.975,df=5)
(Intercept)      conc
  0.75970    1.82522
>
> # el R calcula todo automaticamente
> confint(salida)
      2.5 %    97.5 %
(Intercept)  0.75970  2.276014
conc         1.82522  2.035495

#####
# 6) IC para E(Y) cuando X=6 y X=10
#####
> # con el siguiente comando, que calcula el estimador e IC
> # correspondiente a X=6 y a X=10
> predict(salida,newdata=data.frame(conc = 6), interval="confidence")
  fit      lwr      upr
1 13.1 12.67945 13.52055
> predict(salida,newdata=data.frame(conc = 10), interval="confidence")

```

```
      fit      lwr      upr
1 20.82143 20.22668 21.41618
```

```
#####
# 7) Intervalos de prediccion
#####
```

```
> predict(salida,newdata=data.frame(conc = 6), interval="prediction")
```

```
      fit      lwr      upr
1 13.1 11.91051 14.28949
```

```
> predict(salida,newdata=data.frame(conc = 10), interval="prediction")
```

```
      fit      lwr      upr
1 20.82143 19.55978 22.08308
```

```
>
> #####
> # 8) Validacion de supuestos
> #####
>
```

```
# Graficos para decidir si se satisfacen los supuestos de linealidad
# y homoscedasticidad
```

```
# 1) el grafico que ya tenemos
plot(conc,fluo)
abline(salida,col=2)
```

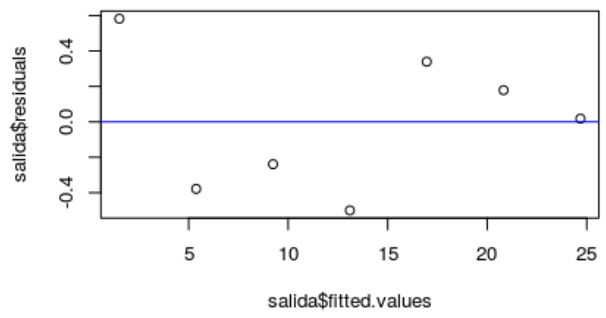
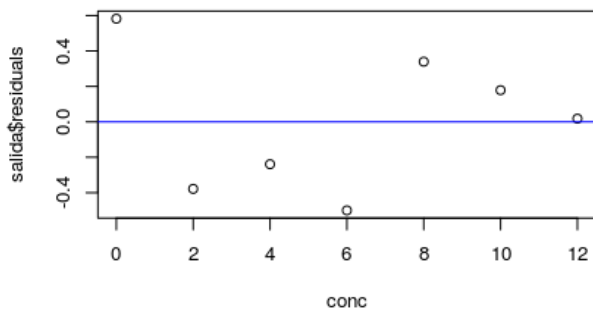
```
> mfrow=c(2,2) #para hacer cuatro graficos juntos
> # 2)
```

```
> mfrow=c(2,2) #para hacer cuatro graficos juntos
> # 2)
> plot(conc,salida$residuals)
> abline(a=0,b=0,col=4)
```

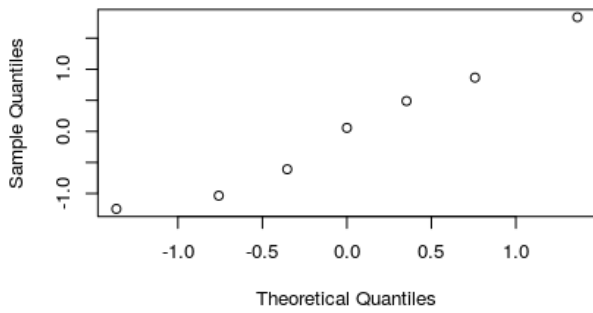
```
> #3)
> plot(salida$fitted.values,salida$residuals)
> abline(a=0,b=0,col=4)
```

```
> #4)
> qqnorm(rstandard(salida),main="qqplot de residuos
+ estandarizados")
```

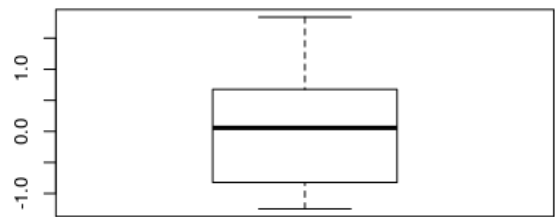
```
> #5)
> boxplot(rstandard(salida),main="boxplot de residuos
+ estandarizados")
```



qqplot de residuos estandarizados



boxplot de residuos estandarizados



```
> shapiro.test(rstandard(salida))
```

Shapiro-Wilk normality test

```
data: rstandard(salida)  
W = 0.95794, p-value = 0.8009
```