

# Extrayendo morfometrías

Alfredo Cortell-Nicolau

Si lo reducimos a lo más básico, la morfometría geométrica es tan solo una forma más sofisticada de extraer medidas. Es la vieja pregunta de comparación de ratios entre distintos artefactos arqueológicos para saber si estas dichas diferencias en ratios pueden darnos alguna información añadida. Esto es cierto, pero solo en parte. En realidad la morfometría geométrica es algo más, ya que nos permite llegar mucho más lejos en cuanto al detalle que pretendemos analizar. Ya no son solo ratios, sino si distintas posiciones de adiciones cerámicas o de pedúnculos de puntas de flecha pueden aportarnos algo de información en términos de variabilidad cultural, geográfica o cronológica.

Para este tutorial, utilizaremos una pequeña parte de una colección de microlitos geométricos. Los microlitos geométricos son una forma específica de puntas de flecha. Aunque existen desde el Paleolítico Superior en Europa (y desde tiempos anteriores en África y Asia) cobran especial importancia con la llegada del Holoceno, con los grupos epipaleolíticos y mesolíticos, así como con los primeros grupos agrícolas. En esencia, los microlitos geométricos son pequeños artefactos líticos con forma geométrica (normalmente formas trapezoidal, triangular o de media luna, aunque en algunos casos también los hay romboidales y cuadrangulares) que se insertan de forma modular en los astiles. Su rápida producción y fácil reemplazo los convierten en un elemento clave de la cultural material a partir del Holoceno. Además, diferentes procesos de variedad regional llevan a la aparición de distintas tendencias morfométricas que ayudan en el análisis arqueológico de las particularidades culturales de los distintos grupos.

En nuestro caso, nos centraremos en comparativas de los microlitos geométricos de los últimos caza-recolectores y de los primeros agricultores de la Península Ibérica. Los geométricos aquí presentados son tan solo una submuestra de los más de 2.000 pertenecientes a la Cueva de la Cocina y a la Cova de l'Or, ambas en el este peninsular. Todos se encuentran publicados, con datos y código de libre acceso.

Lo primero que vamos a hacer es cargar los datos. Estos se encuentran en la carpeta **Datos**, y para leer la información necesaria podemos teclear lo siguiente.

```
## Load and prepare necessary data
geos <- read.csv("./Datos/geos_less.csv")
index <- geos$ID ## Este objeto nos servirá después
```

Adicionalmente, necesitaremos los paquetes **geomorph** y **Momocs**, que hemos instalado en la sesión anterior y ahora podemos cargar con el siguiente comando.

```
library(geomorph) ## para landmarks
library(Momocs) ## para outlines
```

A partir de aquí el proceso va a variar, ya que el código y funciones necesarios para hacer el análisis basado en landmarks o en outlines es diferente. La diferencia fundamental entre landmarks y outlines es que, utilizando landmarks, se exigen puntos específicos de artefactos (y que sean comunes a todos ellos) y se realiza el análisis desde ahí. En outlines, en vez de elegirse puntos específicos, se realiza un análisis sobre toda su forma. Pero vamos a verlo en detalle por separado.

## Análisis basado en landmarks

Para el análisis basado en landmarks, utilizaremos el paquete **geomorph**, desarrollado por Dean Adams y colegas. Conceptualmente, el análisis por landmarks se basa la identificación de puntos específicos dentro del objeto u organismo a analizar (en nuestro caso, normalmente artefactos) que encapsulen el máximo posible de información morfométrica. Es fundamental que los puntos seleccionados (cada landmark) estén presentes en **todos los objetos**. Es decir, si elegimos, por ejemplo, un vértice, ese vértice tiene que estar presente en todos los artefactos. Vamos a ver un ejemplo de cómo haríamos esto.

```
## Para crear la secuencia de landmarks
character_index <- paste0(index, ".jpg")

## Cambiamos al directorio en el que se encuentran los objetos
setwd("./Shapes")

## Digitalizamos los landmarks
digitize2d(character_index, nlandmarks = 7, tpsfile = "geos_landmarks")
```

Como esto, para el total de 97 piezas, nos llevaría buena parte de la sesión, ya he preparado el archivo que necesitáis. Lo podéis cargar así:

