Ejemplo simple de regresión lineal simple y múltiple (para realizar en el aula)

OBJETIVOS:

1. **Familiarización con la terminología y funciones ligadas a un análisis de regresión lineal**
2. **Establecer las pautas a seguir en un análisis de regresión lineal**

FUNCIÓN A USAR EN UN ANÁLISIS DE REGRESIÓN

CONTENIDO

código

Ejemplo simple de regresión lineal simple y múltiple

Función a usar para un análisis de regresión:

lm(formula, data, subset)

Descripción de los argumentos usados:

formula: Describe la ecuación del modelo; es decir, la variable dependiente o respuesta seguida del símbolo ~ y las variables independientes

f$y~f$x

data: Es optativo El data.frame que contiene las variables a usar. En nuestro ejemplo será

f

subset: Es optativo. Permite realizar el análisis sólo en parte de los datos.

Resultados del análisis

Coefficients, residuals, fitted

Que representan los coeficientes, residuos, valores ajustados, respectivamente

1.Regresión simple

1.1.Datos (introducidos editor de data.frames)
Datos

>f=edit(data.frame())

> f

```
  x y
1 1 2
2 2 4
3 3 5
```

1.2.Regresión simple: Coeficientes del modelo ajustado

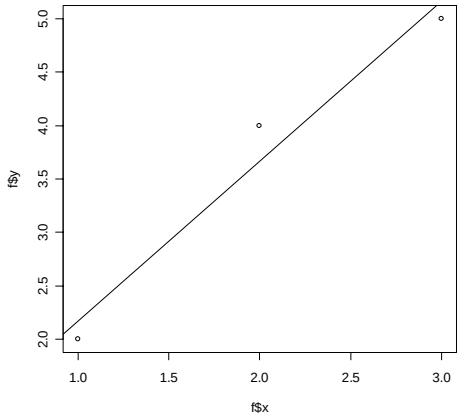
> lm(f$y~f$x) **#equivalente a lm(y~x,data=f)**

```
Call:
lm(formula = f$y ~ f$x)

Coefficients:
(Intercept)          f$x
     0.6667       1.5000
```

1.3.Representación gráfica con la recta ajustada y los datos

> plot(fx,fy)
> abline(lm(f$y~f$x))



1.4.Ver resumen del análisis de regresión

> summary(lm(f$y~f$x))

```
Call:
lm(formula = f$y ~ f$x)

Residuals:
      1       2       3
-0.1667  0.3333 -0.1667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.6667     0.6236   1.069    0.479
f$x           1.5000     0.2887   5.196    0.121

Residual standard error: 0.4082 on 1 degrees of freedom
Multiple R-Squared: 0.9643,     Adjusted R-squared: 0.9286
F-statistic:    27 on 1 and 1 DF,  p-value: 0.1210
```

1.5.Predicción o valores ajustados

> predict(lm(f$y~f$x))

```
       1        2        3
2.166667 3.666667 5.166667
```

> fitted(lm(f$y~f$x))

```
       1        2        3
2.166667 3.666667 5.166667
```

1.6.Otra forma de manipular el objeto “lm”: asignándole un nombre

Hace más manejable el objeto, sin necesidad de especificarlo cada vez que se use, se invoca a parte de los resultados del análisis.

>reg1=lm(f$y~f$x)

> coef(reg1)

```
(Intercept)         f$x
  0.6666667   1.5000000
```

> residuals(reg1)

```
         1          2          3
-0.1666667  0.3333333 -0.1666667
```

> fitted(reg1)

```
       1        2        3
2.166667 3.666667 5.166667
```

> predict(reg1)

```
       1        2        3
2.166667 3.666667 5.166667
```

>plot(reg1) #Presenta diversos gráficos: valores ajustados y residuos; Cuantiles teóricos de una Normal y los residuos estandarizados; Localización escala; residuos y leverage.

