REGRESIÓN

TRABAJO EN EL LABORATORIO

BOOK: Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani

An Introduction to Statistical Learning with Applications in R

Springer, 2013

Chapter 07

Presentación

- Nombre: Raúl Pérez

- Profesor del Dpto. Ciencias de la Computación e I.A.
- Despacho: n. 23 de la planta 4 de la ETSIIT

- email: fgr@decsai.ugr.es

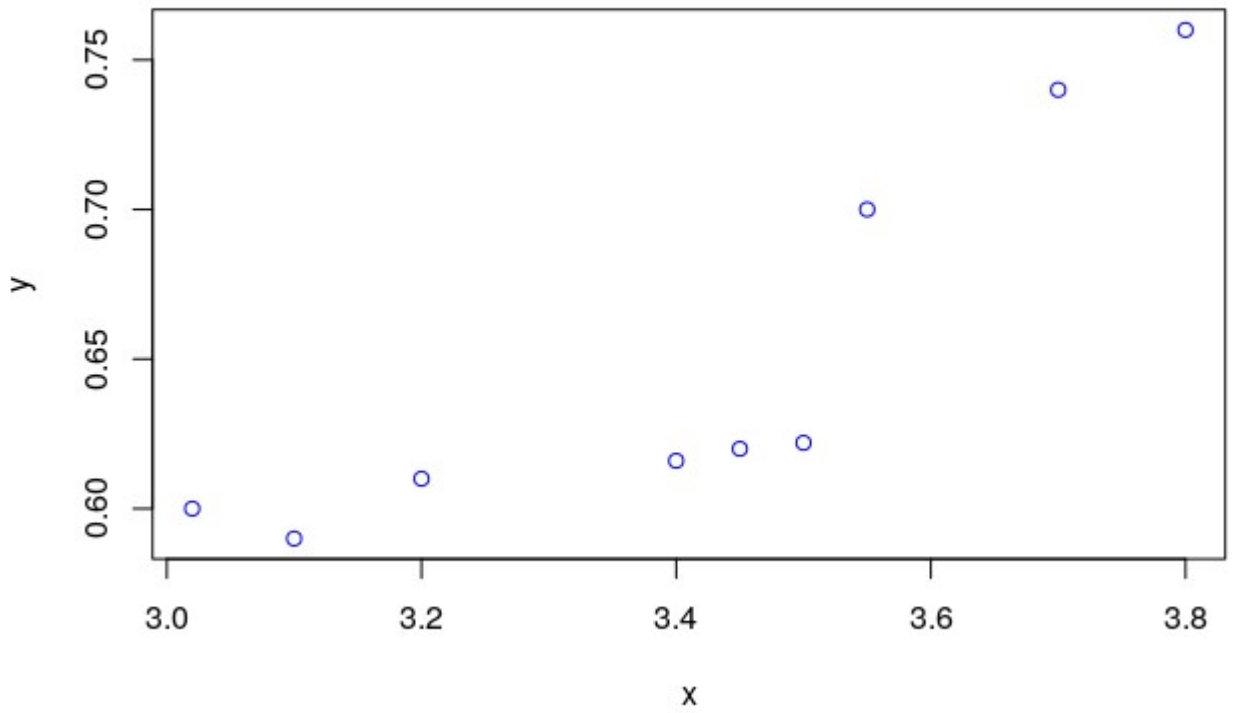
- Tutorías:
  - Lunes de 10:00 a 13:30
  - Martes de 10:00 a 12:30
- Clases:
  - Martes 5 de Diciembre
  - Miércoles 13 de Diciembre
  - Martes 19 de Diciembre

Regresión lineal, multi-lineal y no lineal

- Regresión lineal y multi-lineal
  - Algunos parámetros para determinar la bodad de la aproximación
  - Ejercicio 1
- Regresión no lineal linealizable
  - Modelo polinomial poly()
  - Ejercicio 2
  - Regresión con spline cúbicos
  - Ejercicio 3
- Regresión no lineal
  - Modelo aditivo general gam()
  - Ejercicio 4
- Ejercicio de Repaso

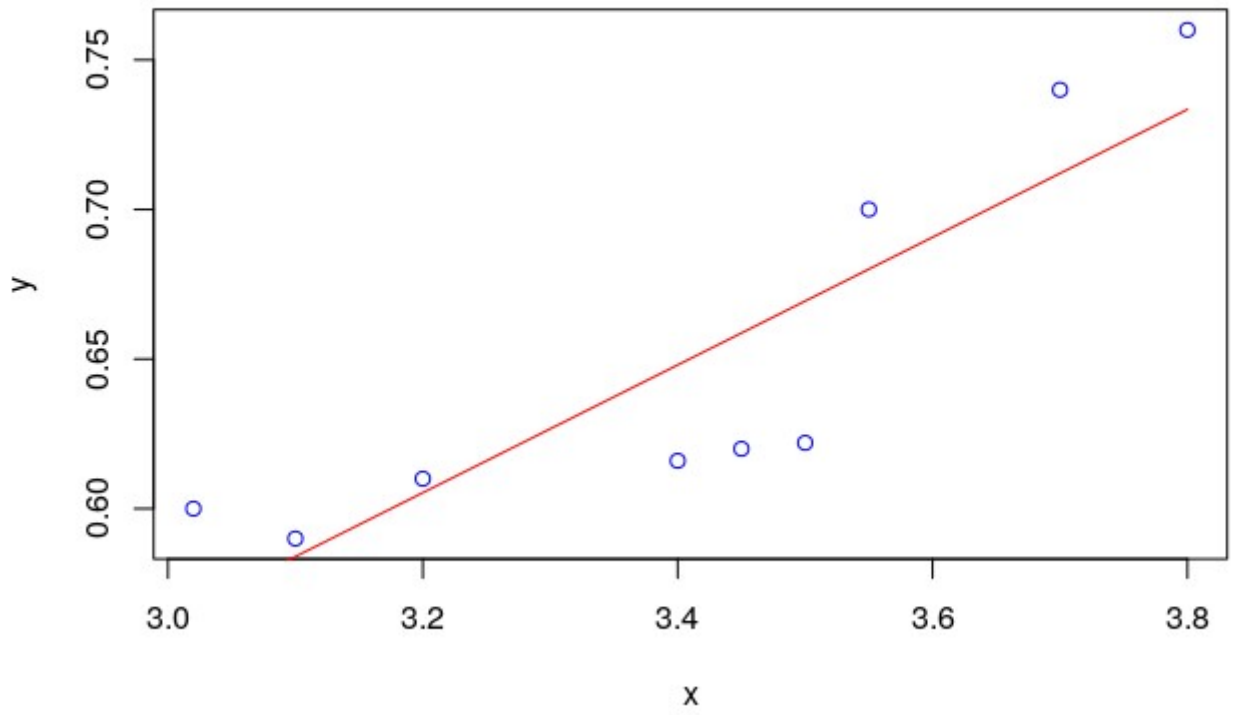
Introducción

- Cuando uno piensa en un problema de regresión, normalmente se tiene una imagen parecida a esta.