La función `readland.tps` pertenece al paquete **geomorph**. Este paquete requiere archivos en formato `.tps` para poder desarrollar su análisis y esta es la forma de cargarlos. Si exploráis el objeto que acabáis de cargar, veréis que es un **array** que incluye siete puntos por subobjeto; es decir, cada una de las coordenadas que se han introducido al marcar los landmarks.

```
class(lm_geo)
head(lm_geo)
```

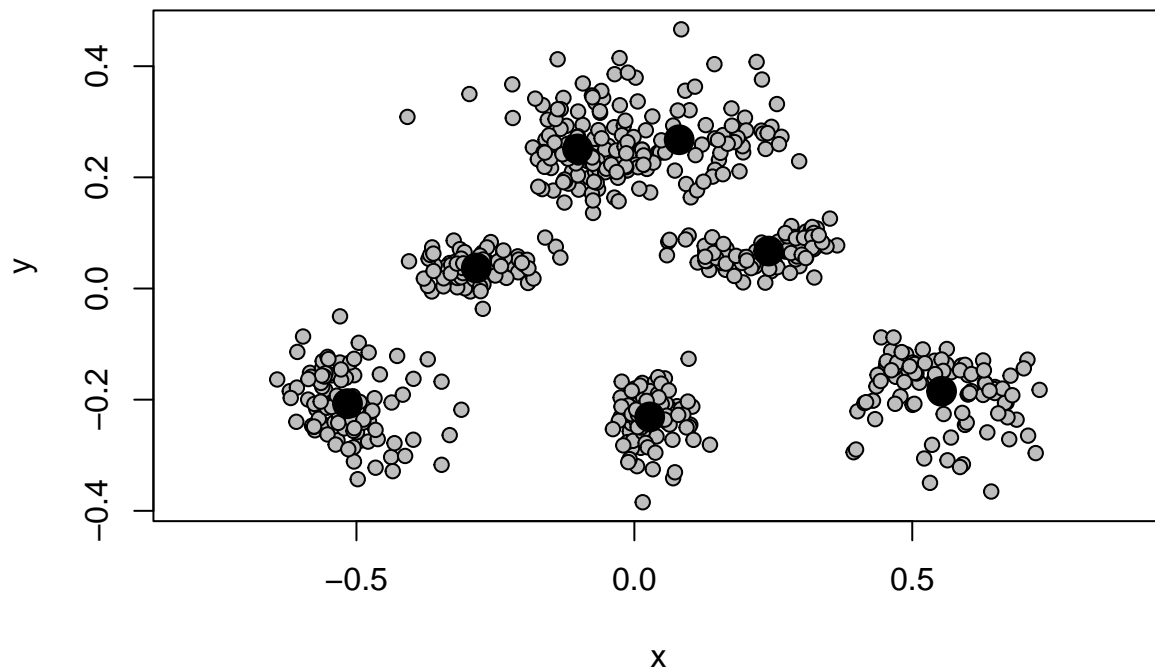
Se presupone que las landmarks habrán estado bien tomadas y que no habrá ninguna faltante, pero si hubiera algún fallo, podéis corregirlo con la siguiente función:

```
lm_geo <- estimate.missing(lm_geo)
```

Fijáos como se ha sobrescrito al objeto anterior. Haciéndolo así (sobrescribiendo objetos que sabes que no vas a reutilizar porque los estás corrigiendo, o modificando definitivamente) se evita sobrecargar el espacio con memoria innecesaria. En todo caso, si se cree que se pueden haber perdido varias landmarks, debe considerarse reempezar todo el proceso de nuevo. Esta función está pensada para utilizar en casos en los que esto no sea posible.

Una parte importante del análisis de morfometría geométrica, y de la que ya habréis oído hablar durante el workshop, es el **General Procrustes Analysis** o GPA. Esencialmente, este análisis consiste en la adecuación en escala de las distintas piezas analizadas. Pensad que, conceptualmente, se están analizando elementos superpuestos, así que es importante que la escala sea la misma para que los distintos landmarks estén *mas o menos* e el mismo lugar. Para hacer un GPA de landmarks, podemos hacer lo siguiente:

```
lm_geo_gpa <- gpagen(lm_geo, print.progress = F)
plot(lm_geo_gpa) ## Vemos los landmarks superpuestos
```