> vcov(reg1) #matriz de varianzas covarianzas de los coeficientes

```
            (Intercept)         f$x
(Intercept)   0.3888889 -0.16666667
f$x          -0.1666667  0.08333333
```

> sqrt(diag(vcov(reg1))) #desviación típica de los coeficientes estimados

```
(Intercept)         f$x
  0.6236096   0.2886751
```

1.7.Predicción de intervalos de respuestas futuras, nuevas o individuales

> predict(reg1, interval="prediction")

```
       fit       lwr       upr
1 2.166667 -4.856952  9.190286
2 3.666667 -2.323096  9.656429
3 5.166667 -1.856952 12.190286
Warning message:
Predictions on current data refer to _future_ responses
 in: predict.lm(reg1, interval = "prediction")
```

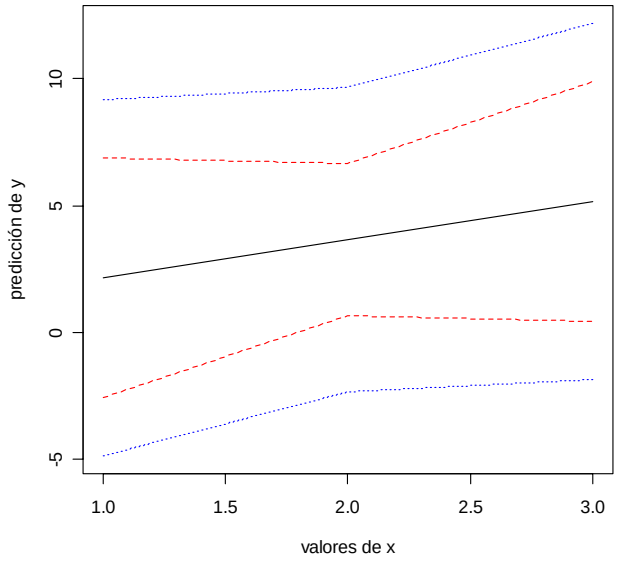
1.8.Predicción de intervalos de respuestas medias

> predict(reg1, interval="confidence")

```
       fit        lwr      upr
1 2.166667 -2.5686563 6.901990
2 3.666667  0.6717855 6.661548
3 5.166667  0.4313437 9.901990
```

1.9.Representación gráfica de los límites y valores ajustados

```
LimMedias<- predict(reg1, interval="confidence")
LimIndv<-predict(reg1, interval="prediction")
matplot(f$x,cbind(LimMedias,LimIndv[,-1]),
      lty=c(1,2,2,3,3), type="l", col=c('black','red', 'red','blue','blue'),xlab="valores de x", ylab="predicción de y")
```



2.Regresión con variables cualitativas

2.1.Datos: Modificaremos el data.frame inicial añadiendo nueva información

```
>f2<-rbind(f,f)
>f2<-edit(f2) #añada una columna z y modifique
>f2
```

```
  x y z
1 1 2 a
2 2 4 a
3 3 5 a
4 1 3 b
5 3 5 b
6 4 9 b
```

2.2.Previa inspección gráfica

Gráfico de los datos clasificados por z

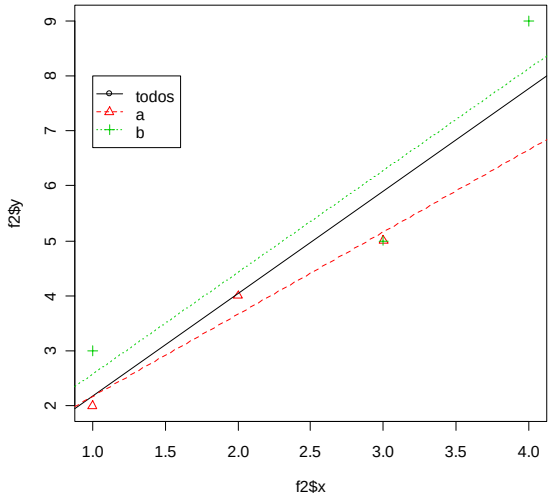
> (f2$z) #Observe que z es de tipo caracter
```
[1] "a" "a" "a" "b" "b" "b"
```

> as.factor(f2$z)
```
[1] a a a b b b
Levels: a b
```

> fz=as.factor(f2$z) #lo convertimos en factor para poder asignarles códigos numéricos que usamos para identificar puntos en un gráfico de dispersión, por ejemplo, definiendo el tipo o color que lo representará

