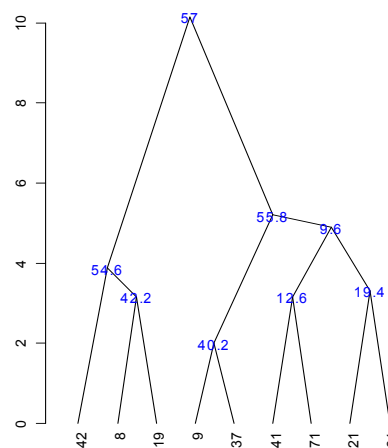
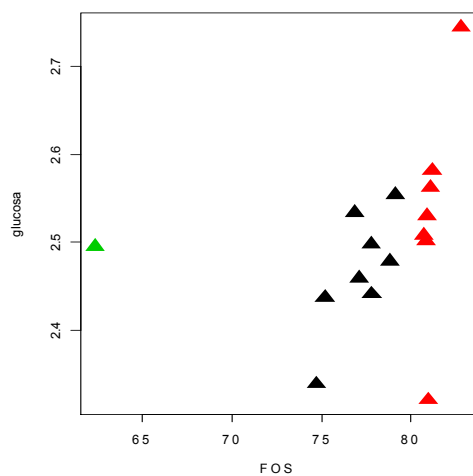
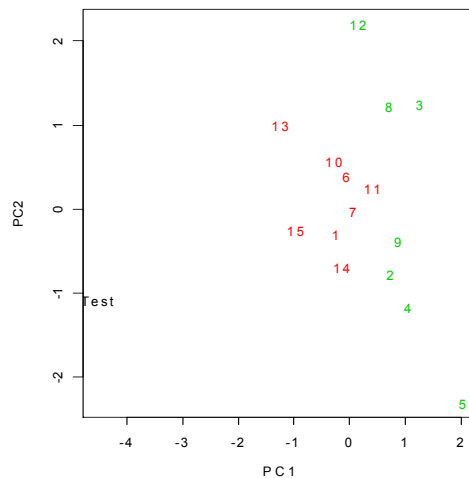
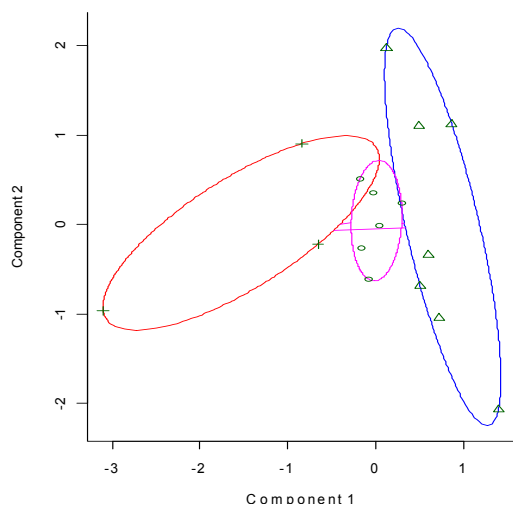


Analisis Multivarial

- Analisis multivarial. Componentes principales
- Clasificacion no supervisada: cluster analysis, dendrograma y consensus (metodos jerarquicos).
- Clasificacion supervisada. Analisis discriminante.y k vecinos mas cercanos.



Componentes principales

R dispone de funciones para hallar las componentes principales a partir de correlaciones o covariancias de las variables observadas.

Para este estudio considere el siguiente archivo: “azucares en yacon.txt”

FOS, glucosa, fructosa y sacarosa

Una variable adicional que es la identificación de cada registro (ID)

```
> azucar <- read.table("azucares en yacon.txt",header=TRUE)
> datos <- azucar[,-1]
> correl <- cor(datos)
> valores <- eigen(correl)
> valores
```

```
$values
[1] 2.14206323 1.24405952 0.59276138 0.02111586
```

```
$vectors
      [,1]      [,2]      [,3]      [,4]
[1,] 0.6688031 0.10715436 0.1680706 0.71622111
[2,] 0.3107702 -0.62950240 -0.7115513 -0.02904034
[3,] -0.1326303 -0.76885749 0.6184480 0.09375192
[4,] -0.6622186 -0.03320897 -0.2880435 0.69093745
```

Para hallar los valores correspondientes a cada componente principal, es necesario primero estandarizar los datos, en este caso “datos”. Si usamos la función estándar descrita anteriormente (Pág. 34). Las 4 columnas se pueden estandarizar

```
> a1 <- estandar(datos,1)
> a2 <- estandar(a1,2)
> a3 <- estandar(a2,3)
> a4 <- estandar(a3,4)
```

Entonces se forma una matriz A.

```
> A <- as.matrix(a4)
```

Las ponderaciones de las componentes se encuentran en el objeto valores, entonces obtenemos la matriz X:

```
> X <- valores$vectors
```

Las componentes principales se obtienen multiplicando la matriz A por X.

```
> CP1 <- A%*%X
```

Utilizando la matriz variancia-covariancia

```
> covar <- cov(datos)
> valores <- eigen(covar)
> valores

$values
[1] 31.054475402  0.400136559  0.096671548  0.007448548

$vectors
      [,1]      [,2]      [,3]      [,4]
[1,]  0.854321447  0.47752777 -0.2042005 -0.02010622
[2,]  0.005110203 -0.01540644 -0.1124739  0.99352209
[3,] -0.013639922 -0.37397769 -0.9207939 -0.10996962
[4,] -0.519540858  0.79490286 -0.3127147 -0.02040282
```

El aporte de cada componente se puede hallar según la proporción respecto al total.