El objeto producido por `gpagen` es relativamente complejo. Observémoslo para entenderlo.

```
str(lm_geo_gpa)
```

```
## List of 15
## $ coords      : num [1:7, 1:2, 1:97] 0.5885 0.297 0.0225 -0.0999 -0.2771 ...
##   ..- attr(*, "dimnames")=List of 3
##     .. ..$ : chr [1:7] "1" "2" "3" "4" ...
##     .. ..$ : chr [1:2] "X" "Y"
##     .. ..$ : chr [1:97] "1" "2" "3" "4" ...
## $ Csize       : Named num [1:97] 178 176 189 210 194 ...
##   ..- attr(*, "names")= chr [1:97] "1" "2" "3" "4" ...
## $ iter        : num 3
## $ consensus   : num [1:7, 1:2] 0.5528 0.2417 0.0806 -0.1022 -0.2843 ...
##   ..- attr(*, "dimnames")=List of 2
##     .. ..$ : chr [1:7] "1" "2" "3" "4" ...
##     .. ..$ : chr [1:2] "X" "Y"
## $ procD       : NULL
## $ p           : int 7
## $ k           : int 2
## $ nsliders    : num 0
## $ nsurf       : num 0
## $ points.VCV  : NULL
## $ points.var  : NULL
## $ data        : NULL
## $ Q           : num 1.06e-05
```

```
## $ slide.method: NULL
## $ call      : language gpagen(A = lm_geo, print.progress = F)
## - attr(*, "class")= chr "gpagen"
```

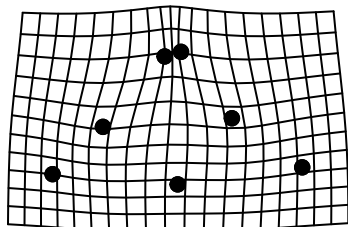
Algo muy útil para estimar la variabilidad de nuestros landmarks es estimar la media de su forma. De ese modo, podemos compararlos entre sí, o con respecto a una base fija, como puede ser la propia media. Esto nos indicaría cuánto se desvían objetos específicos de dicha media.

```
msh <- mshape(lm_geo_gpa$coords) ## Estima la forma media
```

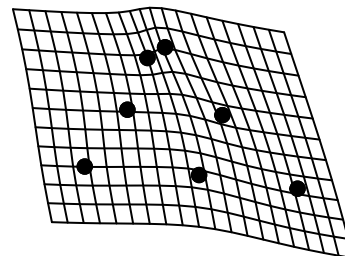
Y podemos utilizar para observar como se relacionan distintas formas y componentes entre sí.

```
par(mfrow = c(1,2))
plotRefToTarget(msh, lm_geo_gpa$coords[, ,5])
title("Mean to geo 5")
plotRefToTarget(msh, lm_geo_gpa$coords[, ,50])
title("Mean to geo 50")
```