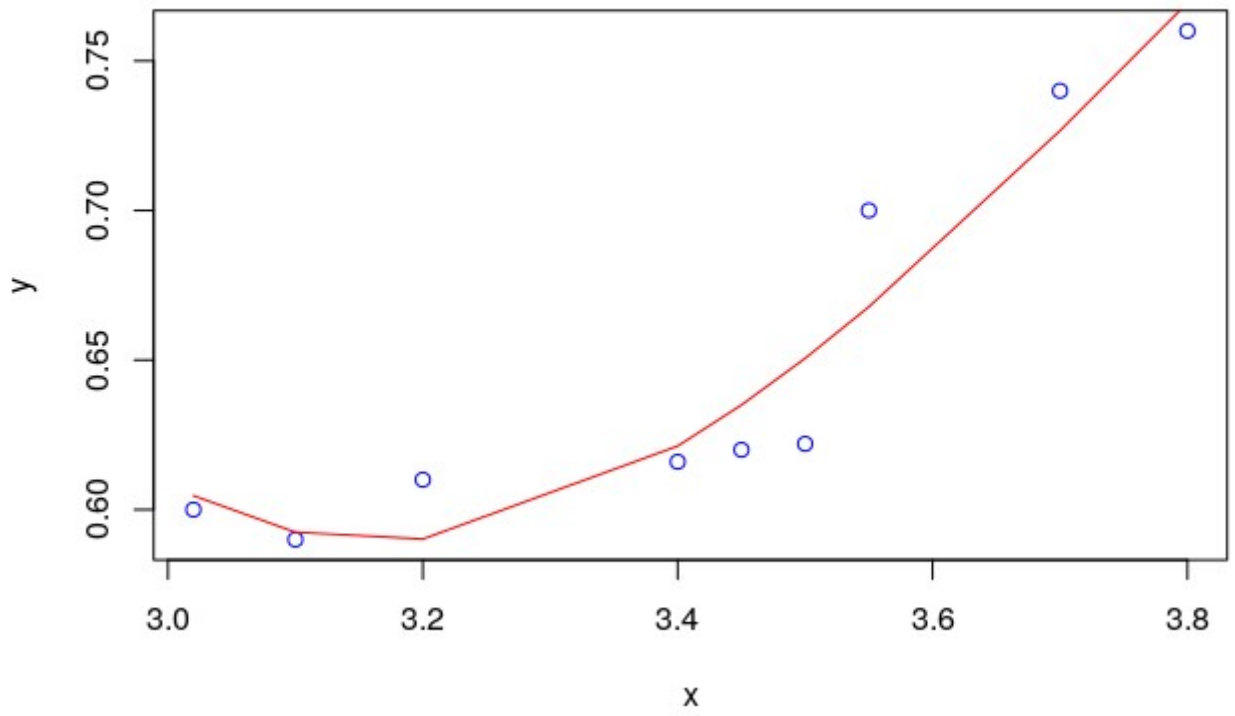
Introducción

- y sobre esos puntos o bien define una recta que pase lo más cerca posible de todos los puntos …



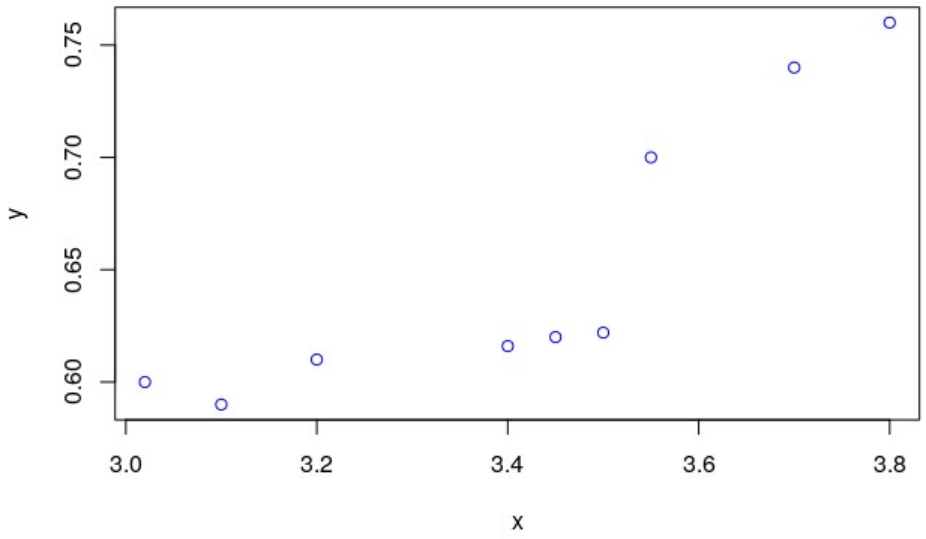
Introducción

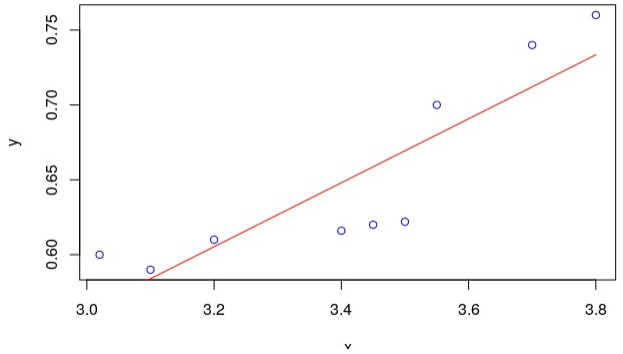
- o bien una curva …

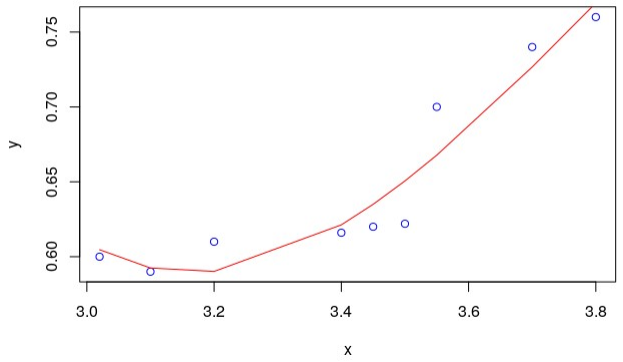


Introducción

- y uno siempre se queda con la sensación con que la aproximación no es lo suficientemente buena.