> as.numeric(fz)
```
[1] 1 1 1 2 2 2
```

>Numz=as.numeric(fz)
>plot(f2$x,f2$y, type="p",pch=Numz+1,col=Numz+1) #Representación de los puntos de cada subgrupo en colores distintos >abline(lm(f2$y~f2$x)) #recta de regression con los 6 casos
>abline(lm(f2$y~f2$x,data=f2, subset=f2$z=="a"),col=2,lty=2)
>abline(lm(f2$y~f2$x,data=f2, subset=f2$z=="b"),col=3,lty=3)
>legend(1,8,c("todos","a","b"),pch=c(1,2,3),lty=1:3,col=1:3)



2.3.Regresión con una variable cualitativa

Datos:
> f2
```
  x y z
1 1 2 a
2 2 4 a
3 3 5 a
4 1 3 b
5 3 5 b
6 4 9 b
```

Modo 1 (que usaremos generalmente)
> lm(f2$y~fz) #equivalente a lm(f2$y~factor(f2$z)) #Cuando los niveles son numéricos (no caracteres) es necesario declarar la variable como factor, de lo contrario usaría los niveles como valores numéricos de

una variable cuantitativa, erróneamente. Si es de tipo carácter se asume como factor, aunque no se especifique.

```
Call:
lm(formula = f2$y ~ fz)

Coefficients:
(Intercept)          fzb
      3.667        2.000
```

Modo 2 (más propio de análisis de varianza o anova)

> lm(f2$y~fz-1)

Call:
lm(formula = f2$y ~ fz - 1)

Coefficients:
  fza   fzb
3.667 5.667

2.4.Regresión con una Cuantitativa y otra cualitativa

> lm(f2$y~f2$x+fz) #equivalente a lm(f2$y~f2$x+factor(f2$z))

Call:
```
lm(formula = f2$y ~ f2$x + fz)

Coefficients:
(Intercept)         f2$x          fzb
     0.1667       1.7500       0.8333
```

2.5.Modelo con interacción de x y z

>lm(f2$y~f2$x+fz+f2$x*fz)

```
Call:
lm(formula = f2$y ~ f2$x + fz + f2$x * fz)

Coefficients:
(Intercept)         f2$x          fzb     f2$x:fzb
     0.6667       1.5000      -1.0000       1.5000
```

Observe que cuando se usa la variable cualitativa como factor, automáticamente el sistema la trata como una variable ficticia. Las variables faz y fazx se han añadido para que pueda comprobarlo. faz es la ficticia de z y fazx es la interacción de x y la ficticia de z

> f3
  x y z faz fazx
1 1 1 2 a   0   0
2 2 2 4 a   0   0
3 3 3 5 a   0   0
4 1 1 3 b   1   1
5 3 3 5 b   1   3
6 4 4 9 b   1   4

Por ejemplo, el modelo:

lm(f3$y~factor(f3$z)+f3$x+f3$x*factor(f3$z))

usando en la notación de los términos factor, equivale a

lm(f3$y~f3$faz+f3$x+f3$fazx)

y también a

lm(f3$y~f3$faz+f3$x+f3$x*f3$faz)

2.6.Resumen de modelos posibles de regresión con variables cuantitativa, cualitativa e interacción

```
>regcat=lm(f2$y~factor(f2$z))
>regsimple=lm(f2$y~f2$x)
>regcatsim=lm(f2$y~f2$x+factor(f2$z))
>reginter=lm(f2$y~factor(f2$z)+f2$x+f2$x*factor(f2$z))  #equivalente a lm(y~x*factor(z), data=f2)
```

Observe cómo la formula x*factor(z) representa a la fórmula expandida: 1 + x + factor(z) + x:factor(z). Es decir, incluye todos los términos de menor orden[1].