**Mean to geo 5**



**Mean to geo 50**

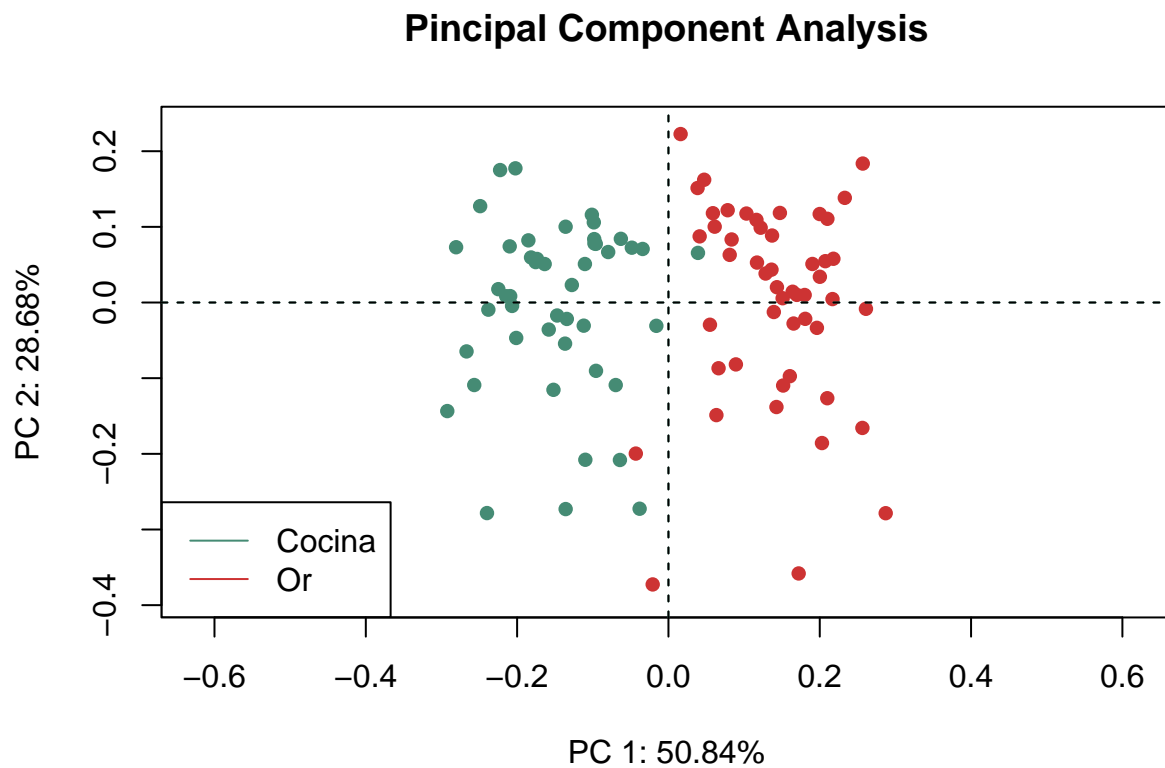


En esta fase de cargar y observar datos, un paso extremadamente común en el análisis de la morfometría geométrica es el análisis de componentes principales (PCA) de las coordenadas. Veremos esto en más detalle en la última sesión pero, por ahora, baste decir que es una forma de reducir la dimensionalidad de elementos que, de otra forma, serían muy difíciles de comprender. Si aplicamos un PCA sobre nuestros objetos, vemos que hay una clara diferencia entre los microlitos geométricos procedentes de la Cueva de la Cocina (Mesolítico) y los procedentes de la Cova de l'Or (Neolítico).

```
## Hacer PCA
lmr_PCA <- gm.prcomp(lm_geo_gpa$coords)

## Crear columna para el color
clr <- ifelse(geos$Site == "Cueva de la Cocina", "aquamarine4", "brown3")

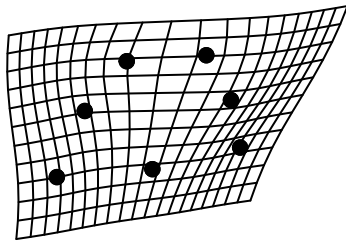
plot(lmr_PCA, pch = 16, col = clr, main = "Pincipal Component Analysis")
legend("bottomleft", legend = c("Cocina", "Or"), lty = c(1,1),
      col = c("aquamarine4", "brown3"))
```



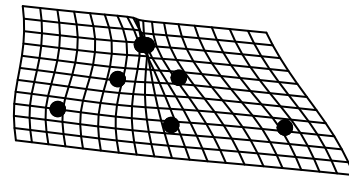
Por último, podemos combinar los dos elementos anteriores y observar cómo se relacionan los componentes principales con las formas medias

```
par(mfrow = c(1,2))
plotRefToTarget(msh, lmr_PCA$shapes$shapes.comp1$max)
title("Mean to max PC1")
plotRefToTarget(msh, lmr_PCA$shapes$shapes.comp1$min)
title("Mean to min PC1")
```

## Mean to max PC1



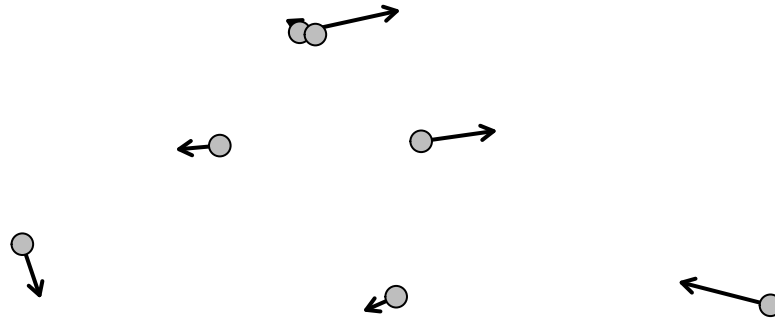
## Mean to min PC1



E, incluso, entender cuales son las tracciones de los distintos landmarks, basándonos en el análisis de componentes principales.

```
par(mfrow = c(1,1))
plotRefToTarget(lmr_PCA$shapes$shapes.comp1$min,
                 lmr_PCA$shapes$shapes.comp1$max, method = "vector",
                 mag = 0.5)
title("Tracciones")
```

