Archaeological Geometric Morphometrics and R

As part of the #StayHomeButStudy Workshop Series

Dr Christian Steven Hoggard (University of Southampton, United Kingdom)

Introduction

This guide provides a "hands-on" step-by-step introduction into the application of geometric morphometric (GMM) methodologies in archaeological science, as conducted the R Environment. Using a published dataset this workflow will guide the reader through four key GMM procedures: 1) data importing (and creation), 2) data transformation, 3) data analysis and 4) data visualisation. A Japanese translation of this documentation is also provided.

I will first demonstrate the actions or functions on Zoom (using this markdown document) and then allow time for all participants to run the function (3-5 minutes per function). To run a 'chunk', a shaded area of function we can press the "Run selected chunk" button, represented by a play button or use the shortcut `ctrl + enter`. Should there be any queries then please let us know in the Slack workspace. Conversely, when you complete a function please could you use a "thumbs up" emoji on Slack. We are allowing time between functions to ensure that all participants keep up, if you finish a particular process early explore the functions in the packages through the 'Help' tab in the 'Packages' window.

This practical constitutes the second workshop of the #StayHomeButStudy event, organised by Dr. Atsushi Noguchi, and is tailored for Japanese archaeologists, researchers and enthusiasts.

About the Code, Packages and Data

The data used throughout this guide originates from Ivanovaitė et al (2020): *"All these Fantastic Cultures? Research History and Regionalization in the Late Palaeolithic Tanged Point Cultures of Eastern Europe"*, published in open-access in the European Journal of Archaeology (https://doi.org/10.1017/eaa.2019.59). The data can be found on a GitHub repository (https://github.com/CSHoggard/-Eastern-Europe-Tanged-Points), in addition to the Open Science Framework (https://osf.io/agrwb/).

All code, and data, including the markdown document (in HTML and PDF format) for this practical can be found on GitHub (https://github.com/CSHoggard/-japanworkshop2020tree/master/workshop_2).

The GMM procedure detailed below is grounded on two-dimensional outline analysis. In conducting outline analysis for this practical the following two packages are necessary:
* **Momocs** (Version 1.3.0) https://cran.r-project.org/web/packages/Momocs/index.html
* **tidyverse** (Version 1.3.0) https://cran.r-project.org/web/packages/tidyverse/index.html ## **Software Installation**

Following the installation of R and RStudio, we can now install the required packages:

```
if(!require("Momocs")) install.packages('Momocs', repos='http://cran.us.r-project.org')
if(!require("tidyverse")) install.packages('tidyverse', repos='http://cran.us.r-project.org')
if(!require("rio")) install.packages('rio', repos='http://cran.us.r-project.org')
```

As the tidyverse and Momocs packages may take time to install given the size of the files *please ensure that these are downloaded prior the workshop.*

To bring the data (from Github) into R/Rstudio we can use the `import` function from the `rio` package, and extract the data from the repository:

```
database <- rio::import("https://github.com/CSHoggard/-workshopjapan2020/raw/master/workshop_2/database

tpsdata <- rio::import("https://github.com/CSHoggard/-workshopjapan2020/raw/master/workshop_2/tpslines.
```

Once installed, the packages can be activated through the `library()` function:

```
library(Momocs)
library(tidyverse)
```

About the Data

This data was composed to assess the robustness of cultural taxonomies in the Final Palaeolithic period of Eastern Europe, as portrayed through tanged point variants. It consists of 250 tanged point outlines and produced in the TPS Suite (https://life.bio.sunysb.edu/morph/soft-dataacq.html), using the **outline object** function. As this dataset was produced in the TPS Suite the file format is **.tps**. In their composition these outlines are semilandmarks, an algorithm-produced series of equidistant points are each shape. A database of all examples and their respective cultural assignment is also provided.

Importing GMM Data: Alternative Approaches

There are a number of ways which landmark and outline morphometric data can be imported into the R Environment. Here, for ease and replicability, the outline data (in .tps format) was stored on a GitHub repository and directly fed into the R environment, utilising the `Momocs::import_tps()` function in a .rds file. Other ways to import .tps data (if saved locally) include the above function, `geomorph::readland.tps()` and rewriting tools in Momocs e.g. `Momocs::rw_rule()`. Data from stereomorph can also be imported through the `Momocs::import_StereoMorph_ldk()` and `Momocs:import_StereoMorph_curve()` functions.

