USO DE HERRAMIENTAS INFORMÁTICAS PARA LA RECOPILACIÓN, ANÁLISIS E INTERPRETACIÓN DE DATOS DE INTERÉS EN LAS CIENCIAS BIOMÉDICAS

Módulo 4
Análisis Multivariado de datos

Alfredo Rigalli
Maela Lupo
María Eugenia Chulibert
Mercedes Lombarte
Patricia Lupión

Centro Universitario de Estudios Medioambientales
Facultad de Ciencias Médicas
Universidad Nacional de Rosario

2019

USO DE HERRAMIENTAS INFORMÁTICAS PARA LA RECOPILACIÓN, ANÁLISIS E INTERPRETACIÓN DE DATOS DE INTERÉS EN LAS CIENCIAS BIOMÉDICAS

Análisis Multivariado de Datos

Alfredo Rigalli
Maela Lupo
María Eugenia Chulibert
Mercedes Lombarte
Patricia Lupión

Centro Universitario de Estudios Medioambientales
Facultad de Ciencias Médicas
Universidad Nacional de Rosario

Uso de herramientas informáticas para la recopilación, análisis e interpretación de datos de
interés en las ciencias biomédicas : análisis multivariado de datos numéricos / Alfredo Rigalli ...
[et al.]. - 1a edición para el alumno - Rosario : Alfredo Rigalli, 2019.
Libro digital, PDF

Archivo Digital: descarga y online
ISBN 978-987-86-1179-2

1. Análisis de Datos. 2. Tecnología Biomédica. I. Rigalli, Alfredo
CDD 610

AUTORES

Chulibert, María Eugenia: Licenciada en nutrición. Estudiante del doctorado en Ciencias Biomédicas de la Facultad de Ciencias Médicas de la Universidad Nacional de Rosario. Becaria doctoral del CONICET.

Lombarte, Mercedes: Licenciada en biotecnología y Doctora en Ciencias Biomédicas. Investigadora del Laboratorio de Biología Ósea y del Centro Universitario de Estudios Medioambientales y docente de la cátedra de Química Biológica de la Facultad de Ciencias Médicas de la Universidad Nacional de Rosario.

Lupión, Patricia: Licenciada en biotecnología. Estudiante del doctorado en Ciencias Biomédicas de la Facultad de Ciencias Médicas de la Universidad Nacional de Rosario. Becaria doctoral del CONICET.

Lupo, Maela: Licenciada en biotecnología y Doctora en Ciencias Biomédicas. Investigadora del Laboratorio de Biología Ósea y del Centro Universitario de Estudios Medioambientales y docente de la cátedra de Química Biológica de la Facultad de Ciencias Médicas de la Universidad Nacional de Rosario.

Rigalli, Alfredo: Bioquímico y Doctor en bioquímica. Investigadora del Laboratorio de Biología Ósea y del Centro Universitario de Estudios Medioambientales y docente de la cátedra de Química Biológica de la Facultad de Ciencias Médicas de la Universidad Nacional de Rosario. Investigador independiente del Consejo de Investigaciones de la UNR y del CONICET

Tabla de contenidos

1.Clase 4.1....7
1.1.Funciones 7
1.2.Modelos lineales....8
1.2.1.Modelo lineal con una variable independiente: Regresión lineal....9
1.2.2.Test de potencia para un modelo lineal de una variable independiente....13
1.3.Modelos no lineales....17
2.Clase 4.2....23
2.1.Revisión modelo lineal con una variable independiente....23
2.2.Modelos lineales con más de una variable independiente....26
2.3.Modelo lineal sin interacción entre las variables independientes....26
2.4.Modelo lineal con interacción entre las variables independientes....29
2.5.Otros recursos con modelos lineales....30
2.5.1.Conocer la fórmula del modelo....31
2.5.2.para conocer los coeficientes....31
2.5.3.conocer residuales....31
2.5.4.Grafica de los residuales....32
2.5.5.Valores modelizados de la variable dependiente....32
2.5.6.Intervalo de confianza....32
2.6.Graficar un modelo lineal....33
2.6.1.Agregado de puntos experimentales....37
3.Clase 4.3....39
3.1.Modelos lineales. Continuación....39
3.2.Multiple regression stepwise analysis....39
3.3.Comparación de dos modelos lineales....42
3.4.Test de potencia para un modelo lineal....44
3.5.modelos lineales con variables categóricas....48
3.6.Reasignación de niveles....51
4.Clase 4.4....53
4.1.Regresión logística....53
4.2.Regresión logística con una variable independiente cualitativa....53
4.3.Regresión logística con variables continuas y categóricas....56
4.4.Elegir las variables más adecuadas para el modelo....57
5.Clase 4.5....58
5.1.Análisis de los componentes principales....58
5.2.Inclusión de variables suplementarias cuantitativas....63
5.3.Inclusión de variables cualitativas....65
6.Clase 4.6....70
6.1.Análisis de las correspondencias múltiples....70
7.Clase 4.7....76
7.1.Análisis de las correspondencias múltiples....76
7.1.1.Construcción de elipses de confianza....82
7.2.Clasificación jerárquica de los componentes principales 86
7.3.Sumary del objeto hcpctablaR471....89
8.Clase 4.8....92
8.1.Introducción al análisis ROC....92
8.2.Análisis ROC con R....94
8.3.Comparación de dos pruebas diagnósticas....99

9.Clase 4.9.....101
9.1.Curvas de sobrevida – análisis de supervivencia.....101
9.2.Análisis de supervivencia clasificando por un factor.....103
9.3.Comparación de dos o más curvas de sobrevida.....105
9.4.Regresión de Poisson.....105

1. Clase 4.1

Video: https://youtu.be/U8P8J5XHzto
Tabla de datos: http://hdl.handle.net/2133/15481

1.1. Funciones

Una función es una relación matemática en que una variable cuantitativa aparecerá como una variable dependiente o bien como una independiente habitualmente multiplicada, dividida o sumada a una constante o parámetros y las variables cualitativas aparecerán afectando el valor de uno de esos parámetro.

Recordemos como se define una función en R. La forma general de definirla y asignarla a un objeto es

nombrefunción<-function(variable){expresión}
Por ejemplo si quisiéramos utilizar en R la función y=2*x+1, el código para introducir la función sería

```
> y<-function(x){2*x+1}
```

la función quedó incorporada en el objeto llamado "y"

Esta función tiene dos parámetros: 2 y 1, una variable dependiente: y y una variable independiente x.

Si deseamos conocer el valor de y para x=2

```
> y(2)
[1] 5
```

Si tuviéramos un vector formado por valores de x

```
> c<-c(1,2,3)
```

y deseamos conocer el valor de la función para cada uno de ellos

```
> y(c)
[1] 3 5 7
```

Si el rango conocido de valores de x fuera [0,5] y en base a este rango hallamos la función con la que estamos trabajando y deseamos calcular

```
> y(3)
[1] 7
```

estaríamos realizando una interpolación

pero si calculáramos

```
> y(8)
[1] 17
```

estaríamos haciendo una extrapolación.

Una función así definida es un modelo matemático que representa la relación entre dos o más variables. En esta primer clase del módulo trabajaremos con funciones que tiene solo dos variables: una dependiente y otra independiente. Utilizando el modelo hallado en base a un rango de valores, nos permite hacer predicciones de que ocurrirían en un rango diferente de valores o en valores del intervalo que no han sido explorados.

Podríamos tener funciones de más de una variable. En el caso siguiente tenemos dos variables independientes: x e y.

```
> z<-function(x,y){2*x+2*y+1}
```

Si deseamos calcular el valor de z para los valores x=1 e y=1

```
> z(1,1)
[1] 5
```

Debemos recordar que para una función de más de una variable, los valores de la variable independiente serán asignados en el orden que fueron definidos en la función. En z<-function(x,y)..... x es la primer variable e y la segunda. Por lo tanto en z(1,1) se sobre entiende que el primer valor es para la x y el segundo para la y.

Si redefinimos la función por el cambio en uno de su parámetros

```
> z<-function(x,y){2*x+3*y+1}
```

el valor de z para los mismos valores de las variables independientes será

```
> z(1,1)
[1] 6
```

para otro par de valores

```
> z(0,-1)
[1] -2
```

Podemos tener funciones más complejas que incluyan otro tipo de operaciones y funciones, como se muestra en la función w, donde las variables independientes las llamamos: a, b y c.

```
> w<-function(a,b,c){2*a+3*log(b)+5/c}
```

si deseamos el valor de w para tres valores de a, b y c

```
> w(1,10,2)
```

obtendremos

```
[1] 11.40776
```

probemos otra terna de valores

```
> w(1,0,2)
```

```
[1] -Inf
```

El valor hallado es un límite y no un resultado. El log de cero no existe, pero es un valor que tiene a valer menos infinito.

Lo mismo ocurre con

```
> w(1,10,0)
```

```
[1] Inf
```

La división por cero tampoco existe.

Las funciones nos servirán como herramientas para modelizar fenómenos observados o datos obtenidos de experimentos. Comenzaremos con los modelos lineales

1.2. Modelos lineales

Un modelo lineal es una ecuación que explica la variación de los valores de una variable cuantitativa (variable dependiente) en función de otras llamadas variables independientes que

pueden ser cuantitativas y/o cualitativas. Las variables independientes están afectadas en los modelos lineales por operaciones algebraicas como sumas, restas, productos y cocientes. Aunque veremos que el método utilizado en R también puede ser aplicado a otros casos. Veremos en primer lugar los modelos lineales de una variable dependiente y una independiente que conocemos como regresión lineal de una variable. Este tema ya fue desarrollado en el módulo 3 con otro enfoque, y se repetirá en este módulo como introducción a los modelos lineales con más de una variable independiente.

1.2.1. Modelo lineal con una variable independiente: Regresión lineal

Para revisar el tema se recomienda revisar la clase 8 del módulo 3.

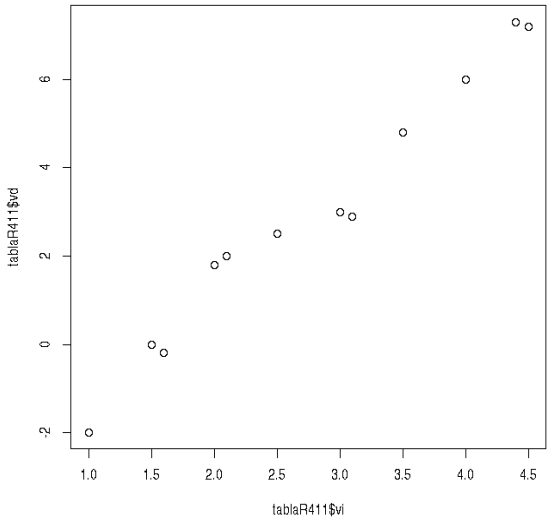
Introduzcamos en nuestro espacio de trabajo los datos de las variables vi y vd de la tablaR411 de la planilla de cálculo tablaR4-1.xls/ods. Las variables vi y vd constituyen pares de valores que han sido medidos simultáneamente sobre un dado sistema o unidad experimental.

```
> tablaR411<-read.table("clipboard",header=T,sep="\t",dec=",",encoding="latin1")
```

```
> tablaR411
   vi   vd
1  1.0 -2.0
2  1.5  0.0
3  2.0  1.8
4  2.5  2.5
5  3.0  3.0
6  3.5  4.8
7  4.0  6.0
8  4.5  7.2
9  1.6 -0.2
10 2.1  2.0
11 3.1  2.9
12 4.4  7.3
```

y graficamos las variables vi y vd. La observación de la gráfica de los datos es importante a la hora de elegir un modelo.

```
> plot(tablaR411$vi,tablaR411$vd)
```



Es claro en este caso que los datos tiene una distribución de manera de estar próximo a una recta, por lo que podemos relacionar a la variable vi y vd a través de la ecuación de una recta. Pero bien podría ser otro tipo de gráficas que describa la relación de una mejor manera.
Una recta tiene parámetros que podemos calcularlos con la función lm()

Recordemos que una recta tiene una ecuación matemática del tipo

y = a*x + h

que para nuestro caso sería

vd= a*vi + h

donde vd es la variable dependiente, vi la variable independiente. a: es la pendiente que da la inclinación de la recta, que deberá ser positiva en este caso.

h: la ordenada al origen, es decir el valor de vd para un valor 0 de vi. En otras palabras el valor en que la recta corta al eje vertical (de la variable vd)

Entonces nos inclinamos porque el modelo que relaciona a vi y vd es un modelo lineal. Aplicamos lm() y su resultado lo asignamos a un objeto

```
> lmtablaR411<-lm(vd~vi,data=tablaR411)
```

La tabla anova() nos indicará si la variación de la variable vd tiene dependencia de vi.

```
> anova(lmtablaR411)
```

```
Analysis of Variance Table
Response: vd
           Df  Sum Sq Mean Sq F value    Pr(>F)
vi          1  92.599  92.599  346.84 4.304e-09 ***
Residuals  10   2.670   0.267
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

El valor Pr(<F)= 4,304e-09 nos indica que la variación de la variable vd se explica (o se asocia) significativamente a variaciones de la variable vi. Esto no necesariamente implica una relación de causa efecto.

Si pedimos un summary() del objeto lmtablaR411 tendremos más información. Veremos una parte de ella que resaltamos en amarillo

```
> summary(lmtablaR411)

Call:
lm(formula = vd ~ vi, data = tablaR411)

Residuals: [1]
     Min       1Q   Median       3Q      Max
-0.86749 -0.31840  0.02216  0.24220  0.75772
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -3.9126     0.3971  -9.852 1.82e-06 ***
vi             2.4775     0.1330  18.624 4.30e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.5167 on 10 degrees of freedom
Multiple R-squared:  0.972,     Adjusted R-squared:  0.9692
F-statistic: 346.8 on 1 and 10 DF,  p-value: 4.304e-09
```

Interpretación del summary()

El valor 4.30e-09, de la columna Pr(>|t|), al ser mucho menor que 0.05 nos indica que la pendiente (vi en la tabla anterior) es significativamente diferente de cero y tiene un valor de 2,4775. Es decir que la vd aumenta 2,4775 unidades por cada una unidad que aumenta vi. Por otra parte el valor 1.82e-06, de la columna Pr(>|t|), al ser mucho menor que 0.05 nos indica que la ordenada al origen (intercept, en la tabla) es significativamente diferente de cero.

Multiple R-squared: 0.972
es el valor del coeficiente de correlación, que recordemos si es cercano a 1 o -1 indica también un buen ajuste. En este caso es muy cercano a 1 y es un ajuste significativo como lo indica el valor de p-value

p-value: 4,304e-09

nos indica un que el ajuste de la recta a los puntos es altamente significativo.

Es decir entonces que nuestro modelo quedará conformado como

vd = 2,4775 * vi – 3,9126

podemos graficar nuestro modelo. Para ello seguiremos unas serie de pasos

Confirmemos el rango de variación de la variable vi.

```
> range(tablaR411$vi)
```

1 Los residuales nos están sosteniendo si el modelo elegido es representativo de nuestros datos. Una mediana cercana a cero, primer y tercer cuartilo (1Q y 3Q) similares en valor absoluto pero de signo contrario y Max y mín con las mismas características avalan la buena elección del modelo.

[1] 1.0 4.5

pidamos datos del modelo hallado. Otros se pueden pedir con summary()

> lmtablaR411

Call:
lm(formula = vd ~ vi, data = tablaR411)
Coefficients:
(Intercept) vi
-3.913 2.477

Si colocamos el nombre del modelo seguido de $ y coefficients, podemos obtener los parámetros ya calculados y vistos anteriormente. Como vemos el parámetro [1] es la ordenada al origen y le [2] la pendiente

> lmtablaR411$coefficients[1]

(Intercept)
-3.912649
> lmtablaR411$coefficients[2]
vi
2.477464

Resumiendo: los parámetros del modelo lineal: pendiente y ordenada al origen podemos hallarlos de tres maneras: summary(lmtablaR411), lmtablaR411 y lmtablaR411$coefficientes[1] y lmtablaR411$coefficientes[2]

construimos nuestro modelo (la función) y lo colocamos en un objeto

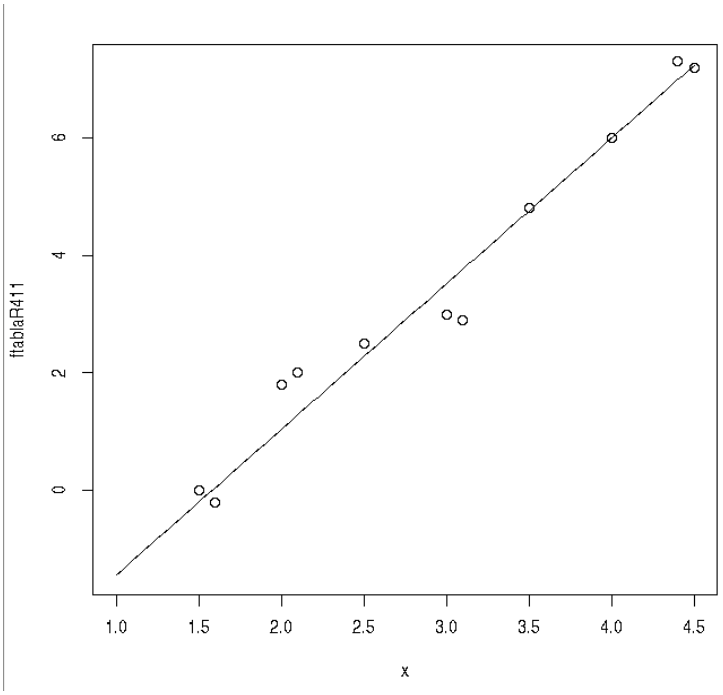
> ftablaR411<-function(vi){lmtablaR411$coefficients[2]*vi+lmtablaR411$coefficients[1]}

graficamos la función para el rango de valores de vi calculado recientemente

> plot(ftablaR411,1,4.5)

luego colocamos nuestros datos con la función points()

> points(tablaR411$vi,tablaR411$vd)



Sin duda el ajuste es bueno.

El error de tipo I del lm() nos asegura que rechazamos bien cualquier otro modelo. Pero cuan acertados estamos en nuestra elección de un modelo lineal. Es decir cual es la potencia de nuestro ensayo?

1.2.2. Test de potencia para un modelo lineal de una variable independiente

El paquete pwr tiene una función que permite calcular la potencia para modelos lineales. La regresión lineal es el modelo más sencillo.

En general la función se escribe

pwr.f2.test(u = NULL, v = NULL, f2 = NULL, sig.level = NULL, power = NULL)

donde

u: grados de libertad del numerador

v: grados de libertad del denominador

f2: effect size

sig.level: probabilidad de error tipo I

power: potencia del ensayo.

recurrimos entonces a la tabla anova

```
> anova(lmtablaR411)
Analysis of Variance Table
Response: vd
           Df  SumSq Mean Sq F value    Pr(>F)
vi          1 92.599  92.599  346.84 4.304e-09 ***
```

```
Residuals    10        2.670          0.267
-
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Los valores de u y v, salen de la columna de grados de libertad (Df)

u= 1

v= 10

El effect size podemos estimarlo por el parámetro de nuestra recta que representa la pendiente. Para ello ejecutamos nuevamente

```
> summary(lmtablaR411)

Call:
lm(formula = vd ~ vi, data = tablaR411)
Residuals:
    Min      1Q  Median      3Q     Max
-0.86749 -0.31840  0.02216  0.24220  0.75772

Coefficients:
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)   -3.9126       0.3971    -9.852   1.82e-06 ***
vi             2.4775       0.1330    18.624   4.30e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5167 on 10 degrees of freedom
Multiple R-squared:  0.972,          Adjusted R-squared:  0.9692
F-statistic: 346.8 on 1 and 10 DF,  p-value: 4.304e-09
```

La línea resaltada en amarillo nos está diciendo que la pendiente es significativamente diferente de cero, por lo que podemos considerar un effect size =large. Es decir lo asumimos importante.

Entonces aplicamos una función de pwr que nos permite conocer la magnitud del efecto. Para ello utilizaremos la función cohen.ES() que nos permite calcular la magnitud del efecto.

```
> cohen.ES(test = "f2", size = "large")

    Conventional effect size from Cohen (1982)

         test = f2

         size = large

   effect.size = 0.35
```

Aplicamos la función para calcular potencia de una regresión lineal

```
> pwr.f2.test(u = 1, v = 10, f2 = 0.35, sig.level = 0.05, power = NULL)


    Multiple regression power calculation
```

u = 1

v = 10

f2 = 0.35

sig.level = 0.05

power = 0.4571621

Nos indica que aceptamos que los puntos son ajustados por una recta con un error de tipo I de 0,05 y por otro lado la certeza de que la recta es una buena opción de ajuste es del 45,7 %.

Por lo tanto nuestro modelo es

vd = 2,4778 * vi - 3,9126

Como anticipamos este modelo nos permite interpolar. Por ejemplo en nuestro experimento vi variaba entre los siguientes valores

> range(tablaR411$vi)

[1] 1.0 4.5

y los valores medidos están dados por el objeto tablaR411

> tablaR411

	vi	vd
1	1.0	-2.0
2	1.5	0.0
3	2.0	1.8
4	2.5	2.5
5	3.0	3.0
6	3.5	4.8
7	4.0	6.0
8	4.5	7.2
9	1.6	-0.2
10	2.1	2.0
11	3.1	2.9
12	4.4	7.3

supongamos que quisiéramos predecir cuanto valdría vd si vi toma el valor 1.8. Este valor no está entre los medidos, pero solo basta realizar el siguiente cálculo para tener un valor en base a nuestro modelo.

vd = 2,4775 * vi - 3,9126

vd = 2,4775 * 1,8 - 3,9126

vd= 0,5459

que también podríamos resolver con nuestra función: ftablaR411

> ftablaR411(1.8)

vi

0.5467852

La pequeña diferencia se debe a que los valores utilizados en el primer cálculo están truncados a 4 decimales. En este caso hemos hecho una interpolación.

También podríamos extrapolar. Por ejemplo queremos predecir cuanto valdría vd si vi = 10. Claramente el valor 10 está fuera del intervalo de valores medidos y en base a los cuales se eligió y validó el modelo. Podría ocurrir que ya para esos valores no valga el modelo. Si suponemos que sí vale el modelo, el cálculo sería sencillo

```
> ftablaR411(10)
    vi
20.86199
```

Podemos hacer también un experimento "in silico". Por ejemplo deseamos tener 10 valores de vd cuando vi=2. Obviamente si aplicamos el modelo al valor 2 obtendremos un solo valor. Pero podemos colocar una función de generación de número aleatorios entre 0 y 1 como la función runif(), que ya hemos utilizado.

para el valor vi=2, vd valdría

```
> ftablaR411(2)
    vi
1.042278
```

pero si a la función la multiplico por (1+runif()), me generará valores cercanos al valor indicado pero aleatoriamente distribuidos, ubicados en vi = 2

```
> ftablaR411(2)*(1+runif(10,-0.5,0.5))
 [1] 1.0772383 1.5118169 0.8242088 1.4579229 0.9049450 0.6320793 1.4649221
 [8] 0.9873621 0.6517581 0.8958611
```

Su cálculo no dará los mismos resultados que los mostrados en la línea anterior, ya que la función runif() genera número aleatorios dentro de un cierto intervalo.

veamos ahora graficados, estos valores aleatorios.

creamos un vector con 10 valores de vi=2, que actúen como abscisa de los valores simulados.

```
> viinsilico<-c(2,2,2,2,2,2,2,2,2,2)
```

graficamos el modelo en el intervalo [1,4.5]

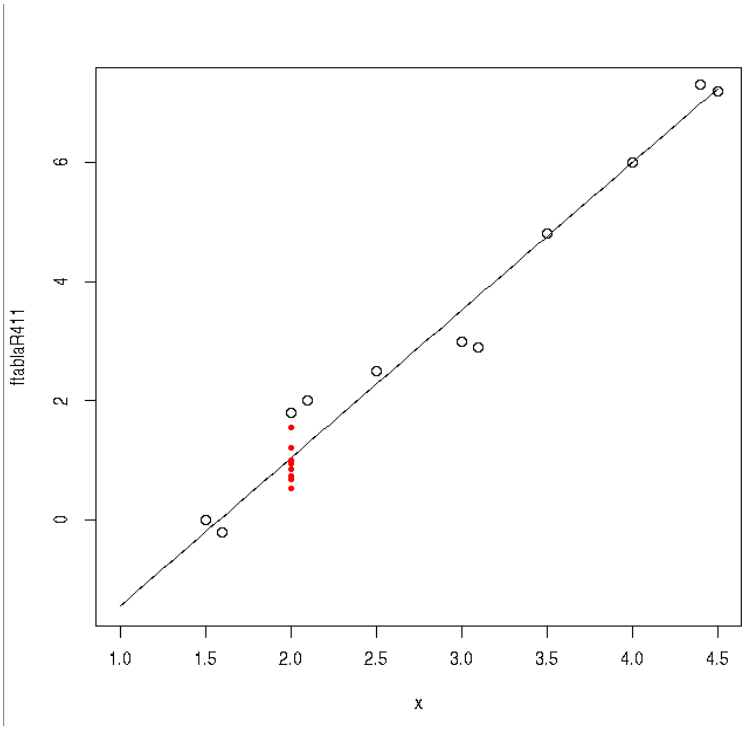
```
> plot(ftablaR411,1,4.5)
```

colocamos los puntos experimentales

```
>points(tablaR411$vi,tablaR411$vd,cex=1)
```

graficamos los valores de vd simulados para un valor de vi=2, que estarán en un rango de +/- el 50% del valor anticipado por el modelo por el modelo.

```
> points(viinsilico,ftablaR411(2)*(1+runif(10,-0.5,0.5)),cex=1,pch=20,col="red")
```



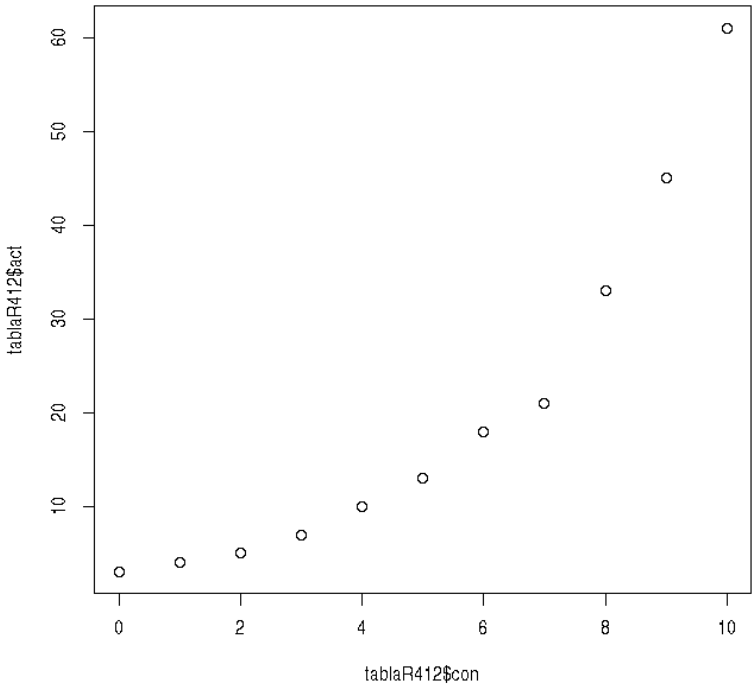
1.3. Modelos no lineales

Un modelo no lineal es aquel en que la relación entre las variables no podrá representarse por una línea recta. Veamos el tema analizando los datos de la tablaR412. Introduzcamos los valores

```
> tablaR412<-read.table("clipboard",header=T,sep="\t",dec=",",encoding="latin1")
> tablaR412
   con act
1    0   3
2    1   4
3    2   5
4    3   7
5    4  10
6    5  13
7    6  18
8    7  21
9    8  33
10   9  45
11  10  61
```

grafiquemos estos valores

```
> plot(tablaR412$con,tablaR412$act)
```



vemos claramente que los puntos no se ubican siguiendo una recta. Entonces si deseamos modelizar el fenómeno estaremos frente a un modelo no lineal. La elección de la ecuación que modelice dicho fenómeno excede a este curso, por lo cual aceptaremos la función propuesta.
Se propone como modelo una función exponencial de la forma

$$act = e^{(a * con)} + b$$

Ecuación 1.1.

donde 'act' es la variable dependiente, 'con' la independiente, a y b los parámetros de la función.

Para asegurarnos si la función puede describir el comportamiento de los datos podemos crear la función en R y graficarla

```
> act<-function(con){exp(a*con)+b}
```

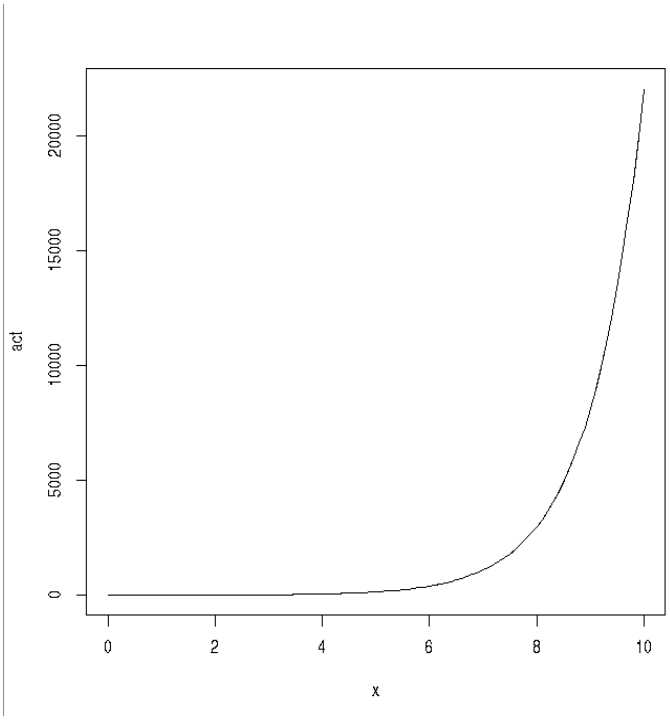
asignamos arbitrariamente valores a y, b

```
> a<-1
> b<-1
```

y graficamos para el intervalo de 'con' de nuestros datos: [0,10].

```
> plot(act,0,10)
```

obteniendo



como vemos el formado de la gráfica parece ser adecuado, aunque los valores obtenidos discrepan de los de la tablaR412. Esta discrepancia está en los valores asignados a los parámetros. Allí está ahora el trabajo, que es hallar los valores óptimos de los parámetros.