## Tracciones



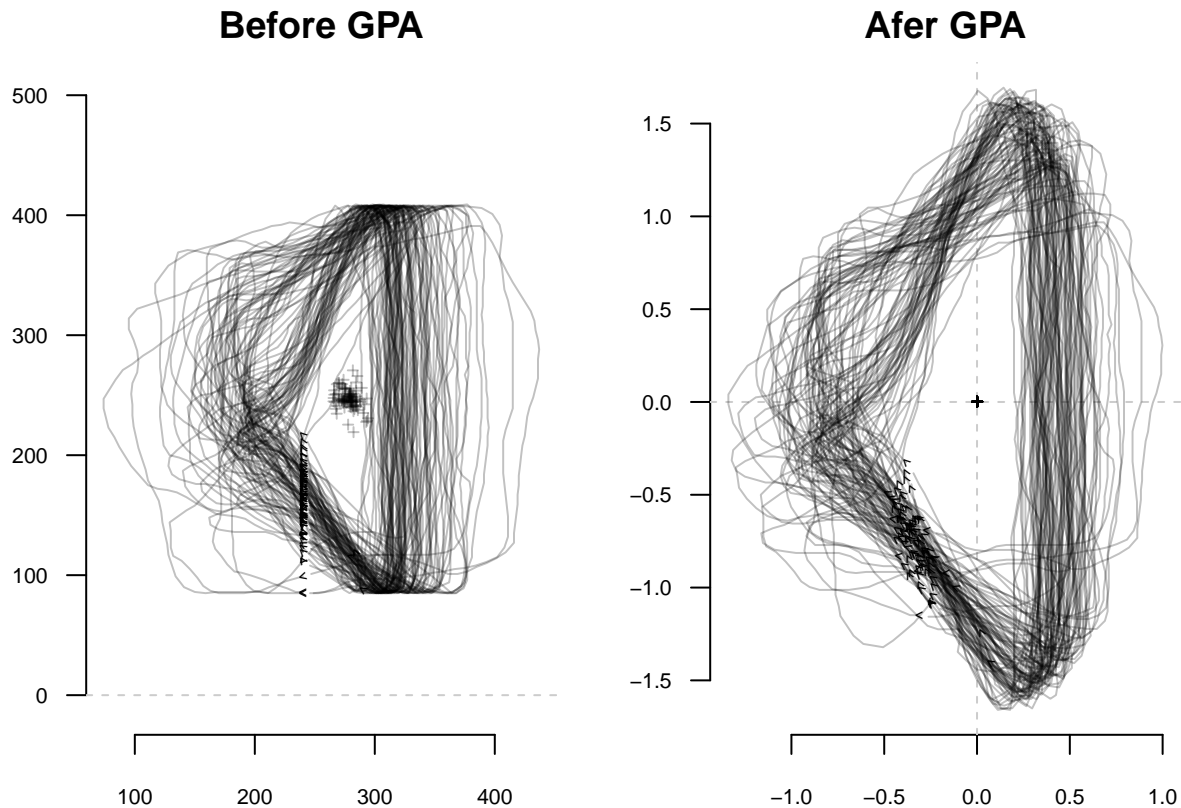
## Análisis basado en outlines

La otra opción es hacer un análisis basado en outlines. En este caso no estaremos utilizando puntos fijos que habremos marcado previamente, sino toda la forma del geométrico. Para esto, el funcionamiento es ligeramente distinto y, por lo tanto, también lo será el paquete que vamos utilizar. Más en concreto, usaremos el paquete `Momocs`, desarrollado y mantenido principalmente por Vincent Bonhomme. Lo primero que tendremos que hacer es cargar los objetos. Esto, esencialmente, consiste en la transformación de una serie de figuras en .jpg (idealmente negras, tal y como las tenéis en la carpeta) a un objeto de clase `coo`, inteligible para `Momocs`. Para eso, hacemos lo siguiente.