Note: for the purpose of this workshop I will detail in-text the function and its constitutent package e.g. `geomorph::readland.tps()`, however only the function is what will be 'used' so-to-speak e.g. 'readland.tps(). This helps you to understand what packages the functions originate from.

Within Momocs, outlines can be extracted from silhouette data through the `Momocs::import_jpg()` and `Momocs::import_jpg1()` functions. See their respective helpfiles for more details on these functions. These will be demonstrated at the end of the workshop.

Examining the Data

With our data now in the R Environment we can now call our tpsdata object through the `base:: View` functions. The `base::View()` function will highlight the three constituent parts of the tps file: the 1) *Coo* (coordinate data), 2) *cur* (the curve data if necessary), and 3) *scale* (the scale data if present). It is the Coo and scale data which we will take forward, with the database, to examine shape variation among our tanged points. It is best not to call the tps data as R will stream all coordinate data for each example. We can also inspect our database using the `head()` function, and examine the different components of our dataset.

```
head(database)
```

```
## # A tibble: 6 x 10
##      ID Site  Context Longitude Latitude Country Archaeological_~ File_Name
##   <dbl> <chr> <chr>       <dbl>    <dbl> <chr>   <chr>            <chr>
## 1     1 Balt~ Baltaš~      24.0     54.0 Lithua~ Baltic Magdalen~ Bal.1
## 2     2 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.1
## 3     3 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.3
## 4     4 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.4
## 5     5 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.5
## 6     6 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.6
## # ... with 2 more variables: Reference <chr>, Notes <chr>
```

We can observe that the group data we want to examine (Archaeological_Unit) is `<chr>`, that is to say of type 'character', and not `<fctr>` ('factor'), as required for our analysis. This can be corrected through the `base::as_factor()`function and confirmed through the `base::is_factor()` function:

```
database$Archaeological_Unit <- as.factor(database$Archaeological_Unit)

is.factor(database$Archaeological_Unit) # check to see the data is now of type 'character'
```

```
## [1] TRUE
```

We can also inspect the number of different archaeological units within our dataset through the `base::summary()` function. This highlights the number of tanged points in each group. With certain taxonomic units rarely used this is reflected in the low sample sizes for certain groups e.g. Vyshegorian.

```
summary(database$Archaeological_Unit)
```

```
##     Baltic Magdalenian Bromme (Eastern Europe) Bromme (Western Europe)
##                     36                       9                      49
##                 Grensk            Krasnosillya              Perstunian
##                     55                      29                       4
##    Pitted Ware (Type A)               Podolian             Vyshegorian
##                     24                      14                       8
##             Wolkushian
##                     22
```

GMM Procedure 1: Outline File Creation

Central to Momocs are a specific suite of shape classes for: 1) *outlines* (OutCoo), *open outlines* (OpnCoo) and *landmarks* (LdkCoo), with often one class specific to your own dataset. While some operations in Momocs are generic and do not depend on one of these classes, many functions require your data to be one of these specific 'S3 objects'. In this instance our tps data is comprised of outlines, and so we wish for our data to be `OutCoo`, as to enable efourier (elliptic Fourier) analyses. Other analyses including rfourier (radii Fourier) or tfourier (tangent angle Fourier) analyses can be conducted through this process but for this workshop we're only going consider elliptic Fourier analysis (EFA).

Through this lens, the coordinate data (coo) must therefore be turned into outline data through the `Momocs::Out()` function for the workflow to work. Once performed, we can then enter the object (here titled 'shape') and examine its properties.

```
shape <- Out(tpsdata$coo, fac = database) # incorporating our database as our factors
shape # call the object
```

```
## Out (outlines)
##   - 250 outlines, 1543 +/- 1370 coords (in $coo)
##   - 10 classifiers (in $fac):
## # A tibble: 250 x 10
##      ID Site  Context Longitude Latitude Country Archaeological_~ File_Name
##   <dbl> <chr> <chr>       <dbl>    <dbl> <chr>   <fct>            <chr>
## 1     1 Balt~ Baltaš~      24.0     54.0 Lithua~ Baltic Magdalen~ Bal.1
## 2     2 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.1
## 3     3 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.3
## 4     4 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.4
## 5     5 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.5
## 6     6 Baro~ Barouka      30.3     53.5 Belarus Grensk           Bor.6
## # ... with 244 more rows, and 2 more variables: Reference <chr>, Notes <chr>
##   - also: $ldk
```