3.Comparación de modelos anidados con anova

La función anova permite comparar modelos anidados o jerárquicos (todos los términos en un modelo previo aparecen en el siguiente modelo, más algunos otros adicionales) para contrastar la significatividad de los términos que van añadiéndose. El orden en que se van añadiendo afectará a los resultados. La contribución al ajuste de cada término que entra en el modelo, depende también de lo que aportan o explican los términos ya introducidos. Por eso antes de decidir qué modelo seleccionar es preciso tener en cuenta esto y probar las posibilidades que pueden presentarse.

```
> reg0=lm(f2$y~1)  #Modelo más simple con sólo intercept
> reg0

Call:
lm(formula = f2$y ~ 1)

Coefficients:
(Intercept)
      4.667


> anova(reg0, regsimple, regcatsim, reginter)
Analysis of Variance Table

Model 1: f2$y ~ 1
Model 2: f2$y ~ f2$x
Model 3: f2$y ~ f2$x + factor(f2$z)
Model 4: f2$y ~ factor(f2$z) + f2$x + f2$x * factor(f2$z)
  Res.Df     RSS Df Sum of Sq       F  Pr(>F)
1      5 29.3333
2      4  3.8636  1   25.4697 18.6040 0.04977 *
3      3  2.9167  1    0.9470  0.6917 0.49307
4      2  2.7381  1    0.1786  0.1304 0.75256
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

La variable x, es significativa al nivel alfa del 5% (pvalor=0.04977)
Observe que el factor z cuando entra a formar parte del modelo, una vez que x ha sido introducido, tiene poca capacidad para añadir a la explicación de la variable dependiente y. Vea que la significatividad de este término es 0,4931. Es decir, no es útil para explicar y.
Con este chequeo anova, el modelo definitivo es el que sólo incluye a x como variable independiente (regsimple).

$$Y = 0.3182 + 1.8636\ X$$

```
> anova(reg0, regcat, regcatsim, reginter)
```

[1] Cuando hay términos interacción significativos, un término de menor orden que sea no significativo, no quiere decir que no sea necesario. Deberán incluirse en el modelo todos los términos de menor orden sean o no significativos (principio jerárquico).

```
Analysis of Variance Table

Model 1: f2$y ~ 1
Model 2: f2$y ~ factor(f2$z)
Model 3: f2$y ~ f2$x + factor(f2$z)
Model 4: f2$y ~ factor(f2$z) + f2$x + f2$x * factor(f2$z)
  Res.Df     RSS Df Sum of Sq       F  Pr(>F)
1      5 29.3333
2      4 23.3333  1    6.0000  4.3826 0.17136
3      3  2.9167  1   20.4167 14.9130 0.06099 .
4      2  2.7381  1    0.1786  0.1304 0.75256
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

En el anova último, el factor z entra a formar parte del modelo cuando x aún no ha sido introducido. Si chequeamos aquí su capacidad explicativa de la variable dependiente, podemos comprobar que tampoco resulta útil, aunque su nivel significativo difiere (pvalor=0.17136; vea resultado primero). Una vez introducido el factor z, la variable x pierde poder explicativo, vea que la significatividad de este término es ahora 0,06099, que tampoco llega a ser significativo, aunque está próximo. Si las variables se introducen en este último orden, ninguna de ellas aparece significativa.

4.Búsqueda automática de los términos significativos del modelo con la función step

Un modo útil de explorar la significatividad de los términos de un modelo cuando no se tiene una propuesta teórica sustantiva, es decir, un modelo coherente teóricamente formulado, es mediante la función step. Sirve como guía en una primera aproximación para encontrar el modelo más parsimonioso posible y que mejor se ajuste a los datos.

>step(reginter)

```
Start:  AIC=3.29
f2$y ~ factor(f2$z) + f2$x + f2$x * factor(f2$z)

                     Df Sum of Sq    RSS    AIC
- factor(f2$z):f2$x  1    0.1786 2.9167 1.6721
<none>                            2.7381 3.2930

Step:  AIC=1.67
f2$y ~ factor(f2$z) + f2$x

               Df Sum of Sq     RSS     AIC
- factor(f2$z)  1    0.9470  3.8636  1.3591
<none>                      2.9167  1.6721
- f2$x          1   20.4167 23.3333 12.1487

Step:  AIC=1.36
f2$y ~ f2$x

       Df Sum of Sq     RSS     AIC
<none>               3.8636  1.3591
- f2$x  1   25.4697 29.3333 11.5218

Call:
lm(formula = f2$y ~ f2$x)

Coefficients:
(Intercept)         f2$x
     0.3182       1.8636
```