Como en el caso anterior, también aquí se tendrá que hacer el GPA.

```
## GPA
geo_out_cntr <- coo_center(coo_scale(geo_out))

## Plot
par(mfrow = c(1,2))
stack(geo_out, title = "Before GPA")
stack(geo_out_cntr, title = "Afer GPA")
```



El análisis de outlines no funciona exactamente igual que el análisis por landmarks. Aquí se hace lo denominado como Elliptical Fourier Analysis. No vamos a entrar en las mates, que son complejas pero, conceptualmente, imaginad que recorréis una piscina (que sería vuestro artefacto) varias veces registrando cada paso que dais. La primera vez que la recorréis solo hacéis un elipse, pero cada vez tenéis que acercaros más a la forma de la piscina hasta que la tengáis al milímetro. Bien, en nuestro caso, cada una de esas vueltas a la piscina se llama armónico, y el problema es que, más armónicos requerirán más tiempo de computación. En ocasiones esa diferencia de computación no es perceptible, pero si tenemos muchos datos podría serlo, así que lo mejor es elegir el mejor número de armónicos para que capte toda la forma del artefacto sin tardar demasiado. En este caso, elegimos siete armónicos.

```
geo_out_F <- efourier(geo_out_cntr, nb.h = 7, norm = F, start = T)
```

De nuevo, podemos utilizar el objeto producido para hacer un análisis de componentes principales

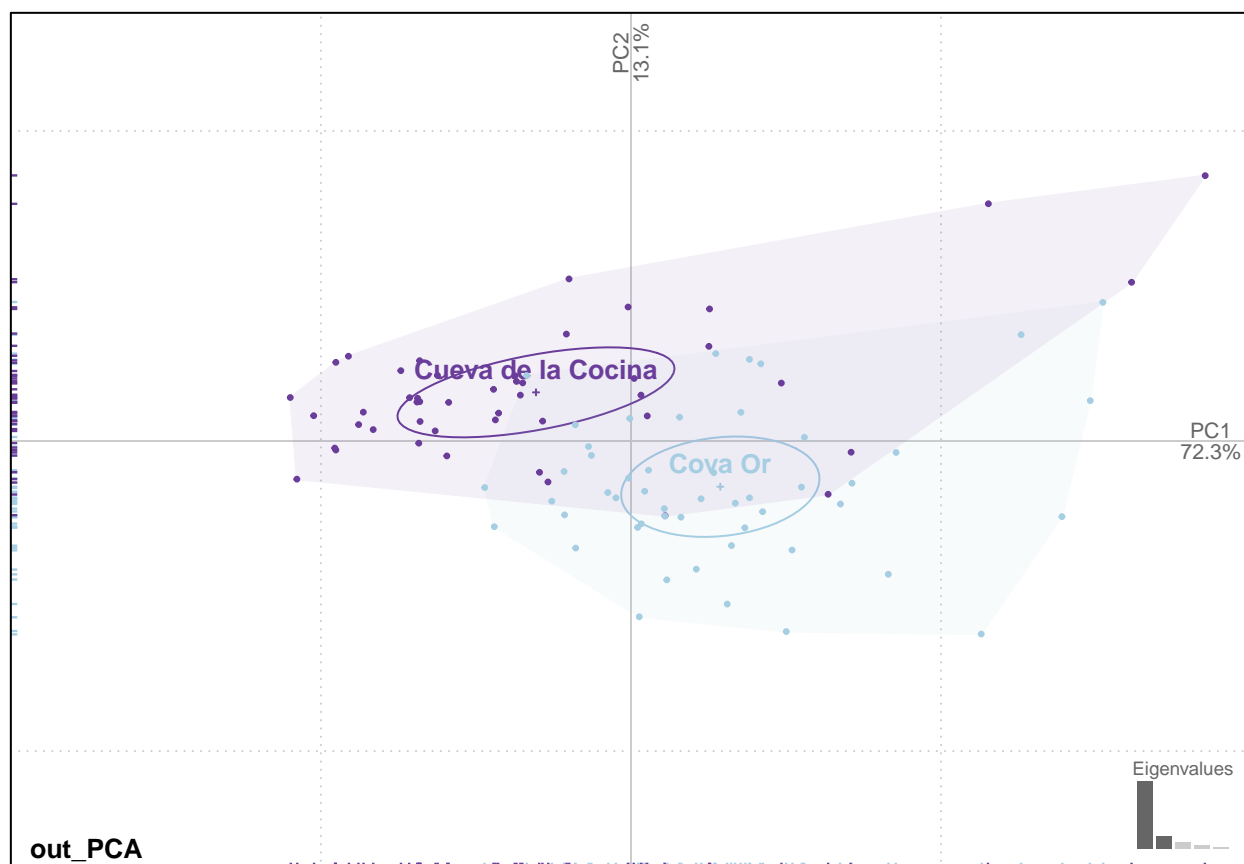
```
## PCA outlines
out_PCA <- PCA(geo_out_F)
```