Para poder hallar los parámetros primero se deben tener valores aproximados de los mismos.

A través de metodologías que exceden este curso se buscan valores aproximados de los parámetros a y b. Estos procedimientos permiten obtener estos valores (que no son los únicos y podrían variar según el mecanismo utilizados)

a= 0,2

b-3

Estos valores se conocen como valores iniciales para el proceso de optimización. Entonces hallaremos los parámetros optimizados utilizando la función nls()

> ajuste<-nls(act~exp(a*con)+b,data=tablaR412,start=list(a=0.2,b=3),trace=T)

4893.992 : 0.2 3.0
684.0876 : 0.3726502 2.5269364
165.5872 : 0.3937845 3.2848308
32.47333 : 0.4088601 4.1372199
31.87251 : 0.4077449 4.1647468
31.87251 : 0.407746 4.163894

La función nls va buscando valores de los parámetros a y b y ajusta la función a los valores de la tablaR412 calculando el error. Como podemos ver en la primer fila de la tabla anterior, la primer columna es el valor del error, la segunda el valor del parámetro a y la tercera la del parámetro b. Como se observa en las líneas siguientes, se reduce el error y los parámetros van tendiendo a un valor. Cuando el error llega a un nivel prefijado, cesa la iteración y tenemos los valores de los

parámetros optimizados y que harían que el modelo planteado represente adecuadamente los datos obtenidos.

Veamos entonces si es correcto.

asignamos a los parámetros a y b los valores hallados

> a<-0.407746

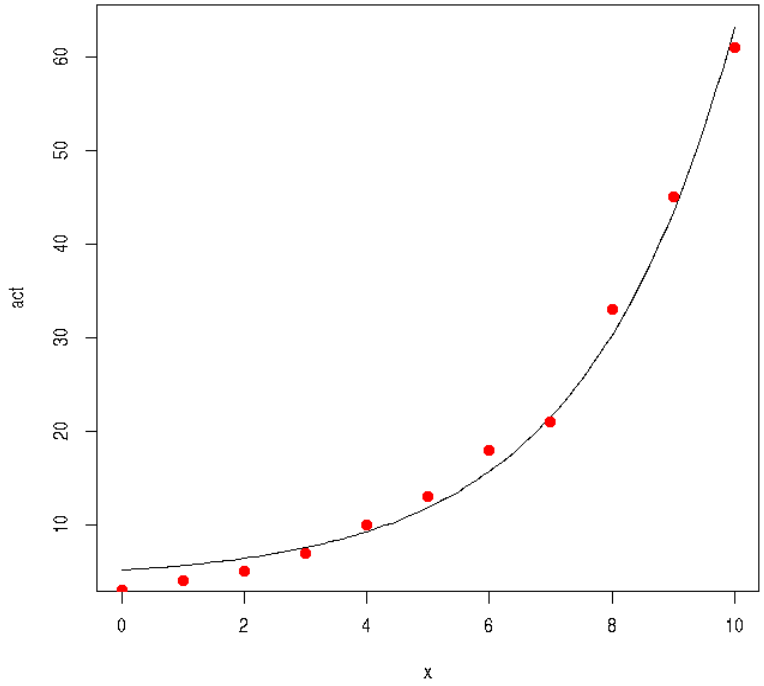
> b<-4.163894

graficamos el modelo. Recuerde que act es la función definida que incluye los parámetros a y b. Ahora a y b tomarán los valores asignados recientemente

> plot(act,0,10)

colocamos los puntos experimentales

> points(tablaR412$con,tablaR412$act,pch=20,col="red",cex=1.5)



Vemos que la línea que representa nuestro modelo se ajusta muy bien a los valores experimentales.

Podemos pedir datos de los resultados del proceso de optimización

> summary(ajuste)
Formula: act ~ exp(a * con) + b
Parameters:

	Estimate	Std. Error	t value	Pr(>|t|)
a	0.407746	0.003157 1	29.137	5.09e-16 ***
b	4.163894	0.698260	5.963	0.000212 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.882 on 9 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 8.16e-07

Supongamos que el valor del parámetro 'a' tuviera valores biológicos acotados, es decir con restricciones que impidieran ciertos valores. Por ejemplo supongamos que 'a' no pueda ser menor de 0,45 y b no puede ser menor de 3. Estas restricciones pueden introducirse en el proceso de optimización y ajuste.

```
>ajuste<-nls(act~exp(a*con)+b,data=tablaR412,start=list(a=0.5,b=3),lower=list(a=0.45,b=3),
trace=T, algorith="port")
  0:     5673.3381: 0.500000  3.00000
  1:     5559.0381: 0.499387  3.00000
  2:     5128.2771: 0.496986  3.00000
  3:     2285.1945: 0.475223  3.00000
  4:     668.46163: 0.450000  3.00000
  5:     668.46163: 0.450000  3.00000
```

Como podemos ver con estas restricciones es error mostrado en la primer columna no se pudo hacer tan pequeño como en el primer caso de optimización y los parámetros se fijaron justamente en los mínimos. Los asignamos a nuestros parámetros

```
> a<-0.45
```

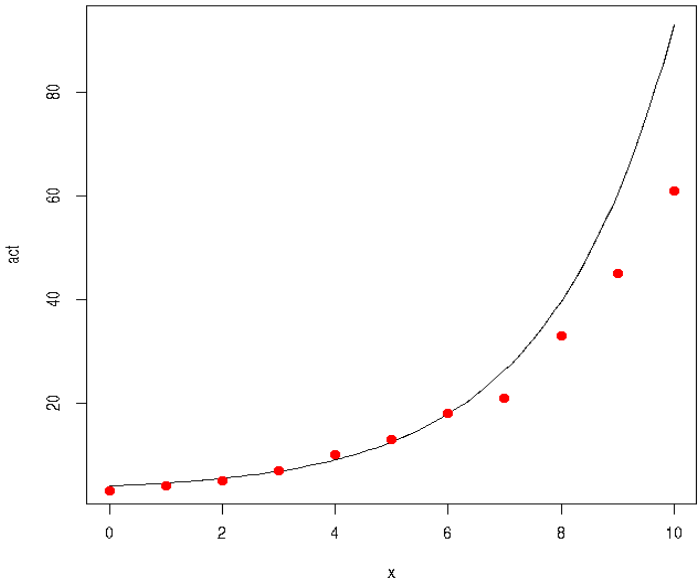
```
> b<-3
```

graficamos nuevamente nuestro modelo

```
> plot(act,0,10)
```

y colocamos nuestros puntos experimentales

```
> points(tablaR412$con,tablaR412$act,pch=20,col="red",cex=1.5)
```



Como podemos ver si bien el modelo elegido daba un muy buen ajuste, cuando los parámetros 'a' y 'b' se fijaban automáticamente. Al introducir restricciones impuestas por la realidad, el modelo ajusta muy bien para valores de la variable independiente menores a 6, pero no tiene buen ajuste para valores mayores. Nuestro modelo sobre-estima los valores reales.

2. Clase 4.2

Video: https://youtu.be/M7MoGE5R_VU
Tabla de datos: http://hdl.handle.net/2133/15482

2.1. Revisión modelo lineal con una variable independiente

En esta clase revisaremos nuevamente los conceptos y fundamentos de modelos lineales, resaltando algunos conceptos adicionales. Un modelo lineal es una relación donde la variable dependiente es cuantitativa y la o las variables independientes pueden ser cuantitativas o cualitativas. Al menos una de ellas debe ser cuantitativa.

Comenzaremos con el modelo más sencillo que es la regresión lineal

Introduzca en su espacio de trabajo la tablaR421 de la planilla de cálculo tablaR4-2.ods/xls

```
> tablaR421<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

```
> tablaR421
   numero  t   A
1       1  0 -12
2       2  1   1
3       3  2  -3
4       4  3   5
5       5  4   9
6       6  5  18
7       7  6  25
8       8  7  13
9       9  8  36
10     10  9  45
11     11 10  48
```

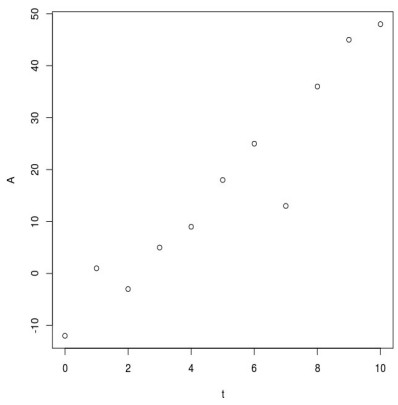
Consideramos que numero y t son variables independientes

A es la variable dependiente.

t y A son cuantitativas mientras que numero es cualitativa

Deseamos modelizar el comportamiento de A en función de t. Entonces graficamos A vs t

```
> plot(A~t,data=tablaR421)
```



Los puntos nos indican una relación creciente: a medida que aumenta t, aumenta A, relación que podríamos aproximar con una recta. Plantearemos un modelo lineal y luego verificaremos si la elección fue adecuada.

Aplicamos un análisis de regresión lineal, para lo cual podemos plantear el problema como un modelo lineal, para lo cual la función lm() es adecuada

```
> lmtablaR421<-lm(A~t,data=tablaR421)

> summary(lmtablaR421)

Call:
lm(formula = A ~ t, data = tablaR421)

Residuals:
   Min     1Q  Median     3Q    Max
-15.182  -1.296   1.182   2.636   6.909

Coefficients:
             Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)  -11.5909     3.4696      -3.341   0.00865 **
t              5.6818     0.5865       9.688   4.65e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.151 on 9 degrees of freedom
Multiple R-squared:  0.9125,        Adjusted R-squared:  0.9028
F-statistic: 93.86 on 1 and 9 DF,  p-value: 4.655e-06
```

¿Qué indica valor Pr(>|t|)?

En la línea (Intercept) el valor Pr(>|t|) = 0,00865, como este valor es menor que 0,05 nos está indicando que la ordenada al origen de la recta es significativamente diferente de cero. Es decir que en nuestro modelo podemos afirmar con una probabilidad de error de tipo I que cuando la variable t tome el valor 0, A tendrá un valor significativamente diferente de cero y que el modelo predice que tendrá una valor de -11,5909..

Por otra parte en la línea "t", el valor Pr(>|t|) = 4.65e-06, el cual es también mucho menor que 0,05, lo que indica que al aumentar t aumenta significativamente el valor de A. Es decir el valor de A varía significativamente con el cambio de t. Dicho de otra manera la pendiente de nuestra recta es significativamente diferente de cero, en este caso el modelo predice que tendrá un valor de 5,6818. En otras palabras cada unidad que varíe t, la variable A variará 5,6818.

Veamos ahora la tabla de ANOVA

```
> anova(lmtablaR421)

Analysis of Variance Table
Response: A
           Df  Sum Sq  Mean Sq  F value   Pr(>F)
t           1  3551.1   3551.1   93.863  4.655e-06 ***
Residuals   9   340.5     37.8
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Al analizar la tabla pongamos la atención en los siguientes valores

SumSq (linea t)= 3551.1 explica la variabilidad de los datos debido al efecto de t sobre A. SumSq de la línea t podemos hallarlo también con el nombre SSR

Sum Sq (linea Residuals) 340.5 explica la variabilidad de los datos debido al error. Sum Sq de la linea Residual podemos hallarla tambien con el nombre SSE.

SST= SSR + SSE = 3551.1+340.5 explica la variabilidad total de los datos de A por t y por error

MSR=SSR/Df= 3551.1/1 = 3551.1 (es el valor de la columna Mean Sq linea t)

MSE=SSE/Df= 340.5/9 = 37,8 (es el valor de la columna Mean Sq linea Residual)

F value= MSR/MSE= 3551.1/37.8 = 93.86

La probabilidad de hallar una valor de F mayor a 93.86 es igual a 0.000004665. Por lo tanto podemos concluir que el modelo planteado describe muy bien las variabilidad de A en función de t. Cuanto mayor sea el F-value mayor explicación de la variabilidad de nuestra variables dependiente en función de la independiente.

Un valor mucho mayor de MSR con respecto a MSE indica buen ajuste de nuestro modelo

El coeficiente de determinación: R^2 es otro predictor del ajuste. Si bien este valor lo podemos ver en la salida del programa cuando utilizamos summary(lmtablaR421), que era de 0,9125. Podemos calcularlo con los datos de la tabla obtenida al aplicar anova(lmtablaR421). Para esto dividimos el valor de SSR (efecto de nuestra variable independiente sobre la dependiente) por SST (efecto total de la variable y el error). Es claro que cuanto menos influya el error y más la variable, R2 será mas cercano a uno. Veamos para nuestro caso

R^2 = SSR/SST

R^2 = 3551.1/(3551+340.5)= 0.9125

Conclusión: el ajuste del modelo lineal planteado, representado por la recta

A=-11.5909 + 5.6818 * t

es un ajuste significativo ya que su R^2 = 0.91 con $p<0.01$.

2.2. Modelos lineales con más de una variable independiente

Como dijimos en un modelo lineal la variable dependiente es cuantitativa y entre las independientes, al menos una debe ser cuantitativa. Entre las variables independientes puede haber interacción o no y esto produce cambios en el modelo a utilizar. La existencia de interacción implica que el efecto de una variable independiente sobre la variable dependiente es influenciada por el valor de las otras independientes. Veremos casos sin y con interacción entre las variables independientes.

2.3. Modelo lineal sin interacción entre las variables independientes

Veamos ahora un modelo lineal en el que incluiremos dos variables independientes.

Introduzcamos en el espacio de trabajo la tablaR422 de la planilla de cálculo tablaR4-2.ods/xls

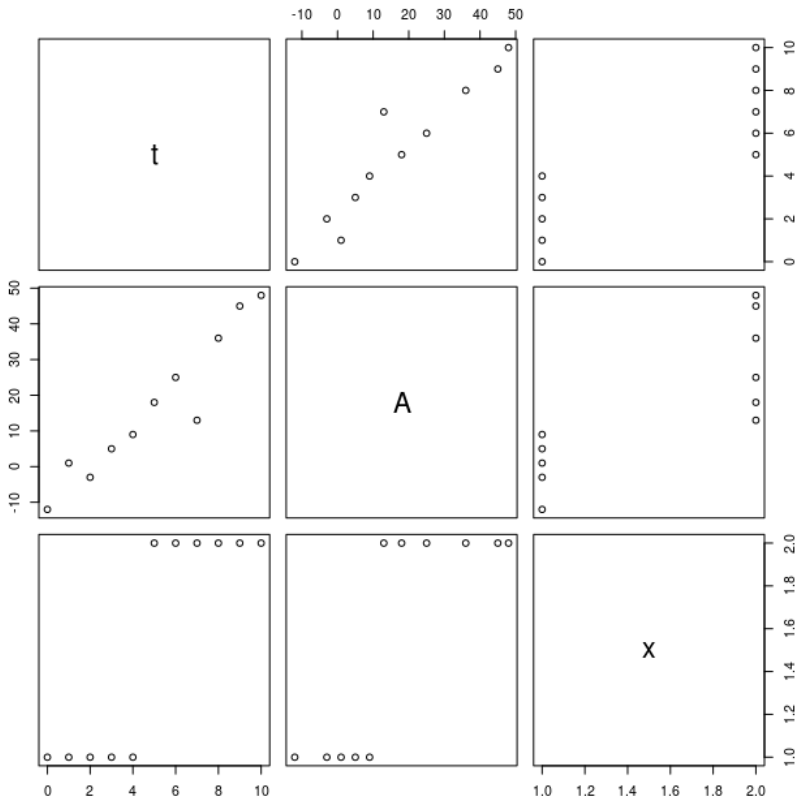
```
> tablaR422<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")

> tablaR422
   t     A   x
```

1 0 -12 a
2 1 1 a
3 2 -3 a
4 3 5 a
5 4 9 a
6 5 18 b
7 6 25 b
8 7 13 b
9 8 36 b
10 9 45 b
11 10 48 b

Graficamos A vs t y x

> plot(tablaR422)



Esta gráfica vista en módulos anteriore muestra cada variable graficada contra todas las variables. Considerando como variable respuesta a A, observamos que al aumentar t aumenta A (grafica de la fila 2, columna 1) y al aumentar x también aumenta A (fila 2, columna 3). Si suponemos que los efectos son lineales podemos plantear el siguiente modelo

> lmtablaR422<-lm(A~t+x,data=tablaR422)

pedimos un summary del análisis realizado

```
> summary(lmtablaR422)

Call:
lm(formula = A ~ t + x, data = tablaR422)

Residuals:
    Min      1Q  Median      3Q     Max
-14.879  -1.864   1.939   2.712   6.909

Coefficients:
            Estimate   Std. Error  t value  Pr(>|t|)
(Intercept)  -11.818        3.824   -3.091  0.01487 *
t              5.909        1.241    4.763  0.00142 **
xb            -1.667        7.879   -0.212  0.83776
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.506 on 8 degrees of freedom
Multiple R-squared:  0.913,          Adjusted R-squared:  0.8912
F-statistic: 41.97 on 2 and 8 DF,  p-value: 5.731e-05
```

pedimos la tabla de anova del análisis realizado

```
> anova(lmtablaR422)
Analysis of Variance Table

Response: A
          Df   Sum Sq   Mean Sq  F value   Pr(>F)
t          1   3551.1    3551.1  83.9001  1.628e-05 ***
x          1      1.9       1.9   0.0447   0.8378
Residuals  8    338.6      42.3
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

SSR= SumSq (linea t de la tabla anterior)= 3551.1 explica la variabilidad de los datos debido al efecto de t sobre A

SSR= SumSq (linea x de la tabla anterior)= 1.9 explica la variabilidad de los datos debido al efecto de x sobre A

SSE= sum Sq (linea Residual de la tabla anterior) 338,6 explica la variabilidad de los datos debido al error.

SST= SSR + SSE = 3551+ 1.9 + 338.6 = 3891.6

MSR (linea t) =SSR/Df= 3551,1

MSR (linea x) =SSR/Df= 1.9/1 = 1.9

MSE=SSE/Df= 338.6/8 = 42.3

F value (linea t) = MSR/MSE= 3551.1/42.3 = 83.9

F value (linea x) = 1,9/42,3 = 0,0447

R^2 (para variable t) = SSR(t)/SST= 3551.1/3891.6 = 0,9125

R^2 (para variable x) = SSR(x)/SST= 1.9/3891.6 = 0,00049

R^2 = 0,9125 + 0,00049 = 0,913

MSR para t es elevado, lo que nos está indicando que la variable t explica en gran medida la variabilidad de los valores de A. Mientras que el valor de MSR (linea x) es pequeño lo que indica que la variable x explica poco la variabilidad de los valores de A. En la tabla anova vemos que la linea t tiene un valor muy bajo de **Pr(>F)= 1.628e-05 ***** mientras que el para la línea x tiene un valor Pr(>F)= 0,8378, que es mayor que 0,05 y por ende nos indica que el valor que tome la variable x no influye sobre los valores de A. Cosa que si ocurre con la variable t.

Estos resultados coinciden con lo hallado al aplicar summary(lmtablaR422), que repetimos

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-11.818	3.824	-3.091	0.01487 *
t	5.909	1.241	4.763	0.00142 **
xb	-1.667	7.879	-0.212	0.83776

Esta tabla nos indica que en nuestro modelo la ordenada al origen vale -11,818 y es significativamente diferente de cero. La pendiente de la variable t también es significativamente diferente de cero, con una valor de 5,909 (p= 0.00142). Por último el parámetro que representa a la la variable x se interpreta de la siguiente manera. En la tabla aparece xb=1,667. Esto significa que cuando x toma el valor b, el valor de A será 1,667 unidades menor que cuando x=a, al que el software le asigna arbitrariamente el valor 0.

2.4. Modelo lineal con interacción entre las variables independientes

Podríamos hacer el análisis considerando que las variables t y x pueden tener interacción, es decir el efecto de t sobre A depende que x valga a o b. planteamos un modelo con interacción. Como verá a continuación las variables independientes se relacionan en el modelo con *.

> lmtablaR422int<-lm(A~t*x,data=tablaR422)

pedimos un summary del modelo lineal

> summary(lmtablaR422int)

Call:
lm(formula = A ~ t * x, data = tablaR422)

Residuals:

Min	1Q	Median	3Q	Max
-14.5048	-1.5000	0.5238	3.9810	5.6000

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-9.200	5.169	-1.780	0.1183
t	4.600	2.110	2.180	0.0656 .
xb	-9.895	13.314	-0.743	0.4815
t:xb	2.057	2.645	0.778	0.4622

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.673 on 7 degrees of freedom
Multiple R-squared:  0.9199,          Adjusted R-squared:  0.8856
F-statistic:  26.8 on 3 and 7 DF,  p-value: 0.0003278
```

Vemos que el ajuste del modelo es bueno R-squared= 0,9199 y p.value= 0,0003278. Sin embargo la tabla nos indica que no hay interacción, indicada en la línea t:xb con Pr(>|t|) = 0,4622.

Además vemos que al introducir la interacción entre las variables si bien el R-squared no cambió demasiado si lo ha hecho el valor de p-value del parámetro t. En conclusión introducir la interacción entre las variables no ha mejorado nuestro modelo.

2.5. Otros recursos con modelos lineales

La representación gráfica de un modelo lineal solo puede hacerse con una o dos variables independientes. Veremos un ejemplo. Para ello introduza en su espacio de trabajo la tablaR423

```
> tablaR423<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")

> tablaR423
   A B C
1  3 2  4
2  4 3  6
3  4 4  8
4  4 3  8
5  4 5 11
6  3 6 13
7  3 7 15
8  3 6 15
9  3 7 17
10 7 8 19
```

realizamos un modelo lineal con variable independientes A y B y variable respuesta la columna C.

```
> lmtablaR423<-lm(C~A+B,tablaR423)
```

pedimos la tabla anova

```
> anova(lmtablaR423)
Analysis of Variance Table
Response: C
          Df  Sum Sq  Mean Sq  F value    Pr(>F)
A          1  19.297   19.297   17.556  0.004086 **
B          1 197.409  197.409 179.597  3.02e-06 ***
Residuals  7   7.694    1.099
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

y un summary del objeto lmtablaR423

```
> summary(lmtablaR423)
Call:
lm(formula = C ~ A + B, data = tablaR423)
```

```
Residuals:
   Min     1Q Median     3Q    Max
-1.0982 -0.6594 -0.2207  0.7062  1.4359

Coefficients:
             Estimate   Std. Error  t value  Pr(>|t|)
(Intercept) -1.03237    1.28173    -0.805    0.447
A            0.09387    0.29585     0.317    0.760
B            2.40699    0.17961    13.401    3.02e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.048 on 7 degrees of freedom
Multiple R-squared:  0.9657,         Adjusted R-squared:  0.9559
F-statistic: 98.58 on 2 and 7 DF,  p-value: 7.464e-06
```

Concluimos que el modelo lineal elegido ajusta bien a los datos experimentales. Fundamentos

R-squared: 0,9657 : valor próximo a 1

p-value= 7.464e-06 : valor menor a 0,05

Ajusted R-squared: 0,9559 : este valor es cercano a 1 y es más representativo del buen ajuste que el R-squared.

Según tabla de anova: A y B tiene efecto significativo sobre el valor de C, lo que queda representado por los valores de Pr(>F)

A Pr(>F) = 0.004086 **

B Pr(>F) = 3.02e-06 ***

los coeficientes correspondientes a la variable A y B fueron (según summary(lmtablaR423))

```
             Estimate   Std. Error  t value  Pr(>|t|)
(Intercept) -1.03237    1.28173    -0.805    0.447
A            0.09387    0.29585     0.317    0.760
B            2.40699    0.17961    13.401    3.02e-06 ***
```

El coeficiente de A no discrepa de cero y el de B si lo hace, esta conclusión la sacamos en base al p-value de la columna Pr(<|t|)

Los datos que obtenemos con la tabla anova() y con la función summary() también podemos obtenerlos parcialmente con algunas funciones que vemos en las secciones siguientes.

2.5.1. Conocer la fórmula del modelo

Una vez que realizamos el modelo lineal y almacenamos este análisis en un objeto podemos conocer la fórmula aplicada. Para el caso anterior que creamos el objeto lmtablaR423, resulta

```
> formula(lmtablaR423)

C ~ A + B
```

2.5.2. para conocer los coeficientes

si deseamos conocer los coeficientes del modelo

```
> coef(lmtablaR423)
(Intercept)           A           B
-1.03236944  0.09387139  2.40699180
```

2.5.3. conocer residuales

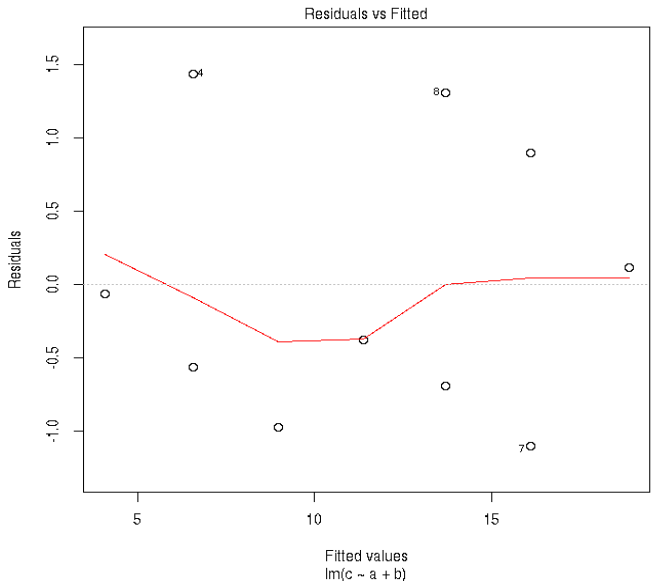
Los residuales es un buen método para ver el ajuste del modelo a los datos experimentales. El residual se calcula como la diferencia para cada valor de la variable independiente entre el valor real medio y el calculado por el modelo. Un buen ajuste es cuando estas diferencias se alternan positivas y negativas. Para nuestro caso si bien no es lo óptimo no está mal.

```
> residuals(lmtablaR423)
         1          2          3          4          5          6
-0.06322831 -0.56409150 -0.97108330  1.43590850 -0.37807510 -0.69119551
         7          8          9         10
-1.09818731  1.30880449  0.90181269  0.11933535
```

2.5.4. Grafica de los residuales

El siguiente comando nos da cuatro gráficas. La primera es la más explicativa y grafica los residuos. La linea roja cercana al cero indica un buen ajuste del modelo a los datos experimentales

```
> plot(lmtablaR423)
```



Solo vemos la primer gráfica generada.

2.5.5. Valores modelizados de la variable dependiente

Podemos calcular que valores tendríamos de la variable C en base al modelo.

```
> fitted.values(lmtablaR423)
       1        2        3        4        5        6        7        8
 4.063228 6.564091 8.971083 6.564091 11.378075 13.691196 16.098187 13.691196
       9       10
16.098187 18.880665
```

El mismo resultado se obtiene con la función

```
> predict(lmtablaR423)
```

2.5.6. Intervalo de confianza

calculamos el intervalo de confianza con nivel de significación 5% de cada uno de los parámetros del modelo. Para ello utilizamos la función confint().

```
> confint(lmtablaR423,level=0.95)
                2.5 %      97.5 %
(Intercept) -4.0631783  1.9984395
A           -0.6056975  0.7934403
B            1.9822874  2.8316962
```

Aquellos valores que incluyen el cero, no son diferentes de cero en forma significativa con p<0,05. En nuestro caso vemos que la intersección y el parametro de la variable A no discrepan de cero, mientras que el parámetro de la variable B si discrepa de cero. Como podemos ver este resultado coincide con lo hallado anteriormente, que repetimos para mejor visualización

```
            Estimate   Std. Error  t value   Pr(>|t|)
(Intercept) -1.03237   1.28173    -0.805    0.447
A            0.09387    0.29585     0.317    0.760
B            2.40699    0.17961    13.401    3.02e-06 ***
```

2.6. Graficar un modelo lineal

Introduzcamos los datos de la tablaR424 de la planilla tablaR4-1.ods/xls

```
> tablaR424<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

```
> tablaR424
    G glucagon insulina
1  1.00     5.0      20
2  1.10     4.3      21
3  1.20     4.8      26
4  1.50     5.0      25
5  1.70     4.6      28
6  1.80     4.1      29
7  1.80     3.9      30
8  1.75     3.5      25
9  1.90     3.4      38
10 2.00     3.1      41
11 2.10     3.0      46
12 2.20     3.0      49
13 2.30     2.1      44
14 2.80     2.6      48
15 2.90     2.5      51
16 2.60     2.0      53
```

planteamos un modelo linean en que los valores de insulina dependen de la concentración de glucosa (G) y de glucagón

```
> lmtablaR424<-lm(insulina~glucagon+G,tablaR424)
```

pedimos un summary() del modelo

```
> summary(lmtablaR424)
Call:
```

```
lm(formula = insulina ~ glucagon + G, data = tablaR424)

Residuals:
   Min     1Q Median     3Q    Max
-9.2547 -2.5464  0.0398  2.0346  7.2855

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   30.744     15.957   1.927   0.0762 .
glucagon      -4.683      2.352  -1.991   0.0679 .
G             11.373      4.194   2.712   0.0178 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4.265 on 13 degrees of freedom
Multiple R-squared:  0.8812,    Adjusted R-squared:  0.863
F-statistic: 48.24 on 2 and 13 DF,  p-value: 9.664e-07
```

y una tabla anova()

```
> anova(lmtablaR424)
Analysis of Variance Table
Response: insulina
           Df  Sum Sq Mean Sq F value    Pr(>F)
glucagon    1 1621.45 1621.45 89.1199 3.495e-07 ***
G           1  133.77  133.77  7.3526    0.0178 *
Residuals  13  236.52   18.19
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

por lo tanto el modelo que obtenemos expresa la concentración de insulina en función de la concentración de glucagón y glucosa por la siguiente ecuación

insulina = 30.744 – 4.683*glucagon + 11.373*G

El modelo nos indica que una aumento de una unidad en el valor del glucagon será acompañado por un descenso de 4.683 unidades en la insulina. Por otra parte un aumento de 1 unidad en la glucemia (G) será acompañada de un aumento de 11,373 unidades de insulina. La tabla anova nos indica que los efetos del glucagón y la G son significativos, como se desprende de los valores de Pr(>F) menores a 0.05.

Para graficarlo construimos una matriz de datos de valores de insulina utilizando el modelo.

creamos dos vectores, uno para G y otro para glucagon. Nos fijamos primero el rango de valores de cada una de las variables

```
> range(tablaR424$G)

[1] 1.0 2.9

> range(tablaR424$glucagon)

[1] 2 5
```

creamos los vectores con ese rango y un incremento de a 0.1 (podría ser otro incremento si se

quisiera), llamamos G al vector que tendrá los valores de glucosa y glucagon al que tendrá los valores de glucagón. Ambos vectores se extienden en el rango medido de las variables.

> G<-seq(1,3,0.1)

> glucagon<-seq(2,5,0.1)

Comprobamos que dichos vectores tengan los datos requeridos

```
> G
 [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
[20] 2.9 3.0
```

glucagon<-seq(2,5,0.1)

```
> glucagon
 [1] 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8
[20] 3.9 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5.0
```

creamos una matriz para los valores de insulina, en principio estará vacia pero tendrá un número de filas igual al numero de elementos de G y un número de columnas con la cantidad de elementos del vector glucagon.

matriztablaR424<-matrix(data=0, nrow=length(G),ncol=length(glucagon))

y luego ejecutamos el script para llenar la matriz. Este escript debe ser escrito en un editor de texto y guardado como archivo txt en el mismo directorio donde se está realizando el gráfico. Para este caso el archivo se llamó matriz.txt.

El tema de scripts será visto en detalle en el módulo 5.