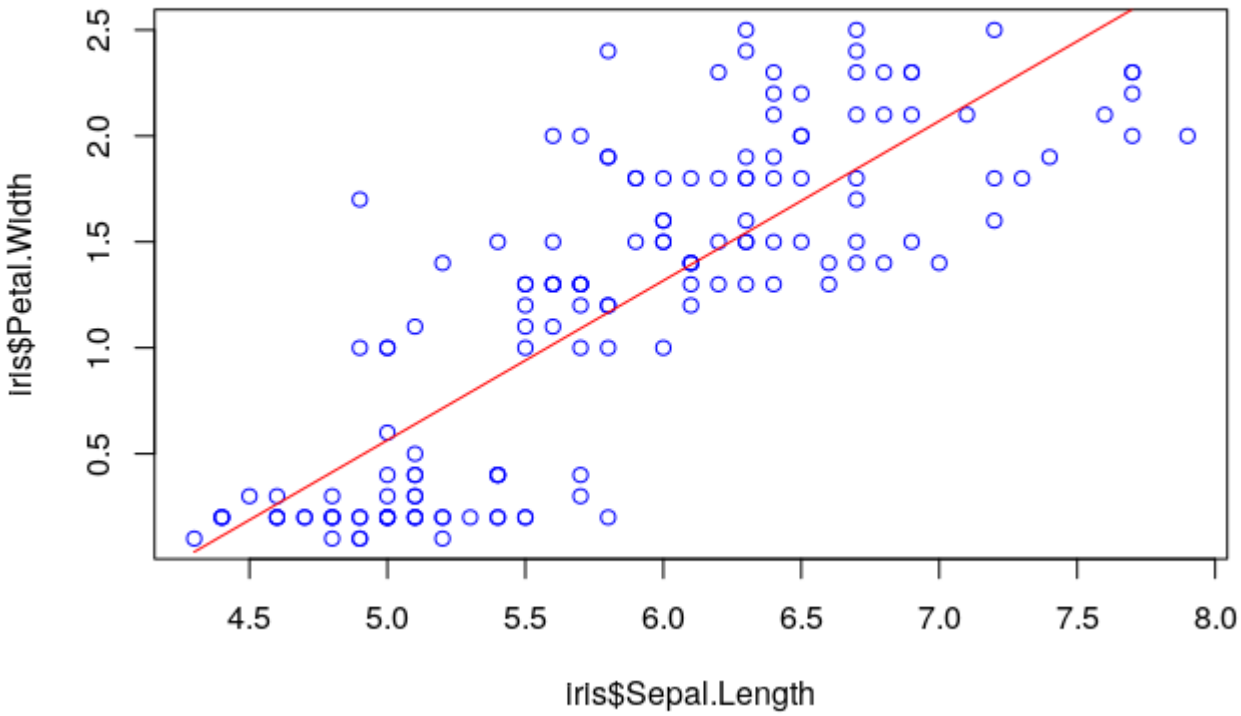






Introducción

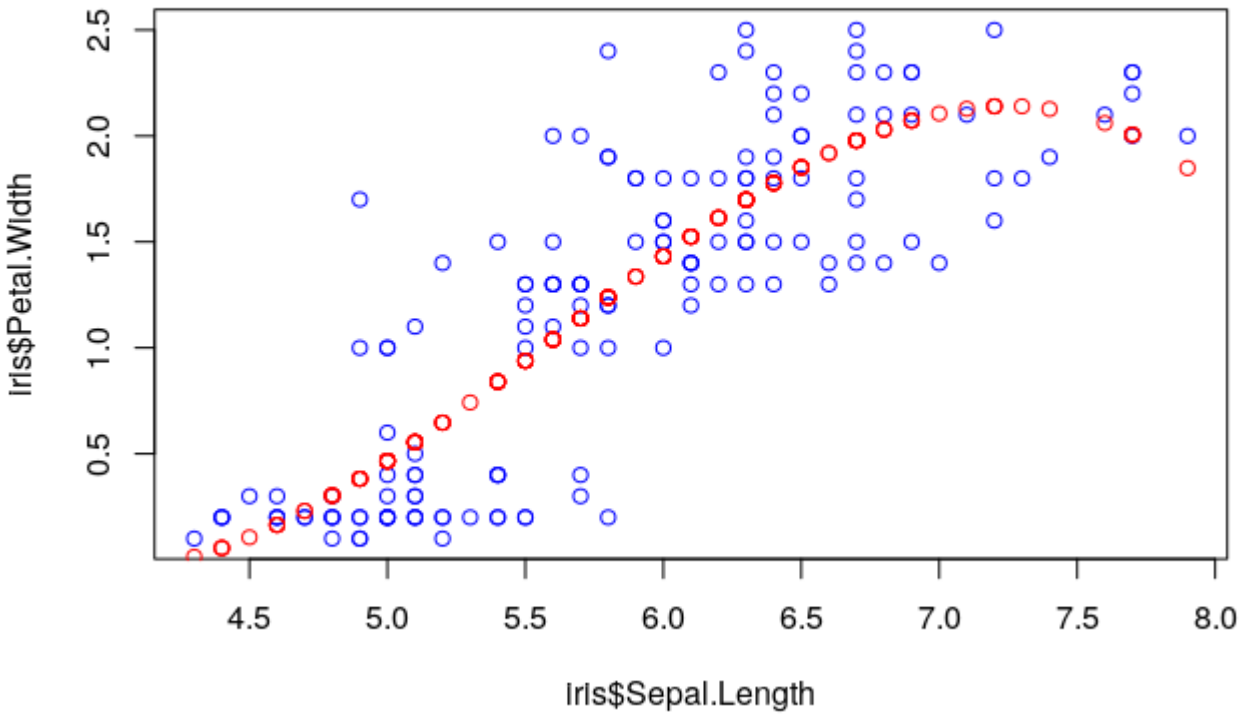
- En un problema real, la regresión se enfrenta a cosas como …



- dónde aquí si que parece que no hay una aproximación buena.

Introducción

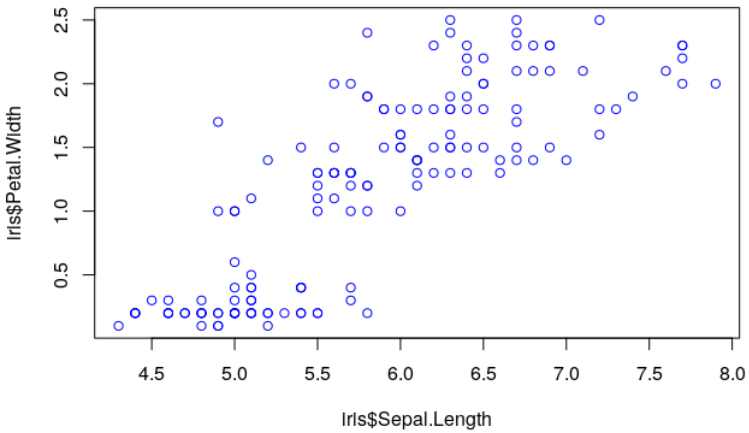
- En un problema real, la regresión se enfrenta a cosas como …

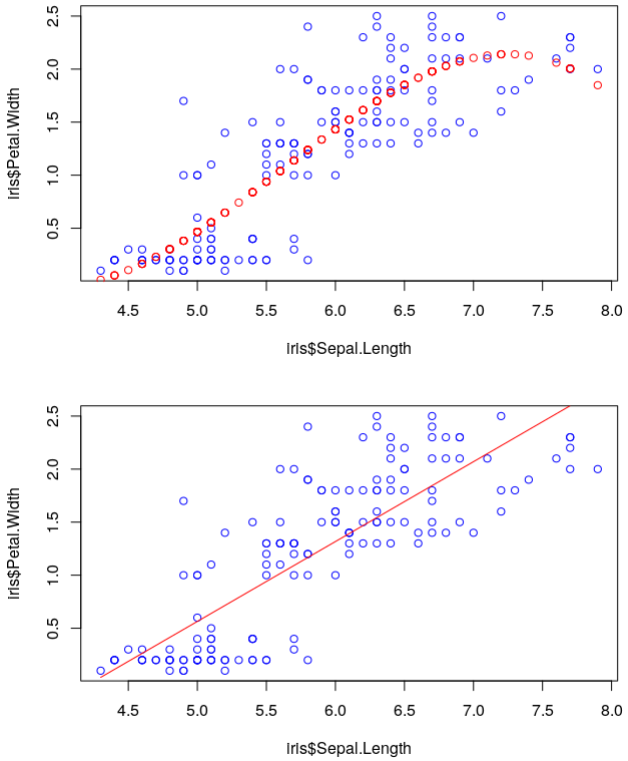


- dónde aquí si que parece que no hay una aproximación buena.

Introducción

- Por tanto, es necesario definir criterios más o menos objetivos (no simplemente visuales) para la valorar la aproximación.





- A continuación mostramos un ejemplo…

Descripción de la metodología con un ejemplo

1. Análisis preliminar
    1. Gráfico de pares
    2. Numérico: coeficientes de correlación
2. Construcción del modelo
    1. Definición del modelo
    2. Contraste de hipótesis sobre la nulidad de los coeficientes
3. Estimación de la bondad del modelo construido
    1. Estimación del sigma (Error Estardar Residula [EER])
    2. ANOVA
    3. Coeficiente de determinación
    4. Normalidad
    5. Homocedasticidad
    6. Incorreción

Descripción de la metodología con un ejemplo Base de Datos (Iris)

- Trabajaremos con la conocida bases de datos IRIS (Lirios).
- Presenta un problema clásico de clasificación, pero nosotros lo transformaremos en un problema de regresión.
- La base de datos original está compuesta por 4 variables predictivas reales y una variable de clasificación nominal. En base a la 4 variables predictivas se trata de discriminar entre 3 variantes de la planta Iris (Setosa, Versicolour y Virginica).



Descripción de la metodología con un ejemplo Base de Datos (Iris)

Tomar del SWAD el fichero “Script1_Pasos_Analisis_Modelo.R”

Seguir la ejecución según vayan apareciendo en las transparencias

- Las 4 variables predictivas miden la longitud y la anchura del Pétalo y el Sépalo de la planta.

- En nuestro caso, eliminaremos la variable de clasificación, y trataremos de estimar la anchura del sépalo en función de las otras 3 variables.

Comando en R

```
library(ISLR)
datos <-data.frame(y=iris$Sepal.Width,x1=iris$Sepal.Length,
                   x2=iris$Petal.Length,x3=iris$Petal.Width)
```

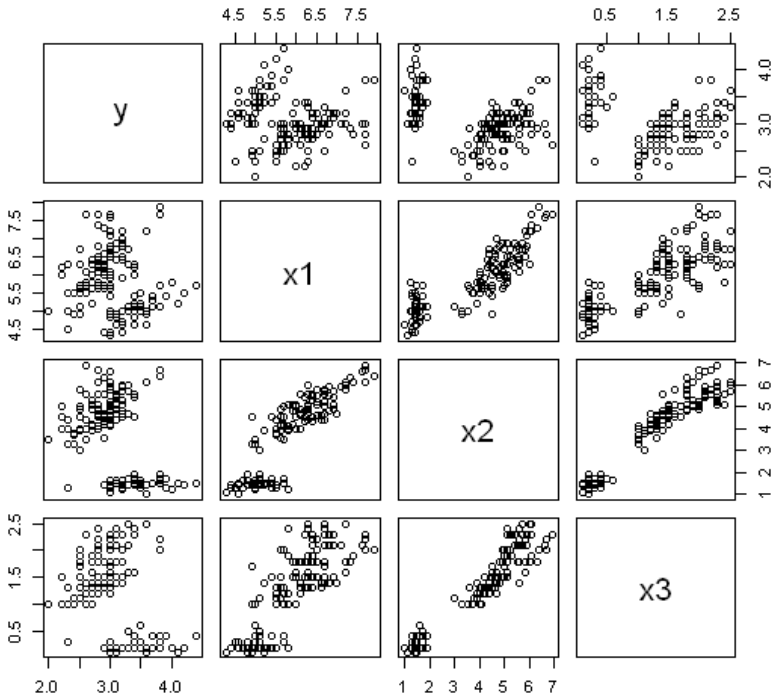
Descripción de la metodología con un ejemplo

Análisis preliminar: Gráfico de pares

- El análisis de los datos suele empezar con un visionado de los datos para establecer el grado de dependencia entre las variables del problema.