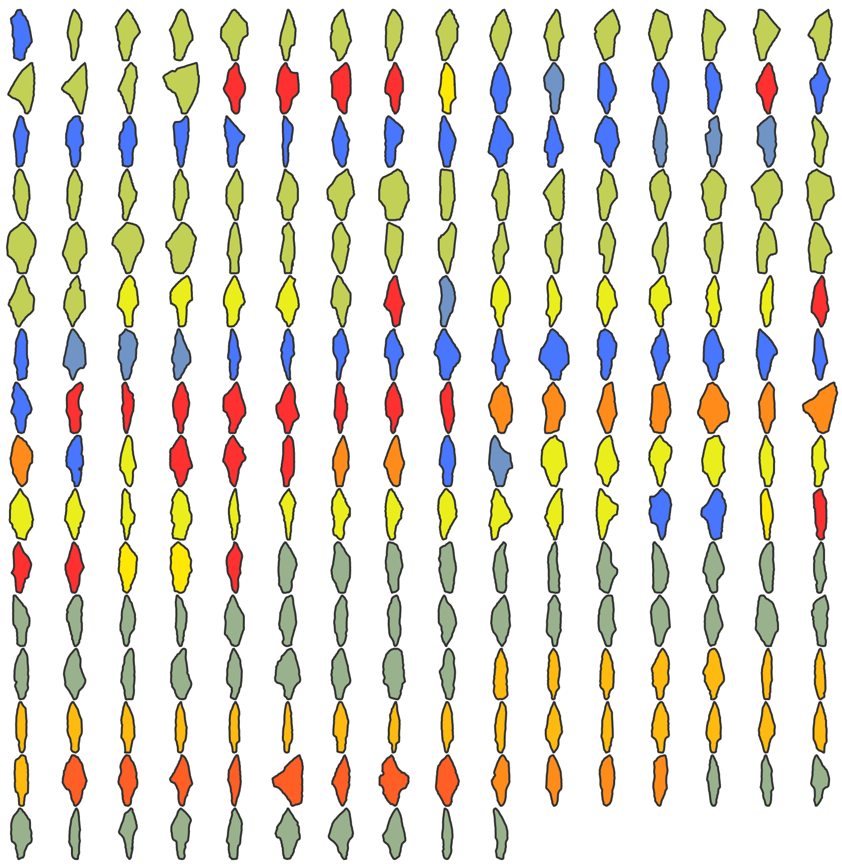
This tells us that in our Out file there are a total of 250 outlines, with a mean number of 1543 landmarks and 10 different factors (longitude, Latitude, Archaeological_Unit, etc.). We are only going consider Archaeological_Unit within these factors.

GMM Procedure 2: Outline Visualisation

Now our data is in the R environment and in the appropriate class required for Momocs, we can examine the outline shapes. We can first look at all outlines through the `Momocs::panel()` function. Factors can also be coloured in using the `fac` argument.

An example using the `Momocs::panel()` function is seen below.

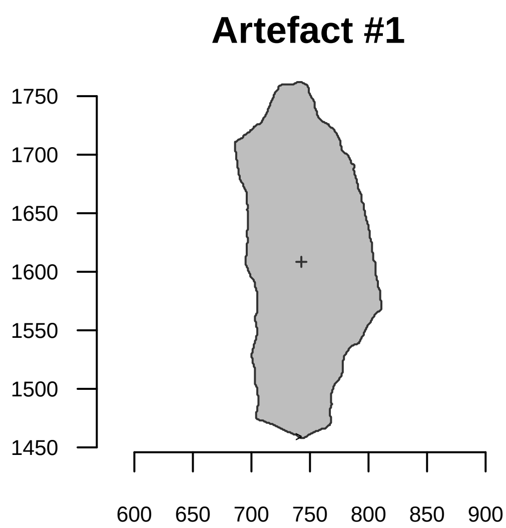
```
panel(shape, main = "", fac = 'Archaeological_Unit')
```



An alternative to the `Momocs::panel()` function is `Momocs::mosaic()`, an updated display function (which will soon replace panel). This does include a legend, unlike the panel function, however the legend drawing options are limited, and are currently being improved for further package versions.