Momocs ofrece varias opciones para realizar visualizar nuestro PCA con sus propias funciones. Vemos algunas!

```
## PCA plot
plot(out_PCA, 1, pos.shp = "range_axes", palette = col_qual,
     ellipses = TRUE, morphospace = FALSE)
```

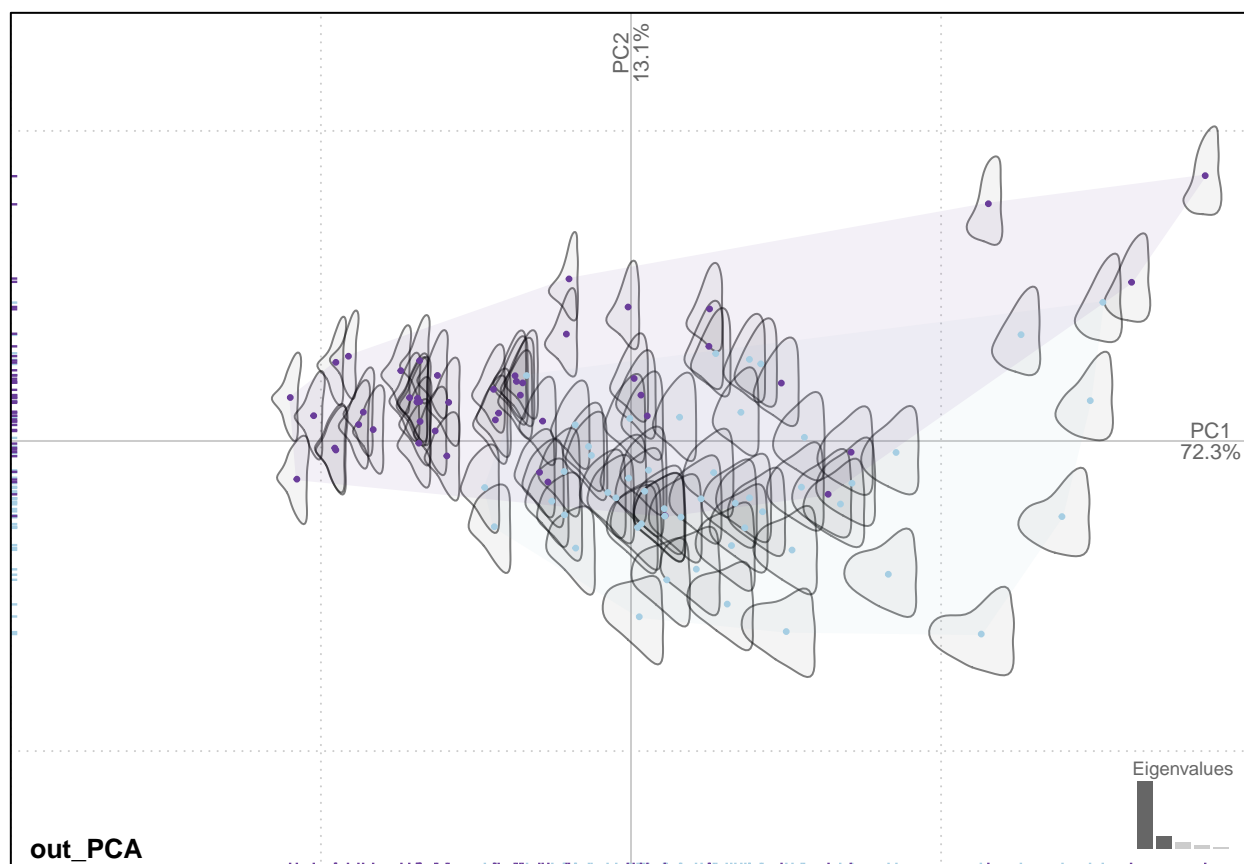
```
## will be deprecated soon, see ?plot_PCA
```





```
## PCA con las formas reales
plot(out_PCA, 1, pos.shp = "xy", palette = col_qual, labelsgroups = FALSE)
```

## will be deprecated soon, see ?plot\_PCA



```
## PCA con labels y formas circulares
plot(out_PCA, 1, pos.shp = "circle", stars = TRUE, palette = col_qual,
      labelsgroups = FALSE, labelspoints = TRUE)
```

## will be deprecated soon, see ?plot\_PCA