Comando en R

```
# 1a. Análisis preliminar (Gráfico)
plot(datos)
```



Descripción de la metodología con un ejemplo

Análisis preliminar: Numérico (Correlación)

- Un análisis numérico nos puede ayudar a mejorar la percepción de interrelación entre las variables como por ejemplo usando la correlación.
-
-

Comando en R

```
> cor(datos)
           y         x1         x2         x3
y  1.0000000 -0.1175698 -0.4284401 -0.3661259
x1 -0.1175698  1.0000000  0.8717538  0.8179411
x2 -0.4284401  0.8717538  1.0000000  0.9628654
x3 -0.3661259  0.8179411  0.9628654  1.0000000
```

- Valores cercanos a 1 y -1 indican alto grado de correlación entre las variables, mientras que valores cercanos a 0 indican independencia.

Descripción de la metodología con un ejemplo
Construcción del modelo: Definición del modelo

- En este caso, parece que hay poca correlación entre las variables predictivas y la variable a estimar. A pesar de eso, vamos a definir un modelo lineal en base a las 3 variables.

Comando en R

```
# 2. Construccion del modelo

reg_lineal <-lm(y~x1+x2+x3, data= datos)
resumen_reg_lineal <-summary(reg_lineal)
resumen_reg_lineal
```

Descripción de la metodología con un ejemplo
Construcción del modelo: Definición del modelo

- El resultado del modelo es el siguiente

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = datos)

Residuals:
      Min        1Q    Median        3Q       Max
-0.88045 -0.20945  0.01426  0.17942  0.78125

Coefficients:
             Estimate Std. Error t value   Pr(>|t|)
(Intercept)   1.04309    0.27058   3.855   0.000173 ***
x1            0.60707    0.06217   9.765    < 2e-16 ***
x2           -0.58603    0.06214  -9.431    < 2e-16 ***
x3            0.55803    0.12256   4.553    1.1e-05 ***
 ---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3038 on 146 degrees of freedom
Multiple R-squared: 0.524, Adjusted R-squared: 0.5142
F-statistic: 53.58 on 3 and 146 DF, p-value: < 2.2e-16
```

¿Son todas las variables relevantes para la estimación de los datos?

Test de hipótesis sobre Pr(>|t|):
Si el valor es menor a 0.05 se rechaza la hipótesis de nulidad de la variable.

En este caso, todas aportan al modelo.

Descripción de la metodología con un ejemplo

Construcción del modelo: Definición del modelo

- El resultado del modelo es el siguiente

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = datos)

Residuals:
      Min        1Q    Median        3Q       Max
-0.88045 -0.20945  0.01426  0.17942  0.78125

Coefficients:
             Estimate Std. Error t value   Pr(>|t|)
(Intercept)  1.04309    0.27058   3.855    0.000173 ***
x1           0.60707    0.06217   9.765    < 2e-16 ***
x2          -0.58603    0.06214  -9.431    < 2e-16 ***
x3           0.55803    0.12256   4.553    1.1e-05 ***
 ---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3038 on 146 degrees of freedom
Multiple R-squared: 0.524, Adjusted R-squared: 0.5142
F-statistic: 53.58 on 3 and 146 DF, p-value: < 2.2e-16
```

Modelo Lineal

$$Y = 1.04309 + 0.60707\,x1 - 0.58603\,x2 + 0.55803\,x3$$

Descripción de la metodología con un ejemplo
Estimación de la bondad del modelo (1. Estimación de Sigma)

- Ya tenemos un modelo lineal que trata de describir los datos y queremos saber como de buena es esta aproximación. Para ello, pasaremos algunos test al modelo.

- Empezamos la estimación de Sigma (Error Estándar Residual), que mide si la discrepancia entre el modelo y los datos es normal.
- La forma de cálculo del sigma es la siguiente:

Comando en R

```
cv <- 100*(resumen_reg_lineal$sigma/(mean(datos$y)))
cv
```

- Si el valor obtenido es inferior al 10 se considera un error estándar residual aceptable. En nuestro caso es 9.93, menor que 10 y por tanto aceptable.

Descripción de la metodología con un ejemplo
Estimación de la bondad del modelo (2. ANOVA)

- La segunda verificación tiene que ver con contestar a la siguiente pregunta:
- ¿Pueden anularse simultáneamente todos los coeficientes del modelo sin que empeore significativamente la aproximación?
- La respuesta a esta pregunta la encontramos en *summary* del modelo y en su *p-value* asociado.

- Si *p-value* es inferior a 0.05 se rechaza la hipótesis. En nuestro caso el valor es <2.2e-16.

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = datos)

Residuals:
      Min        1Q    Median        3Q       Max
-0.88045 -0.20945  0.01426  0.17942  0.78125

Coefficients:
             Estimate Std. Error t value   Pr(>|t|)
(Intercept)   1.04309    0.27058   3.855   0.000173 ***
x1            0.60707    0.06217   9.765    < 2e-16 ***
x2           -0.58603    0.06214  -9.431    < 2e-16 ***
x3            0.55803    0.12256   4.553    1.1e-05 ***
 ---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3038 on 146 degrees of freedom
Multiple R-squared: 0.524, Adjusted R-squared: 0.5142
F-statistic: 53.58 on 3 and 146 DF, p-value: < 2.2e-16
```

Descripción de la metodología con un ejemplo
Estimación de la bondad del modelo (ANOVA)

- La segunda verificación tiene que ver con contestar a la siguiente hipótesis:
- ¿Pueden anularse simultáneamente todos los coeficientes del modelo sin que empeore significativamente la aproximación?
- La respuesta a esta pregunta la encontramos en *summary* del modelo y en su *p-value* asociado.

- Si *p-value* es inferior a 0.05 se rechaza la hipótesis. En nuestro caso el valor es <2.2e-16.

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = datos)

Residuals:
      Min        1Q    Median        3Q       Max
-0.88045 -0.20945  0.01426  0.17942  0.78125

Coefficients:
             Estimate Std. Error t value   Pr(>|t|)
(Intercept)   1.04309    0.27058   3.855   0.000173 ***
x1            0.60707    0.06217   9.765    < 2e-16 ***
x2           -0.58603    0.06214  -9.431    < 2e-16 ***
x3            0.55803    0.12256   4.553    1.1e-05 ***
 ---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.3038 on 146 degrees of freedom
Multiple R-squared: 0.524, Adjusted R-squared: 0.5142
F-statistic: 53.58 on 3 and 146 DF,  p-value: < 2.2e-16
```

Descripción de la metodología con un ejemplo

Estimación de la bondad del modelo (3. Coeficiente de Determinación R^2)

- La tercera verificación hace referencia al coeficiente de Determinación. Este coeficiente indica que porcentaje de la variabilidad de los datos la explica el modelo.

Comando en R

```
resumen_reg_lineal$r.squared
```

- Si el resultado obtenido es superior a 80% se considera un valor aceptable. En nuestro caso es 52.4%, menor que 80 y **por tanto no aceptable**.
- El valor de R^2 se ve afectado por número de variables predictivas involucradas en el modelo. Para reducir esta dependencia, se calcula el R^2 ajustado

Comando en R

```
resumen_reg_lineal$r.squared
```

- Si los valores no son parecidos, entonces hay influencia en el número de variables y este segundo valor es más fiable.

Descripción de la metodología con un ejemplo

Estimación de la bondad del modelo (4. Error Cuadrático Medio)

- La última verificación es calcular el Error Cuadrático Medio (MSE) del modelo frente a los datos.

Comando en R	Resultado

# MSE 	
reg_lineal.fit <- predict(reg_lineal, newdata = datos)
 	
sum(((datos$y-reg_lineal.fit)^2))/length(reg_lineal.fit)
	[1] 0.089826

- Esta medida se utiliza mucho en los trabajos de regresión, pero no da una medida estandarizada y contrastable como las tres anteriores. En este caso, se usa en comparación con otro modelo.

- A pesar de ser muy usada, no es una medida robusta del modelo usada de forma aislada. Entre modelos que satisfacen las medidas anteriores, aquel que tiene menor MSE suele ser el que mejor aproxima los datos.

Descripción de la metodología con un ejemplo
Estimación de la bondad del modelo (5. Normalidad)

- Una vez que hemos establecido la bondad de la aproximación obtenida (en nuestro caso satisface 2 de las 3), ahora queremos determinar que buenas propiedades nos puede ofrecer el uso de este modelo. Para ello, vamos a evaluar 4 criterios.
- Empezaremos con el de Normalidad que trata de asegurar que los errores del modelo se distribuyen siguiendo una distribución normal.