We can also draw individual shapes of interest using the `Momocs::coo_plot()` function. A number of aesthetic or stylistic changes (including line colour and fill) are possible.

```
coo_plot(shape[1], col = "grey", main = "Artefact #1")
```



GMM Procedure 3: Outline Normalisation

Normalisation, as stressed by Claude (2008), has long been an issue in the elliptic Fourier process. Normalisation can be performed through the actual elliptic Foruier transformation (using what is known as the "first ellipse"). As we noted in the first workshop, this process (normalisation and elliptic fitting to coefficients) is equivalent to the Procrustes Superimposition for landmark data./

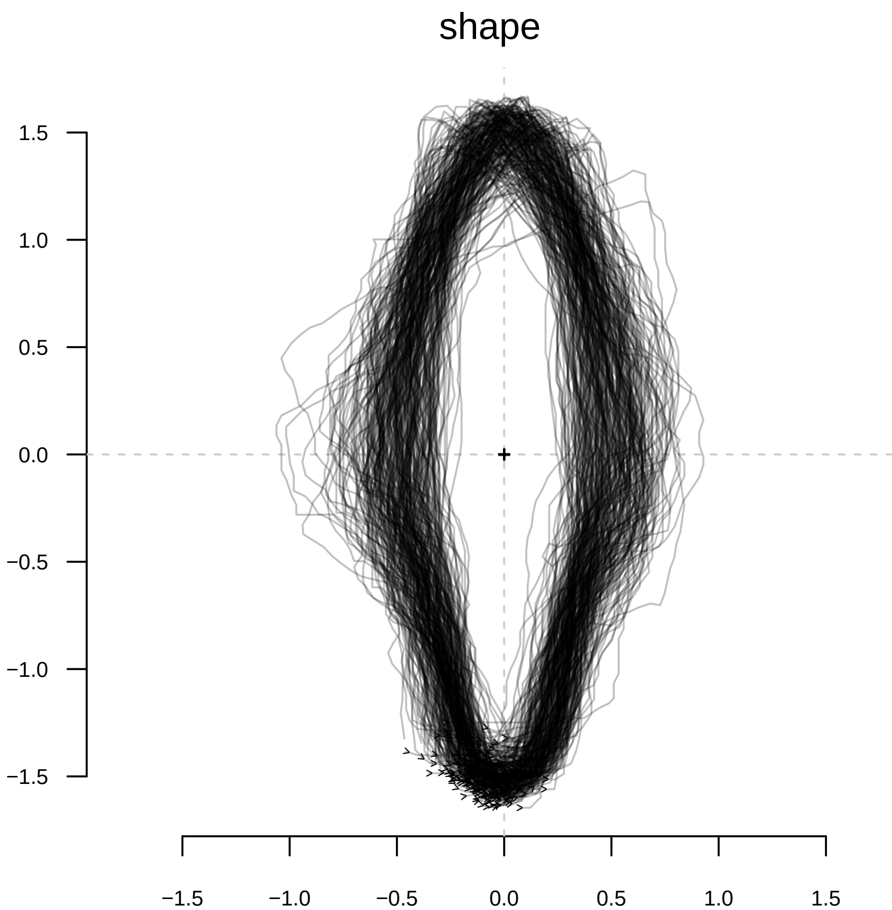
It is recommended to normalise (standardise) and align your shapes before the `Momocs::efourier()` process. Rotation was considered before outline digitisation, however rotation could also be explored in Momocs through the `Momocs::coo_aligncalliper()` function. Here we will explore three transformation processes: 1) `Momocs::coo_center()`, 2) `Momocs::coo_scale()` and 3) `Momocs::coo_close()`.

These three functions perform the following actions:
* `Momocs::coo_center()`: This action centres coordinates on a common origin (common centroid).
* `Momocs::coo_scale()`: This action scales the coordinates by their 'scale' if provided, or centroid size if 'scale is not provided.
* `Momocs::coo_close()`: Closes unclosed shapes (precautionary).

We can then use the `Momocs::stack()` function to inspect all outlines, now according to a common centroid and of a common scale:

```
shape <- coo_center(shape)
shape <- coo_scale(shape)
shape <- coo_close(shape)

stack(shape, main = "")
```



GMM Procedure 4: Elliptic Fourier Transformation

Elliptic Fourier Analysis (EFA) is one of a number of Fourier based methods of curve composition derived from the first series by Jean Baptiste Joseph Fourier (1768-1830), and developed by Giardina and Kuhl (1977) and Kuhl and Giardina (1982). In practice, a set of four parametric equations (grounded on sine and cosine transformations) are used to define the x and y Cartesian landmarks into curves (Fourier harmonic amplitudes). The coefficients (termed A,B,C and D), when summed together, represent the approximation of artefact form, and are the framework for further analyses. This level of detail depends on the number of harmonics you use. The first harmonic (first ellipse) is responsible for rotation and defines an ellipse in the plane, with which all other harmonics fit onto. The greater the number of harmonics, the greater the level of detail, and the closer the curves resemble the shape. However, a considerable level of statistical noise is produced if there is too much detail (and thus too many harmonics), and so an appropriate level of harmonics are necessary.