En el caso de correlación:

```
> valores$value*100/sum(valores$value)
[1] 53.5515808 31.1014881 14.8190346  0.5278965
```

En covariancia:

```
> valores$value*100/sum(valores$value)
[1] 98.40216440  1.26791076  0.30632266  0.02360218
```

Y las componentes pueden ser calculadas como:

```
> X <- valores$vectors
```

Las componentes principales se obtienen multiplicando la matriz A por X.

```
> CP2 <- A%%X
```

Otra función que permite hallar las componentes principales es: `princomp()`. Si usa correlaciones el parámetro `cor=TRUE`, si usa covariancias es `cor=FALSE`.

```
> componentes<-princomp(datos, cor=TRUE)
> summary(componentes)
```

Importance of components:

	Comp.1	Comp.2	Comp.3	Comp.4
Standard deviation	1.4635789	1.1153742	0.7699100	0.145312978
Proportion of Variance	0.5355158	0.3110149	0.1481903	0.005278965
Cumulative Proportion	0.5355158	0.8465307	0.9947210	1.000000000

```
> names(componentes)
[1] "sdev"      "loadings" "center"    "scale"     "n.obs"     "scores"    "call"
```

Cada objeto proporciona información sobre las componentes principales.

La matriz de componentes esta en: `componentes$scores` que tiene la misma información obtenida en CP para correlaciones. Para obtener el mismo valor y signo, se debe particularizar cada matriz; por ejemplo si se normaliza ambas matrices CP1 y CP2 para la primera componente se tiene:

```
> CP1[,1]/CP1[1,1]
> componentes$scores[,1]/ componentes$scores[1,1]
```

Ambos vectores son iguales en signo y magnitud.

De igual forma para covariancias.

```
> componentes<-princomp(datos, cor=FALSE)
> summary(componentes)
```

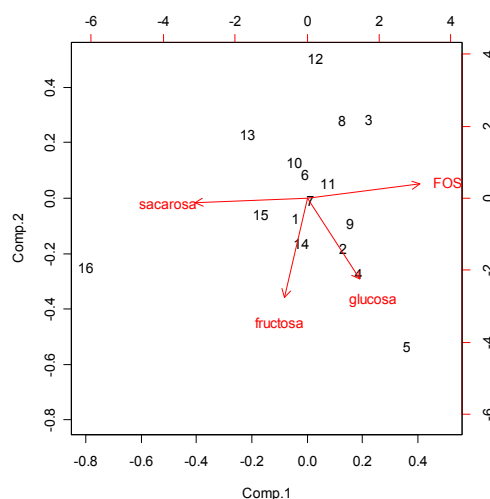
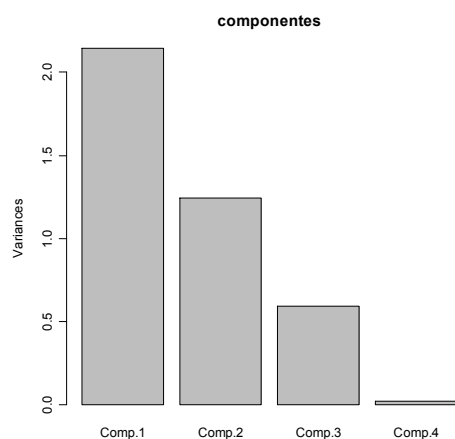
Importance of components:

	Comp.1	Comp.2	Comp.3	Comp.4
Standard deviation	5.3956993	0.61247696	0.301047465	0.0835644307
Proportion of Variance	0.9840216	0.01267911	0.003063227	0.0002360218
Cumulative Proportion	0.9840216	0.99670075	0.999763978	1.0000000000

Las proporciones acumulativas son las mismas encontradas con el procedimiento anterior.

Otra información es obtenida de:

```
> plot(componentes)
> biplot(componentes)
```



Clasificación no supervisada. Conglomerados (cluster).

Permite formar grupos similares según las medidas que uno considere para el agrupamiento.

Hay muchas funciones para ello, las principales son:

dist() para el calculo de la distancia entre individuos
hclust() procedimiento para los agrupamientos.
cutree() para hacer los cortes del dendrograma.

y otras funciones de la librería cluster y agricolae como se indica

```
plclust()  
rect.hclust()  
daisy()  
pam()  
clusplot()  
as.dendrogram()  
cut()  
consensus()  
hcut()
```

Subir los paquetes

```
> library(cluster)  
> library(agricolae)
```

Análisis cluster

Métodos para calcular distancia.

“euclidian”.- la distancia entre los vectores x e y
“maximum”.- la distancia Máxima entre dos componentes x,y
“manhattan”: la distancia Absoluta entre los dos vectores
“canberra”.- $\text{sum}(|x_i - y_i| / |x_i + y_i|)$.
“minkowki”.- la norma de p componentes
“binary”.- distancia como una proporción de frecuencias.

“**binary**”.- La distancia es determinada como una proporción entre el total de celdas de presencia del marcador (una sola vez) entre el total de celdas con presencia del marcador.

Ejemplo: Para hallar la distancia entre A y B, se tiene 5 celdas con presencia del marcador { (1,1), (1,1), (1,0), (1,1), (1,1) } y celdas con una sola vez { (1,0) }, entonces la distancia es $1/5 = 0.2$. En otro caso, Distancia (B, D), total con presencia del maracador { (1,1), (1,0), (0,1), (1,0), (1,1) } y celdas con una sola vez { (1,0), (0,1), (1,0) }, entonces la distancia (B, D) es igual a $3/5 = 0.6$

Métodos para hallar dendrogramas.