Comando en R

```
par(mfrow=c(2,2))
plot(reg_lineal)
```

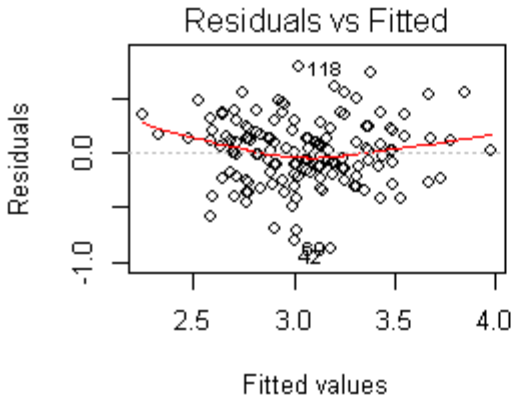
- Nos fijaremos en el gráfico “Normal Q-Q”

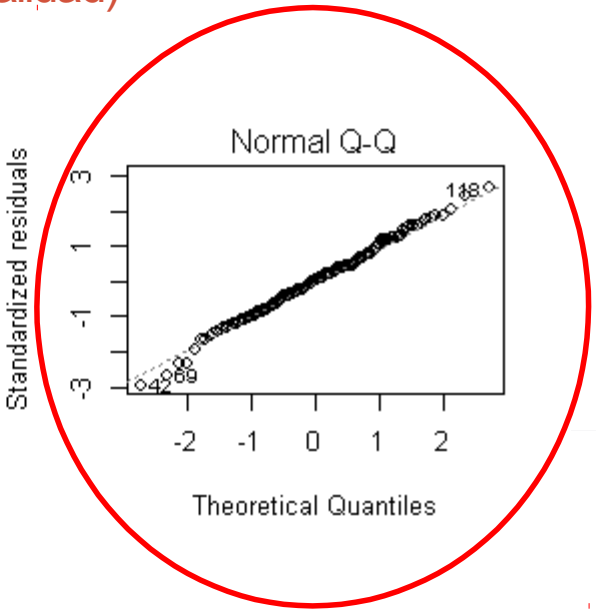
Descripción de la metodología con un ejemplo

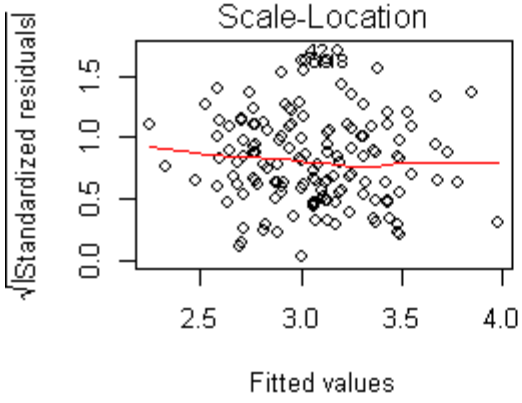
Estimación de la bondad del modelo (5. Normalidad)

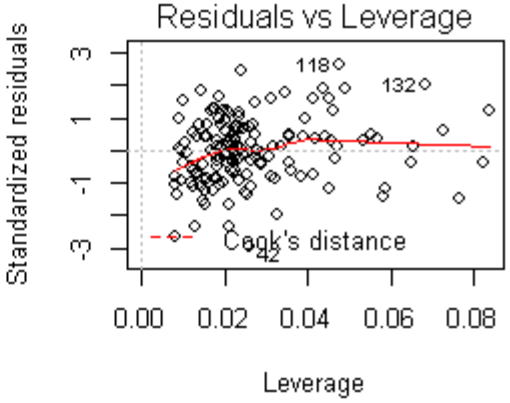
Queremos que los datos se ajusten lo más posible a la línea discontinua.

En nuestro caso, se aproximan bien, excepto los primeros datos.









Descripción de la metodología con un ejemplo
Estimación de la bondad del modelo (5. Normalidad)

- Para asegurar más este diagnostico, haremos un histograma sobre los errores (residuos).

Comando en R

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

e <- residuals(reg_lineal)
d <- e/resumen_reg_lineal$sigma

hist (d, probability = T, xlab = "Errores estandar", main = "", xlim = c(-3,3))

d.seq <- seq(-3,3,length = 50)

lines(d.seq, dnorm(d.seq, mean(d), sd(d)), col="red")
```

Descripción de la metodología con un ejemplo

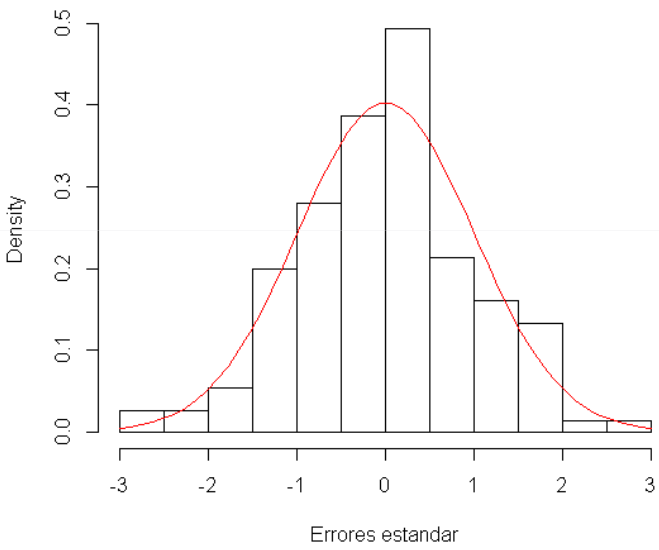
Estimación de la bondad del modelo (5. Normalidad)

- El gráfico muestra que el histograma tiene cierto parecido con la campana de normalidad, pero los errores cercanos a cero positivos parecen excesivos.

- Aún podemos usar una tercera vía para mostrar la normalidad de los errores. Consiste en aplicar el test de Shapiro-Wilk, tiene como hipótesis nula

H0: error~N(mean,sd)

Histograma de Residuos



Comando en R

```
# El test de normalidad Shapiro-Wilk
shapiro.test(e)
```

Resultado

data: e W = 0.9944, p-value = 0.8389

p-value>0.05, se acepta la hipótesis

Descripción de la metodología con un ejemplo

Estimación de la bondad del modelo (6. Homocedasticidad)

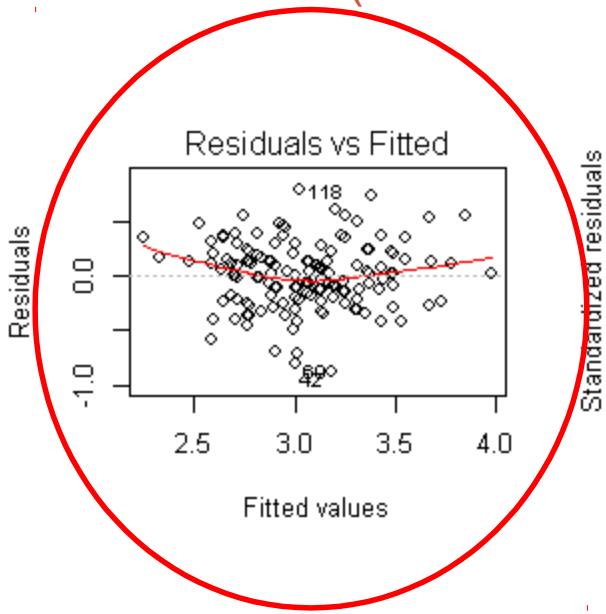
La homocedasticidad hace referencia a que la varianza de los datos se mantiene constante. Para ello Mirar el grafico de

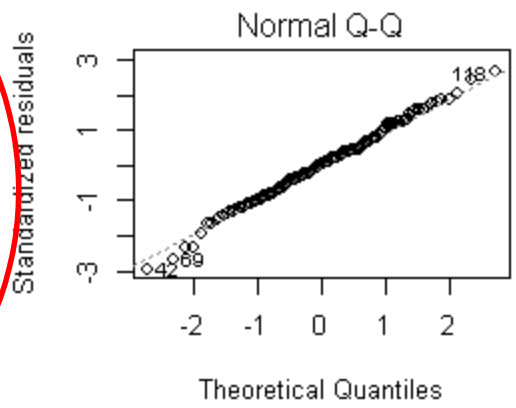
Comando en R

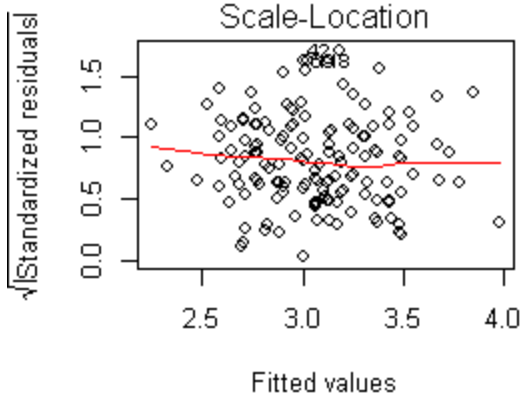
```
plot(reg_lineal)
```

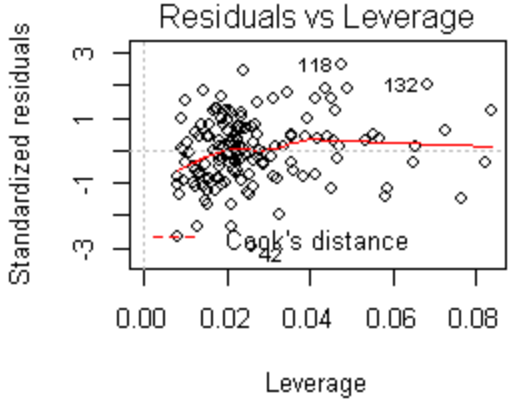
"Residuals vs Fitted"

Se debe observar que la anchura de los datos es aproximadamente igual.









Descripción de la metodología con un ejemplo
Estimación de la bondad del modelo (6. Homocedasticidad)

- Parece que se satisface, aunque visualmente pueden quedar dudas.
- Con el fin de poder afirmarlo, usaremos el test de Breusch-Pagan que tiene como **H0: homocedastidad**

Comando en R