```
#texto del script
for (i in seq(1,length(G),0.1)){
for (j in seq(1,length(glucagon),by=0.1)){
matriztablaR424[i,j]<-30.744 -4.683*glucagon[j]+11.373*G[i]
}
}
```

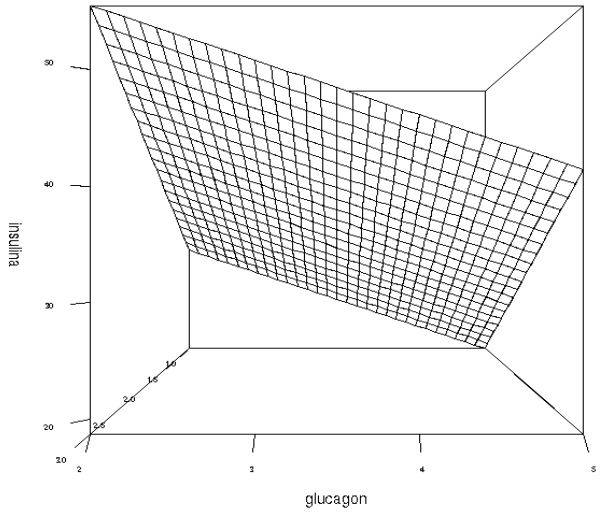
ejecutamos el script
source("matriz")

nos quedará una matriz con los valores de insulina calculados para cada valor de G y glucagon.

luego graficamos persp con el siguiente código

```
>
persp(x=seq(1,3,length.out=nrow(matriztablaR424)),y=seq(2,5,length.out=ncol(matriztablaR424))
,matriztablaR424,phi=0,theta=90,xlab="",ylab="glucagon",zlab="insulina",xlim=range(G),ylim=r
ange(glucagon),box=TRUE,axes=T,d=1,r=2,nticks=4,ticktype="detailed",cex.axis=0.5)
```

que nos da el siguiente gráfico



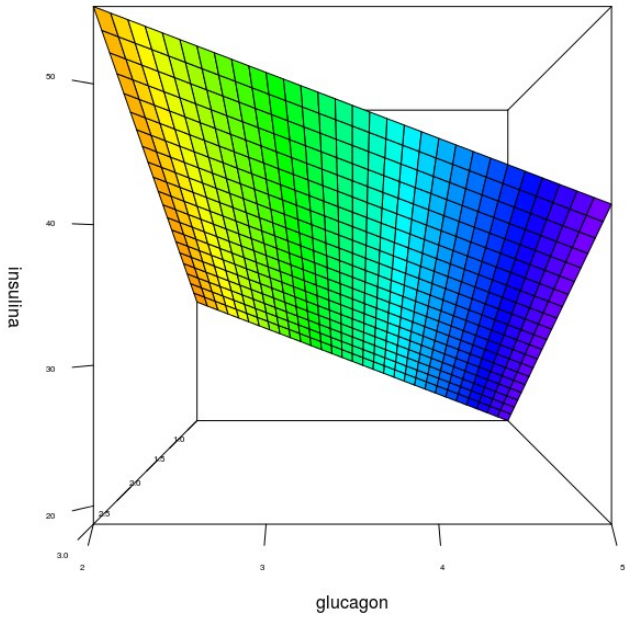
En este gráfico tenemos a insulina en el eje vertical, glucagón en el horizontal anterior y G se halla en el horizonal lateral. El gráfico nos indica que al aumentar el glucagón baja la insulina y al aumentar la G aumenta la insulina.

Podemos asignarle color al gráfico para ver mejor los incrementos o descensos de insulina

> colorvect<-rainbow(length(matriztablaR424),start=0.1,end=0.8)

y luego hacemos el grafico

>
persp(x=seq(1,3,length.out=nrow(matriztablaR424)),y=seq(2,5,length.out=ncol(matriztablaR424)),matriztablaR424,phi=0,theta=90,xlab="",ylab="glucagon",zlab="insulina",xlim=range(G),ylim=range(glucagon),box=TRUE,axes=T,d=1,r=2,nticks=4,ticktype="detailed",cex.axis=0.5,col=colorvect)



2.6.1. Agregado de puntos experimentales

A la grafica del modelo podemos agregarle los datos experimentales, en nuestro caso los datos de la tablaR424

se debe crear primero un objeto con la función persp

> persptablaR424<-persp(x=seq(1,3,length.out=nrow(matriztablaR424)),y=seq(2,5,length.out=ncol(matriztablaR424)),matriztablaR424,phi=0,theta=90,xlab="",ylab="glucagon",zlab="insulina",xlim=range(G),ylim=range(glucagon),box=TRUE,axes=T,d=1,r=2,nticks=4,ticktype="detailed",cex.axis=0.5)

en este caso se generó el gráfico sin color

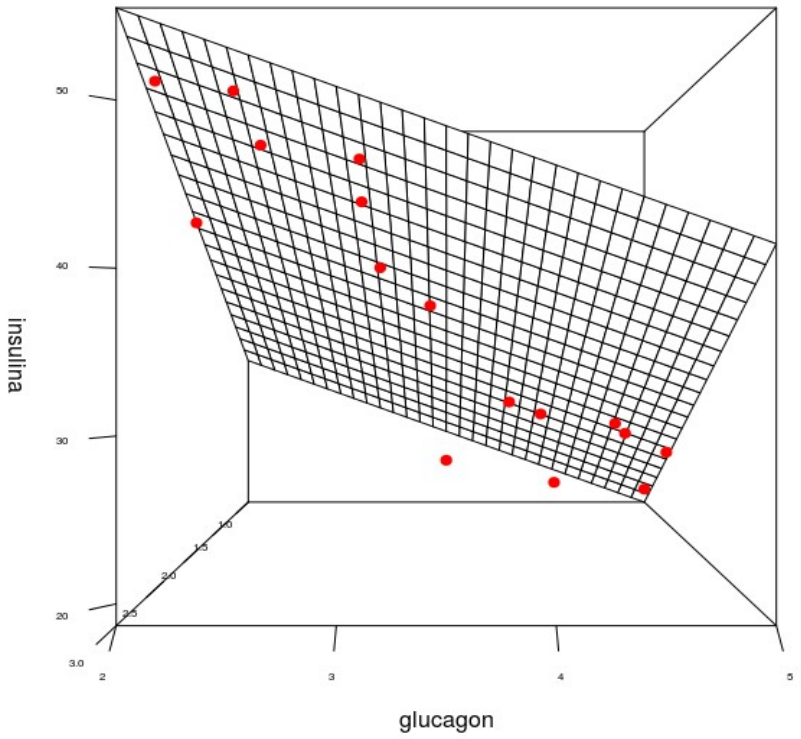
luego creamos un objeto con los puntos de la tablaR424, pero haciendo referencia al objeto creado anteriormente

mispuntos<-trans3d(tablaR424$G,tablaR424$glucagon,tablaR424$insulina,pmat=persptablaR424)

luego con la función points, agregamos los puntos

points(mispuntos,pch=20,cex=1, col="red")

obtendremos la siguiente gráfica



En la gráfica vemos el plano indica la variación de la insulina en función de la glucosa y el glucagón según el modelo planteado y los puntos en rojo son nuestros datos experimentales, que muestran que el modelo ajusta bien los datos, ya que se halla en su mayoría sobre el plano.

3. Clase 4.3

Video: https://youtu.be/X7Pf2jFIjZM
Tabla de datos: http://hdl.handle.net/2133/15483

3.1. Modelos lineales. Continuación

Al aplicar modelos lineales a datos con más de una variable independiente es común que tengamos dificultades a la hora de seleccionar las variables que mejor representan la variación de nuestra variable dependiente. Es importante conocer que cuanto más variables independientes coloquemos en el modelo mejor puede parecer el ajuste, ya que el valor de R-squared aumenta aproximándose al valor 1. Sin embargo, esto puede ser erróneo. A continuación veremos algunos métodos que pueden ayudar al momento de la elección de un buen modelo lineal.

3.2. Multiple regression stepwise analysis

Se require la biblioteca MASS. Instálela.

```
library(MASS)
```

Este método sirve para seleccionar las variables que más pesan sobre un modelo. Se basa en el índice AIC. **Cuanto más pequeño AIC, mejor es el modelo.**

introduzcamos los datos de la tablaR431 de la planilla de cálculo tablaR4-3.ods/xls

```
> tablaR431<-read.table("clipboard",header=TRUE,sep="\t",dec=",",encoding="latin1")
```

```
> tablaR431
   glucemia insulinemia glucagonemia  PC
1      1.00          20           30  41
2      0.90          17           35  34
3      0.80          17           37  35
4      1.50          35            8  70
5      1.40          42            5  81
6      1.30          25           10  50
7      1.20          20           28  40
8      1.00          12           51  23
9      0.70           6           55  12
10     0.58           5           60  11
11     1.70          58            4 115
```

Tomamos la variables glucemia como variable dependiente y utilizamos las otras tres variables como variables independientes.

```
> lmtablaR431<-lm(glucemia~insulinemia+glucagonemia+PC,tablaR431)
```

```
> summary(lmtablaR431)
Call:
lm(formula = glucemia ~ insulinemia + glucagonemia + PC, data = tablaR431)

Residuals:
     Min       1Q   Median       3Q      Max
-0.147572 -0.069713 -0.008802  0.072024  0.176769

Coefficients:
                Estimate   Std. Error   t value  Pr(>|t|)
```

```
(Intercept)      1.112243  0.268749   4.139  0.00436 **
insulinemia      0.053976  0.083540   0.646  0.53881
glucagonemia    -0.008403  0.004507  -1.864  0.10453
PC              -0.022094  0.042808  -0.516  0.62166
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1288 on 7 degrees of freedom
Multiple R-squared:  0.9063,     Adjusted R-squared:  0.8662
F-statistic: 22.58 on 3 and 7 DF,  p-value: 0.0005639
```

La tabla nos indica que salvo la ordenada al origen los otros parámetros no difieren significativamente de cero. Pero, el R-squared es alto (0,9063) y el p-value (0,0005639) muy inferior a 5%. Por otra parte la tabla de anova nos indica que la insulinemia influye significativamente sobre la variabilidad de glucemia.

```
> anova(lmtablaR431)
Analysis of Variance Table

Response: glucemia
             Df  Sum Sq  Mean Sq F value    Pr(>F)
insulinemia   1 1.06449 1.06449 64.1327 9.055e-05 ***
glucagonemia  1 0.05526 0.05526  3.3293    0.1108
PC            1 0.00442 0.00442  0.2664    0.6217
Residuals     7 0.11619 0.01660
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Se nos crea la duda si dejar solo la insulinemia o las demás. Si deseamos sacar variables, cuales pueden mejorar significativamente el modelo y cuales no.

Para ayudarnos en esta selección utilizaremos funciones y herramientas del paquete MASS. Aplicamos la función stepAIC y nos apoyaremos en los valores del índice AIC. Esta función va secuencialmente sacando y colocando variables que nos permiten decidir si cada variable independiente aporta o no al modelo.

```
> steplmtablaR431<-stepAIC(lmtablaR431,direction="both")

Start:  AIC=-42.05
glucemia ~ insulinemia + glucagonemia + PC

               Df Sum of Sq     RSS     AIC
- PC            1  0.004422 0.12061 -43.644
- insulinemia   1  0.006929 0.12312 -43.418
<none>                      0.11619 -42.055
- glucagonemia  1  0.057697 0.17389 -39.620

Step:  AIC=-43.64
glucemia ~ insulinemia + glucagonemia

               Df Sum of Sq     RSS     AIC
<none>                      0.12061 -43.644
```

```
+ PC              1       0.004422      0.11619 -42.055
- glucagonemia   1        0.055261      0.17587 -41.495
- insulinemia    1         0.059921     0.18053 -41.207
```

El cálculo comienza con un valor de AIC de -42,05, indicado en Start y en la tercer línea de la primer tabla. La tabla con -PC y -insulinemia, nos indica que al sacar del modelo esas variables el valor de AIC disminuyó, lo que haría mejor el modelo. En la columna AIC se muestran los valores del índice sin dichas variables.

Luego en la tabla siguiente muestra el AIC sin el PC (en la columna <none>) y lo coloca nuevamente, viendo que el AIC aumenta, lo que confirma que PC no contribuye. En la misma tabla vemos que sacar glucagonemia e insulinemia aumentan AIC, por lo que no sería favorable para el modelo, sacar estas variables.

Por lo tanto nos quedaríamos con el modelo siguiente

```
> lmtablaR431<-lm(glucemia~insulinemia+glucagonemia,tablaR431)

> summary(lmtablaR431)
Call:
lm(formula = glucemia ~ insulinemia + glucagonemia, data = tablaR431)

Residuals:
        Min        1Q    Median        3Q       Max
-0.165896 -0.074146  0.002077  0.0Multiple R-squared:  0.9028,        Adjusted        R-squared:
0.8785
F-statistic: 37.14 on 2 and 8 DF,  p-value: 8.94e-05 62322  0.203554

Coefficients:
                Estimate   Std. Error    t value Pr(>|t|)
(Intercept)      1.082539     0.250188   4.327   0.00252 **
insulinemia      0.010961     0.005498   1.994   0.08132 .
glucagonemia -0.008189     0.004277  -1.915  0.09189 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.1228 on 8 degrees of freedom
Multiple R-squared:  0.9028,          Adjusted R-squared:  0.8785
F-statistic: 37.14 on 2 and 8 DF,  p-value: 8.94e-05

> anova(lmtablaR431)
Analysis of Variance Table
Response: glucemia
                Df    Sum Sq   Mean Sq   F value    Pr(>F)
insulinemia      1    1.06449  1.06449      70.6075 3.061e-05 ***
glucagonemia   1     0.05526 0.05526      3.6655   0.09189 .
Residuals        8     0.12061 0.01508
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Vemos que con este modelo insulinemia explica en gran parte la variancia de la glucemia, mientras que la glucagonemia por el valor de Pr(>F)= 0,09189, no alcanza el valor 0,05.

Si calcularamos los R-squared para cada variable, utilizando las columnas SumSq de la tabla anterior, tenemos

SumSq insulinemia: 1.06449
SumSq glucagonemia. 0.05526
SumSq residuals: 0.12061
SumSq total= 1.6449 + 0.05526 + 0.12061 = 1.24036
dividiendo cada SumSq por SumSq total obtenemos los R-squared para cada variable y el R-squared total

	Sum Sq	R-Squared variables	R-squared
insulinemia	1.06449	1.06449/1.21036 = 0.8582105195	0,9027621013
glucagonemia	0.05526	0.05526/1.21036 = 0.0445515818	
residual	0.12061	0.12061/1.21036 = 0.0972378987	
total	1.24036		

valores que nuevamente nos indican que la insulinemia contribuye al modelo mucho más que la glucagonemia.

Como vimos anteriormente la función stepAIC nos permitió tomar la decisión de eliminar la variable PC. Podemos centrar nuestra atención en los valores de R-squared, adjusted R-squared y p-value para cada modelo.

Para el modelo con todas las variables, planteado de la siguiente manera

```
> lmtablaR431<-lm(glucemia~insulinemia+glucagonemia+PC,tablaR431)
```

Multiple R-squared: 0.9063, Adjusted R-squared: 0.8662

F-statistic: 22.58 on 3 and 7 DF, p-value: 0.0005639

y para el modelo sin incluir la variable PC

```
> lmtablaR431<-lm(glucemia~insulinemia+glucagonemia,tablaR431)
```

Multiple R-squared: 0.9028, Adjusted R-squared: 0.8785

F-statistic: 37.14 on 2 and 8 DF, p-value: 8.94e-05

Vemos que si bien el R-squared descendió en su tercer decimal al quitar PC de las variables, el Adjusted R-squared aumentó. Este valor es más adecuado para ver la bondad del modelo. Además vemos que al haber quitado PC el p-value es más pequeño.

Podemos comparar ambos modelos para ver si realmente hay diferencias significativas entre ellos. Para ello creamos dos objetos con los dos modelos incluyendo o no la variable PC

```
> lmtablaR431conPC<-lm(glucemia~insulinemia+glucagonemia+PC,tablaR431)
```

```
> lmtablaR431sinPC<-lm(glucemia~insulinemia+glucagonemia,tablaR431)
```

3.3. Comparación de dos modelos lineales

Con la función anova() comparamos los modelos

```
> anova(lmtablaR431conPC,lmtablaR431sinPC)
Analysis of Variance Table
Model 1: glucemia ~ insulinemia + glucagonemia + PC
Model 2: glucemia ~ insulinemia + glucagonemia
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)

```
1          7   0.11619
2          8    0.12061   -1      -0.0044216  0.2664  0.6217
```

el valor de Pr(>F)= 0,6217 nos indica que los dos modelos no difieren significativamente, aunque el modelo sin PC explica mejor la variabilidad de la glucemia en función de las variables independientes.

Otra forma de seleccionar las variables independientes de un modelo lineal es utilizando la biblioteca leaps

library(leaps)

de la cual utilizamos la función regsubsets()

```
>leapslmtablaR431<-
regsubsets(glucemia~insulinemia+glucagonemia+PC,data=tablaR431,nbest=10)
```

```
> summary(leapslmtablaR431)
Subset selection object
Call: regsubsets.formula(glucemia ~ insulinemia + glucagonemia + PC,
   data = tablaR431, nbest = 10)

3 Variables  (and intercept)
                  Forced in  Forced out
insulinemia        FALSE      FALSE
glucagonemia       FALSE      FALSE
PC                 FALSE      FALSE
10 subsets of each size up to 3
Selection Algorithm: exhaustive
          insulinemia  glucagonemia  PC
1  ( 1 )      "*"          " "          " "
1  ( 2 )      " "          " "          "*"
1  ( 3 )      " "          "*"          " "
2  ( 1 )      "*"          "*"          " "
2  ( 2 )      " "          "*"          "*"
2  ( 3 )      "*"          " "          "*"
3  ( 1 )      "*"          "*"          "*"
```

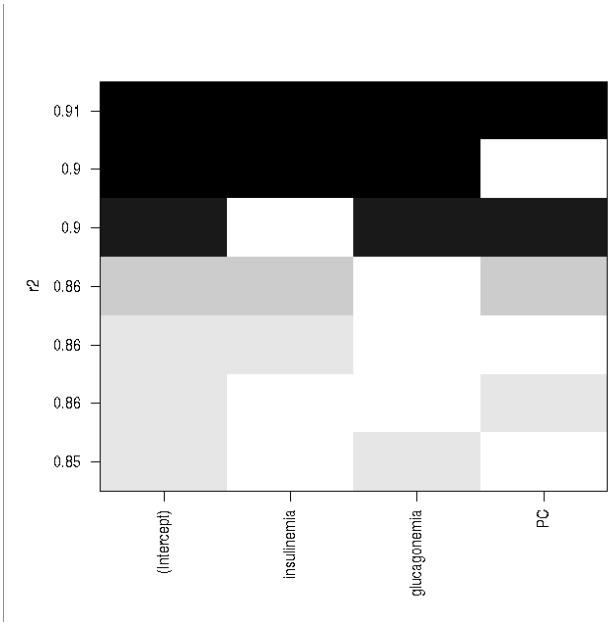
nbest: mostrará los 10 primeros modelos elegidos en función del set de datos probando todas las combinaciones. Para nuestro caso tenemos tres variables, por lo que serán posible solo 7 modelos: 3 con cada variables independiente, 3 con combinaciones de a dos variables y 1 con las tres variables. En la tabla anterior están indicadas la cantidad de variables por la primer columna

vemos que de a 1 variable el de mejor ajuste es tomando insulinemia como independiente

tomando dos variables, el mejor ajuste es con insulinemia y glucagonemia

La gráfica del objeto también puede ayudarnos en la elección

```
> plot(leapslmtablaR431,scale="r2")
```



como se interpreta? miremos de abajo hacia arriba

el peor R2 es con intercept y glucagonemia. r2= 0,85

sigue intercept + PC, r2= 0,86

sigue insulinemia + intercept r2= 0,86

de los que usan dos variables el mejor ajuste es con intercept + glucagonemia e insulinemia r2=0,90

Por supuesto es mejor ajuste es con las tres variables, r2= 0,91. Pero como vimos poco aporta al modelo PC ya que AIC aumenta y Adjusted R-squared disminuye.

3.4. Test de potencia para un modelo lineal.

Ya hemos visto en clases anteriores el test de potencia para un modelo lineal con una sola variable independiente. Cuando hay más de una variable independiente se puede utilizar el mismo test.

Veamos los modelos lineales realizados anteriormente sobre los datos de la tablaR431. Habíamos realizado dos análisis. Luego de aplicar la función stepAIC() nos inclinamos porque el modelo era mejor sin incluir PC. Veamos un resumen de dichos análisis.

Modelo planteado con las tres variables: insulinemia – glucagonemia -PC

```
> lmtablaR431<-lm(glucemia~insulinemia+glucagonemia+PC,tablaR431)
```

resumimos datos del summary y tabla anova

```
> summary(lmtablaR431)

Call:
lm(formula = glucemia ~ insulinemia + glucagonemia + PC, data = tablaR431)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.147572	-0.069713	-0.008802	0.072024	0.176769

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	1.112243	0.268749	4.139	0.00436	**
insulinemia	0.053976	0.083540	0.646	0.53881	
glucagonemia	-0.008403	0.004507	-1.864	0.10453	
PC	-0.022094	0.042808	-0.516	0.62166	

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1288 on 7 degrees of freedom
Multiple R-squared: 0.9063, Adjusted R-squared: 0.8662
F-statistic: 22.58 on 3 and 7 DF, p-value: 0.0005639

> anova(lmtablaR431)
Analysis of Variance Table
Response: glucemia

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
insulinemia	1	1.06449	1.06449	64.1327	9.055e-05	***
glucagonemia	1	0.05526	0.05526	3.3293	0.1108	
PC	1	0.00442	0.00442	0.2664	0.6217	
Residuals	7	0.11619	0.01660			

Modelo planteado con las dos variables: insulinemia – glucagonemia

lmtablaR431<-lm(glucemia~insulinemia+glucagonemia,tablaR431)
> anova(lmtablaR431)

resumen del summary
Multiple R-squared: 0.9028, Adjusted R-squared: 0.8785
F-statistic: 37.14 on 2 and 8 DF, p-value: 8.94e-05

Analysis of Variance Table
Response: glucemia

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
insulinemia	1	1.06449	1.06449	70.6075	3.061e-05	***
glucagonemia	1	0.05526	0.05526	3.6655	0.09189	.
Residuals	8	0.12061	0.01508			

La función stepAIC nos sugería sacar PC, ya que el índice AIC se hacia menor, por otra parte al sacar PC vemos que: aumenta adjuted R-squared y desciende el p-value del modelo.

Nos inclinamos entonces por el segundo modelo. Entonces rechazamos cualquier modelo distinto al lineal, quedándonos con el lineal con p-value menor al 5%.
¿Cual es la potencia de nuestro ensayo? ¿Cual es la probabilidad que realmente sea un buen modelo?

Aplicamos el test de potencia del paquete pwr()

pwr.f2.test(u = NULL, v = NULL, f2 = NULL, sig.level = NULL, power = NULL)

donde

u: grados de libertad del numerador

v: grados de libertad del denominador

f2: effect size

sig.level: probabilidad de error tipo I

power: potencia del ensayo.

veamos la tabla anova del segundo modelo

Analysis of Variance Table
Response: glucemia

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
insulinemia	1	1.06449	1.06449	70.6075	3.061e-05	***
glucagonemia	1	0.05526	0.05526	3.6655	0.09189	.
Residuals	8	0.12061	0.01508			

u = 1 + 1 =2 (Df de insulinemia + Df glucagonemia)
v= 11 – 2- 1 = 8
Los números utilizados en el cálculo de v son: número de datos de la tabla – u -1). El resultado hallado coincide con el Df de Residual.

calculamos la magnitud del efecto con la función de Cohen. Esta función en su forma general es

cohen.ES(test = c("p", "t", "r", "anov", "chisq", "f2"), size = c("small", "medium", "large"))

utilizamos el valor f2 del argumento test, que corresponde a modelo lineal y size, tomemos el valor medium, dado que glucagonemia no parece ser una variable que represente la variabilidad de la glucemia de manera importante.

```
> cohen.ES(test = c("f2"), size = c("medium"))
   Conventional effect size from Cohen (1982)
         test = f2
         size = medium
         effect.size = 0.15
```

Reemplanzado los valores de u, v, f2 y sig.level

```
> pwr.f2.test(u = 2, v = 8, f2 = 0.15, sig.level = 0.05, power = NULL)

    Multiple regression power calculation
             u = 2
             v = 8
            f2 = 0.15
     sig.level = 0.05
         power = 0.1460902
```

tenemos una potencia del 14%. Aunque el error de tipo I fue bajo la potencia es reducida. Quizás el modelo solo incluyendo la insulinemia es más adecuado.

Lo probamos

planteamos el modelo solo con insulinemia

```
> lmtablaR431<-lm(glucemia~insulinemia,tablaR431)


> summary(lmtablaR431)
Call:
lm(formula = glucemia ~ insulinemia, data = tablaR431)

Residuals:
      Min       1Q   Median       3Q      Max
-0.16833 -0.09171 -0.04387  0.14904  0.17045

Coefficients:
             Estimate   Std. Error  t value  Pr(>|t|)
(Intercept)  0.621438   0.077128    8.057    2.09e-05 ***
insulinemia  0.020405   0.002765    7.381     4.19e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1398 on 9 degrees of freedom
Multiple R-squared:  0.8582,          Adjusted R-squared:  0.8425
F-statistic: 54.47 on 1 and 9 DF,  p-value: 4.188e-05
```

el valor de p-value es más bajo aun que los otros modelos

```
> anova(lmtablaR431)
Analysis of Variance Table

Response: glucemia
             Df  Sum Sq   Mean Sq   F value    Pr(>F)
insulinemia      1  1.06449  1.06449     54.474   4.188e-05 ***
Residuals        9  0.17587  0.01954
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Calculamos la magnitud del efecto con el argumento size=large, dado que la insulimia tiene un efecto importante sobre la glucemia como se deprende de la tabla obtenida con anova()

```
> cohen.ES(test = c("f2"), size = c("large"))

    Conventional effect size from Cohen (1982)
         test = f2
         size = large
   effect.size = 0.35
```

aplicamos el test de potencia

```
> pwr.f2.test(u = 1, v = 9, f2 = 0.35, sig.level = 0.05, power = NULL)

    Multiple regression power calculation
```

```
              u = 1
              v = 9
             f2 = 0.35
      sig.level = 0.05
          power = 0.4181865
```

Una potencia de 41% con p-value menor a 0,05 y R-squared 0,85, es lo mejor que podemos tener.

3.5. modelos lineales con variables categóricas

Ya hemos visto la utilización de variables categóricas. En este caso nos centraremos en ellas y en la escritura de los modelos, gráficas en dos dimensiones y en la reasignación de niveles.