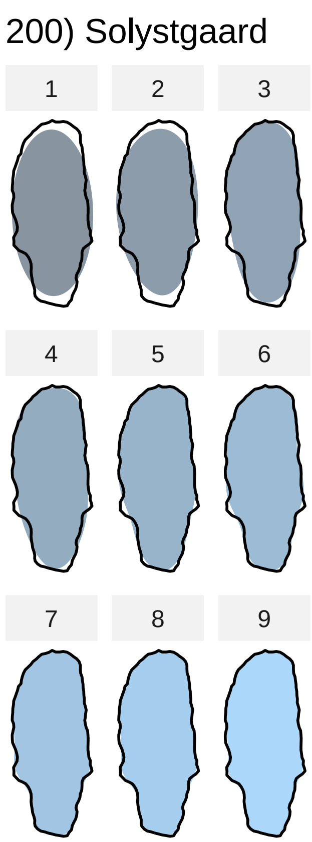
When a level of harmonic power is determined by the researcher (95%, 99%, 99.9%, 99.99%), a series of procedures can be implemented to test how many harmonics are necessary:
* `Momocs::calibrate_harmonicpower_efourier()`: This function estimates the number of harmonics required for the elliptic Fourier process (and all other Fourier processes).
* `Momocs::calibrate_reconstructions_efourier()`: This procedure calculates reconstructed shapes for a series of harmonic numbers. This process best demonstrates the harmonic process.
* `Momocs::calibrate_deviations_efourier`(): This procedure calculates deviations from the original and reconstructed shapes for a series of harmonic numbers.

```r
calibrate_harmonicpower_efourier(shape, id = 4, nb.h = 20, plot = FALSE)
```

```
## $gg
## [1] NA
##
## $q
##                  h1      h2      h3       h4       h5      h6       h7       h8
## 4) Barouka 6.27197 77.4212 80.2774 97.35366 97.45275 98.0372 98.43572 99.14469
##                   h9      h10      h11      h12      h13      h14      h15
## 4) Barouka 99.25864 99.40648 99.55404 99.68398 99.72722 99.81056 99.89088
##                  h16      h17      h18 h19
## 4) Barouka 99.92353 99.93879 99.96744 100
##
## $minh
##   90%   95%   99% 99.9%
##     5     5     9    17
```

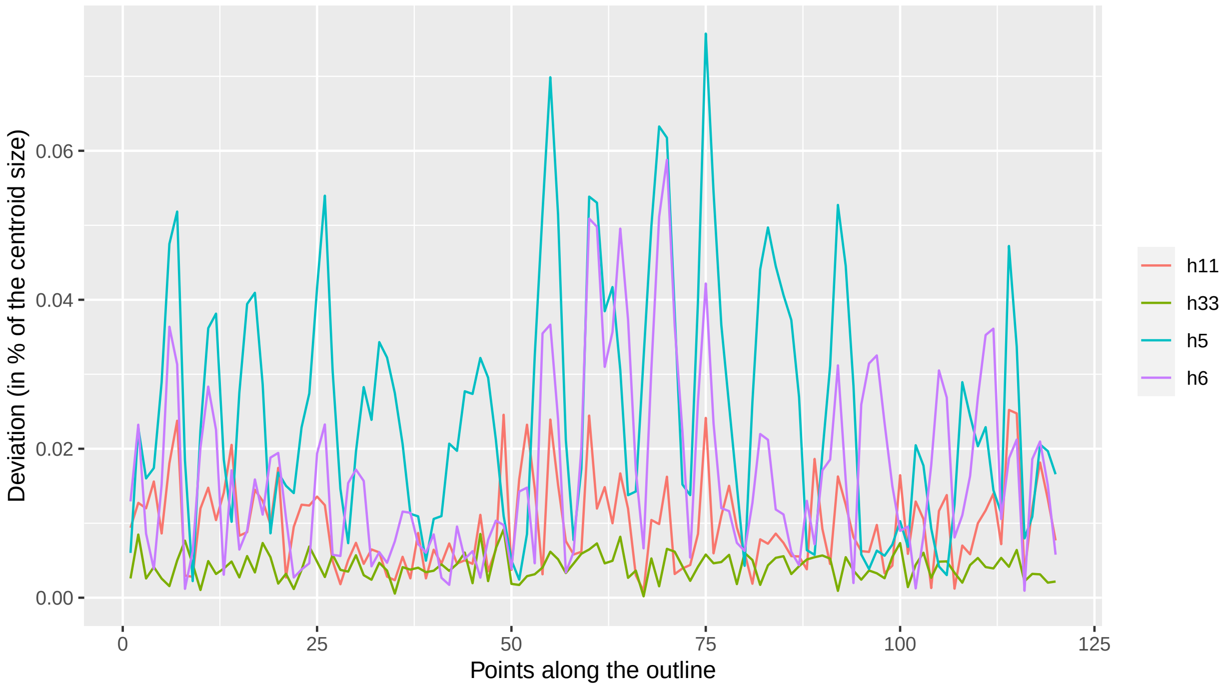
This first procedures highlights how much shape (harmonic power) is represented by the individual harmonics. Here we assessed it on one example and only considered the first twenty harmonics. Typically, the process is performed on all shapes, however one is used here to detail the components obtained from the function. These three calibrate functions can also take some time to process, please be patient while they load.

```r
calibrate_reconstructions_efourier(shape)
```