```
# test de Breusch-Pagan
library(lmtest)
bptest(reg_lineal)
```

Resultado

```
studentized Breusch-Pagan test

data: reg_lineal
BP = 0.7348, df = 3, p-value = 0.865
```

- p-value es mayor que 0.05 y por consiguiente se acepta la hipótesis de homocedasticidad.

Descripción de la metodología con un ejemplo

Estimación de la bondad del modelo (7. Incorrelación)

- La incorrelación trata de demostrar que los errores producidos por el modelo no están correlados. Visualmente se puede ver de la siguiente forma

Comando en R

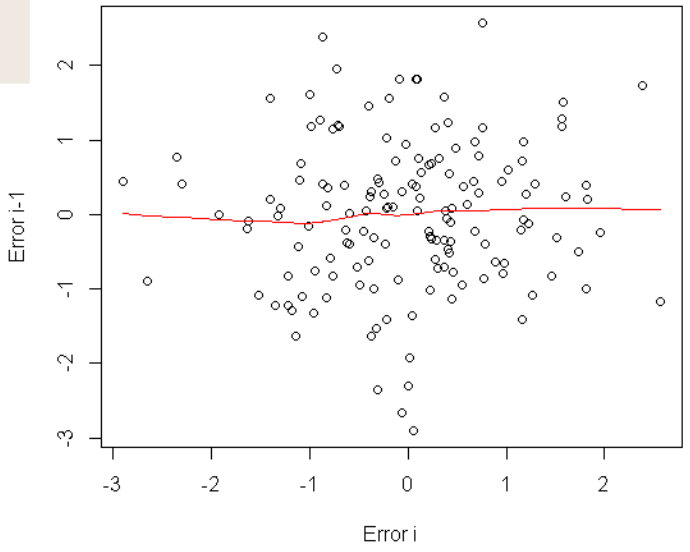
```
n <- length(d)

plot(d[1:n-1],d[2:n], xlab = "Error i", ylab = "Error i-1")

lines(lowess(d[1:n-1],d[2:n]),col="red")
```

No debería verse ninguna tendencia ascendente o descendente en la línea roja.

En este caso, parece claro que no la hay.



Descripción de la metodología con un ejemplo
Estimación de la bondad del modelo (7. Incorrelación)

- De forma analítica podemos mostrar la incorrelación haciendo uso del test de Durbin-Watson, con **H0: correlacion 0**.

Comando en R

```
# Test de Durbin-Watson
library(lmtest)
dwtest(reg_lineal, alternative = "two.sided")
```

Resultado

```
Durbin-Watson test

data: reg_lineal
DW = 1.8887, p-value = 0.4258
alternative hypothesis: true autocorrelation is not 0
```

- p-value es mayor que 0.05 y por consiguiente se acepta la hipótesis de incorrelación.

Descripción de la metodología con un ejemplo
Representación de la aproximación

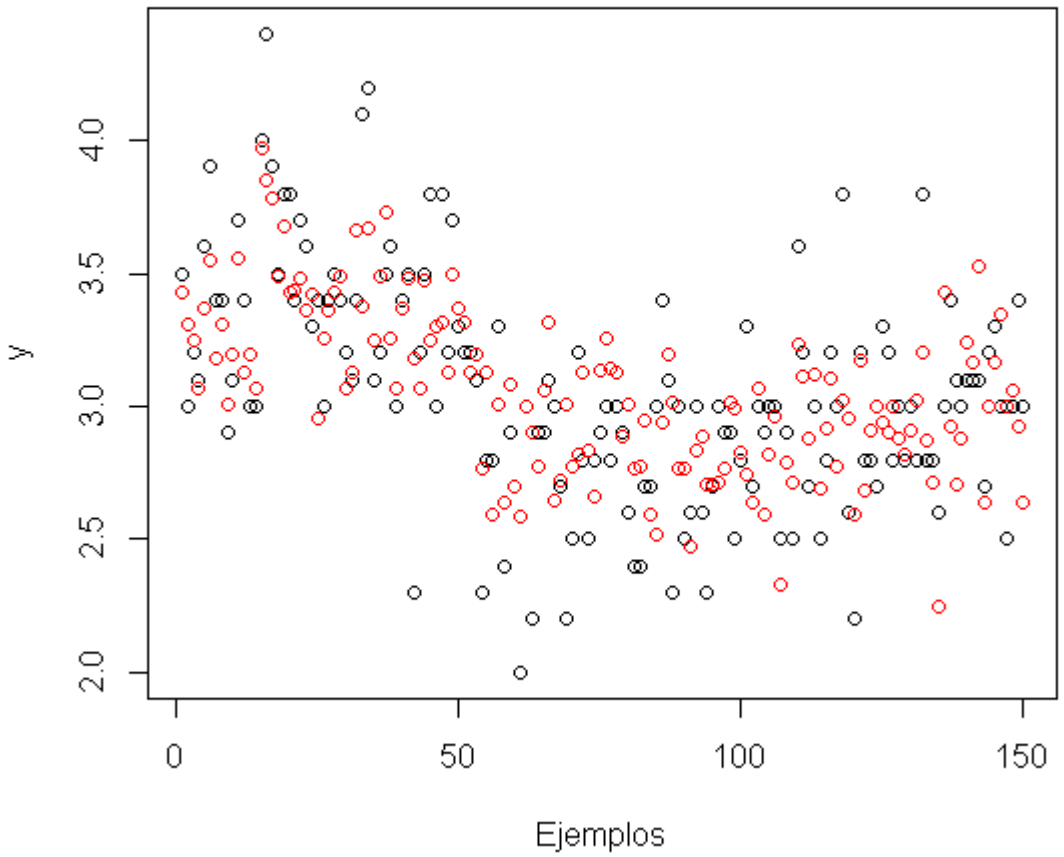
- No es fácil mostrar visualmente el resultado del modelo obtenido cuando este tiene más de 2 variables implicadas.
- La forma más simple sería la siguiente:

Comando en R

```
plot(1:dim(datos)[1],datos[,1], xlab ="Ejemplos" , ylab="y")
pred <- predict(reg_lineal, newdata = datos)
points(1:dim(datos)[1],pred, col="red")
```

Descripción de la metodología con un ejemplo

Representación de la aproximación (sin ordenar)



Descripción de la metodología con un ejemplo
Representación de la aproximación

- No es fácil mostrar visualmente el resultado del modelo obtenido cuando este tiene más de 2 variables implicadas.
- La forma más simple sería la siguiente:

Comando en R