Introduzcamos los datos de tablaR432

```
> tablaR432<-read.table("clipboard",header=T,dec=",",sep="\t",encoding="latin1")
> summary(tablaR432)
 tratamiento    variable1        variable2        tiempo       nivel
 control:20     Min.   :0.100   Min.   :0.200    Min.   : 0    a:12
 T1     :20     1st Qu.:0.475   1st Qu.:1.675    1st Qu.: 0    b:12
                Median :0.900   Median :2.300    Median : 5    c:16
                Mean   :1.073   Mean   :2.360    Mean   : 5
                3rd Qu.:1.550   3rd Qu.:2.625    3rd Qu.:10
                Max.   :2.700   Max.   :6.000    Max.   :10
```

planteamos el modelo incluyendo las tres variables

```
> lmtablaR432<-lm(variable1~variable2+tiempo+nivel,tablaR432)
> summary(lmtablaR432)
Call:
lm(formula = variable1 ~ variable2 + tiempo + nivel, data = tablaR432)

Residuals:
      Min       1Q   Median       3Q      Max
-1.05699 -0.18840 -0.08451  0.05960  1.04301

Coefficients:
             Estimate   Std. Error  t value   Pr(>|t|)
(Intercept) -0.24006    0.17867  -1.344   0.187736
variable2    0.26050    0.05289   4.926   2.01e-05 ***
tiempo      -0.01017    0.02775  -0.366    0.716299
nivelb       0.83207    0.19830   4.196    0.000176 ***
nivelc       1.24746    0.33252   3.751    0.000636 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.4309 on 35 degrees of freedom
Multiple R-squared:  0.7262,          Adjusted R-squared:  0.6949
F-statistic: 23.21 on 4 and 35 DF,  p-value: 1.959e-09
```

Aunque el modelo tiene un p-value muy bajo, el R-squared no es tan elevado y aun más bajo el Adjusted R-squared. Podríamos aplicar stepAIC. Pero claramente vemos que la variable tiempo

influye muy poco en el modelo, por lo que podemos eliminarla

```
> lmtablaR432<-lm(variable1~variable2+nivel,tablaR432)

> summary(lmtablaR432)
Call:
lm(formula = variable1 ~ variable2 + nivel, data = tablaR432)

Residuals:
      Min       1Q   Median       3Q      Max
-1.05384 -0.18510 -0.09253  0.04812  1.04616

Coefficients:
             Estimate Std. Error  t value  Pr(>|t|)
(Intercept) -0.25461    0.17210  -1.479    0.148
variable2    0.26650    0.04968   5.365    4.91e-06 ***
nivelb       0.80428    0.18099   4.444    8.11e-05 ***
nivelc       1.14218    0.16529   6.910    4.31e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4257 on 36 degrees of freedom
Multiple R-squared:  0.7251,        Adjusted R-squared:  0.7022
F-statistic: 31.66 on 3 and 36 DF,  p-value: 3.367e-10
```

Vemos que al eliminar el tiempo el Adjusted R-squared aumentó, por lo cual nos quedamos con este modelo al que podríamos escribir de la siguiente manera, utilizando los coeficientes estimados del modelo, de la tabla anterior:

La tabla de Coefficientes, nos presenta los mismos para la variable "nivel" con nivelb y nivelc, adjudicando el valor 0 al nivel a.

variable 1 = -0,25461 + 0,26650 * variable 2 + 0 para nivel a

variable 1 = -0,25461 + 0,26650 * variable 2 + 0,80427 para nivel b

variable 1 = -0,25461 + 0,26650 * variable 2 + 1,14218 para nivel c

Grafiquemos estos modelos y puntos, por nivel. Para el nivel a

para ellos definimos los modelos como funciones

```
> nivela<-function(variable2){-0.25461+0.26650*variable2}

> nivelb<-function(variable2){-0.25461+0.26650*variable2+0.80427}

> nivelc<-function(variable2){-0.25461+0.26650*variable2+1.14218}
```

el rango de la variable2 es

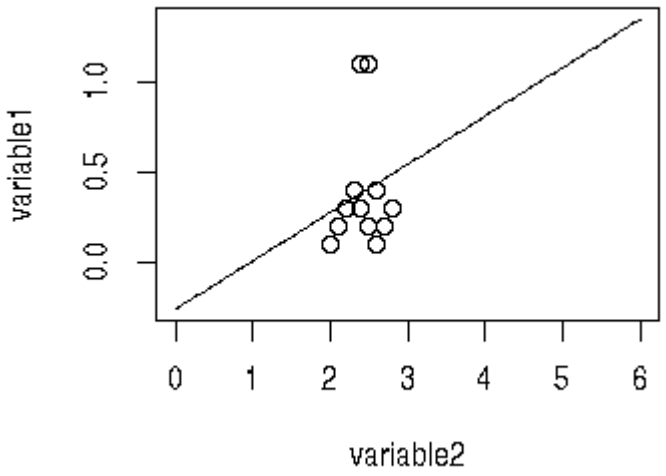
```
> range(tablaR432$variable2)

[1] 0.2 6.0
```

graficamos primero la función nivela y luego los puntos

```
> plot(nivela,0,6,ylab="variable1",xlab="variable2")
```

>
points(tablaR432$variable2[tablaR432$nivel=="a"],tablaR432$variable1[tablaR432$nivel=="a"])



de la misma manera podríamos obtener gráficas para los valores con nivel b y c.

Con las funciones lines, spline, points y otras aprendidas podríamos obtener una gráfica de todos los valores divididos por nivel

definimos un vector para la variable2

> variable2fit<-seq(0.2,6,0.1)

calculamos vectores de la variable1 con cada función según el nivel a, b o c

> variable1fita<-nivela(variable2fit)

> variable1fitb<-nivelb(variable2fit)

> variable1fitc<-nivelc(variable2fit)

graficamos los puntos de la tabla para el nivel a

>
plot(tablaR432$variable2[tablaR432$nivel=="a"],tablaR432$variable1[tablaR432$nivel=="a"],pch=20,col="black",xlab="variable2",ylab="variable1",xlim=c(0.1,6.1),ylim=c(0,3))

luego en otro color los puntos para los niveles b y c

>
points(tablaR432$variable2[tablaR432$nivel=="b"],tablaR432$variable1[tablaR432$nivel=="b"],pch=20,col="red")

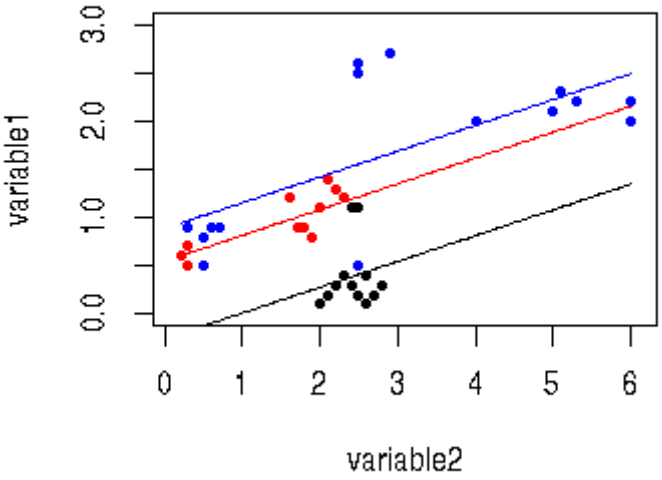
>
points(tablaR432$variable2[tablaR432$nivel=="c"],tablaR432$variable1[tablaR432$nivel=="c"],pch=20,col="blue")

luego las lineas de cada modelo con los colores correspondientes a los puntos

> lines(spline(variable2fit,variable1fita),col="black")

> lines(spline(variable2fit,variable1fitb),col="red")

> lines(spline(variable2fit,variable1fitc),col="blue")



3.6. Reasignación de niveles

Como vimos en el ejemplo anterior, la función lm() asigna el coeficiente cero al nivel a y los valores correspondientes a los niveles b y c. Si deseáramos cambiar dicho orden por cuestiones de presentación de resultados, utilizaremos la función relevel()

En primer lugar es importante que la variable sea factor

> is.factor(tablaR432$nivel)

[1] TRUE

> levels(tablaR432$nivel)

[1] "a" "b" "c"

asignamos ahora al nivel como nivel de referencia

> tablaR432$nivel<-relevel(tablaR432$nivel,ref="b")

chequeamos como quedó el orden de los niveles.

> levels(tablaR432$nivel)

[1] "b" "a" "c"

vemos que b quedó en primer lugar y será tomado como referencia

realizamos nuestro modelo lineal

> lmtablaR432<-lm(variable1~variable2+nivel,tablaR432)

pedimos un resumen del objeto creado

> summary(lmtablaR432)

```
Call:
lm(formula = variable1 ~ variable2 + nivel, data = tablaR432)

Residuals:
    Min      1Q  Median      3Q     Max
-1.05384 -0.18510 -0.09253  0.04812  1.04616
Coefficients:
             Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)  0.54967    0.14142   3.887    0.000419 ***
variable2    0.26650    0.04968   5.365    4.91e-06 ***
nivela      -0.80428    0.18099  -4.444    8.11e-05 ***
nivelc       0.33790    0.18133   1.863    0.070576 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4257 on 36 degrees of freedom
Multiple R-squared:  0.7251,          Adjusted R-squared:  0.7022
F-statistic: 31.66 on 3 and 36 DF,  p-value: 3.367e-10
```

Como vemos de la tabla anterior, aparecen nivel a y nivel c con sus coeficientes. Al nivel b se le asigna el coeficiente 0.

4. Clase 4.4

Video: **https://youtu.be/cdOI3Y4tThk**
tablas: http://hdl.handle.net/2133/15484

4.1. Regresión logística

En la regresión logística es un tipo de modelo en que la variable dependiente es cualitativa. En este modelo se busca una función que relacione variables independientes cuantitativas y/o categóricas con la probabilidad que un variable dependiente cualitativa tome uno u otro valor de dos posibilidades. Es decir la variable dependiente será dicotómica.

Por ejemplo podemos tener unidades experimentales en las que se ha medido sobrepeso y en tal caso tendremos unidades con sobrepeso y otras sin sobrepeso. En estas unidades también se pueden haber medido otras variables como Kcal consumidas por día, sexo, edad, hábitos de actividad física, etc y el objetivo de la regresión logística es hallar en qué grado cada variable mencionada incide sobre la probabilidad de tener o no sobrepeso.

Nos dará respuestas por ejemplo a este tipo de preguntas:

¿Qué probabilidad existe que una rata padezca osteopenia si su valor de remodelado óseo es de 0,013? En este caso la variable dependiente es la presencia o no de osteopenia. Habitualmente se coloca el valor 1 a las unidades experimentales con osteopenia y 0 a las unidades experimentales normales o sin osteopenia.

Utilizaremos de R la función glm (generalized linear model) del paquete stat, que normalmente se instala al instalar R.

Introduzcamos en nuestro espacio de trabajo la tablaR441 de la planilla de cálculo tablaR4-4

```
> tablaR441<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

4.2. Regresión logística con una variable independiente cualitativa

En este ejemplo sencillo asignamos el valor 0 a la categoría "sano"y 1 a la categoría "enfermo". Aplicamos 2 tratamientos T1 y T2

```
> summary(tablaR441)
      var         trat       estado
 Min.   : 23.00   T1:29   Min.   :0.00
 1st Qu.:  61.75  T2:31   1st Qu.:0.00
 Median : 166.00          Median :0.00
 Mean   : 305.70          Mean   :0.45
 3rd Qu.: 292.50          3rd Qu.:1.00
 Max.   :2366.00          Max.   :1.00
```

Nuestra variable respuesta será "estado", la cual debemos pasar a factor

```
> tablaR441$estado<-as.factor(tablaR441$estado)
```

ejecutamos nuevamente summary() para comprobar el cambio

```
> summary(tablaR441)
      var         trat    estado
 Min.   :  23.00  T1:29   0:33
 1st Qu.:  61.75  T2:31   1:27
 Median : 166.00
 Mean   : 305.70
 3rd Qu.: 292.50
```

```
 Max.   :2366.00
```

Es decir tenemos 27 unidades "enfermas" y 33 "sanos".
Descripción de la tabla: en la primer columna, var, tiene los valores de una variable medida en las unidades experimentales. Estas unidades experimentales recibieron uno de dos tratamiento: T1 o T2, cuyos objetivos es la cura de una enfermedad. Si luego del tratamiento la unidad experimental no presenta la enfermedad decimos que es sana y la variables estado toma el valor 0. En cambio si permanece enferma, la variable estado toma el valor 1.

Supongamos que la unidades experimentales son ratas. Queremos saber si la probabilidad que una rata esté en estado 0 o 1 tiene el mismo valor dependiendo de si se aplicó T1 o T2. Si uno de los tratamientos fuera más efectivo para la cura de la enfermedad, entonces la probabilidad de que un animal esté en estado 0, cuando recibió dicho tratamiento debería ser mayor que si no lo recibió.

El modelo de regresión logística se puede escribir:

log(P/(1-P))= a + b * trat

que en códigos de R se escribe

```
> rltablaR441<-glm(estado~trat,data=tablaR441,family="binomial")

> summary(rltablaR441)
Call:
glm(formula = estado ~ trat, family = "binomial", data = tablaR441)
Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.6459  -0.5448  -0.5448   0.7726   1.9905

Coefficients:
             Estimate   Std. Error   z value   P r(>|z|)
(Intercept)  -1.8326     0.5385        -3.403   0.000666 ***
tratT2        2.8886     0.6771         4.266   1.99e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.577  on 59  degrees of freedom
Residual deviance: 58.672  on 58  degrees of freedom
AIC: 62.672
Number of Fisher Scoring iterations: 4
```

Vemos que el coeficiente correspondiente al tratamiento T2 es significativamente diferente de cero. El coeficiente para el tratamiento T1 es cero.

En base a esto podemos escribir las ecuaciones del modelo para los animales que recibieron y que no recibieron tratamiento, reemplazando coeficientes estimados en el modelo

log(P/(1-P))= a + b * trat

para T1

a= -1,8326

b=0

entonces resulta

log(P/(1-P))= -1,8326

despejamos P

P/(1-P) = 0,0147

P = 0,0147 – 0,0147* P

1,0147 P = 0,0147

P = 0,014

que indica que la probabilidad de estar enfermo luego de aplicar el tratamiento T1 es de 1.4%.

para las unidades que recibieron el tratamiento T2

a= -1,8326

b= 2,8886

log(P/(1-P))= -1,8326 + 2,8886

log(P/(1-P))= 1,056

P/(1-P) = 11,3763

P = 11,3763- 11,3763*P

12,3763P = 11,3763

P = 0,92

La probabilidad que una rata esté enferma si recibió el tratamiento T2 es de 92%

Calculemos ahora los Odds ratio (OR)

```
> exp(cbind(OR = coef(rltablaR441), confint(rltablaR441)))
                 OR          2.5 %      97.5 %
(Intercept)  0.16000    0.04711747  0.4122268
tratT2      17.96875    5.20059349 76.9374470
```

la tabla siguiente nos muestra el valor de OR

El OR para los animales en los que se aplicó el tratamiento T2 es OR= 17.96875

El valor de OR si vale 1, indica el riesgo relativo de permanecer enfermo luego de un tratamiento respecto al otro no es diferente

Si el valor de OR es mayor que 1 indica que el riesgo relativo de permanecer enfermo luego del tratamiento T2 es mayor que con el tratamiento T1

Si el valor de OR es menor que 1 el riesgo relativo de padecer la enfermedad luego del tratamiento T2 es menor que si aplicamos el tratamiento T1.

¿Cómo sabemos si el OR es o no diferente de 1? a través del intervalo de confianza. El intervalo de confianza del OR no debe contener el valor 1.

En este caso el intervalo de confianza (5.20059349 - 76.9374470) no incluye a 1 y por lo tanto podemos decir que el riesgo relativo de permanecer enfermo luego de aplicar T2 es mayor que si aplicamos T1.

Si bien en casos más complejos no será posible hacer el razonamiento que haremos a continuación, servirá para interpretar los valores de OR.

En este caso podemos hacer una interpretación del OR. Ejecutemos y analicemos

```
> table(tablaR441$trat,tablaR441$estado)

      0  1
 T1 25  4
 T2  8 23
```

Si dividimos el número de enfermos por sanos con cada tratamiento, obtenemos lo que se conoce como relación de odds. El cociente de odds de T2 y T1 nos da el OR

$$OR = \frac{\frac{23}{8}}{\frac{4}{25}} = 17,96$$

Ecuación 4.1.

El OR nos indica cuantas veces mayor es la relación enfermo/sano con T2 que con T1. Da una información similar a la relación de riesgo relativos entre ambos tratamientos.

Si calculamos el riesgo relativo de padecer enfermedad en cada tratamiento

RR_{T2}= 23/31 = 0,74

RR_{T1}= 4/29 = 0,137

con estos datos podemos calcular el riesgo relativo de padecer la enfermedad entre los que recibieron T2 respecto a T1

RR = 0,74/0,137 = 5.4

Concluimos que OR es un estimador de la relación de los riesgos relativos.

4.3. Regresión logística con variables continuas y categóricas.

Utilizaremos los datos de la tablaR441, pero incluiremos en este caso la variable var y trat. Creamos un objeto al que llamamos rltablaR441completo, ya que incluye todas las variables.

```
> rltablaR441completo<-glm(estado~trat+var,data=tablaR441,family="binomial")

> summary(rltablaR441completo)
Call:
glm(formula = estado ~ trat + var, family = "binomial", data = tablaR441)
 Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9322  -0.4642  -0.3201   0.6198   2.2362

Coefficients:
             Estimate   Std. Error   z value   Pr(>|z|)
(Intercept) -3.195426     0.828326  -3.858     0.000114 ***
tratT2       1.887555     0.823161   2.293     0.021845 *
```

```
var             0.009080   0.003314   2.740    0.006149 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82.577  on 59  degrees of freedom
Residual deviance: 43.893  on 57  degrees of freedom
AIC: 49.893

Number of Fisher Scoring iterations: 7

> exp(cbind(OR = coef(rltablaR441completo), confint(rltablaR441completo)))
Waiting for profiling to be done...
                      OR          2.5 %     97.5 %
(Intercept) 0.04094907  0.005827194  0.1658354
tratT2      6.60320632 1.326887677 36.6418319
var         1.00912103 1.003582117  1.0173484
```

y calculamos los intervalos de confianza para los parámetros del modelo

```
> cbind(coef=coef(rltablaR441completo),confint(rltablaR441completo))
Waiting for profiling to be done...
                coef                  2.5 %     97.5 %
(Intercept) -3.195426181 -5.145219697 -1.7967598
tratT2       1.887555337  0.282836108  3.6011905
var          0.009079683  0.003575716  0.0171996
```

4.4. Elegir las variables más adecuadas para el modelo

Para seleccionar el mejor modelo utilizamos la función stepAIC de la biblioteca MASS. En esta función colocamos el modelo completo y descartamos aquellas variables que hacen que el índice AIC disminuya al retirarla del modelo.

```
> library(MASS)
> steprltablaR441 <-stepAIC(rltablaR441completo,direction="both")
Start:  AIC=49.89
estado ~ trat + var

       Df Deviance    AIC
<none>      43.893 49.893
- trat    1   49.172 53.172
- var     1   58.672 62.672
```

vemos que con las dos variables trat y var el modelo tiene AIC= 49,893. Retirando cualquiera de las dos de las variables el índice AIC aumenta, por lo tanto no es conveniente sacar ninguna variable y el mejor modelo será:

estado ~ var + trat

Es decir que la probabilidad de estar sano o enfermo estará explicado por el valor de ambas variables.

5. Clase 4.5

Video: https://youtu.be/CKVxrC6gVXI
tablas: http://hdl.handle.net/2133/15485

5.1. Análisis de los componentes principales

El análisis de los componentes principales permite observar la relación entre diferentes variables cuanti y cuantitativas. Es de gran utilidad en el análisis exploratorio de grandes tablas de datos.

El método cuya matemática y fundamentos obviaremos, busca componentes que explican la mayor variabilidad de los datos y en general analiza los dos primeros componentes. Es de esperar que estos dos primeros componentes expliquen un porcentaje importante de la variabilidad de los datos.

Con R, el análisis de los componentes principales (de ahora en adelante PCA por principal component analysis) se puede realizar descargado la biblioteca FactoMineR, que se halla en los repositorios utilizados habitualmente.

Cargue la biblioteca FactoMineR con el siguiente código

```
> library(FactoMineR)
```

Ejecute luego las funciones necesarias.

Para introducirnos en el tema trabajaremos con una base de datos que contiene 12 variables y aproximadamente 200 unidades experimentales en las que se midieron estas 12 variables.

Introduzca los datos de la tablaR451 de la planilla de cálculo tablaR4-5.xls/ods.

```
> tablaR451<-read.table("clipboard",header=TRUE,dec=".",sep="\t",encoding="latin1")
```

vea los nombres de las columnas

```
> names(tablaR451)
 [1] "peso"        "edad"        "cfx"         "cfmax"       "cenergia"
 [6] "crigidez"    "cyoung"      "BVTV"        "tbth"        "tbn"
[11] "tbsp"        "tratamiento"
```

Figuran en la tabla peso y edad (columnas 1 y 2) , luego variables biomecánicas del hueso trabecular (columnas 3-7), luego variables histomorfométricas óseas trabeculares (columnas 8-11) y una columna sobre el tratamiento aplicado. Las variables 1-11 son cuantitativas y la 12 es cualitativa.

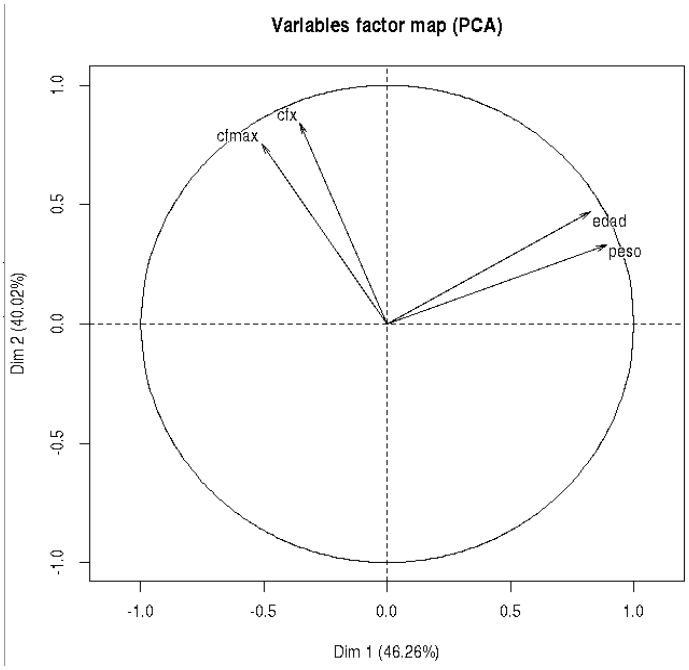
El objetivo de nuestro estudio es hallar relaciones entre las variables, trabajo que podríamos hacer utilizando análisis de correlación. Sin embargo este análisis nos serviría para observar de a pares de variables. Contrariamente PCA nos permitirá realizar un análisis global de los datos y tomar decisiones respecto del conjunto de los datos.

Advertencia: Cuando realizamos el análisis PCA la tabla no puede utilizar columnas que estén separadas por columnas sin utilizar en el análisis. Comencemos analizando la relación entre las variables 1 a 4. Como las columnas 1 a 4 son consecutivas utilizamos directamente la tablaR451

```
> pcatablaR451<-PCA(tablaR451[,1:4], scale.unit = TRUE, ind.sup = NULL, quanti.sup = NULL, quali.sup = NULL,row.w = NULL,col.w = NULL,graph = TRUE,axes = c(1,2))
```

obtendrá dos gráficos:

1) Gráfico de las variables



qué nos indica el gráfico?

1- peso y edad estarían muy correlacionadas, ya que los vectores apuntan en misma dirección y sentido.

2- cfx y cfmax estarían correlacionadas directamente, ya que ambos vectores apuntan en igual dirección y sentido.

3- peso (o edad) no estaría correlacionado con cfx (o cfmax) porque sus direcciones son perpendiculares.

4- En sentido vertical se delimitarían dos semiplanos: hacia la derecha mayor peso y edad, hacia la izquierda lo contrario

5- en sentido vertical también tenemos dos semiplanos: hacia arriba mayor cfx y cfmax, hacia abajo lo contrario.

Podríamos a modo de ejemplo corroborar los puntos 1, 2 y 3 aplicando un test de correlación

punto 1: ¿Existe correlación entre edad y peso? Hacemos para ello un análisis de correlación

```
> cor.test(tablaR451$edad,tablaR451$peso)

        Pearson's product-moment correlation

data:  tablaR451$edad and tablaR451$peso
t = 17.232, df = 182, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7253528 0.8367442
sample estimates:
      cor
```

0.7873925
El valor de p-value y el coeficiente de correlación nos confirman lo observado en la figura.

punto 3: ¿Están correlacionadas edad y cfx?

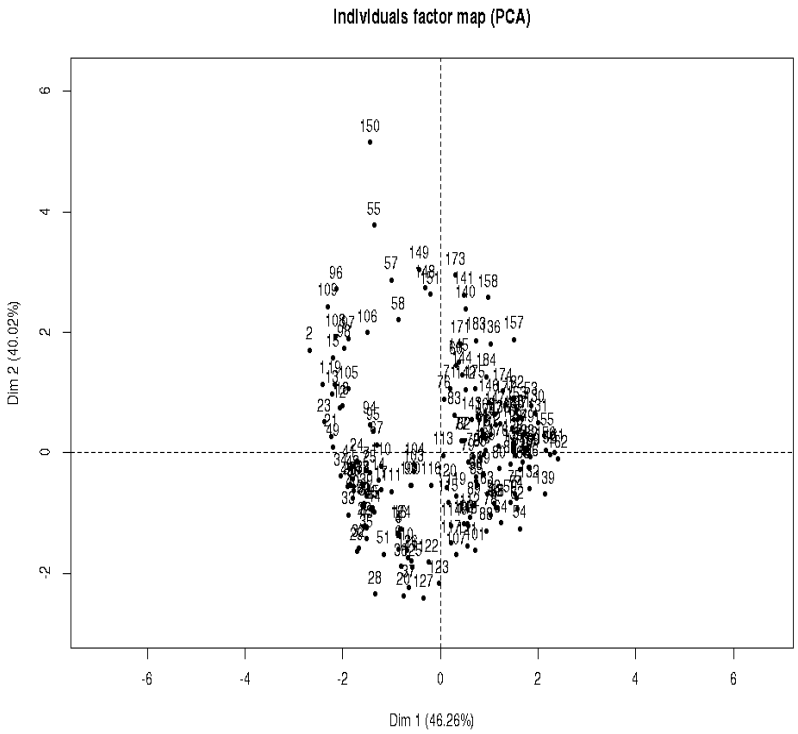
> cor.test(tablaR451$edad,tablaR451$cfx)

Pearson's product-moment correlation

data: tablaR451$edad and tablaR451$cfx
t = 1.0163, df = 177, p-value = 0.3109
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.0712984 0.2203806
sample estimates:
cor
0.07617043
El valor de p-value y el coeficiente de correlación nos confirman la falta de correlación, interpretada en el gráfico de las variables como vectores perpendiculares.

Sorprendente! pruebe las otras correlaciones si aun no se convenció.

2) Gráfico de los individuos



Este gráfico muestra con un punto (además nos muestra el número de cada individuo, que sería la fila en la que se encuentra en el data.frame) en los semiplanos que mencionamos.

Según lo que dijimos anteriormente, deberíamos encontrar hacia la derecha individuos de más edad y pesados que hacia la izquierda y hacia arriba individuos con mayor cfx y cfmax.

Hagamos una comprobación.

Mire la gráfica, tomaremos dos grupos de individuos

grupo A: 51,20 y 28 que están abajo y la izquierda.

grupo B: 136, 157 y 158 que está arriba y un poco a la derecha.

Qué esperaríamos?

Se espera que el grupo A por estar abajo y la izquierda tenga en promedio menos peso, edad, cfx y cfmax que el grupo B.

Veamos entonces solo edad y cfx, ya que sabemos como correlacionan estas con las otras variables.

grupo A

busquemos el individuo que ocupa el lugar 51 en nuestra tabla utilizando recursos ya aprendidos

> tablaR451[51,]

peso edad cfx cfmax cenergia crigidez cyoung BVTV tbth tbn tbsp tratamiento

51 207 66 20.08 38.37 3.03 65.7 0.023 22.813 43.641 5.225 147.987 Sham

asi buscamos los otros individuos del grupo A (20 y 28) y del grupo B (136, 157 y 158) y construimos la tabla con los valores de peso y cfx. Las líneas sombreadas nos indican las medias de los dos grupos formados.

	peso	cfx
51	207	20,08
20	219	3,23
28	167	7,68
media	197,67	10,33
136	445	50,56
157	490	65,63
158	490	50,56
media	475	55,58

los individuos 51, 20 y 28 están abajo y la izquierda, esperaría que tengan menor peso y menor cfx que los individuos 136, 157 y 158. Lo que esperaba se comprueba al analizar las medias de cada variable para cada grupo.

Pruebe a incluir otras variables. Para ello con el mismo código agregue columnas. Es importante tener presente que las columnas tienen que estar consecutivas. En caso que desee tomar columnas que no están consecutivas cree un nuevo data.frame con el código que ya conoce para obtener columnas específicas.

por ejemplo, el código siguiente incluye a la columna 5. Directamente amplia el rango ya que está utilizando las columnas 1 a la 5 que son consecutivas.

> pca<-PCA(pcacurso[,1:5], scale.unit = TRUE, ind.sup = NULL, quanti.sup = NULL, quali.sup = NULL,row.w = NULL,col.w = NULL,graph = TRUE,axes = c(1,2))

pero si quisiera incluir la columna 6 y 7, ademas de las 1 a la 4, primero debería cortar la tabla como ya conoce. A modo de ayuda se da el código para hacerlo

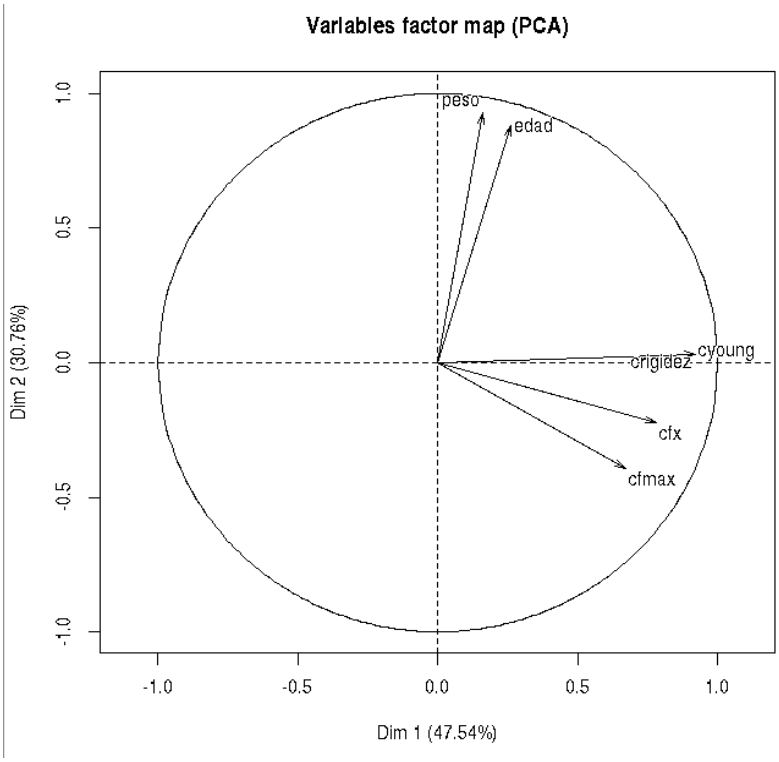
> slicetablaR451<-tablaR451[,c(1:4,6,7)]

slicetablaR451, será un data.frame con 6 columnas.

luego ejecuta

> pcaslicetablaR451<-PCA(slicetablaR451[,1:6], scale.unit = TRUE, ind.sup = NULL, quanti.sup = NULL, quali.sup = NULL,row.w = NULL,col.w = NULL,graph = TRUE,axes = c(1,2))

obtendremos los gráficos de variables e individuos



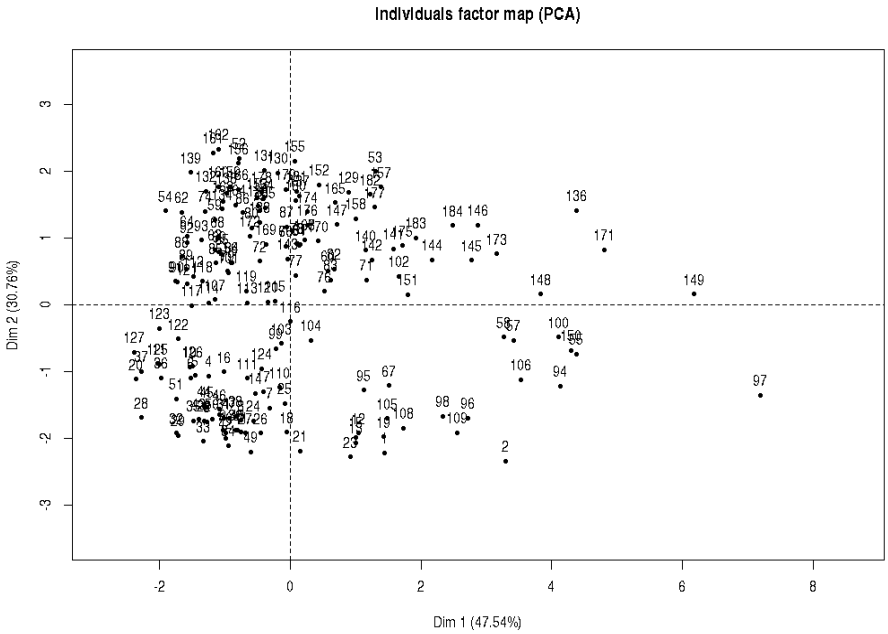
Interpretemos nuevamente. Los dos componentes incorporados (ejes) explican aproximadamente el 77% de las variabilidad de las variables involucradas.

Las variables peso y edad siguen mostrándose correlacionadas, de la misma manera que cyoung, crigidez, cfx y cfmax. Las variables cfx y cfmax son dos variables que miden la resistencia del

hueso en estudio. Podemos ver que cyoung y crigidez correlacionan con ambas variables (por hallarse en la misma dirección), pero las cuatro variables no correlacionan con peso y edad. En un rápido y primer análisis podemos decir que la resistencia ósea no correlaciona con la edad. La gráfica también nos indica que en el semiplano superior hallaremos los individuos más pesados y de mayor edad, mientras que en el de la derecha hallaremos los individuos con mayor resistencia ósea. De esta manera también conocemos que en el semiplano superior derecho tendremos a los individuos de mayor edad y con mayor resistencia ósea, mientras que en inferior izquierdo hallaríamos a los de menor edad, peso y resistencia ósea.

¿Qué individuos espera hallar en el semiplano de arriba a la izquierda? Por supuesto, los de mayor edad pero con baja resistencia ósea.

veamos el gráfico de los individuos



Esperaríamos que por ejemplo el individuo 136 fuera uno de los de máś edad y mayor resistencia ósea (vea el punto en el cuadrante superior derecho), mientras que el 28 fuera más joven y de menor resistencia (vea el punto en el cuadrante inferior izquierdo)

5.2. Inclusión de variables suplementarias cuantitativas

Si quisiéramos saber como se ubicaría otra variable sin cambiar la relación entre las existentes podemos proceder de la siguiente manera.

Supongamos que me interesa ver como se ubica la variable cuantitativa *cenergia. Para ell*o deberemos construir una nueva tabla que incluya a dicha variable. En el análisis anterior teníamos las variables edad, peso, cfx,cfmax,crigidez y cyoung.

Veamos el total de las variables nuevamente