Partiendo del modelo reginter (en este caso particular, hemos tomado el más complejo) se procede a dejar en el modelo sólo los términos que sean relevantes según sea o no significativo el descenso producido en el valor del estadístico AIC[2] (Criterio de Información de Akaike).
Cuanto más se reduzca AIC al incluir un término en el modelo, mejor indicio para la importancia de éste, pero eso no necesariamente conlleva que la reducción sea

[2] AIC = -2 máx log verosimilitud + 2 nºparámetros

significativa. Al quitar un término de un modelo puede ocurrir que se aumente o se reduzca AIC. La decisión para mantener o quitar un término es en función de su significatividad.

El proceso generado permite seleccionar automáticamente el modelo de regresión simple que incluye a x y excluye a z.

Nota: una vez seleccionados los términos del modelo mediante step, es aconsejable ajustar de nuevo el modelo y contrastar la hipótesis de significatividad de los términos seleccionados con el test t.

```
> summary(regsimple)

Call:
lm(formula = f2$y ~ f2$x)

Residuals:
       1        2        3        4        5        6
-0.18182 -0.04545 -0.90909  0.81818 -0.90909  1.22727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.3182     0.9371   0.340  0.75127
f2$x          1.8636     0.3629   5.135  0.00681 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Los resultados corroboran que el modelo seleccionado es el adecuado. X es significativa (p-valor < 0.01) por tanto, sirve para explicar y.

Esta función tiene también otros argumentos importantes como scope y direction.

5.Regresión usando solo un subconjunto de datos del data frame

```
>rega=lm(f2$y~f2$x, subset=f2$z==”a”)   #equivalente a rega=lm(y~x, data=f2, subset=z==”a”)
>regb=lm(f2$y~f2$x, subset=f2$z==”b”)
```

5.1.La función update

Esta función permite modificar el modelo ya ajustado. Toma como argumento el modelo ya ajustado y el resto de los argumentos añadidos representan modificaciones de los argumentos primitivos del objeto modelo ajustado. Por ejemplo, datos, fórmula, etc.)

La regression regb, una vez que se ha ejecutado rega, es equivalente a usar la opción **update**, es decir , actualizar el análisis con los datos que cumplen la nueva condición impuesta:

```
>regb=update(rega, subset= f2$z==”b”)
```

```
> rega

Call:
lm(formula = f2$y ~ f2$x, subset = f2$z == "a")

Coefficients:
(Intercept)         f2$x
     0.6667       1.5000

> regb

Call:
lm(formula = f2$y ~ f2$x, subset = f2$z == "b")

Coefficients:
(Intercept)         f2$x
     0.7143       1.8571
```

Update resulta muy útil cuando se ha ejecutado regresión y aparecen algunos puntos anómalos con residuos grandes y se desea realizar el análisis sin dichos casos. En este caso se actualiza el data quitando los puntos que distorsionan el ajuste.

También es útil para hacer pequeñas modificaciones sobre la fórmula del modelo ajustado. Por ejemplo, para añadir o quitar términos.

update(regsimple, .~. +factor(f2$z)) #añade el factor z. Es decir, la fórmula es ahora y~x+factor(z)

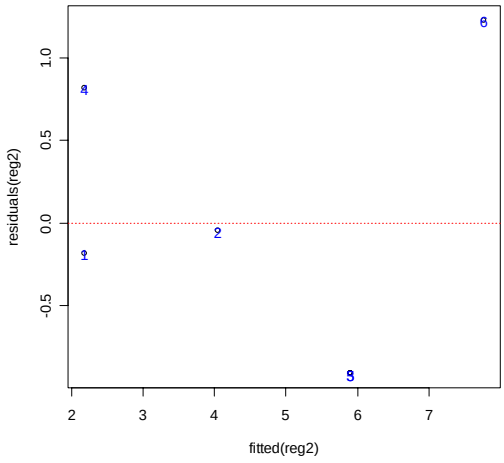
6.Diagnóstico de residuos

Tanto los valores ajustados como los residuos pueden inspeccionarse para decidir sobre la bondad del ajuste del modelo. Pueden etiquetarse los casos con el número de fila en que se encuentran para identificarlos, por si fuera necesario eliminarlos del análisis.

6.1.Gráfico de valores ajustados frente a residuos:

Para el modelo con solo una variable cuantitativa