```
plot(1:dim(datos)[1],datos[,1], xlab ="Ejemplos" , ylab="y")
pred <- predict(reg_lineal, newdata = datos)
points(1:dim(datos)[1],pred, col="red")
```

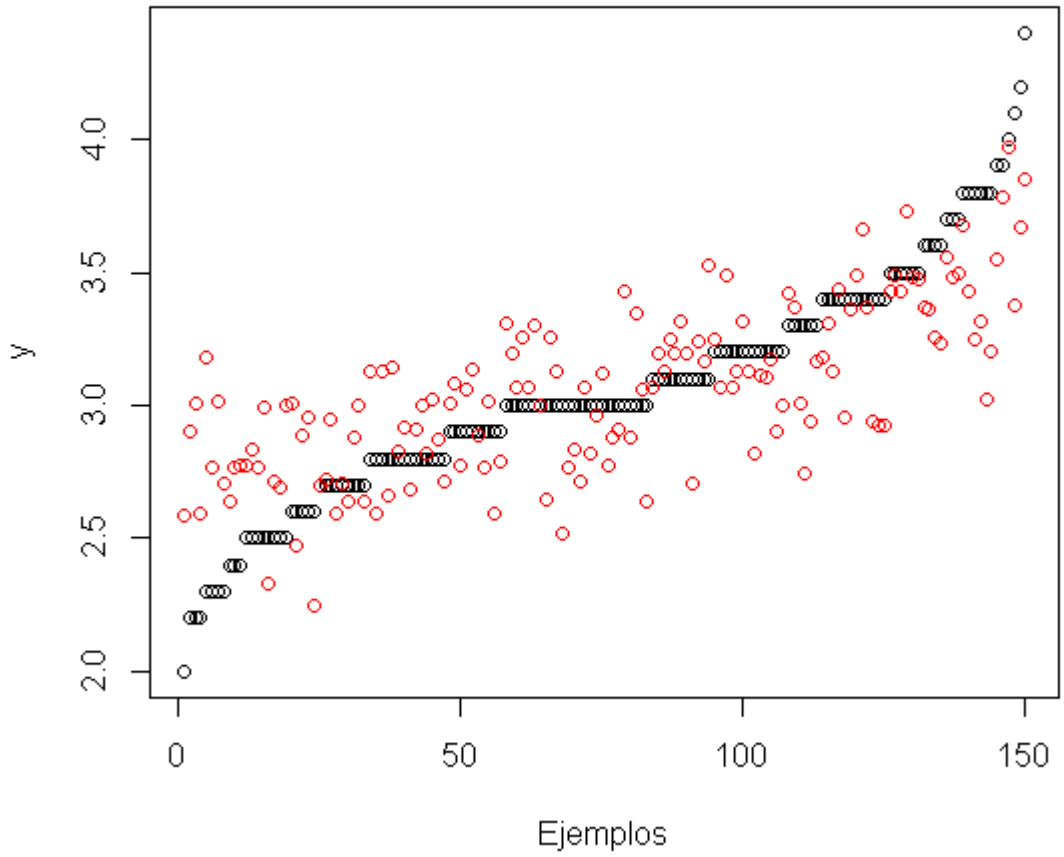
-

Comando en R

```
datos_ord <-datos[sort(datos[,1], index.return=TRUE)$ix,]
reg_lineal = lm(y~x1+x2+x3, data= datos_ord)
plot(1:dim(datos_ord)[1],datos_ord[,1], xlab ="Ejemplos" , ylab="y")
pred <- predict(reg_lineal, newdata = datos_ord)
points(1:dim(datos_ord)[1],pred, col="red")
```

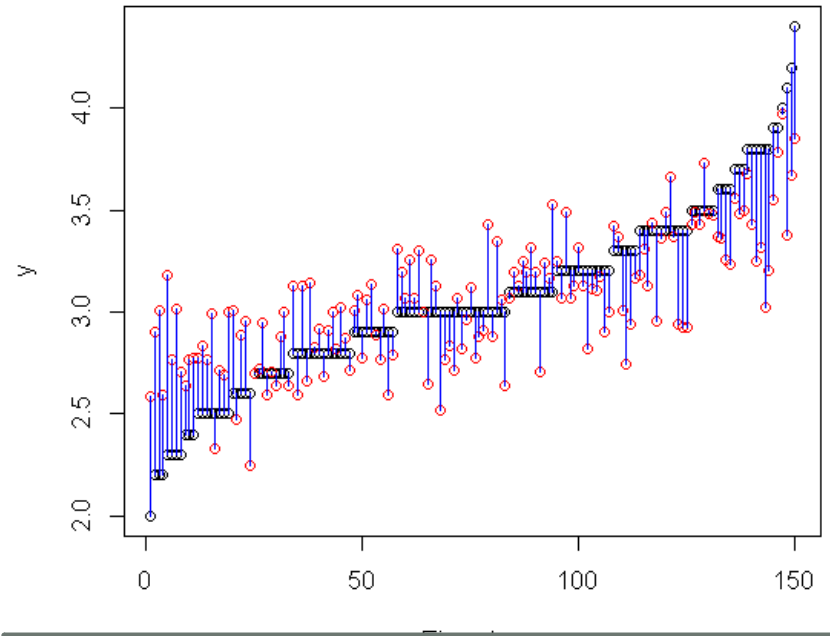
Descripción de la metodología con un ejemplo

Representación de la aproximación (ordenando la salida)



Descripción de la metodología con un ejemplo

Representación de la aproximación (magnitud de los errores)



Puedo observar la magnitud de los errores que se comenten añadiendo el siguiente comando

Comando en R

```
segments(1:dim(datos_ord)[1], datos_ord$y, 1:dim(datos_ord)[1], pred,col="blue")
```

Descripción de la metodología con un ejemplo
Ejercicios

Ejercicio 1: Considera la existencia de la función `MSE <- function (datos, regresion)` que devuelve el error cuadrático medio, dado un conjunto de datos «datos» y un modelo de «regresion» y de la función `ValidacionCruzada <- function (datos, particiones, model)`

que devuelve el resultado de hacer la validación cruzada del modelo «model» sobre el conjunto de ejemplos «datos» usando un número de particiones = «particiones».

Hacer una función `Analisis <- function (datos, model)` que realice el estudio realizado previamente aplicando los test para verificar la idoneidad del modelo.

En «Script2_FuncionAnalisis.R» podéis encontrar la implementación de las funciones anteriores y la estructura de la función Analisis dónde tenéis que rellenar los huecos.

Descripción de la metodología con un ejemplo
Ejercicios

Ejercicio 2: Haciendo uso de las funciones definidas anteriormente, realizar el estudio de la idoneidad de una aproximación multilineal sobre la base de datos iris, pero en este caso para estimar la *Longitud del Pétalo* (*Petal_Length*) en base a la otras tres variables (descartando la de clasificación).

- ¿Es apropiada una aproximación lineal en este caso?
- ¿Dirías que este segundo modelo es mejor que el primero?

	EER	Anova	R2	Norm	Homo	Incor	MSE	CV	%R_{ango}
Ejer1	Si	Si	No	Si	Si	Si	0.09	0.094	10.1%
Ejer2	Si	Si	Si	Si	Si	Si	0.09	0.1	4.2%

Descripción de la metodología con un ejemplo
Ejercicios

Ejercicio 3: Repetir el estudio anterior, pero en este caso para la base de datos trees. Queremos saber si una aproximación multilineal es apropiada para determinar el «Girth» en base a «Height» y «Volume».

	EER	Anova	R2	Norm	Homo	Incor	MSE	CV	%R_{ango}
Ejer1	Si	Si	No	Si	Si	Si	0.09	0.094	10.1%
Ejer2	Si	Si	Si	Si	Si	Si	0.09	0.1	4.2%
Ejer3	Si	Si	Si	No	Si	No	0.56	0.79	5.8%

Regresión no lineal linealizable

1. Modelo polinomial poly()

2. Regresión con “spline”

Regresión no lineal linealizado
Modelo polinomial

- Los modelos de regresión no lineal linealizados son aquellos que utilizan la regresión lineal estándar, pero introducen como argumentos de la regresión, funciones no lineales de las variables predictivas.

- De entre estas extensiones del modelo lineal para definir modelos no lineales están aquellas que hacen uso de polinomios.

Regresión no lineal linealizado
Modelo polinomial

- En R, la sintaxis para definir una familia de polinomios es

Sintaxis

```
poly(x, ..., degree = 1, coefs = NULL, raw = FALSE)

donde:
x          es un vector numérico para evaluar el polinomio
degree     es el grado del polinomio
```

Por consiguiente, un modelo no lineal linealizado usando polinomios sería tan simple como

```
model1 <-lm(y~poly(x1,3), data= datos)
```

Regresión no lineal linealizado
Modelo polinomial

Ejercicio 4: Tomar la base de datos iris (sin la variable de clasificación) y definir un modelo no lineal linealizado usando polinomios de hasta grado 3 que permita aproximar la anchura del sépalo (Sepal_Width) a partir de las otras tres variables usando.

En concreto, estudiar los siguientes 5 modelos

Comando en R