This second function best exemplifies the harmonic concept: as the number of harmonics increase, so the approximation of shape is closer to the digitised artefact. This function also highlights the elliptic fitting in the first harmonic.

```r
calibrate_deviations_efourier(shape)
```



The third and final function provides another means of examining the role of harmonic power on deviation in shape.

Once we know how many harmonics are required we can use the `Momocs::efourier()` function to generated out OutCoe (outline coefficients) object.

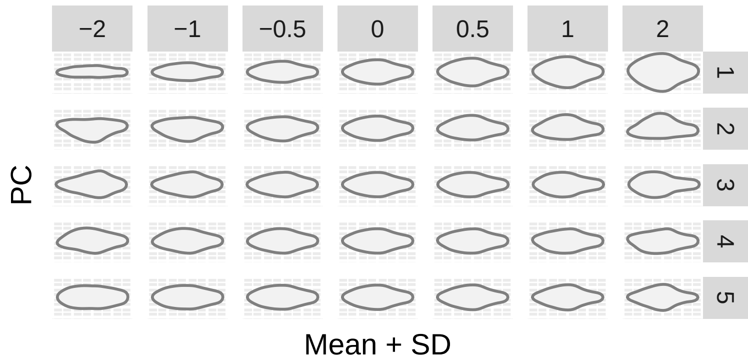
```r
efashape <- efourier(shape, nb.h = 11, smooth.it = 0, norm = TRUE)
```

GMM Procedure 5: Principal Component Analysis (PCA)

With our elliptic fourier coefficients we can now begin the exploratory and analytical procedure. We will start by exploring the main theoretical differences in shape through a Principal Component Analysis (PCA). Please refer to the first workshop for a detailed explanation of PCA. We first need to convert out OutCoe class object to a PCA class object through the `Momocs::PCA()` function. We can then explore the main sources of shape variation through the `Momocs::PCcontrib()` function.

The proportion can also be retrieved through calling the `Momocs::Scree()` function.

```r
pcashape <- PCA(efashape)
PCcontrib(pcashape, nax = 1:5)
```



We can see through this function that Principal Component 1 (PC1), i.e. the main source of shape variation among the tanged points, range from thin tanged points to wider-tanged examples, and that Principal Component 2 (PC2), i.e. the second main source of shape variation, extends from left-exaggerated tangs to right-exaggerated tangs. This function can be set to display as many sources of shape variation as required by the researcher.