```
> reg2=lm(f2$y~f2$x)
> plot(fitted(reg2),residuals(reg2))
> abline(h=0,lty=3,col=2)
> text(fitted(reg2),residuals(reg2),labels=rownames(f2),col=4)
```



Regresión con variable cualitativa y cuantitativa

```
> reg3=lm(f2$y~f2$x+factor(f2$z))
> reg3

Call:
lm(formula = f2$y ~ f2$x + factor(f2$z))

Coefficients:
  (Intercept)           f2$x  factor(f2$z)b
       0.1667         1.7500          0.8333
```

Representación en la misma pantalla de los dos gráficos

```
op=par(mfrow=c(1,2))

plot(fitted(reg2),residuals(reg2))
abline(h=0,lty=3,col=2)
```

```
text(fitted(reg2),residuals(reg2),labels=rownames(f2),col=4)

plot(fitted(reg3),residuals(reg3))
abline(h=0,lty=3,col=2)
text(fitted(reg3),residuals(reg3),labels=rownames(f2),col=4)

par(op)
```

6.2.Análisis gráfico de los residuos y valores ajustados para todos los modelos propuestos

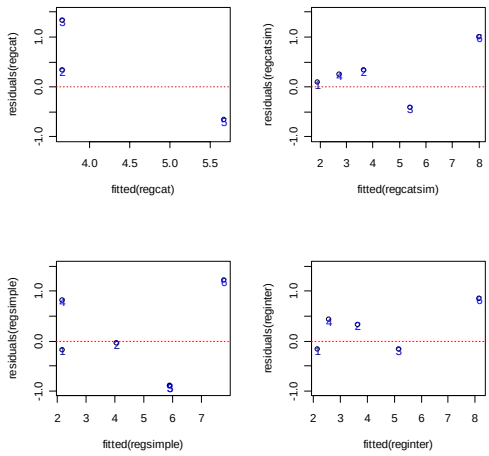
```
layout(matrix(1:4, nrow = 2))

plot(fitted(regcat),residuals(regcat),ylim=c(-1,1.5))
abline(h=0,lty=3,col=2)
text(fitted(regcat),residuals(regcat),labels=rownames(f2),col=4)

plot(fitted(regsimple),residuals(regsimple),ylim=c(-1,1.5))
abline(h=0,lty=3,col=2)
text(fitted(regsimple),residuals(regsimple),labels=rownames(f2),col=4)

plot(fitted(regcatsim),residuals(regcatsim),ylim=c(-1,1.5))
abline(h=0,lty=3,col=2)
text(fitted(regcatsim),residuals(regcatsim),labels=rownames(f2),col=4)

plot(fitted(reginter),residuals(reginter),ylim=c(-1,1.5))
abline(h=0,lty=3,col=2)
text(fitted(reginter),residuals(reginter),labels=rownames(f2),col=4)
```



6.3.Resumen de funciones usadas en el ejemplo de análisis de regresión:

```
f=edit(data.frame())

lm(f$y~f$x)
lm(f2$y~factor(f2$z))
summary(lm(f$y~f$x))
fitted(lm(f$y~f$x))
residuals(lm(f$y~f$x))
regb=update(rega, subset= f2$z==”b”)
update(regsimple, .~. +factor$f2z)  #añade el factor z.  Es decir, la fórmula es ahora y~x+factor(z)
anova(reg0, regsimple, regcatsim, reginter)
step(reginter)
```

```
plot(f$x,f$y)
abline(lm(f$y~f$x))
text(fitted(reg2),residuals(reg2),labels=rownames(f2),col=4)
```