```
> names(tablaR451)
 [1] "peso"      "edad"      "cfx"       "cfmax"     "cenergia"
 [6] "crigidez"  "cyoung"    "BVTV"      "tbth"      "tbn"
```

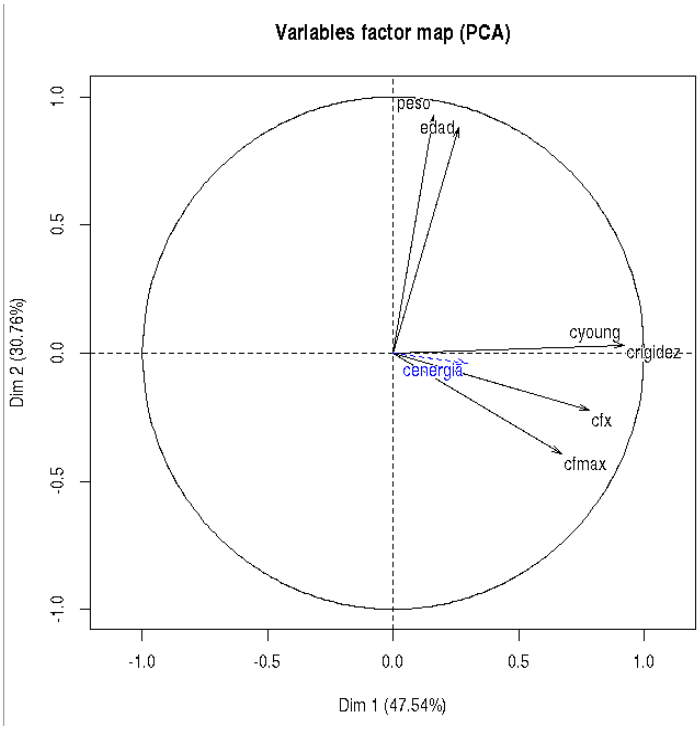
[11] "tbsp" "tratamiento"
nos interesa un slice del data.frame que tenga las variables: 1-4 y 6 y 7 que ya habíamos incluido. Agregaremos ahora la variables 5.

> slicetablaR451<-tablaR451[,c(1:4,6,7,5)]

> names(slicetablaR451)
[1] "peso" "edad" "cfx" "cfmax" "crigidez" "cyoung" "cenergia"
ejecutamos el siguiente código, incluyendo las 7 variables, pero indicando en el argumento quanti.sup, que la variable 7 del data.frame slicetablaR451 es una variable cuantitativa suplementaria

> pcaslicetablaR451<-PCA(slicetablaR451[,1:7], scale.unit = TRUE, ind.sup = NULL, quanti.sup = 7, quali.sup = NULL,row.w = NULL,col.w = NULL,graph = TRUE,axes = c(1,2))

veamos el gráfico de las variables



Sin haber modificado la relación anterior entre las variables, vemos que cenergía correlacionaría con las variables que indican resistencia ósea.

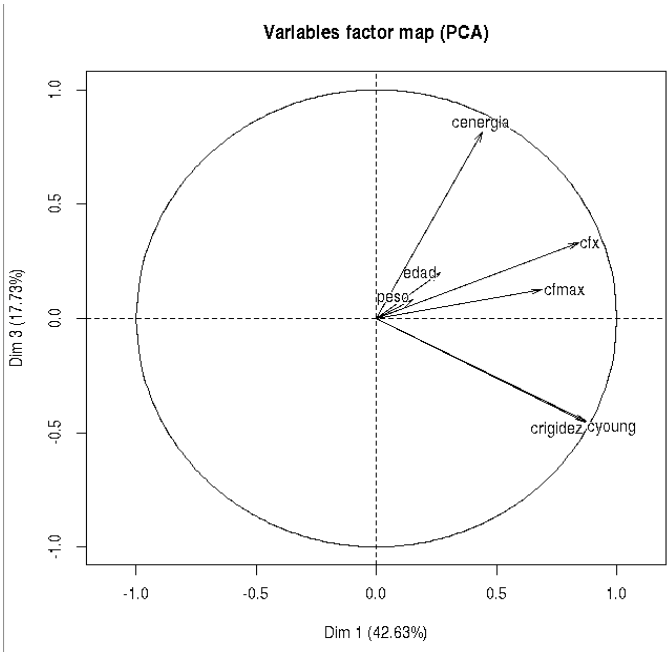
Si ejecutamos el código con el argumento cuanti.sup=NULL, la variable cenergía quedaría incluida en el análisis.

Cabe en este momento hacer un comentario importante. Cuando los vectores apuntan para el mismo lado pero ya no tienen un largo que llega prácticamente al borde puede ser engañoso. El ser más corto quiere decir que se halla en el plano pero tiene una importante componente hacia atrás o adelante. Por lo tanto cenergia podría no estar correlacionando con las variables de resistencia, las cuales si están relacionadas por estar en misma dirección y llegar casi hasta el borde. Lo mismo se interpreta con peso y edad.

Para verificar esto pueden tomarse otras componentes que permitan ver mejor la situación.

Tomaremos los ejes 1 y 3, cambiando el código como se indica abajo

> pcaslicetablaR451<-PCA(slicetablaR451[,1:7], scale.unit = TRUE, ind.sup = NULL, quanti.sup = NULL, quali.sup = NULL,row.w = NULL,col.w = NULL,graph = TRUE,axes = c(1,3))



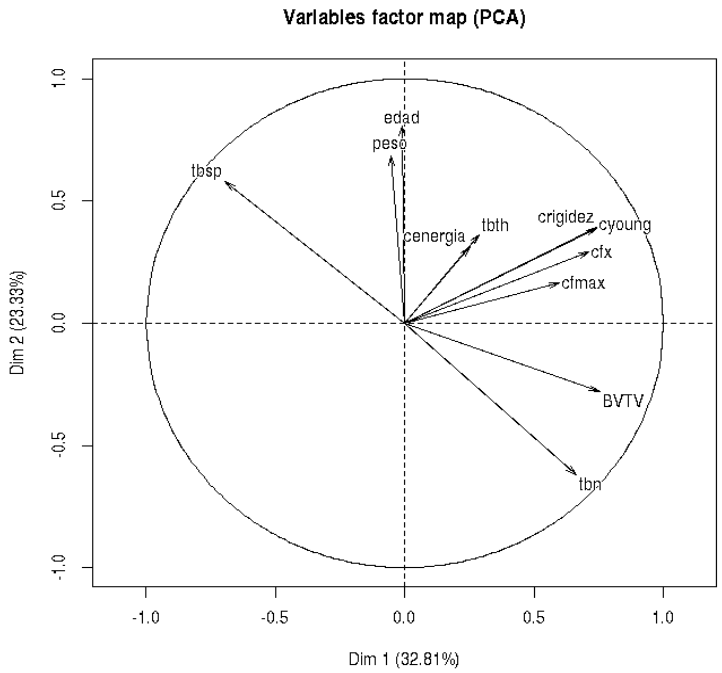
tal como anticipamos, no hay tan fuertes correlaciones. Es más, cenergia no correlaciona con crigidez ni con cyoung. Note que esta proyección no muestra claramente la relación entre peso y edad como si lo hacen los ejes 1 y 2.

5.3. Inclusión de variables cualitativas

La tablaR451 tiene una variable cualitativa que es el tratamiento. El tratamiento tiene dos niveles: ratas ovariectomizadas (OVX) y ratas normales (Sham). Incluiremos todas las variables y veremos que información tenemos.

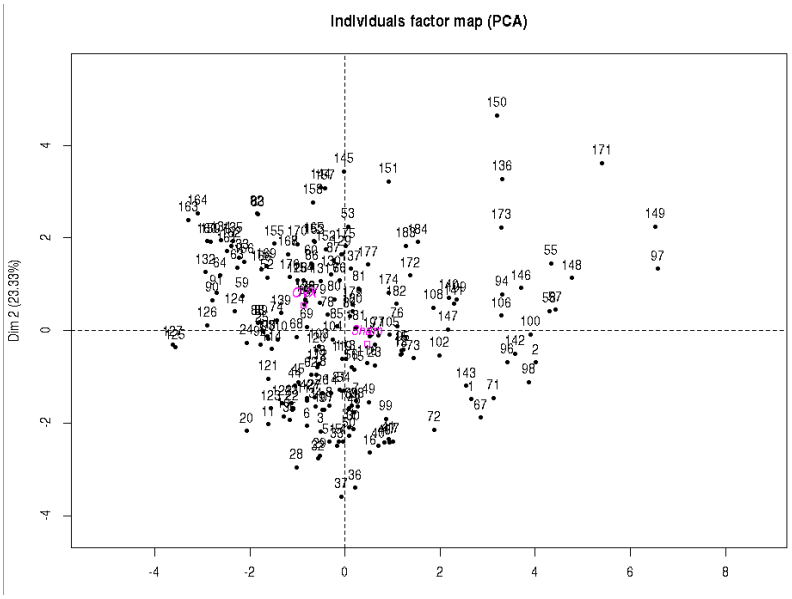
> pcatablaR451<-PCA(tablaR451[,1:12], scale.unit = TRUE, ind.sup = NULL, quanti.sup = NULL, quali.sup = 12,row.w = NULL,col.w = NULL,graph = TRUE,axes = c(1,2))

gráfico de las variables



las variables crigidez, cyoung, cfx, cfmax, BVTV, tbth, cenergía y tbn son variables que indican la presencia de un hueso más fuerte. De mirar la orientación de los vectores podemos ver que el semiplano de la derecha se asocia entonces a huesos mas fuertes y el de la izquierda por lo consiguiente a huesos más débiles. Como el peso y edad son perpendiculaes a las variables indicadas podemos concluir que en estos datos la resistencia de los huesos no depende de la edad.

Veamos el gráfico de los individuos que muestra los individuos y la variable cualitativa.



Vemos en color violeta los nombres de los tratamientos. OVX quedó en el cuadrante superior izquierdo y Sham en el semiplano derecho. El resultado es claro. Las ratas ovariectomizadas se ubican en el semiplano izquierdo asociado a una menor resistencia ósea. Contrariamente las sham están a la derecha asociadas a mayor resistencia ósea.

Si la gráfica no es fácilmente visible se puede ejecutar el código

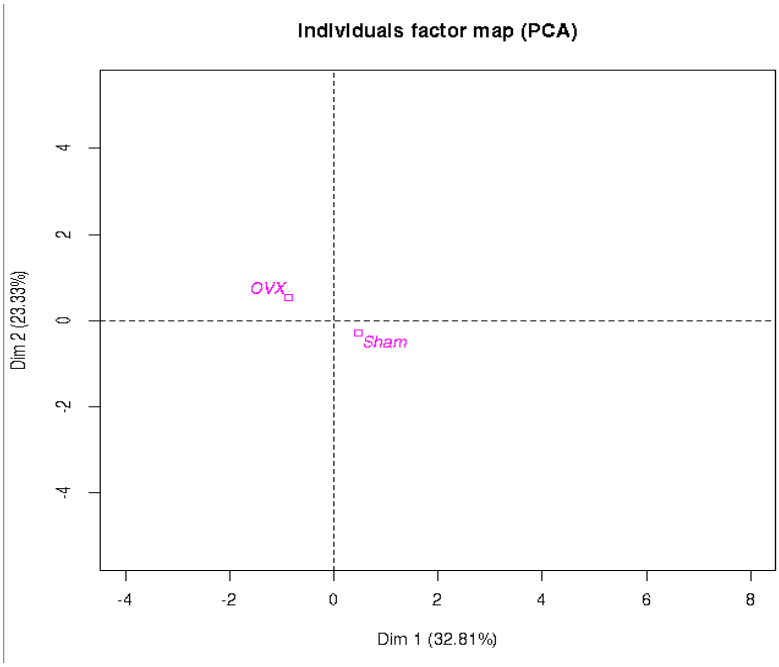
> plot.PCA(pcatablaR451, axes=c(1, 2), choix="ind", xlim=c(-4,8), ylim=c(-4,4),invisible="ind")

en el que con

choix="ind" pedimos el gráfico de los individuos

xlim e ylim fija los límites en función de la gráfica realizada anteriormente.

invisible="ind", no muestra los puntos de los individuos.



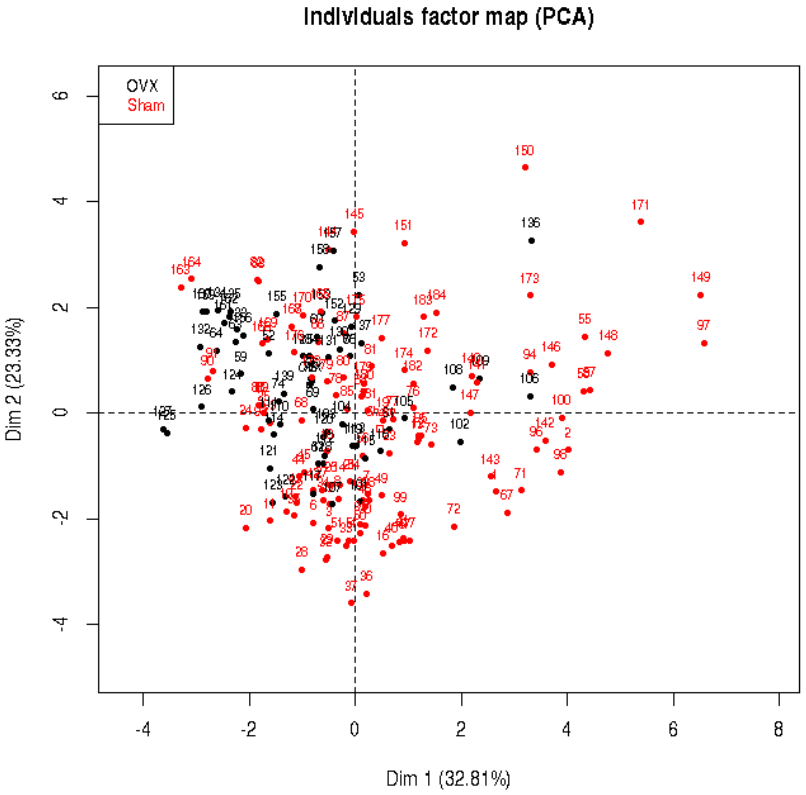
claramente muestra que las Sham están asociadas a alta resistencia ósea que las OVX.

Podemos también colorear los individuos de cada tratamiento de manera de ver su distribución

Para ello utilizamos el código

> plot.PCA(pcatablaR451, axes=c(1, 2), choix="ind", habillage=12, cex=0.7)

habillage=12 indica que la variable 12 de tablaR451 es la cualitativa



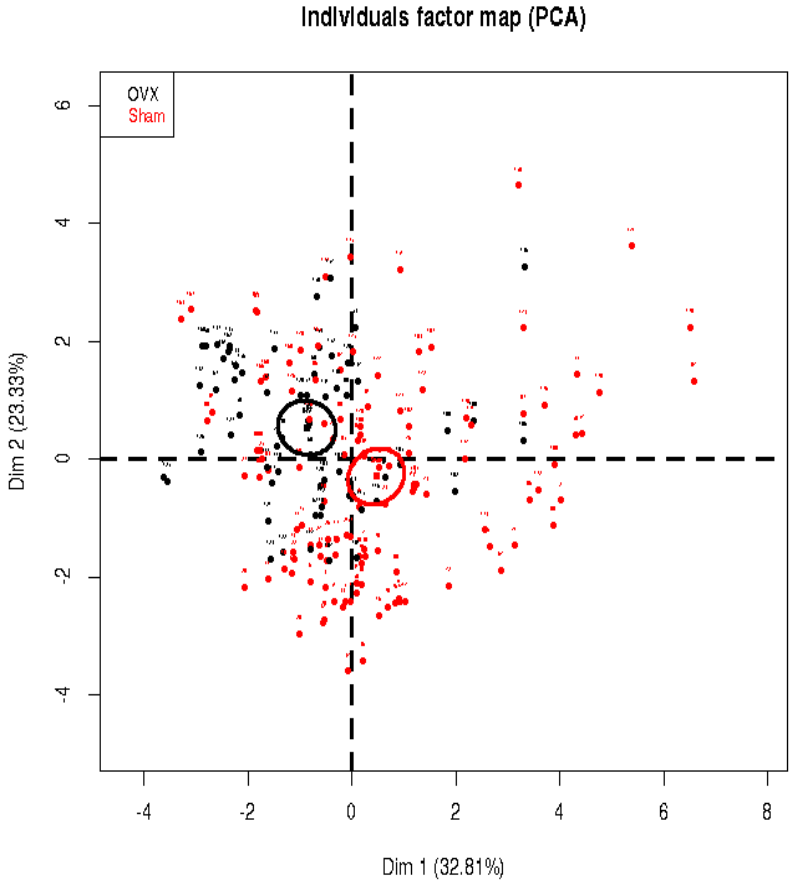
claramente se ve que las ratas normales (sham) se desplazan hacia la derecha de las ovariectomizadas (OVX), aunque se observa superposición de las poblaciones.

Se puede a su vez ubicar elipses de confianza, que serían áreas delimitadas que agrupan a un porcentaje importante de los individuos de cada categoría. Si las elipses no se cruzan interpretaremos la situación como que la variable cualitativa en estudio tiene un efecto importante en las variables cuantitativas.

> concat = cbind.data.frame(tablaR451[,12], pcatablaR451indcoord)

> ellipse.coord = coord.ellipse(concat, bary = TRUE, level.conf = 0.99)

> plot.PCA(pcatablaR451, habillage = 12, ellipse = ellipse.coord, cex = 0.3,lwd=3)



podemos comprobar que si aumentamos el nivel de significación, las elipses aumentan. cuando se toquen interpretaremos que no hay diferencias en las categorías investigadas.

6. Clase 4.6

Video: https://youtu.be/rr-TZlh1Rmc
Tablas: http://hdl.handle.net/2133/15486

6.1. Análisis de las correspondencias múltiples

El análisis de las correspondencias múltiples, es una técnica que permite abordar grandes tablas de datos con variables de tipo cualitativa o categórica. Esta técnica permite tener un panorama de las relaciones que entre ellas existen, orientándonos a los ensayos de hipótesis posteriores.

Realizaremos una primer aproximación al tema utilizando una base de datos sencilla

Introduzcamos los datos de la tablaR461 de la planilla de cálculo tablaR4-6.ods/xls. Primero cargue la biblioteca FactoMineR

```
> library(FactoMineR)
```

```
> tablaR461<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

```
> summary(tablaR461)
  individuo  actividadfisica IMC    sexo   edad
 1     : 1         no:14        B: 3    H:16   A :16
 2     : 1         si:16        N:11    M:14   AM:14
 3     : 1                      O: 5
 4     : 1                      S:11
 5     : 1
 6     : 1
```

La tabla siguiente se originó de una encuesta entre adultos de ambos sexos en que obtuvimos información sobre algunas variables categóricas:

a- Hace actividad física?: si – no

Considerando si, cuando realiza más de 150 min/semana

b- IMC: se obtuvo el peso y la estatura de cada individuo, luego se calculó el IMC y con este valor se clasificó a los individuos como

Bajo peso: B

Peso normal: N

Sobre peso: S

Obesidad: O

c- sexo: M (mujer), H (hombre)

d- en base a la edad se clasificaron en:

Adultos: A : menores de 65 años

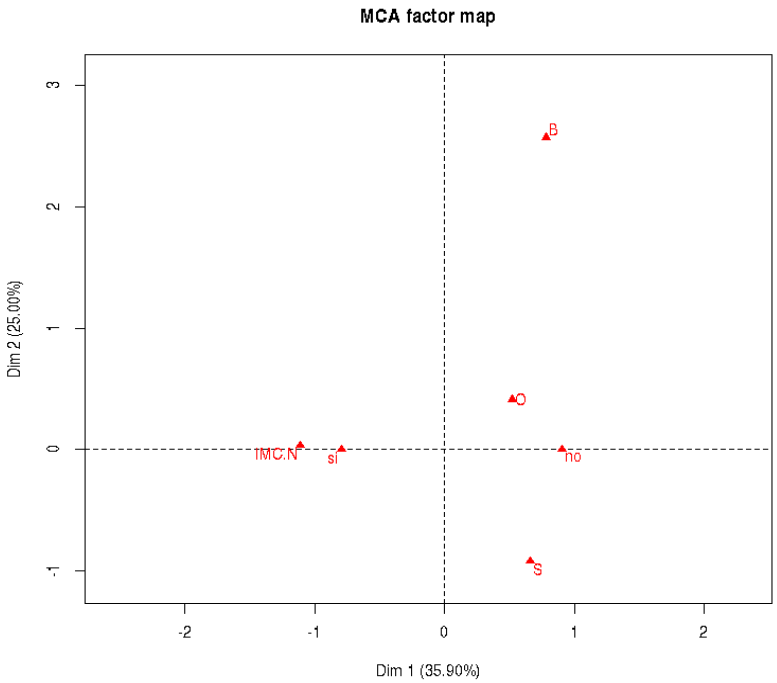
Adultos mayores: AM: mayores de 65 años

Las variables son categórica y se deben hallar como factor. Si alguna no lo fuera corríjalo con la función "as.factor"

En primer lugar, con el fin de construir el entendimiento de este análisis, veremos que relación existe entre IMC y la actividad física (con el fin de ir comprendiendo el análisis).

```
> mcatablaR461<-MCA(tablaR461[,c(2:3)])
```

obtenemos varias gráficas, veamos la gráfica de las variables



En la gráfica tenemos puntos que corresponden a las categorías de las dos variables incluidas en el estudio:

niveles de IMC: IMC.N, B, S,O

niveles de actividad física: si, no.

¿Qué nos indica la gráfica?

1- Referido a actividad física "no" y "si" se hallan sobre la misma linea horizontal, pero uno a la derecha y otro a la izquierda. Por lo que podemos pensar que el semiplano derecho contiene individuos que no hacen actividad física y el izquierdo que si hacen actividad física.

2- Con respecto al IMC, a la izquierda tenemos N y a la derecha las otras categorías.

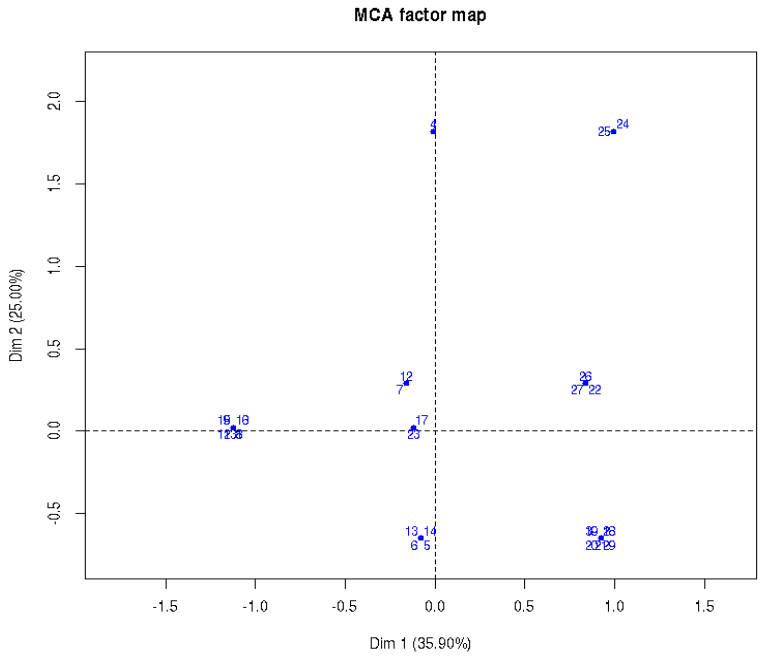
3- además en el semiplano superior tenemos los individuos de bajo peso y obesos y abajo los de sobrepeso estando los normales sobre el eje horizontal.

4- IMC.N está cerca del punto "si" de actividad física, mientras que O y S están cerca del "no" de actividad física. Por otra parte B (bajo peso) no está cerca de ninguno.

A primera vista podríamos decir que el IMC N se asocia a actividad física, mientras que S y O se asocian con falta de actividad física.

Hagamos algunas comprobación.

Veamos el gráfico de los individuos. En esta gráfica vemos los individuos identificados con el número de fila en la tabla



En esta gráfica vemos los individuos numerados de 1-30, donde algunos están superpuestos. Tomemos los individuos 22, 26 y 27.

Se hallan a la derecha y arriba del eje horizontal. Según la gráfica de las variables este sector correspondería a individuos con IMC= O y que no hacen actividad física.

Pedimos esos individuos con el siguiente código

```
> tablaR461[c(22,26,27),]
   individuo actividadfisica IMC  sexo edad
22        22            no   O     H   AM
26        26            no   O     M    A
27        27            no   O     H   AM
```

Hagamos otra comprobación, esta vez en sentido contrario. Seleccionemos de la tabla individuos que simultáneamente hacen actividad física y que tiene IMC=S

```
> tablaR461[tablaR461$actividadfisica=="si"& tablaR461$IMC=="S",]
   individuo actividadfisica IMC sexo edad
5         5            si   S    M   AM
6         6            si   S    H    A
13       13            si   S    H    A
14       14            si   S    H   AM
```

Intentemos buscar los individuos de la tabla en la gráfica de los individuos. Como podemos ver están abajo y tirando a la izquierda, zona que coincide con la ubicación del nivel sobrepeso de la variable IMC y hacia la izquierda coincidiendo con el nivel actividadfisica: si.

Asociación entre variables

De la gráfica de las variables deducimos que existe alguna asociación entre actividad física (si -no) e IMC (B, N, S y O). Surge de la gráfica una asociación entre IMC N y actividad física y por otro lado IMC S u O y actividad física no.

Veamos primero una tabla de contigencia

```
> table(tablaR461$IMC,tablaR461$actividadfisica)
   no si
B   2  1
N   2  9
O   3  2
S   7  4
```

Se puede ver más individuos O y S entre los que no hacen actividad física y más N entre los que si lo hacen

Para verificar si dicha asociación es significativa, podemos hacer una prueba chisq.test

```
> chisq.test(table(tablaR461$IMC,tablaR461$actividadfisica))
        Pearson's Chi-squared test
data:  table(tablaR461$IMC, tablaR461$actividadfisica)
X-squared = 5.6981, df = 3, p-value = 0.1273
Warning message:
In chisq.test(table(tablaR461$IMC, tablaR461$actividadfisica)) :
  Chi-squared approximation may be incorrect
```

No podemos afirmar con p<0,05 que exista dicha asociación.