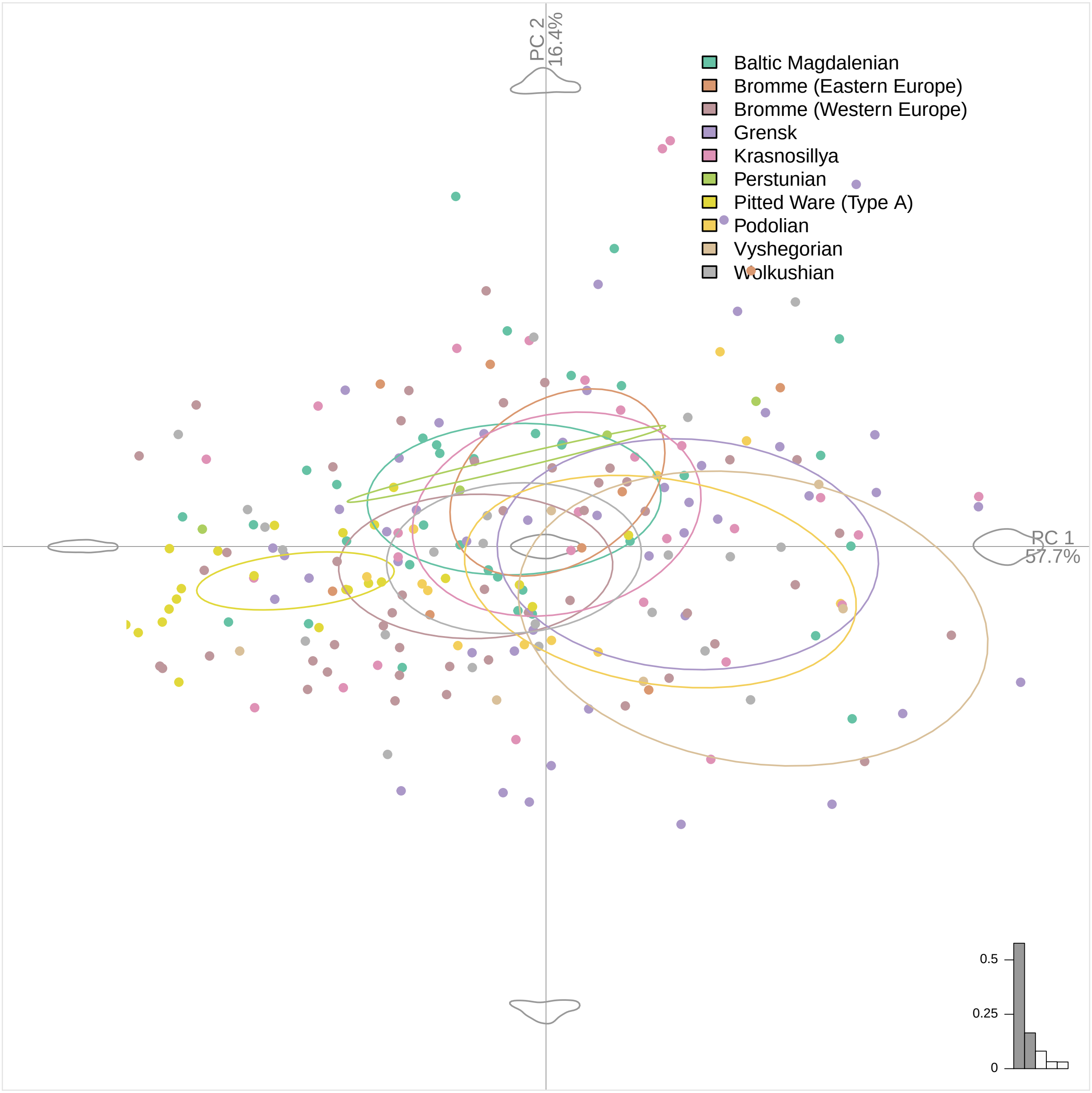
While we can observe the main changes in artefact shape, at present we are unsure how much variation these components account for. Using the `Momocs::Scree()` function we can find out that PC1 accounts for 57.7% of all shape variation, and that the first two axes account for 74.1% (almost three quarters of all shape variation within our dataset). 95% of all shape variation can be accounted for in the first ten principal components (an observation we will come back to afterwards).

```
scree(pcashape)
```

```
## # A tibble: 44 x 3
##     axis proportion cumsum
##    <int>      <dbl>  <dbl>
##  1     1    0.577    0.577
##  2     2    0.164    0.741
##  3     3    0.0806   0.821
##  4     4    0.0319   0.853
##  5     5    0.0307   0.884
##  6     6    0.0251   0.909
##  7     7    0.0195   0.929
##  8     8    0.0106   0.939
##  9     9    0.00918  0.948
## 10    10    0.00719  0.956
## # ... with 34 more rows
```

Now we know the main sources of shape variation, and the importance of each axis, we can now observe how each tanged point is reflected in the theoretical shape space through the `Momocs::plot_PCA()` function.

```
plot_PCA(pcashape, axes = c(1,2), ~Archaeological_Unit, morphospace_position = "full_axes", zoom = 2, c
```