```
model1 <-lm(y~poly(x1,3), data= datos)
model2 <-lm(y~poly(x2,3), data= datos)
model3 <-lm(y~poly(x3,3), data= datos)
model4 <-lm(y~poly(x1,3)+poly(x2,3)+poly(x3,3), data= datos)
model5 <-lm(y~poly(x1,3)*poly(x2,3)*poly(x3,3), data= datos)
```

Regresión no lineal linealizado

Modelo polinomial

Ejercicio 4:

Comando en R

```
model1 <-lm(y~poly(x1,3), data= datos)
model2 <-lm(y~poly(x2,3), data= datos)
model3 <-lm(y~poly(x3,3), data= datos)
model4 <-lm(y~poly(x1,3)+poly(x2,3)+poly(x3,3), data= datos)
model5 <-lm(y~poly(x1,3)*poly(x2,3)*poly(x3,3), data= datos)
```

¿Cuál de los modelos consideras que realiza una mejor estimación?

	EER	Anova	R2	Norm	Homo	Incor	MSE	CV	$\%R_{ango}$
model1	No	Si	No	Si	No	No	0.18	0.19	14.1%
model2	No	Si	No	Si	No	Si	0.11	0.12	11.4%
model3	No	Si	No	Si	No	Si	0.12	0.12	11.4%
model4	Si	Si	No	Si	Si	Si	0.07	0.08	9.5%
model5	Si	Si	Si	No	Si	Si	0.03	1.05	20.6%

Regresión no lineal linealizado
Modelo polinomial

Ejercicio 4:

Comando en R

```
model5 <-lm(y~poly(x1,3)*poly(x2,3)*poly(x3,3), data= datos)
```

Interpretación en R del modelo 5

Coefficients	Estimate	Interpretación
(Intercept)	16.59	16.59
poly(x1,3)1	447.19	+ 447.19 * x1
poly(x1,3)2	84.53	+ 84.53 * x1^2
…..	….	….
poly(x1,3)1:poly(x2,3)1	-6172.70	- 5172.7 * x1 * x2
…..	…..	…..
poly(x1,3)2:poly(x2,3)1:poly(x3,3):3	-6818.4	- 6818.4 * x1^2 * x2 * x3^3

Regresión no lineal linealizado
splines

- Los “splines” son un tipo particular de polinomio muy utilizado en aproximación de datos por sus especiales propiedades de adaptación.

- En R, podemos encontrar multitud de ellos, pero nosotros nos centraremos por ahora en los splines cúbicos.

Sintaxis

```
bs(x, df = NULL, knots = NULL, degree = 3, intercept = FALSE,
Boundary.knots = range(x))
```

donde:

x	es un vector numérico para evaluar el polinomio
degree	es el grado del spline, en nuestro caso siempre 3

Regresión no lineal linealizado
splines

Ejercicio 5: Tomar la base de datos **iris** (sin la variable de clasificación) y definir un modelo no lineal linealizado usando splines cúbicos que permita aproximar la anchura del sépalo (Sepal_Width) a partir de las otras tres variables usando.

En concreto, estudiar los siguientes 5 modelos

Comando en R

```
library(splines)

model1 <-lm(y~bs(x1), data= datos)
model2 <-lm(y~bs(x2), data= datos)
model3 <-lm(y~bs(x3), data= datos)
model4 <-lm(y~bs(x1)+bs(x2)+bs(x3), data= datos)
model5 <-lm(y~bs(x1)*bs(x2)*bs(x3), data= datos)
```

Regresión no lineal linealizado

splines

Ejercicio 5:

Comando en R

```
model1 <-lm(y~bs(x1), data= datos)
model2 <-lm(y~bs(x2), data= datos)
model3 <-lm(y~bs(x3), data= datos)
model4 <-lm(y~bs(x1)+bs(x2)+bs(x3), data= datos)
model5 <-lm(y~bs(x1)*bs(x2)*bs(x3), data= datos)
```

	EER	Anova	R2	Norm	Homo	Incor	MSE	CV	$\%R_{ango}$
model1	No	Si	No	Si	No	No	0.18	0.19	14.1%
model2	No	Si	No	Si	No	Si	0.11	0.12	11.4%
model3	No	Si	No	Si	No	Si	0.12	0.12	11.3%
model4	Si	Si	No	Si	Si	Si	0.07	0.08	9.5%
model5	Si	Si	**Si**	No	Si	Si	0.03	**2.37**	**24.6%**

Regresión no lineal
GAM (Modelo General Aditivo)

- GAM representa la autentica regresión no lineal y construye la aproximación a partir de la suma de funciones no lineales.
- Su sintaxis es semejante a la regresión lineal

Sintaxis

```
gam(formula, family = gaussian, data, weights, subset, na.action, start,
etastart, mustart, control = gam.control(...), model=FALSE, method,
x=FALSE, y=TRUE, ...)
```

- Los dos parámetros relevantes son <formula> y <data> que especifican la forma en la que los factores intervienen en la aproximación y los datos respectivamente.

- Su uso también es semejante a lo que ya hemos visto y lo pondremos en práctica con el siguiente ejercicio.

Regresión no lineal
GAM (Modelo General Aditivo)

Ejercicio 6: Tomar la base de datos iris (sin la variable de clasificación) y definir un modelo no lineal linealizado usando polinomios de hasta grado 3 que permita aproximar la anchura del sépalo (Sepal_Width) a partir de las otras tres variables usando.

En concreto, estudiar estos 4 modelos

Comando en R

```
model1 <-lm(y~ns(x1,4)+ns(x2,4)+ns(x3,4), data= datos)
model2 <-gam(y~s(x1,4)+s(x2,4)+s(x3,4), data= datos)
model3 <-gam(y~s(x1,16)+s(x2,16)+s(x3,16), data= datos)
model4 <-gam(y~s(x1,16)*s(x2,16)*s(x3,16), data= datos)
```

donde

- ns(x,g) representa un spline natural de grado “g” sobre la variable “x”
- s(x,g) representa un spline suavizado de grado “g” sobre la variable “x”

Regresión no lineal
GAM (Modelo General Aditivo)

Ejercicio 6:

Comando en R

```
model1 <-lm(y~ns(x1,4)+ns(x2,4)+ns(x3,4), data= datos)
model2 <-gam(y~s(x1,4)+s(x2,4)+s(x3,4), data= datos)
model3 <-gam(y~s(x1,16)+s(x2,16)+s(x3,16), data= datos)
model4 <-gam(y~s(x1,16)*s(x2,16)*s(x3,16), data= datos)
```

	EER	Anova	R2	Norm	Homo	Incor	MSE	CV	%R_{ango}
model1	Si	Si	No	Si	Si	Si	0.07	0.08	9.5%
model2	No	--	No	Si	Si	Si	0.06	0.08	8.9%
model3	No	--	No	Si	Si	Si	0.04	0.1	9.7%
model4	No	--	No	Si	Si	Si	0.03	0.09	9.2%

	Modelo	EER	Anova	R2	Norm	Homo	Incor	MSE	CV	%R$_{ango}$
Ejer1	*lineal*	Si	Si	No	Si	Si	Si	0.09	0.09	10.1%
Ejer4	*model4*	Si	Si	No	Si	Si	Si	0.07	0.08	9.5%
Ejer5	*model3*	No	Si	No	Si	No	Si	0.12	0.12	11.3%
Ejer5	*model4*	Si	Si	No	Si	Si	Si	0.07	0.08	9.5%
Ejer6	*model1*	Si	Si	No	Si	Si	Si	0.07	0.08	9.5%
Ejer6	***model2***	No	--	--	Si	Si	Si	0.06	0.08	**8.9%**
Ejer6	*model3*	No	--	--	Si	Si	Si	0.04	0.1	9.7%
Ejer6	*model4*	No	--	--	Si	Si	Si	**0.03**	0.09	9.2%

Ejercicio de Repaso

Ejercicio 7: Dada la base de datos “trees” que contiene tres variables (Girth, Height, Volume), se pide

A. Encontrar la mejor aproximación de “Volume” a partir de “Girth” y “Height” usando un modelo de regresión polinomial (Usando “poly()” hasta grado 3). El criterio de decisión será aquella que tenga menor porcentaje de error tras la validación cruzada.

B. Repetir el apartado A) pero en este caso haciendo uso de regresión con “bs()” spline cúbicos. ¿Es mejor que el modelo polinomial encontrando?.

C. Repetir el apartado A) pero en este caso haciendo uso de regresión “gam()” con spline suavizados d hasta grado 4 “s()”. ¿Se consigue alguna mejora en relación a los dos modelos anteriores?