Sin embargo en la tabla podemos observar un reducido número de individuos con IMC=B,

podríamos eliminar dichos individuos. Creamos una nueva tabla sin los individuos de IMC= B a la que llamamos slicetablaR461

```
>  slicetablaR461<-tablaR461[tablaR461$IMC!="B",]
```

```
> table(slicetablaR461$IMC,slicetablaR461$actividadfisica)
   no si
B  0  0
N  2  9
O  3  2
S  7  4
```

Hemos eliminados los valores pero no el nivel B. Entonces eliminamos el nivel B

```
> slicetablaR461$IMC<-droplevels(slicetablaR461$IMC)
```

comprobamos que así sea

```
> table(slicetablaR461$IMC,slicetablaR461$actividadfisica)
   no si
N  2  9
O  3  2
S  7  4
```

realizamos el chisq.test

> chisq.test(table(slicetablaR461$IMC,slicetablaR461$actividadfisica))

Pearson's Chi-squared test

data: table(slicetablaR461$IMC, slicetablaR461$actividadfisica)

X-squared = 5.2036, df = 2, p-value = 0.07414

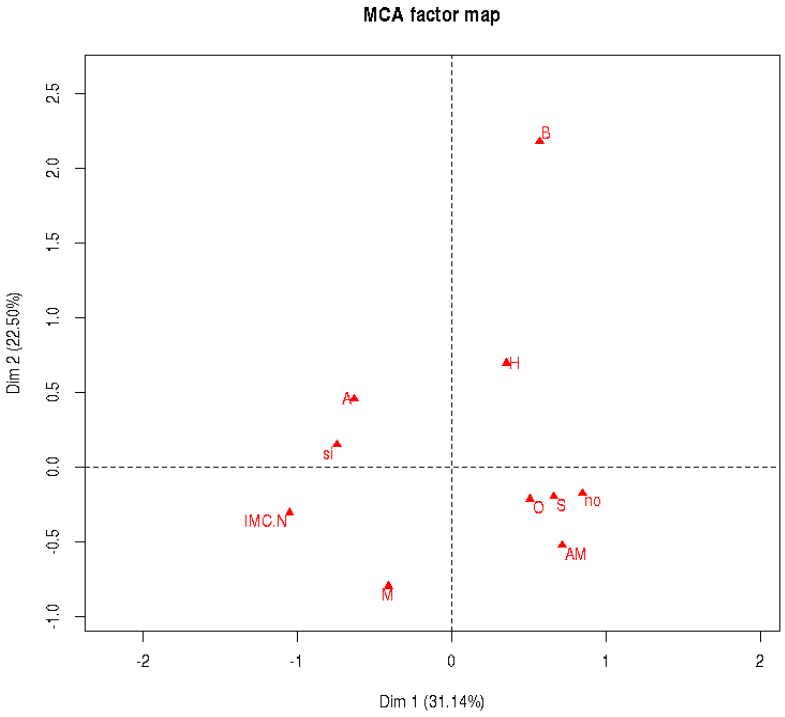
Warning message:

In chisq.test(table(slicetablaR461$IMC, slicetablaR461$actividadfisica)) :

Con un error de tipo I de 0,074 podemos afirmar la asociación entre IMC y actividad física.

Comprendido el funcionamiento de MCA y las gráficas de las variables y los individuos, veremos el resultado del análisis incluyendo todas las variables, menos el número de individuo.

>mcatablaR461<-MCA(tablaR461[,c(2:5)])



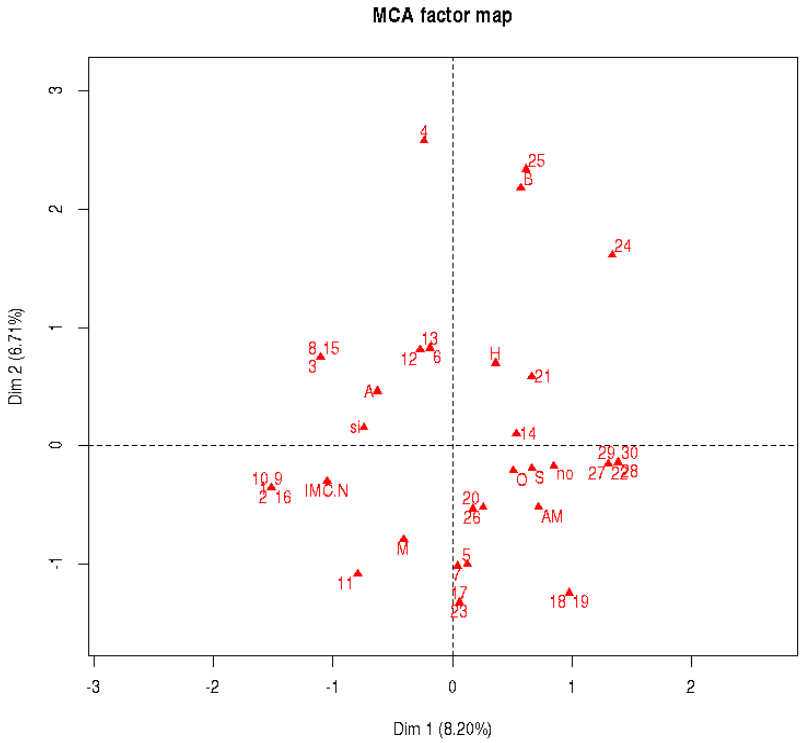
Ubique los diferentes niveles de IMC, sexo, actividad física y edad.

Vemos el área superior izquierda asociada a edad= A y actividad física= si

El área inferior derecha asociada a edad = AM y actividad física= no

Comprobemos distribución de los individuos

Tomemos dos individuos de cada área. Para ello veamos la gráfica de los individuos



Tomemos los individuos 8 y 15 del área superior izquierda (zona Adulto y actividad física si) y 18 y 19 del área inferior derecha (zona S y O, actividad física no y AM)

```
> tablaR461[c(8,15,18,19),]
   individuo actividadfisica IMC sexo edad
8          8              si   N    H    A
15        15              si   N    H    A
18        18              no   S    M   AM
19        19              no   S    M   AM
```

Si no se convenció haga otras comprobaciones.

7. Clase 4.7

Vídeo: https://youtu.be/n-eofO3blr0
tablas: http://hdl.handle.net/2133/15487

7.1. Análisis de las correspondencias múltiples

Análisis de las correspondencias múltiples. Continuación

Como ya vimos en la clase anterior el análisis de las correspondencias múltiples, es una técnica que permite observar la asociación entre variables de tipo cualitativa para luego realizar una comprobación de dicha asociación.

Continuaremos el desarrollo con los datos de la clase anterior, en la que utilizamos datos de una encuesta entre adultos de ambos sexos en que obtuvimos información sobre algunas variables categóricas:

a- Hace actividad física?: si – no

b- IMC: que se categorizó como: Bajo peso: B, Peso normal: N, Sobre peso: S y Obesidad: O

c- sexo: M (mujer), H (hombre)

d- clasificación de adultos como: Adultos (A) y Adultos mayores (AM)

Los datos se tabularon y se halla en el archivo tablaR4-7.ods/xls. Introduzca los valores de la tabla en su espacio de trabajo.

```
> tablaR471<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

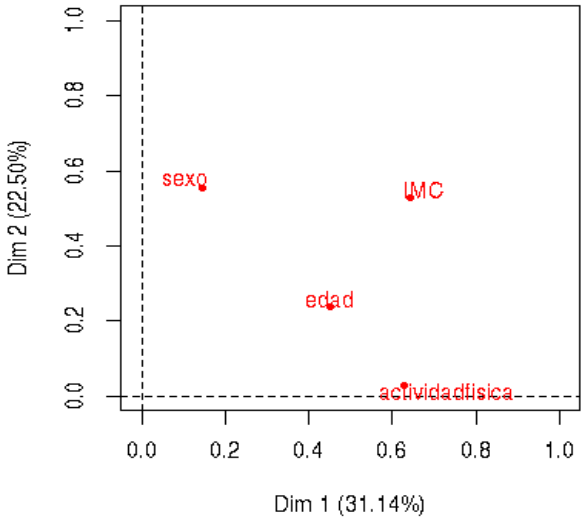
```
> tablaR471$individuo<-as.factor(tablaR471$individuo)
```

```
> summary(tablaR471)
   individuo  actividadfisica IMC    sexo    edad
 1     : 1          no:14        B: 3   H:16    A :16
 2     : 1          si:16        N:11   M:14    AM:14
 3     : 1                       O: 5
 4     : 1                       S:11
 5     : 1
 6     : 1
```

Realicemos el análisis con las columnas 2 a 5.

```
> mcatablaR471<-MCA(tablaR471[,c(2:5)])
```

Como ya vimos nos dará varias gráficas (variables, categorías e individuos), una de ellas es la relación de las variables a las dimensiones

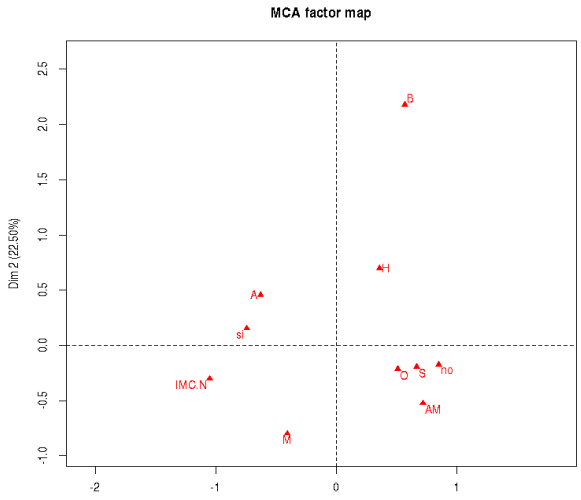


Esta gráfica nos indica que la variables "actividad física" es muy ligada a la dimensión 1. Como esta variables tiene dos categorías: si – no. Es muy probable que las mismas estén cerca del cero de la dimensión 2, estando una a la derecha y otra a la izquierda, cerca del eje de la dimensión 1.

Con respecto a la variables "sexo", la vemos ligada a la dimensión 2. Como esta variable tiene dos categorías, seguramente hallaremos una arriba y otra abajo próximas al eje de la dimensión 2.

Las otras variables están igualmente relacionadas a ambas dimensiones. Las categorías de IMC las encontraremos en los cuatro semiplanos, pero más dispersas que las categorías correspondientes a edad.

Compruebe esto en el gráfico de las variables.

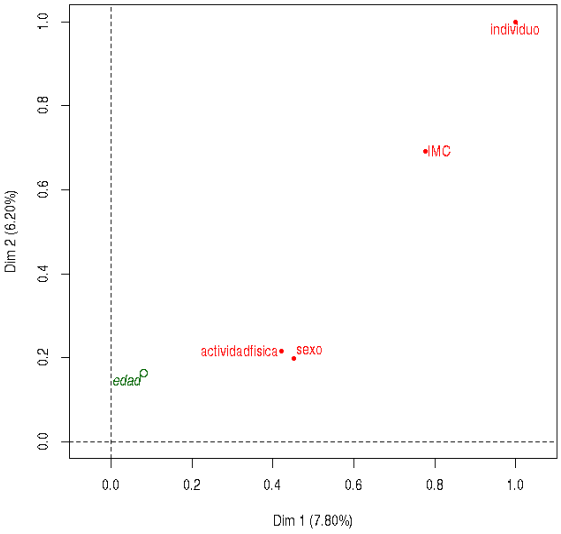


Analicemos el uso de variables cualitativas suplementarias. Supongamos que la edad (A o AM) la tomamos como suplementaria.

y realicemos el análisis con todas las variables.

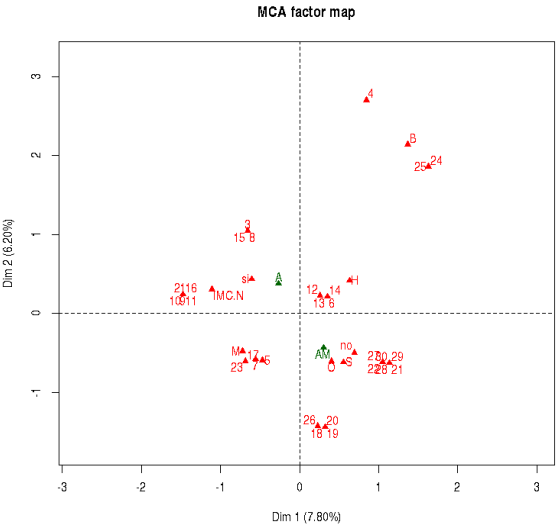
> mcatablaR471<-MCA(tablaR471[,c(1:5)],quali.sup=5)

veamos el gráfico de las variables



al tomar la edad como suplementaria, vemos que la actividad física está más relacionada al sexo. Dicho de otra manera independientemente de la edad considerada parece haber una asociación entre actividad física y sexo.

Si vemos la posición de las variables en el gráfico de los individuos, vemos una asociación entre adulto mayor (AM), actividad física (no) y obesidad (O) y sobrepeso (S).



ahora ejecutemos el análisis pero no permitamos la construcción automática de graficos

> mcatablaR471<-MCA(tablaR471[,c(2:5)],graph=FALSE)

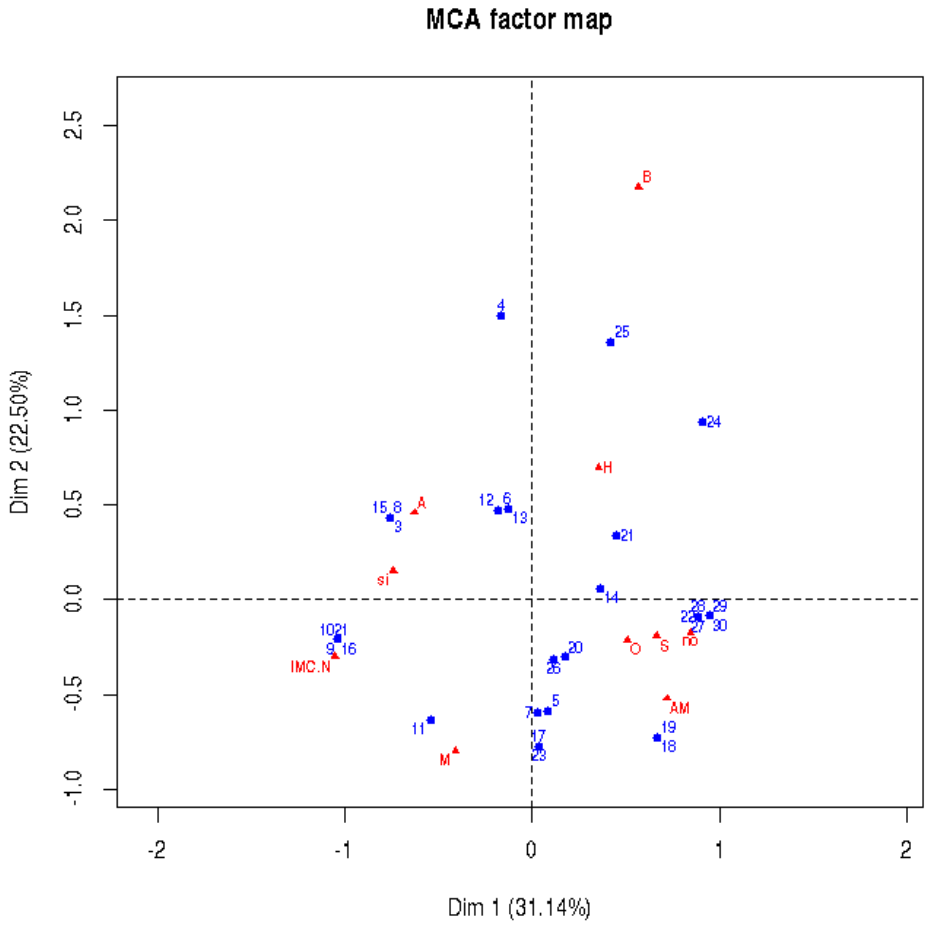
Ahora graficaremos:

1- las categorías

2- los individuos

3- reduciremos el tamaño de los puntos y rótulos

> plot.MCA(mcatablaR471,choix="ind",cex=0.7) #ind selecciona individuos y categorías



Ahora graficaremos:

1- solo los individuos

2- reduciremos el tamaño de los puntos y rótulos

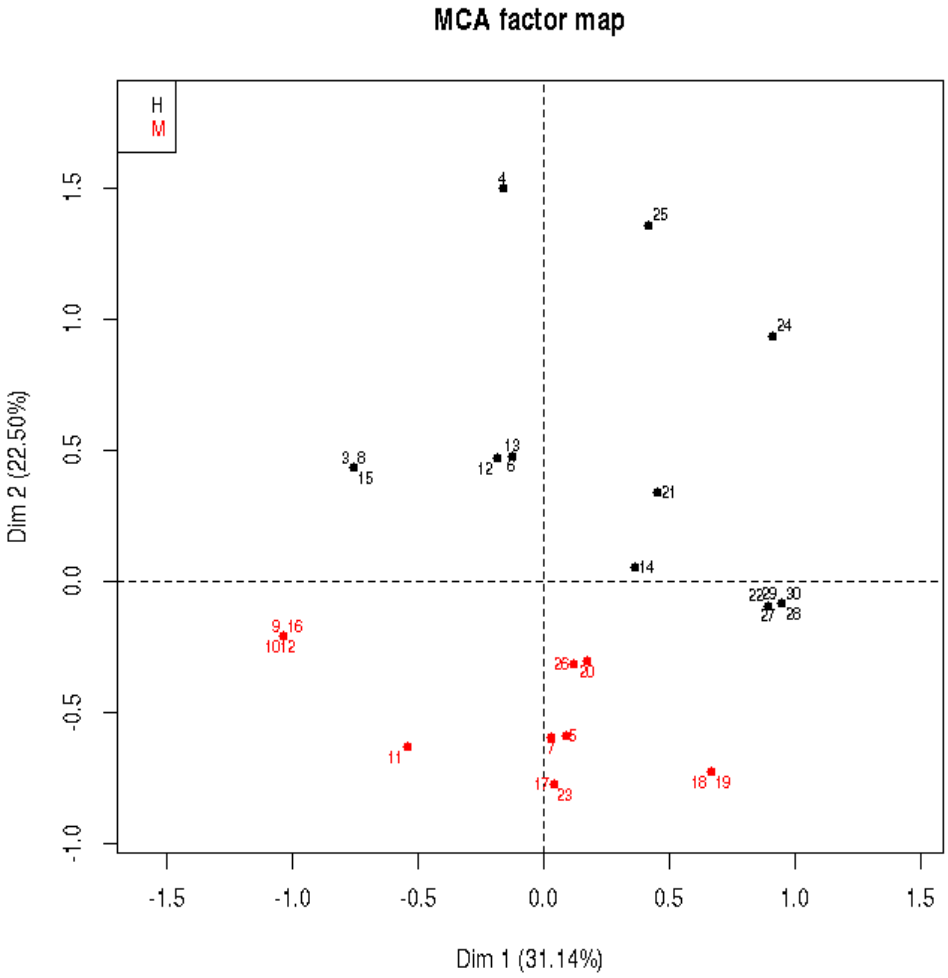
3- resaltaremos en diferente color los individuos de la categoría sexo

>plot.MCA(mcatablaR471,choix="ind",invisible="var",habillage=3,cex=0.7)

recuerde que como MCA lo aplicamos de la columna 2 a 5, la columna 3 es sexo. Compruébelo

> names(datosmca)

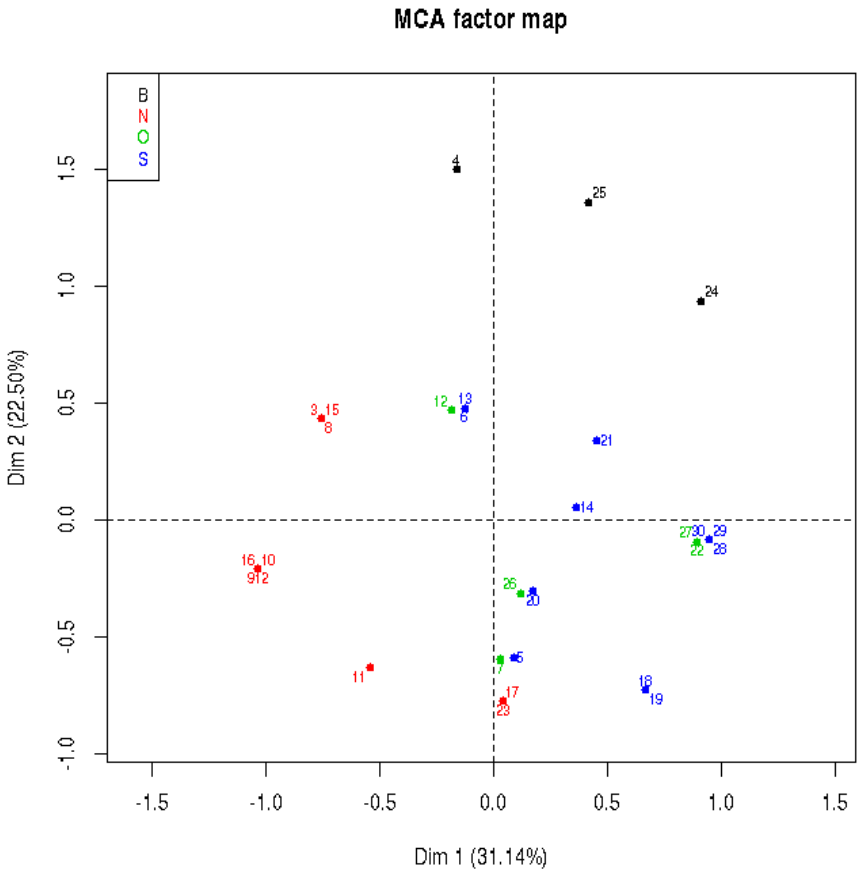
[1] "individuo" "actividadfisica" "IMC" "sexo"
[5] "edad"



Vemos la mujeres ubicadas hacia la zona de actividad física, mientras que los hombres hacia la contraria. Por supuesto hay superposición

Pruebe ahora distinguir individuos por la categoría IMC

> plot.MCA(mcatablaR471,choix="ind",invisible="var",habillage=2,cex=0.7)



Podemos comprobar la distribución de los individuos acorde a las categorías, existiendo una clara separación en N y B, mientras que se observa separación de N y B pero superposición entre O y S

7.1.1. Construcción de elipses de confianza

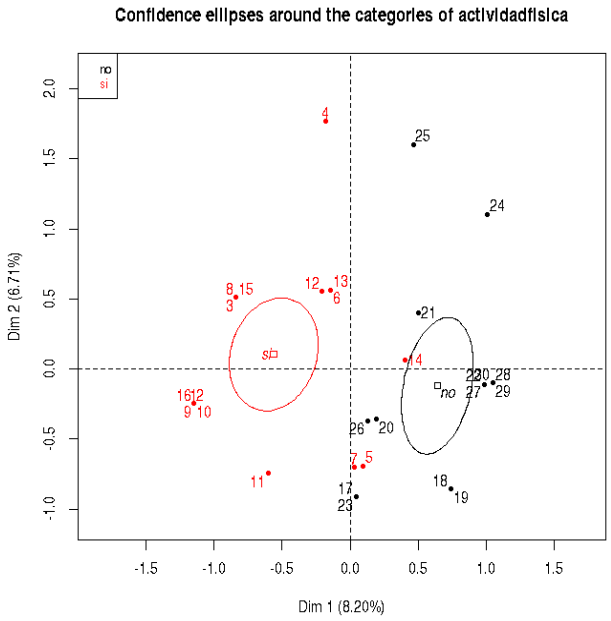
Las elipses de confianza nos indicarán con más claridad si los niveles de un factor o de todos influyen en la distribución de los individuos. Construimos un objeto con el análisis incluyendo las columas 2 a 5.

```
> mcatablaR471<-MCA(tablaR471[,c(2:5)],graph=FALSE)
```

Analicemos que ocurre con actividad física, mostrando las elipses de confianza, con un alfa 0,05.

```
> plotellipses(mcatablaR471,keepvar=1,level=0.95)
```

obtenemos la siguiente gráfica

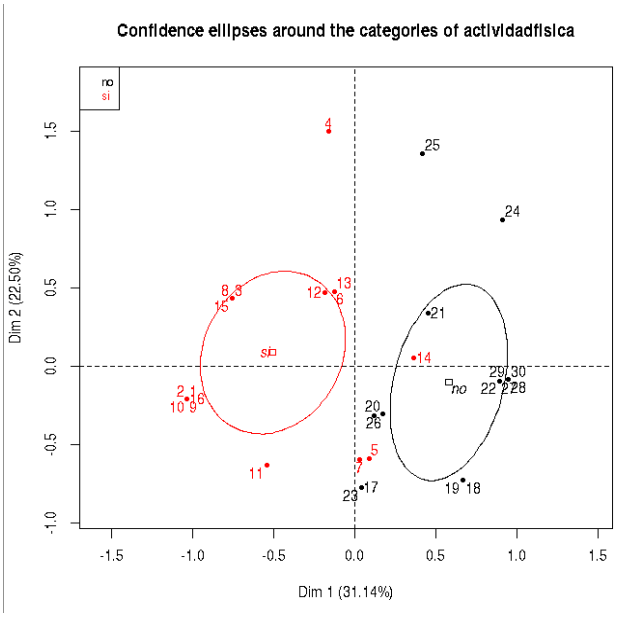


que nos estaría indicando que los niveles (si-no) de actividad física separan a los individuos,

Si probamos disminuir alfa por ejemplo a 0,001

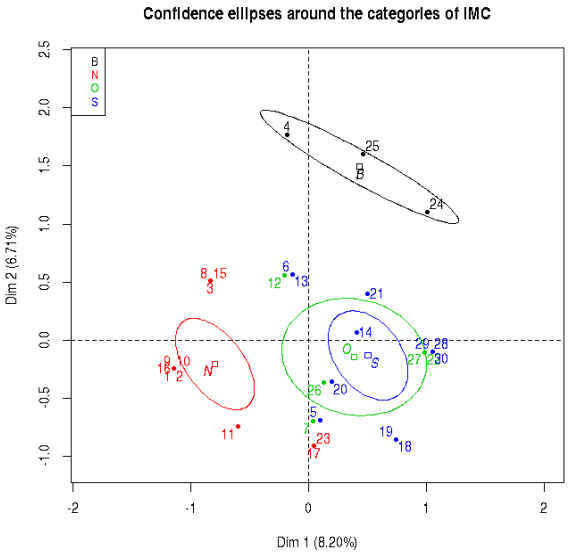
> plotellipses(mcatablaR471,keepvar=1,level=0.999)

vemos que las elipses tiende a acercarse. Cuando las elipses se tocan podríamos decir que los niveles del factor analizado no tienen efecto sobre la distribución de los individuos.



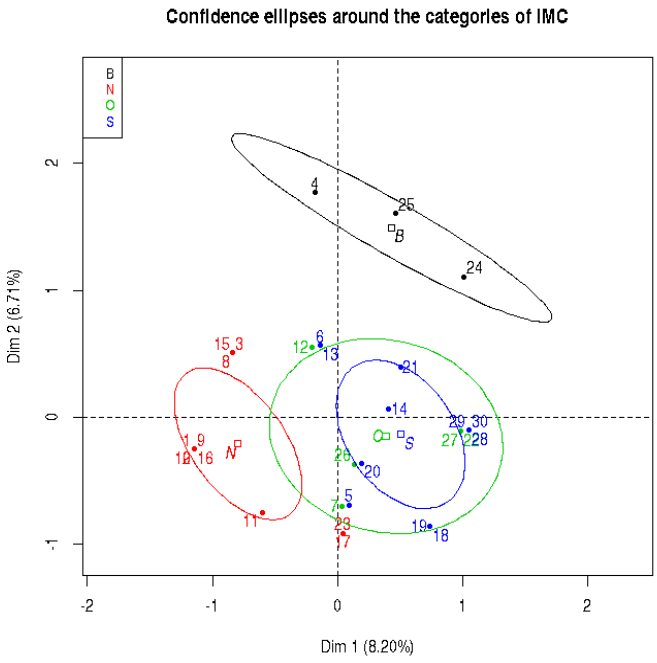
Analicemos los niveles de la variables IMC

> plotellipses(mcatablaR471,keepvar=2,level=0.95)



Nos estaría indicando una diferencia entre individuos N, S, O y B, pero que S y O no serían muy diferentes. Si aumentamos el nivel de significación por ejemplo a 0,999

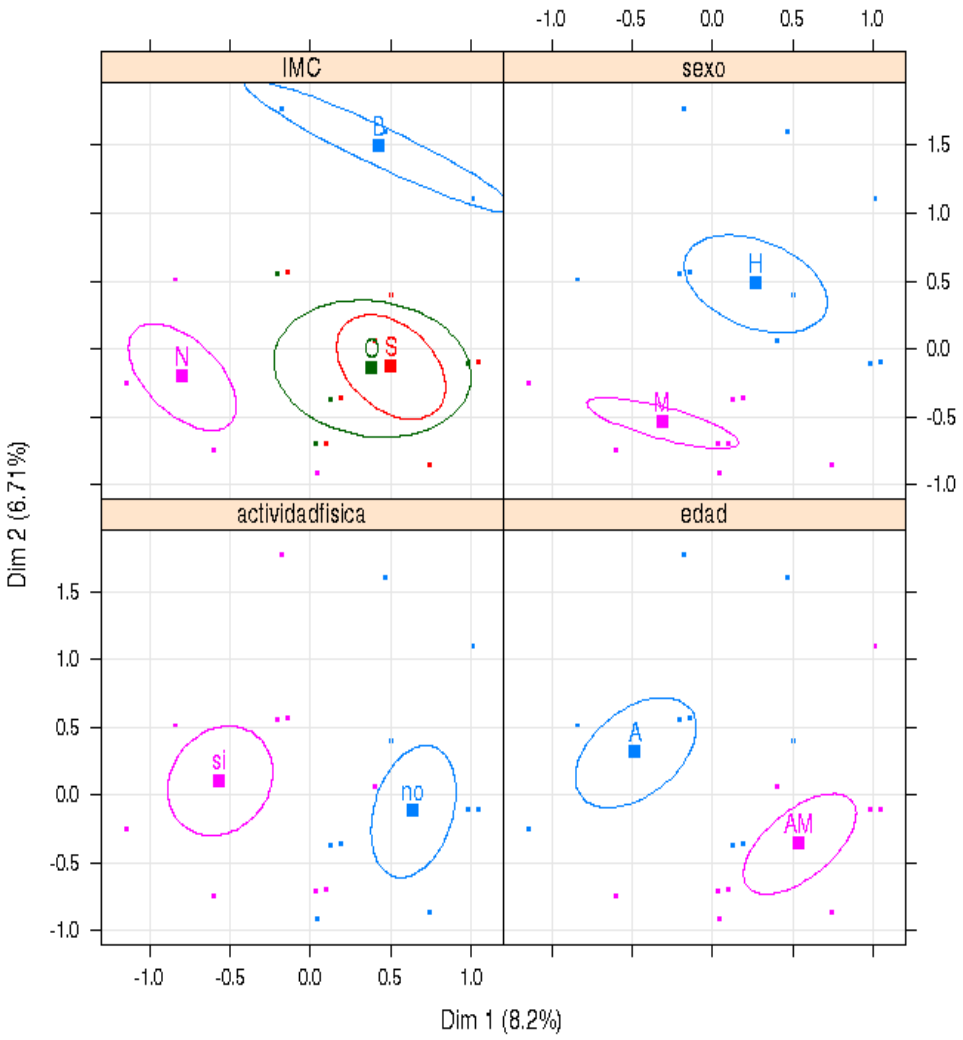
> > plotellipses(mcatablaR471,keepvar=2,level=0.999)



Vemos que ya no podríamos afirmar lo mismo para N y O aunque si entre N y S o N y B.

Podemos ver esto para todas las variables cualitativas juntas, que en nuestro caso son 4, desde la columna 2 a la 5.

>
plotellipses(mcatablaR471,keepvar="all",level=0.95,magnify=1,cex=1.5,type="p",keepnames=T)



Pruebe ahora algunos códigos extra de plotellipses

>
plotellipses(mcatablaR471,keepvar="all",level=0.95,magnify=1,cex=1.5,type="p",keepnames=T)

level puede variar (0,1)

magnify=[1,∞)

cex=(0,∞)

type= p o g

keepnames= T – F

7.2. Clasificación jerárquica de los componentes principales

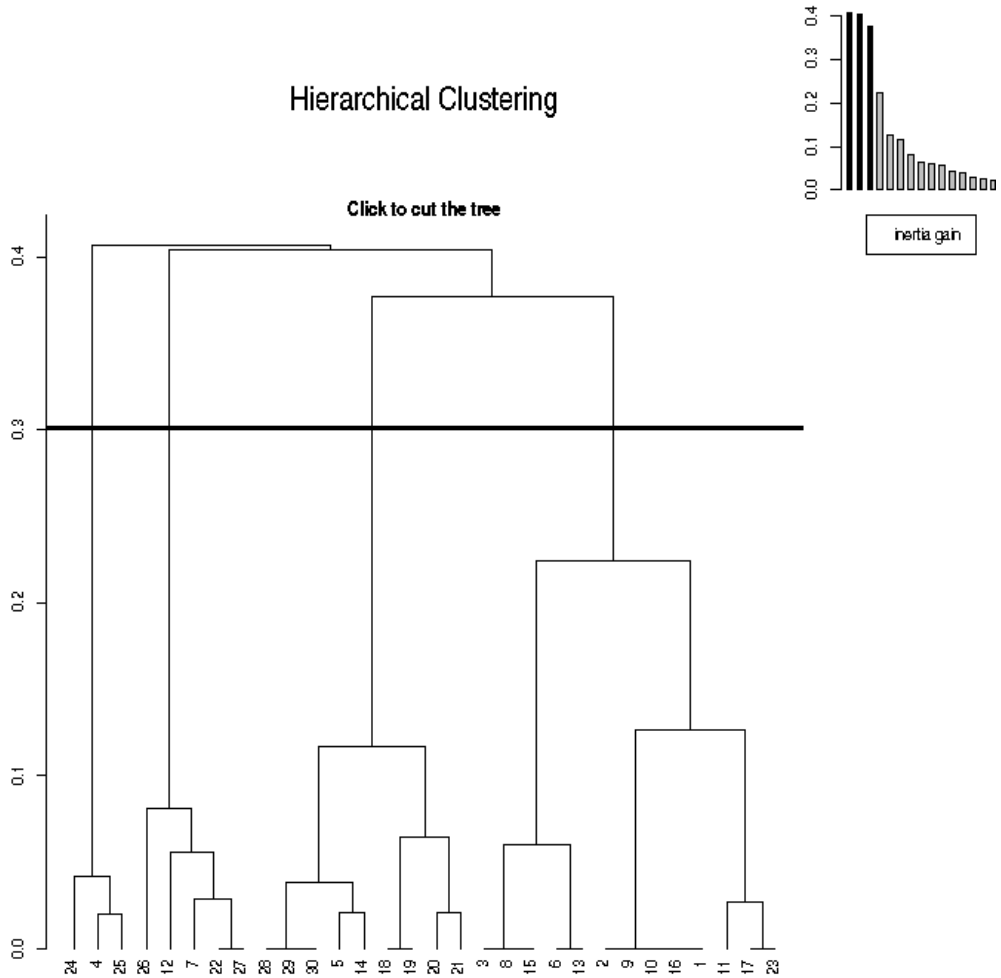
Este análisis pretende agrupar individuos de una muestra en función de múltiples variables cualitativas, de manera que se formen grupos en base a los niveles de cada factor.

Este estudio realizado sobre el análisis MCA realizado permite agrupar a los individuos por sus similitudes formando grupos más o menos homogéneos según el nivel de corte que realicemos

Cuando ejecutamos el comando siguiente

> hcpctablaR471<-HCPC(mcatablaR471)

obtendremos varias gráficas pero las que más nos interesa es la siguiente



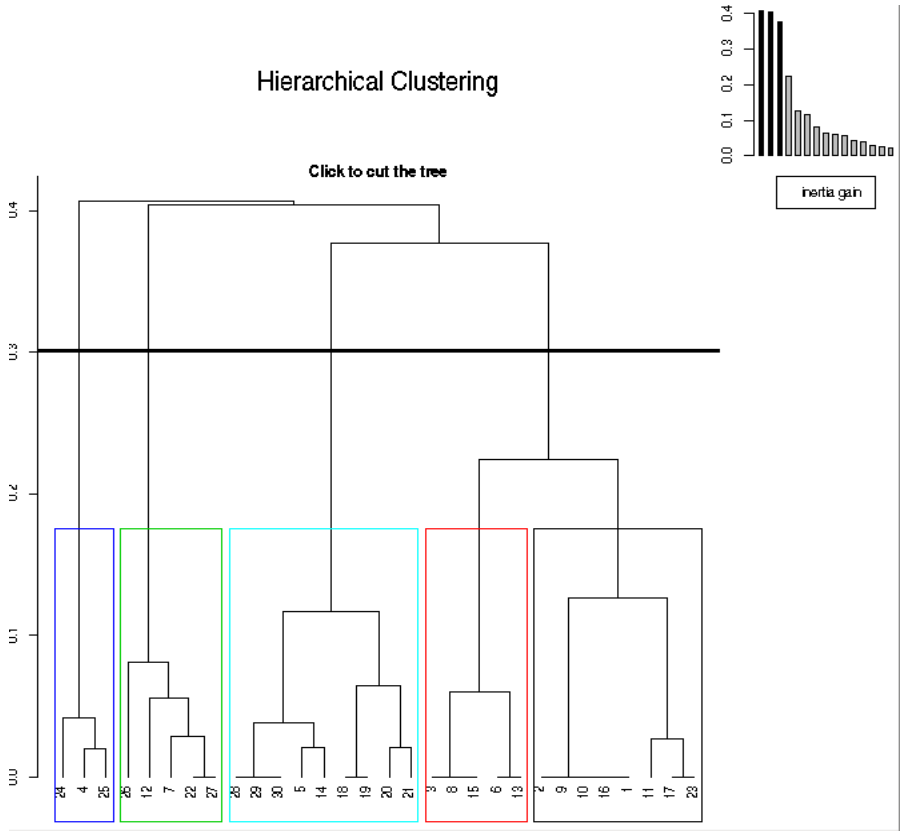
Cuando se ejecuta la función

> hcpctablaR471<-HCPC(mcatablaR471)

vimos que en la gráfica aparece un rótulo: "Click to cut the tree" y el cursor se coloca en cruz, haciendo click en el gráfico cortamos el gráficos y haremos clusters más o menos densos. Cuanto menos individuos sean más parecidos serán en los factores de las variables y viceversa si contienen más individuos, pero serán diferentes los grupos con alguna tendencia de agrupación.

Luego del "click" aparecerán tres gráficas

Veamos la primera, tendiendo en cuenta que el click lo realizamos a la altura de los rectángulos coloreados. Es decir el software nos hizo 5 clusters.



Como interpretamos esta gráfica?

En el eje horizontal están los números de nuestros individuos. Analicemos los individuo 22 y 27, vemos que están unidos por una linea horizontal, esto nos indica que comparten todas las categorías de las variables estudiadas.

Seleccionemos esos individuos

tablaR471[c(22,27),]

	individuo	actividadfisica	IMC	sexo	edad

22 22 no O H AM
27 27 no O H AM
Comprobado!

Ahora los individuos 22, 27 y 7 ya no tendrán todas las categorías iguales pero si muchas similitudes

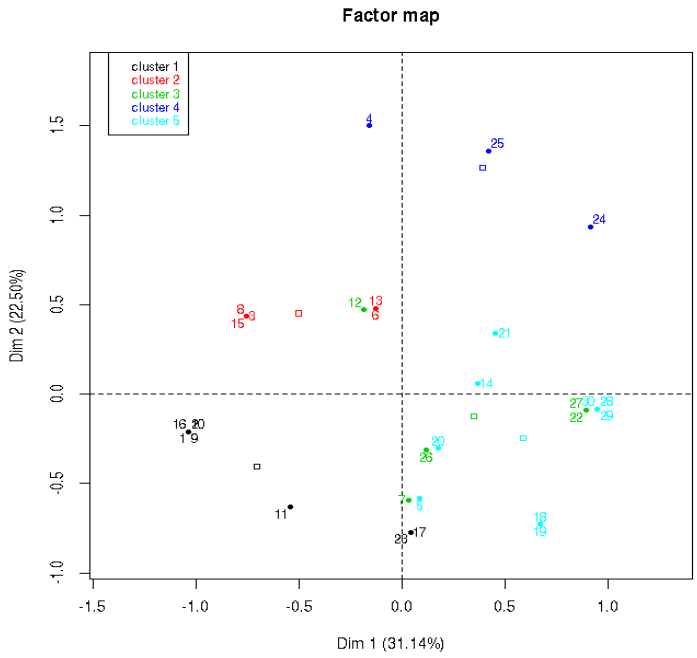
```
> tablaR471[c(7,22,27),]
   individuo actividadfisica IMC sexo edad
7          7              si   O    M   AM
22        22              no   O    H   AM
27        27              no   O    H   AM
```

todos son obesos y AM aunque no tienen el mismo sexo y la actividad física.

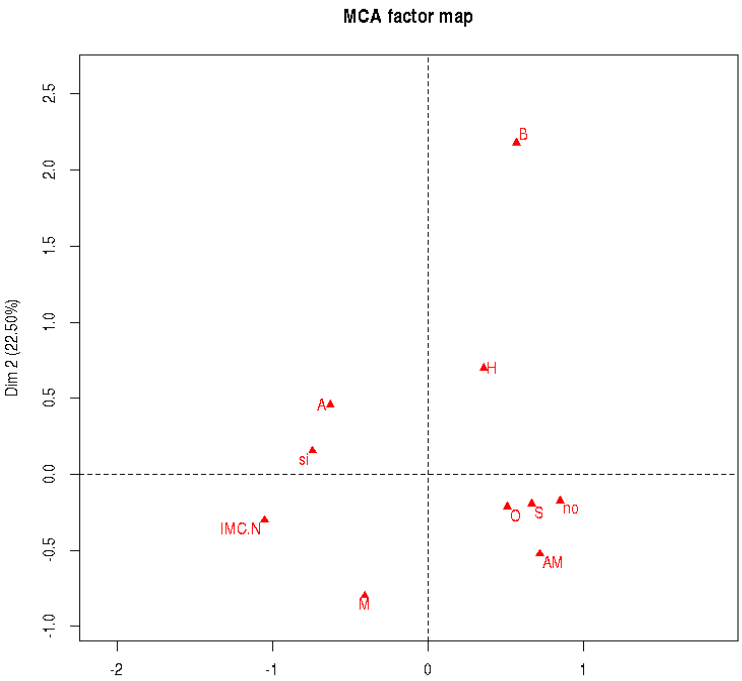
También se puede ver en otra gráfica de individuos, donde los colores son los mismos clusters y quizás podría ayudarnos para ver las características de los clusters formados, la gráfica de los factores.

Tomemos el cluster de la derecha que tiene los individuos 2, 9 , 10, 16, 1, 11,17, 23. ES indudable por lo explicado que en este cluster hay básicamente tres grupos homogéneos: uno de 6 individuos, otro de dos y uno que es un solo individuo. Veamos en la otra gráfica donde se hallan, estarán en color negro.

Los hallamos abajo y a la izquierda.



En función de la gráfica de los factores,



es probable que abunden en este cluster las mujeres adultas que hacen actividad física y tengan IMC N. Comprobemoslo!!

buscamos los individuos

```
> datosmca[c(2,9,10,16,1,11,17,23),]
    individuo  actividadfisica  IMC sexo edad
2         2              si   N       M    A
9         9              si   N       M    A
10       10              si   N       M    A
16       16              si   N       M    A
1         1              si   N       M    A
11       11              si   N       M   AM
17       17              no   N       M   AM
23       23              no   N       M   AM
```

Confirmado,

M: 8/8

actividad física si: 6/8

IMC N: 8/8

edad A: 5/8

7.3. Sumary del objeto hcpctablaR471

En el objeto generado con la función HCPC() tenemos abundante información.

Al llamar el objeto tendremos diferente información

> hcpctablaR471

> hcpctablaR471$data.clust

es un data frame con los datos de nuestros individuos, en los que se especifica el cluster en que quedó cada uno. Dándonos la misma información de la gráfica obtenida.

$data.clust

	actividadfisica	IMC	sexo	edad	clust
1	si	N	M	A	1
2	si	N	M	A	1
3	si	N	H	A	2
4	si	B	H	A	4
5	si	S	M	AM	5
6	si	S	H	A	2
7	si	O	M	AM	3
8	si	N	H	A	2
9	si	N	M	A	1
10	si	N	M	A	1
11	si	N	M	AM	1
12	si	O	H	A	3
13	si	S	H	A	2
14	si	S	H	AM	5
15	si	N	H	A	2
16	si	N	M	A	1
17	no	N	M	AM	1
18	no	S	M	AM	5
19	no	S	M	AM	5
20	no	S	M	A	5
21	no	S	H	A	5
22	no	O	H	AM	3
23	no	N	M	AM	1
24	no	B	H	AM	4
25	no	B	H	A	4
26	no	O	M	A	3
27	no	O	H	AM	3
28	no	S	H	AM	5
29	no	S	H	AM	5
30	no	S	H	AM	5

Si deseamos obtener los individuos de un cluster en particular, por ejemplo el 1 podemos aplicar el siguiente código

> hcpctablaR471$data.clust[hcpctablaR471$data.clust$clust==1,]

	actividadfisica	IMC	sexo	edad	clust
1	si	N	M	A	1
2	si	N	M	A	1
9	si	N	M	A	1
10	si	N	M	A	1
11	si	N	M	AM	1
16	si	N	M	A	1
17	no	N	M	AM	1
23	no	N	M	AM	1

Si desea conocer el número de individuos en un cluster, por ejemplo en el cluster 2

```
> nrow(hcpctablaR471$data.clust[hcpctablaR471$data.clust$clust==2,])
[1] 5
```

ordenar los individuos del data.frame segun cluster creciente

```
hcpctablaR471$data.clust[order(hcpctablaR471$data.clust[,5]),]
   actividadfisica IMC sexo edad clust
1               si   N    M    A     1
2               si   N    M    A     1
3               si   N    H    A     1
8               si   N    H    A     1
9               si   N    M    A     1
10              si   N    M    A     1
11              si   N    M   AM     1
15              si   N    H    A     1
16              si   N    M    A     1
17              no   N    M   AM     1
23              no   N    M   AM     1
7               si   O    M   AM     2
12              si   O    H    A     2
22              no   O    H   AM     2
26              no   O    M    A     2
27              no   O    H   AM     2
4               si   B    H    A     3
24              no   B    H   AM     3
25              no   B    H    A     3
5               si   S    M   AM     4
6               si   S    H    A     4
13              si   S    H    A     4
14              si   S    H   AM     4
18              no   S    M   AM     4
19              no   S    M   AM     4
20              no   S    M    A     4
21              no   S    H    A     4
28              no   S    H   AM     4
29              no   S    H   AM     4
30              no   S    H   AM     4
```

8. Clase 4.8

Vídeo: https://youtu.be/wESrjOFLiDA
tablas: http://hdl.handle.net/2133/15488

8.1. Introducción al análisis ROC

El análisis ROC es un tipo de análisis que es de utilidad para determinar el valor de corte de una variable que tiene utilidad diagnóstica. Por ejemplo, es conocido que valores de glucemia en ayunas superiores a 2 g/l son diagnóstico de diabetes mellitus. La pregunta es ¿Cómo se halla dicho valor?. El análisis ROC da esta información.

Básicamente se debe tener un diagnóstico de una situación con un método validad o bien por la sumatoria de valores de variables relacionadas y una variable a la que le queremos asignar la utilidad de poder diagnosticar en base a su valor la situación de cada unidad experimental.

Pongamos un ejemplo sencillo. Supongamos que tenemos animales de experimentación que por el examen clínico y bioquímico los clasificamos como normales y anormales. Por otra parte en estos animales medimos una variable, la cual creemos que puede tener valor diagnóstico. ¿Qué significa que tenga valor diagnóstico? Esto significa que en base a dicha variable podamos decidir si una rata es normal o anormal, con la sola observación de esa variable. El análisis ROC nos permite determinar el valor de la variable por encima o debajo del cual la unidad experimental tiene uno u otro diagnóstico. A ese valor lo llamamos valor de corte o threshold.

Veamos un ejemplo concreto en primer lugar para comprender la metodología e interpretación

supongamos que tenemos 20 unidades experimentales que hemos clasificados como sanos (0) y enfermos (1) según un criterio previo preestablecido y por otra parte medimos una variable cuantitativa "var1" en cada unidad. La variable var1 es la que utilizaremos como diagnóstico. Estos datos los tenemos en la tablaR481

```
> tablaR481<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

la variable condición debe estar como factor

```
> tablaR481$condicion<-as.factor(tablaR481$condicion)
```

```
> summary(tablaR481)
condicion      var1
 0:10       Min.   : 2.00
 1:10       1st Qu.: 4.75
            Median : 5.50
            Mean   : 7.00
            3rd Qu.:10.00
            Max.   :15.00
```

Para comprender los términos que utilizaremos en este análisis supondremos que el valor de corte de la variable var1 toma el valor 7, de tal manera que si var1 es menor o igual a 7 la unidad experimental es sana (0) y si fuera mayor que 7 enfermo (1). En base a este corte podemos clasificar a los individuos

hacemos esta selección con el siguiente código

```
> table(tablaR481$var1<=7,tablaR481$condicion)
             0  1
  FALSE      0  7
  TRUE      10  3
```

La tabla nos está clasificando a los individuos como sanos o enfermos por dos criterios distintos. Las columnas por un criterio de selección ya establecido (sano=0 o enfermo=1) y las filas por el valor de la variable var1. La columna FALSE serían aquellos individuos que tiene valor de la variable mayor que 7 (es decir unidades experimentales que según nuestra variable son enfermos) y la columna TRUE tiene a los individuos con var1<7 es decir que serían sanos por el valor de la variable.

Para comprender mejor la tabla cambiemos FALSE por (+) es decir individuos que dieron enfermos por el valor de la variable. Cambiemos TRUE por (-) es decir individuos que resultaron sanos si los clasificamos por nuestra variable.

	0	1
+	0	7
-	10	3

Definiremos algunos términos. Recordemos entonces

1: indica enfermo y 0: sano según un criterio establecido

+: indica enfermo y -: sano según el valor de corte de nuestra variable var1.

Definimos entonces

verdaderos positivos: tp: individuos que son enfermos por el criterio de clasificación y por el valor de la variable. Es decir que tiene el valor "enfermo o 1" de la condición y +. En nuestra tabla vemos que son 7 individuos.

verdaderos negativos: tn: individuos que son "sanos o 0" por criterio y por el valor de la variable son "-". En nuestro caso son 10 individuos.

falsos positivos: fp: individuos que son sanos por el criterio, pero enfermos por el valor de nuestra variable. En nuestro caso son 0 individuos

falsos negativos: fn: individuos que son enfermos por el criterio, pero sanos por el valor de la variable. En nuestro caso son 3 individuos.

podemos representar esto en la tabla

	0	1
+	fp=0	tp= 7
-	tn=10	fn= 3

Vemos en la tabla que el total de enfermos según el criterio de clasificación utilizado es de 10 individuos (tp + fn), mientras que el total de sanos por el criterio utilizado es de 10 individuos (fp + tn). Por otra parte el total de enfermos detectados por nuestra variable var1 (+) es de 7 individuos (fp+tp), mientras que el total de sanos detectados por la clasificación en base al valor de corte de la variable es de 13 (tn + fn)

Definiremos otros términos de importancia en este análisis.

Sensibilidad: tp/(numero total de enfermos) = 7/10 = 0,7

La sensibilidad variará entre 0 y 1. Si vale cero quiere decir que nuestra variable no detecta ningún enfermo. En cambio si vale 1 quiere decir que los individuos que son detectados como positivos por nuestra variable son todos los enfermos.

Especificidad: tn/(número total de sanos) = 10/10 = 1

La especificidad varía entre 0 y 1. Si la especificidad es 0 indica que ningún individuo sano fue detectado como tal por nuestra variable y contrariamente todos los sanos nos dieron positivos. Es decir la prueba no es para nada específica porque da positiva cuando no está la enfermedad. En el punto contrario si todos los sanos fueran detectados como sanos, la especificidad sería 1. Es decir ningún sano fue detectado como enfermo.

valor predictivo de test positivo: positive predictive value: ppv: tp/total de + = tp/(fp+tp) 7/7 =1

el valor predictivo de test positivo nos indica que probabilidad tenemos que un individuo sea enfermo cuando la prueba diagnóstica así lo indica. En nuestro caso tenemos 7 individuos que dieron una prueba positiva de enfermedad y todos ellos a su vez tenían la enfermedad. Es decir que no hubo individuos que hubieran dada positiva la prueba estando sanos. En este caso podemos decir que el valor predictivo del test positivo es 1. Es decir si nos da positiva nuestra prueba diagnóstica, existe una probabilidad del 100% que esté enfermo.

valor predictivo de test negativo: negative predictive value: npv: tn/total de negativos= tn/(tn + fn) = 10/(10+3) = 10/13 = 0,77

El valor predictivo de test negativo nos indica la probabilidad de que un individuo sea sano cuando nuestra prueba dio negativa. En nuestro caso dieron negativos 13 individuos, pero solo 10 eran sanos. Es decir que el valor predictivo de test negativo es del 0,77, es decir que si la prueba nos da negativa la probabilidad que el individuo sea realmente sano es del 77%.

Es evidente que nuestro valor de corte será mejor y nuestra variable predictiva de la enfermedad será mejor cuando la S, E, VPP y VPN sean más cercanos a 1, así como los fp y fn sean más cercanos a 0.

El análisis ROC nos permite hallar el valor de corte de la variable que nos da los mejores valores de las variables mencionadas.

8.2. Análisis ROC con R

En el análisis de los datos utilizaremos el paquete pROC del software R.

instalar la biblioteca pROC y luego cargar la biblioteca para utilizarla en el espacio de trabajo.

```
> library(pROC)
```

es importante que la variable predictiva sea continua mientras que el criterio de clasificación sea un factor.

Hagamos los cálculos con los datos de la tablaR481, que vemos a continuación

```
> tablaR481
   condicion var1
1          0    2
2          0    2
3          0    4
4          0    5
5          0    4
6          0    5
7          0    5
8          0    6
9          0    7
10         0    2
11         1    5
12         1    5
```

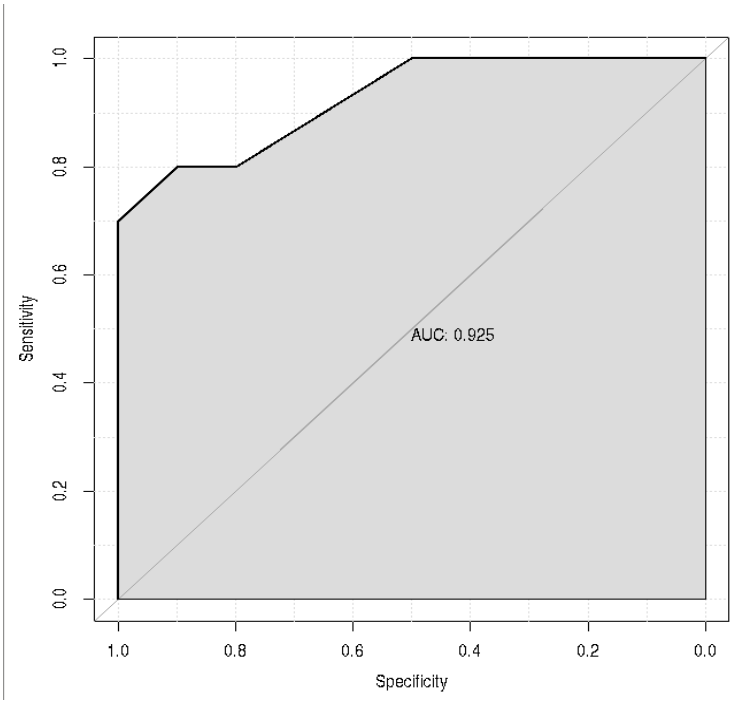
13	1	8
14	1	10
15	1	12
16	1	11
17	1	15
18	1	15
19	1	10
20	1	7

```
> summary(tablaR481)
condicion      var1
 0:10          Min.   : 2.00
 1:10          1st Qu.: 4.75
               Median : 5.50
               Mean   : 7.00
               3rd Qu.:10.00
               Max.   :15.00
```

El primer análisis recomendado es hallar la curva ROC y el área bajo la curva

roc(response = tablaR481$condicion, predictor = tablaR481$var1,smooth = FALSE, plot = TRUE, grid = TRUE, print.auc = TRUE,auc.polygon = TRUE)

nos arroja la siguiente curva y análisis



Data: tablaR481$var1 in 10 controls (tablaR481$condicion 0) < 10 cases (tablaR481$condicion 1).

Area under the curve: 0.925

En este punto es importante el valor de "Area under the curve", que llamaremos de acá en adelante AUC.

Si el valor de AUC es mayor a 0,5 podemos decir que la prueba diagnóstica es de utilidad y tiene futuro como prueba diagnóstica. Dicho en otras palabras podemos decir que con el valor de var1 podemos clasificar a las unidades experimentales como sanas (0) o enfermas (1). Cuanto más cercano a 1 sea el valor de AUC, mejor la prueba diagnóstica. En general si AUC >0,8 la prueba tiene buen poder diagnóstico. En este caso es 0.925

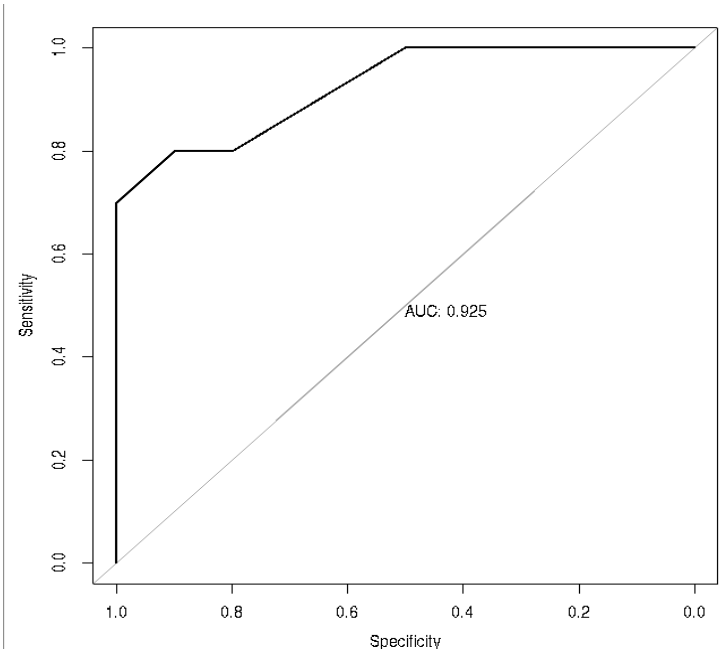
Sin duda el valor elevado del AUC justifica el análisis completo, el cual mostramos a continuación

Creamos primero un objeto roc para alojar nuestro análisis

> roctablaR481<-roc(response = tablaR481$condicion, predictor = tablaR481$var1)

pedimos la curva ROC, incluyendo el valor del AUC.

> plot(roctablaR481,print.auc=TRUE)



podemos pedir al objeto creado que solo nos de el valor del AUC

> auc(roctablaR481)

Area under the curve: 0.925

el intervalo de confianza del AUC

> ci(roctablaR481)

95% CI: 0.8181-1 (DeLong)

Si el CI incluye al 1 indica que es una buena prueba diagnóstica.

Pedimos los valores de corte de la variable var1, sensibilidad, especificidad, fn, fp, vp,tp,tn. Estos son los datos de más relevancia.

```
>
coords(roctablaR481,"best",ret=c("threshold","sensitivity","specificity","tp","tn","fn","fp","ppv","
npv"),as.list=F,drop=F)
```

```
             best
threshold   6.5000000
sensitivity 0.8000000
specificity 0.9000000
tp          8.0000000
tn          9.0000000
fn          2.0000000
fp          1.0000000
ppv         0.8888889
npv         0.8181818
```

Analicemos los resultados.

threshold: es el valor de corte de la variable var1. Nos indica que si tomamos como corte var1=6,5. Aquellos individuos con valor mayor serán enfermos según nuestra variables y los con var1<6,5 los consideraremos sanos.

Sensitivity: 0,8. Nos indica que nuestro ensayo detecta como enfermos el 80 % del total de los realmente enfermos según el criterio que habíamos utilizado para clasificarlos. Esto nos indica que habrá falsos negativos. Es claro en este ejemplo entender esto ya que son 10 los individuos clasificados como enfermos por el criterio. Como la Sensibilidad de nuestro ensayo es del 80 %, habría un 20% que no es detectado. Eso justifica que el valor fn sea de 2, el 20 % de los enfermos no fueron detectados por la medida de var1.

La especificidad fue de 0,9. Nos indica que nuestro ensayo detecta como sanos el 90% de los individuos clasificados como sanos según nuestro criterio original. Es decir que hay un 10% de sanos que serán clasificados como enfermos según var1. Como en nuestro ejemplo son 10 los sanos segun el criterio, si la especificidad fue del 90% hay un 10 % de sanos que estarán entre los enfermos, es decir serán falsos positivos. Esto justifica el valor 1 para la variable fp.

El valor de ppv= 0,89, indica una predictibilidad de prueba positiva del 89 %. Es decir que si con el valor de var1 decimos que un individuo es enfermos tenemos un 89% de probabilidad de estar en lo cierto.

El valor de npv= 0,82, indica que la predictibilidad de prueba negativa es del 82%, es decir que si por el valor de var1 clasificamos a un paciente como sano, tenemos una probabilidad del 82 % de estar diciendo lo correcto.

Puede comprobar los valores haciendo una tabla utilizando el valor de corte de la variable var1

```
> table(tablaR481$var1>6.5,tablaR481$condicion)
          0  1
 FALSE  9  2
 TRUE   1  8
```

cambiemos y modifiquemos la tabla

	0	1
-	tn= 9	fn= 2
+	fp= 1	tp= 8

Algunas consideraciones importantes.

El análisis ROC me permite elegir el valor de corte que da mejor valor de los parámetros. Sin embargo en algunos casos ese valor puede y debe ser modificado en función de la Sensibilidad y especificidad.

Supongamos que la variable que estamos desarrollando para medir un determinado estado, será utilizada para diagnosticar una enfermedad que, en el caso de no detectarla las consecuencias fueran dramáticas para la vida del paciente. En tal caso es preferible un ensayo con Sensibilidad lo mas cercano a 1 aunque perdamos especificidad. Es decir desearíamos tener un valor de corte que nos permita detectar todos los enfermos aunque algunos sanos nos den positivos.

En este caso es conveniente ejecutar el analisis roc con cambiando el argumento "best" por "all"

>
coords(roctablaR481,"all",ret=c("threshold","sensitivity","specificity","tp","tn","fn","fp","ppv","npv"),as.list=F,drop=F)

	all	all	all	all	all	all	all
threshold	-Inf	3.0000000	4.5000000	5.5	6.5000000	7.5000000	9.0000000
sensitivity	1.0	1.0000000	1.0000000	0.8	0.8000000	0.7000000	0.6000000
specificity	0.0	0.3000000	0.5000000	0.8	0.9000000	1.0000000	1.0000000
tp	10.0	10.0000000	10.0000000	8.0	8.0000000	7.0000000	6.0000000
tn	0.0	3.0000000	5.0000000	8.0	9.0000000	10.0000000	10.0000000
fn	0.0	0.0000000	0.0000000	2.0	2.0000000	3.0000000	4.0000000
fp	10.0	7.0000000	5.0000000	2.0	1.0000000	0.0000000	0.0000000
ppv	0.5	0.5882353	0.6666667	0.8	0.8888889	1.0000000	1.0000000
npv	NaN	1.0000000	1.0000000	0.8	0.8181818	0.7692308	0.7142857

	all	all	all	all
threshold	10.500	11.5000000	13.5000000	Inf
sensitivity	0.400	0.3000000	0.2000000	0.0
specificity	1.000	1.0000000	1.0000000	1.0
tp	4.000	3.0000000	2.0000000	0.0
tn	10.000	10.0000000	10.0000000	10.0
fn	6.000	7.0000000	8.0000000	10.0
fp	0.000	0.0000000	0.0000000	0.0
ppv	1.000	1.0000000	1.0000000	NaN
npv	0.625	0.5882353	0.5555556	0.5

en la tabla anterior vemos que con un valor de corte de 4.5 tenemos una sensibilidad de 1, lo que indica que no tendremos falsos negativos. Es decir todos los enfermos serán detectados. Sin embargo habrá algunos sanos que serán detectados como enfermos. Otra prueba diagnóstica adicional puede confirmar el estado de estos individuos.

Es conveniente adoptar este criterio cuando se dispone de dos técnicas. Una de ellas sencilla y económica y otra costosa. Con la prueba sencilla se puede hacer un primer análisis (screening) con el cual sabremos que los que nos dieron sanos, realmente lo son. Los enfermos detectados con esta

primer prueba sabemos que pueden algunos no serlo y los sometemos a otro ensayo. Esto redundará en muchos casos en ahorro de recursos.

8.3. Comparación de dos pruebas diagnósticas

Supongamos que tenemos dos variables: var1 y var2 que puedo utilizar como prueba diagnóstica y deseo comparar cual es mejor. Haré el ensayo ROC para ambas como hemos aprendido en el inciso anterior. Introduzcamos los datos de la tablaR482 y pasemos a factor la condición

```
> tablaR482<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

```
> tablaR482$condicion<-as.factor(tablaR482$condicion)
```

comprobamos los datos de nuestra tabla. La variable diagnóstico de referencia (en este caso condicion) debe ser categórica y var1 y var2 continuas

```
> summary(tablaR482)
 condicion      var1            var2      
 0:10       Min.   : 2.00   Min.   :15.0  
 1:10       1st Qu.: 4.75   1st Qu.:18.0  
            Median : 5.50   Median :22.5  
            Mean   : 7.00   Mean   :22.2  
            3rd Qu.:10.00   3rd Qu.:25.5  
            Max.   :15.00   Max.   :29.0
```

hacemos los análisis roc para cada variable, asignándolo a sendos objetos

```
> roctablaR482var1<-roc(response = tablaR482$condicion, predictor = tablaR482$var1)
```

```
> roctablaR482var2<-roc(response = tablaR482$condicion, predictor = tablaR482$var2)
```

Calculamos las áreas bajo la curva para ambas variables.

```
> auc(roctablaR482var1)
```

```
Area under the curve: 0.925
```

```
> auc(roctablaR482var2)
```

```
Area under the curve: 0.875
```

y los intervalos de confianza

```
> ci(roctablaR482var1)
```

```
95% CI: 0.8181-1 (DeLong)
```

```
> ci(roctablaR482var2)
```

```
95% CI: 0.6982-1 (DeLong)
```

vemos que los intervalos de confianza de superponen, por lo que ambas pruebas no difieren significativamente. Sin embargo si tenemos que inclinarnos por una lo haremos por la var1, debido a que el área bajo la curva es más cercana a 1. Por supuesto en la elección de var1 o var2 pueden jugar también motivos económicos o de facilidad de la prueba para obtener los valores de var1 o var2.

Podemos comparar ambas pruebas con la función roc.test() que compara ambas curvas ROC

```
> roc.test(roctablaR482var1,roctablaR482var2)
        DeLong's test for two correlated ROC curves
data:  roctablaR482var1 and roctablaR482var2
```

Z = 0.49055, p-value = 0.6237
alternative hypothesis: true difference in AUC is not equal to 0
sample estimates:
AUC of roc1 AUC of roc2
0.925 0.875

El valor de p-value nos indica que no hay diferencias entre ambas pruebas diagnósticas aun cuando var1 tiene una AUC mayor. La elección de una u otra variable para el diagnóstico dependerá de la sencillez, costo, impacto de los valores fp y fn del ensayo.

9. Clase 4.9

Vídeo: https://youtu.be/dCp3SGoMy88
Tablas: http://hdl.handle.net/2133/15489

9.1. Curvas de sobrevida – análisis de supervivencia

El análisis de supervivencia o curvas de sobrevida se refieren a un grupo de técnicas para el modelado de sucesos que ocurren a lo largo del tiempo. Si al momento del análisis no ha ocurrido un evento se dice que el dato debe ser censurado.

Básicamente, en el modelo más sencillo la curva de supervivencia nos sirve para evaluar la cantidad de sobrevivientes de una población o la cantidad de muertos a un dado tiempo. Sin embargo el evento: muerte – sobrevida puede ser aplicado a otros eventos que pueden ocurrir con el transcurso del tiempo. Por ejemplo lo podemos aplicar al estudio de los alumnos que se reciben a lo largo del tiempo de una dada cohorte. El evento sería: se recibió – no se recibió. Otros ejemplos podrían ser la cantidad de huevos de un ave que eclosionan a lo largo del tiempo, la cantidad de trabajos científicos publicados sobre un tema, etc. Otro ejemplo de aplicación sería la caída de frutas de una planta a lo largo del tiempo. La fruta puede permanecer en la planta al momento de análisis (en este caso sería un dato censurado) o bien haber caído.

En general le colocamos 0 si el dato es censurado y 1 si no lo es, aunque puede ser realizado con funciones especificas del paquete.

Analizaremos para comenzar una curva de supervivencia de animales machos (m) y hembras (h) a lo largo del tiempo en presencia de una deficiencia vitamínica en la dieta. La tablaR491 de la planilla de cálculo tablaR4-9.ods/xls, contiene los datos que utilizaremos.

Utilizaremos el paquete survival que puede bajar de cualquier repositorio de R.

```
> library(survival)
```

Introducimos los datos

```
> tablaR491<-read.table("clipboard",header=TRUE,dec=",",sep="\t",encoding="latin1")
```

```
> summary(tablaR491)
 sexo    diasdevida       estado
 h:46    Min.   :  5.0    Min.   :0.00
 m:54    1st Qu.: 79.5    1st Qu.:1.00
         Median :150.0    Median :1.00
         Mean   :155.0    Mean   :0.82
         3rd Qu.:221.5    3rd Qu.:1.00
         Max.   :300.0    Max.   :1.00
```

estado= 0, indica que aun no ha muerto

estado=1, se produjo la muerte en el tiempo indicado en diasdevida

creamos un objeto "surv" con la función Surv()

```
> survtablaR491 <- Surv(tablaR491$diasdevida,tablaR491$estado==1)
```

este objeto como vemos tiene los 100 individuos en que se detalla el valor de diasdevida con + si es un dato censurado, es decir que aun no se ha producido la muerte.

```
> survtablaR491
  [1]  50  300+ 150  300+ 180  120  110   80   70   10  300+ 160  110   78   43
 [16]  15  180  300+ 300+  25   58   69   95   97  150  170  180  190  200  201
```

```
[31] 202  300+ 300+ 150  300+  75   76   89   98  150  153  180  300+ 300+ 280
[46] 300+  25  300+ 120  300+ 120  110   90   58   70    5  300+ 150  100   60
[61]  42   10  150  290  300+  26   50   61   90   90  110  160  110  150  150
[76] 151  160  300+ 290  150  300+  65   60   89   89  150  153  180  290  280
[91] 280  300+  65   60   89   89  150  153  180  290
```

realizaremos una curva de sobrevida general de los animales, sin discriminar por sexo. Para ello utilizamos el objeto survtablaR491 generado anteriormente y la función survfit()

```
> curva1tablaR491 <- survfit(survtablaR491~1, data=tablaR491)
```

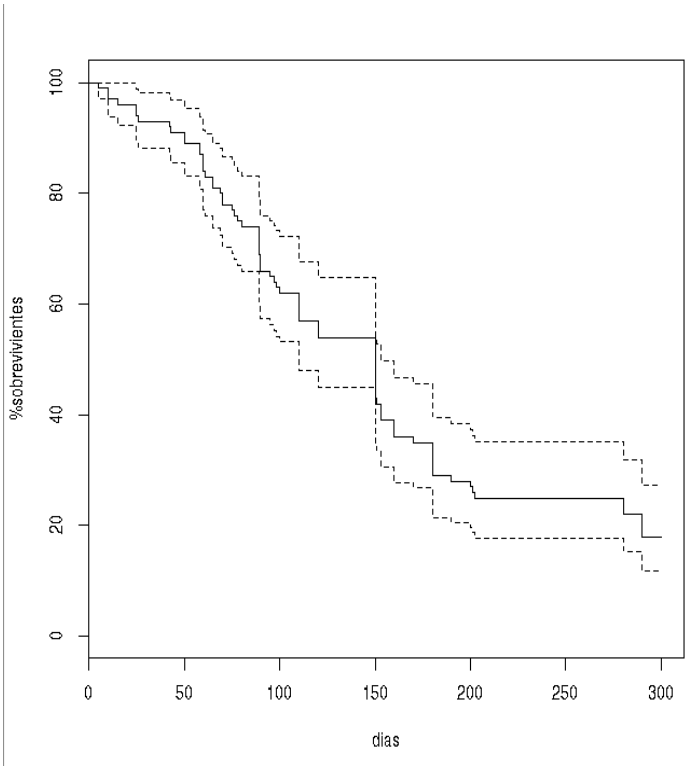
En este objeto hallamos ordenado por tiempo el número de individuos en riesgo de sufrir el evento, en ese caso la muerte y el número de muertes producidos hasta ese tiempo. Además tenemos el error estándar y el intervalo de confianza.

```
>  summary(curva1tablaR491)
Call: survfit(formula = survtablaR491 ~ 1, data = tablaR491)

 time n.risk n.event survival          std.err    lower 95% CI  upper 95% CI
   5    100       1                0.99 0.00995        0.971        1.000
  10     99       2                0.97 0.01706        0.937        1.000
  15     97       1                0.96 0.01960        0.922        0.999
  25     96       2    0.94    0.02375        0.895        0.988
  26     94       1                0.93 0.02551        0.881        0.981
  42     93       1                0.92 0.02713        0.868        0.975
  43     92       1                0.91 0.02862        0.856        0.968
  50     91       2                0.89 0.03129        0.831        0.953
 .........
 160     39       3                0.36 0.04800        0.277        0.468
 170     36       1                0.35 0.04770        0.268        0.457
 180     35       6                0.29 0.04538        0.213        0.394
 190     29       1                0.28 0.04490        0.204        0.383
 200     28       1        0.27    0.04440        0.196        0.373
 201     27       1                0.26 0.04386        0.187        0.362
 202     26       1                0.25 0.04330        0.178        0.351
 280     25       3                0.22 0.04142        0.152        0.318
 290     22       4                0.18 0.03842        0.118        0.273
```

Realizamos entonces una gráfica de la curva de sobrevida.

```
> plot(curva1tablaR491,xlab="dias",ylab="%sobrevivientes",yscale=100)
```



La gráfica anterior nos muestra el procentaje de supervivientes a lo largo del tiempo en líneas continuas con el intervalo de confianza en línea de puntos.

9.2. Análisis de supervivencia clasificando por un factor

Realicemos el análisis de supervivencia por sexo. Nuestros individuos son de sexo h o m.

```
> curva2tablaR491 <- survfit(survtablaR491~sexo, data=tablaR491)

> summary(curva2tablaR491)
Call: survfit(formula = survtablaR491 ~ sexo, data = tablaR491)

                sexo=h
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
   10     46       1    0.978  0.0215        0.937        1.000
   15     45       1    0.957  0.0301        0.899        1.000
   25     44       1    0.935  0.0364        0.866        1.000
   43     43       1    0.913  0.0415        0.835        0.998
   50     42       1    0.891  0.0459        0.806        0.986
   58     41       1    0.870  0.0497        0.777        0.973
   69     40       1    0.848  0.0530        0.750        0.958
   70     39       1    0.826  0.0559        0.724        0.943
   75     38       1    0.804  0.0585        0.698        0.928
   76     37       1    0.783  0.0608        0.672        0.911
   78     36       1    0.761  0.0629        0.647        0.895
   80     35       1    0.739  0.0647        0.623        0.878
   89     34       1    0.717  0.0664        0.598        0.860
```

95	33	1	0.696	0.0678	0.575	0.842
97	32	1	0.674	0.0691	0.551	0.824
98	31	1	0.652	0.0702	0.528	0.805
110	30	2	0.609	0.0720	0.483	0.767
120	28	1	0.587	0.0726	0.461	0.748
150	27	4	0.500	0.0737	0.375	0.668
153	23	1	0.478	0.0737	0.354	0.647
160	22	1	0.457	0.0734	0.333	0.626
170	21	1	0.435	0.0731	0.313	0.604
180	20	4	0.348	0.0702	0.234	0.517
190	16	1	0.326	0.0691	0.215	0.494
200	15	1	0.304	0.0678	0.197	0.471
201	14	1	0.283	0.0664	0.178	0.448
202	13	1	0.261	0.0647	0.160	0.424
280	12	1	0.239	0.0629	0.143	0.400

sexo=m

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
5	54	1	0.981	0.0183	0.9462	1.000
10	53	1	0.963	0.0257	0.9139	1.000
25	52	1	0.944	0.0312	0.8853	1.000
26	51	1	0.926	0.0356	0.8586	0.998
42	50	1	0.907	0.0394	0.8333	0.988
50	49	1	0.889	0.0428	0.8089	0.977
58	48	1	0.870	0.0457	0.7852	0.965
60	47	3	0.815	0.0529	0.7175	0.925
61	44	1	0.796	0.0548	0.6958	0.911
65	43	2	0.759	0.0582	0.6534	0.882
70	41	1	0.741	0.0596	0.6326	0.867
89	40	4	0.667	0.0642	0.5521	0.805
90	36	3	0.611	0.0663	0.4940	0.756
100	33	1	0.593	0.0669	0.4750	0.739
110	32	3	0.537	0.0679	0.4192	0.688
120	29	2	0.500	0.0680	0.3829	0.653
150	27	7	0.370	0.0657	0.2616	0.524
151	20	1	0.352	0.0650	0.2450	0.505
153	19	2	0.315	0.0632	0.2124	0.467
160	17	2	0.278	0.0610	0.1807	0.427
180	15	2	0.241	0.0582	0.1499	0.387
280	13	2	0.204	0.0548	0.1202	0.345
290	11	4	0.130	0.0457	0.0649	0.259

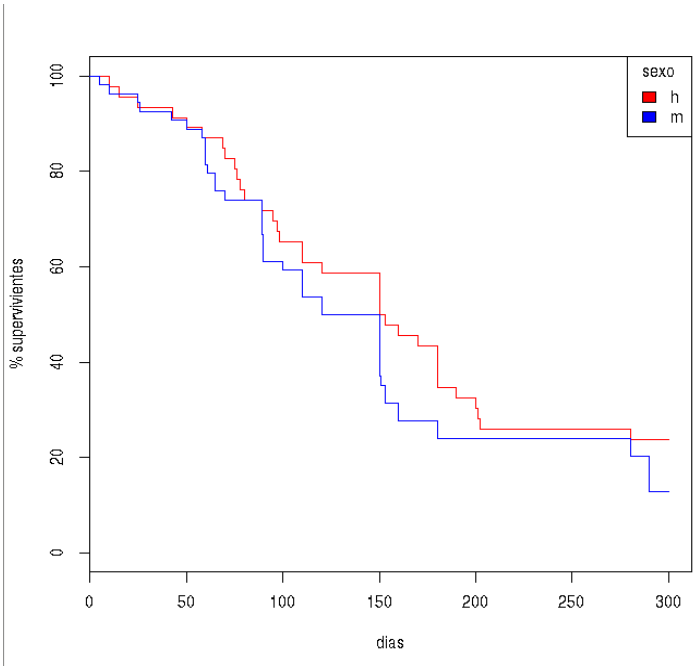
graficamos las curvas de sobrevida de h y m, asignando color rojo a "m" y azul a "h"

```
> plot(curva2tablaR491,xlab="dias",ylab="% supervivientes",yscale=100,col=c("red","blue"))
```

agregamos la leyenda en el vértice superior derecho de la gráfica

```
> legend("topright",title="sexo",c("h","m"),fill=c("red","blue"))
```

indicamos c("h","m") en dicho orden porque en ese orden lo tomó el programa. Ver tabla anterior



9.3. Comparación de dos o más curvas de sobrevida

Vemos que el porcentaje de muertes es mayor en machos que hembras. Podemos comparar si realmente hay diferencias entre machos y hembras. Para ello utilizamos la función survdiff().

> survdiff(Surv(diasdevida, estado==1) ~ sexo,data=tablaR491,rho=0)
rho=0 indica que se utiliza el log-rank test o Mantel Haenszel test.

Call:
survdiff(formula = Surv(diasdevida, estado == 1) ~ sexo, data = tablaR491)

	N	Observed	Expected	(O-E)^2/E	(O-E)^2/V
sexo=h	46	35	40.7	0.799	1.7
sexo=m	54	47	41.3	0.788	1.7

Chisq= 1.7 on 1 degrees of freedom, p= 0.192

El valor de p-value>0,05 nos indica que no hay diferencia entre las sobrevidas de h y m.

9.4. Regresión de Poisson

La regresión de Poisson es un modelo donde la variable dependiente es un conteo y las variables predictivas son variables continuas o categóricas.

Ejemplos donde se puede aplicar este modelo se describen a continuación:

1- Se pretende relacionar el número de goles convertidos por un futbolista a lo largo de un campeonato en función del consumo de oxígeno medido en sus pruebas médicas, el porcentaje de grasa muscular y el número de horas de entrenamiento.

variable dependiente: número de goles

variables independientes: consumo de oxígeno, % grasa corporal, número de horas de entrenamiento.

2- En un test de aprendizaje con animales se toman datos de la cantidad de veces que durante el ensayo, el animal se dirige y explora un dado objeto. Se estima que esta cantidad de veces depende de los mg de tiamina que consume por día, de las horas de entrenamiento que recibe el animal en el experimento y de la edad.

variable dependiente: cantidad de exploraciones

variables independientes: mg de tiamina, horas de entrenamiento, edad.

3- La cantidad de trabéculas óseas con terminales libres es una medida de la discontinuidad del tejido óseo y de la calidad del mismo. Las variables explicativas parecen ser los niveles de parathormona sérica, el porcentaje de calcio en la dieta, la edad y el peso corporal.

variable dependiente: cantidad de terminales libres

variables independientes: niveles de parathormona, % de calcio, edad, peso corporal.

4- El número de hijos de una pareja se halla bajo estudio y se sospecha que puede explicarse por variables como el coeficiente intelectual medido por un test estándar, el nivel socioeconómico medido a través de reglas establecidas por un departamento de estadística y algunas variables biológicas como son el recuento de espermatozoides en un espermograma y los niveles de estrógenos de la mujer

variable dependiente: número de hijos

variables independientes: coeficiente intelectual, nivel socioeconómico, número de espermatozoides y niveles de estrógenos.

Veremos la aplicación de este modelo utilizando datos de la tablaR493. En un grupo de alumnos seguidos durante un semestre se registró el número de evaluaciones aprobadas, variable que se sospecha influenciada por otras variables determinantes como son la cantidad de horas de estudio, el porcentaje de asistencia y el dinero invertido semanalmente específicamente en el estudio. La cantidad de exámenes aprobados se obtuvo de las planillas de los profesores, así como el porcentaje de asistencia. La cantidad e horas de estudio se le solicitó a los alumnos a través de una encuesta, así como una estimación del dinero invertido en temas relacionados estrictamente con el estudio. Introduzca la tablaR493 de planilla de cálculo tablaR4-9.ods/xls. en su espacio de trabajo

```
> tablaR493<-read.table('clipboard',header=TRUE,dec=',',sep='\t',encoding='latin1')
```

comprobamos el ingreso

```
> head(tablaR493)
  individuo horasestudio asistencia examenesaprobado dineroinvertido
1         1           8         90                8              528
2         2           9         98                9              769
3         3           4         56                7              519
4         4           5         56                5              773
5         5           6         65                5              717
6         6           7         66                5              233
```

Podemos hacer una primera aproximación al problema a través de un análisis bivariado, correlacionando la cantidad de examenesaprobados con cada una de las variables. Este análisis no es correcto ya que la correlación es un análisis que se realiza con datos cuantitativos medidos con un cierto error en cada variable. En este caso el número de examenes es cuantitativa pero no

continua y se mide prácticamente sin error.
Veamos la correlación con el número de horas de estudio

```
> cor.test(tablaR493$horasestudio,tablaR493$examenesaprobado)

        Pearson's product-moment correlation

data:  tablaR493$horasestudio and tablaR493$examenesaprobado
t = 3.6341, df = 90, p-value = 0.0004638
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1649933 0.5241371
sample estimates:
    cor
0.357722
```

Si bien el p-value es significativo, el valor de R2= 0.127965, nos estaría indicando que la variación en el número de exámenes aprobados solo sería explicado en un 12 % por las horas de estudio. Sin embargo en este análisis la correlación hallada es positiva y significativa. A más horas de estudio, mejor rendimiento.

Correlación con el porcentaje de asistencia

```
> cor.test(tablaR493$asistencia,tablaR493$examenesaprobado)
        Pearson's product-moment correlation
data:  tablaR493$asistencia and tablaR493$examenesaprobado
t = 6.5134, df = 90, p-value = 4.108e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4085555 0.6907494
sample estimates:
     cor
0.5660096
```

Un valor de p-value mucho menor y un coeficiente de correlación mayor nos están indicando que la cantidad de exámenes es más dependiente de la asistencia que del estudio.

Correlación con el dinero invertido

```
> cor.test(tablaR493$dineroinvertido,tablaR493$examenesaprobado)

        Pearson's product-moment correlation

data:  tablaR493$dineroinvertido and tablaR493$examenesaprobado
t = -0.86918, df = 90, p-value = 0.3871
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2906239  0.1157429
sample estimates:
       cor
-0.09123743
```

Si bien el valor de R es negativo que nos estaría induciendo a pensar que a menos dinero invertido, mayor cantidad de exámenes aprobados, el valor de p-value nos indica que el dinero invertido en el estudio no parece ser determinante.
Estos análisis los realizamos sin poner la vista en forma multivariada. Aplicaremos la regresión de

Poisson con las tres variables independientes intervinientes

> poissontablaR493 <- glm(examenesaprobado ~ horasestudio + asistencia+dineroinvertido, family="poisson", data=tablaR493)

pedimos un summary() del objeto que tiene los resultados del modelo

```
> summary(poissontablaR493)
Call:
glm(formula = examenesaprobado ~ horasestudio + asistencia +
    dineroinvertido, family = "poisson", data = tablaR493)
Deviance Residuals:
       Min        1Q    Median        3Q       Max
-1.18515  -0.41930   0.03832   0.32733   1.25796

Coefficients:
                  Estimate    Std. Error   z value   Pr(>|z|)
(Intercept)      1.415e+00     1.775e-01   7.971     1.57e-15 ***
horasestudio     5.577e-03     2.578e-02   0.216     0.82874
asistencia       6.823e-03     2.576e-03   2.648     0.00809 **
dineroinvertido -5.732e-05     1.482e-04  -0.387     0.69901
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for poisson family taken to be 1)
Null deviance: 38.709  on 91  degrees of freedom
Residual deviance: 26.313  on 88  degrees of freedom
AIC: 375.73
Number of Fisher Scoring iterations: 4
```

En la tabla anterior focalizamos la atención en coefficients. Estos valores son los coeficientes del modelo planteado. Los coeficientes para horasestudio y asistencia son positivos, que nos estarían indicando que a más horas estudiadas y mayor porcentaje de asistencia, será mayor la cantidad de exámenes aprobados. El dinero invertido contrariamente tiene un coeficiente negativo. Si observamos en la misma tabla los valores de p-value de cada coeficiente, que hallamos en la columna Pr(>|z|) , vemos que el único coeficiente que tiene un valor significativamente diferente de cero son la intersección y la asistencia, mientras que horasestudio y dineroinvertido tienen valores muy superiores a 0.05.

Como este análisis nos indicaba que dineroinvertido no tiene efecto significativo, podríamos eliminarlo y realizar nuevamente el análisis en búsqueda de otro modelo. El mejor o peor ajuste lo veremos en el valor de AIC, cuanto menor es este número mejor es el ajuste del modelo

> poissontablaR493 <- glm(examenesaprobado ~ horasestudio + asistencia, family="poisson", data=tablaR493)
>

```
Call:
glm(formula = examenesaprobado ~ horasestudio + asistencia, family = "poisson",
    data = tablaR493)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.23322  -0.41755   0.01166   0.34005   1.32297
```

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.386616	0.162077	8.555	< 2e-16 ***
horasestudio	0.004649	0.025623	0.181	0.85604
asistencia	0.006921	0.002564	2.699	0.00695 **

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 38.709 on 91 degrees of freedom
Residual deviance: 26.462 on 89 degrees of freedom
AIC: 373.88

Number of Fisher Scoring iterations: 4

El valor de AIC es algo menor lo que nos indica que es un mejor modelo

Como podemos ver, las horas de estudio sigue siendo un factor que no explica la cantidad de examenes aprobados. Por lo tanto convendría eliminarlo del análisis generando un modelo sin dicho factor

> poissontablaR493 <- glm(examenesaprobado ~ asistencia, family="poisson", data=tablaR493)

pedimos un summary()

> summary(poissontablaR493)
Call:
glm(formula = examenesaprobado ~ asistencia, family = "poisson",
data = tablaR493)
Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.20498	-0.41364	-0.00184	0.33434	1.32072

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.398506	0.148113	9.442	< 2e-16 ***
asistencia	0.007193	0.002083	3.453	0.000554 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 38.709 on 91 degrees of freedom
Residual deviance: 26.495 on 90 degrees of freedom
AIC: 371.91

Number of Fisher Scoring iterations: 4

Asi nuestro modelo quedaría

numeroexamenesaprobados= 1,398506 + 0,007193*asistencia

En modo predictivo, podríamos estimar el número de exámenes que aprobaría un alumno que asista el 50% de las clases

numeroexamenesaprobados= 1,398506 + 0,007193*50 =1.76

Podemos entonces anticipar que aprobaría aproximadamente 2 exámenes.

Un detalle adicional, como en otros modelos lineales generalizados estudiados. La elección del mejor modelo podemos hacerlo basándonos en el indice AIC. Cuanto más chico este índice mejor es el modelo. Como podemos comprobar si incluimos las tres variables AIC fue 375.73, cuando dejamos dos variables, AIC=373.88 y con una sola variable, AIC= 371.91

USO DE HERRAMIENTAS INFORMÁTICAS PARA LA RECOPILACIÓN, ANÁLISIS E INTERPRETACIÓN DE DATOS DE INTERÉS EN LAS CIENCIAS BIOMÉDICAS

La investigación científica requiere de diversas herramientas y habilidades para resolver los interrogantes que se enfrentan en los proyectos de investigación.

Tan importante como las ideas gestoras de un proyecto son las herramientas que se elegirán para obtener los datos y luego analizarlos e interpretarlos.

R es un completo kit de herramientas que le permitirán recopilar la información científica surgida de las observaciones planteadas, permitiendo un ordenamiento flexible de manera que luego pueda abordarla con herramientas de análisis e interpretación.

La posibilidad de utilizar bibliotecas con las más diversas finalidades así como la posibilidad de programación y construcción de bibliotecas propias hacen de R un entorno inigualable e insuperable en el terreno de la investigación científica.

ISBN 978-987-86-1179-2