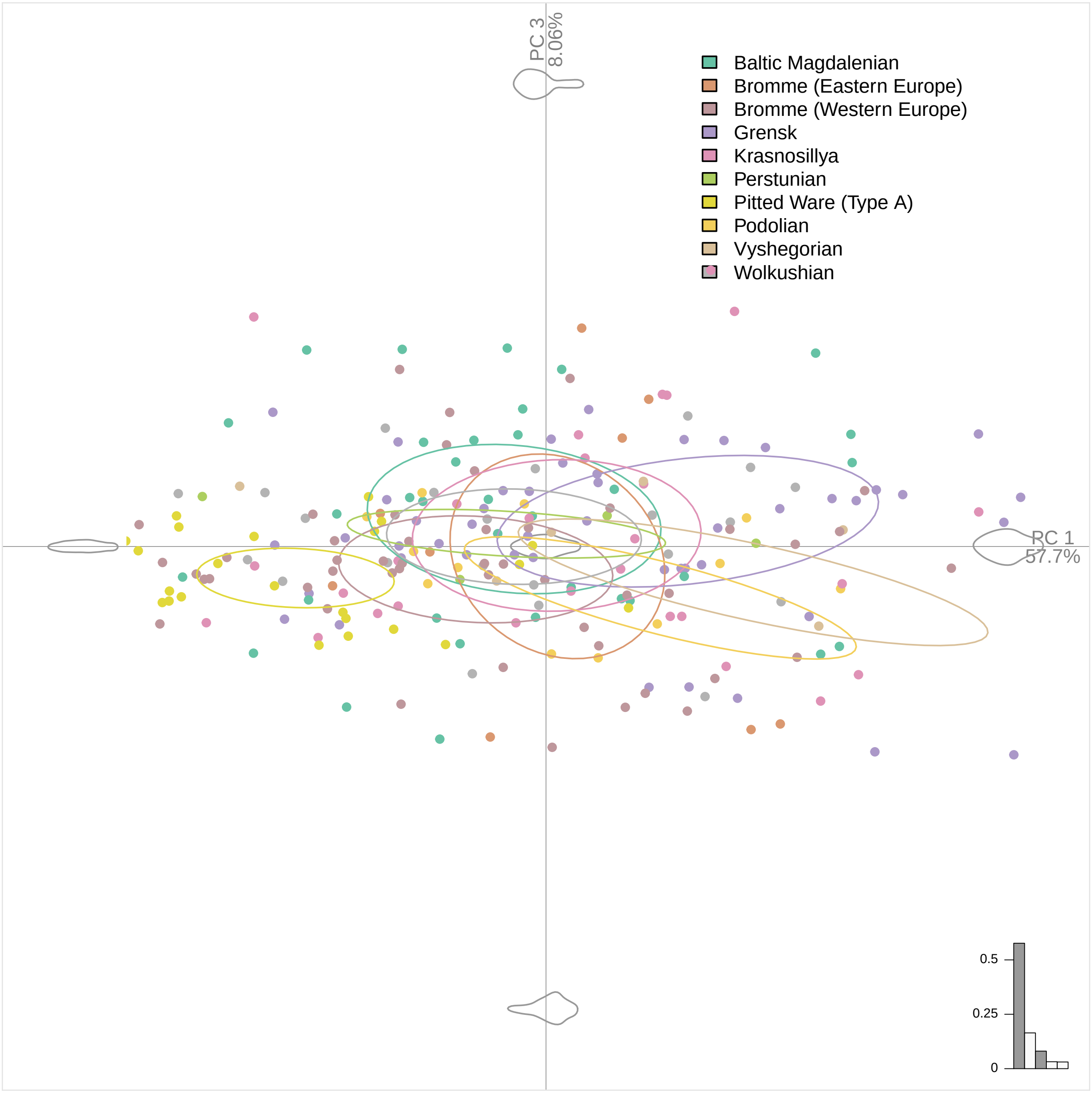


In this diagram we can observe the different distributions of each archaeological unit within the morphospace, and the relative clustering of each unit within this graph. It's important to remember that this graph only represents the first two principal components, we may wish to examine other sources of shape variation (some which may be of importance to archaeologists).

Note: pipes (%>%) are used here to processes multiple arguments at the same time. Momocs supports piping with the whole process able to be 'piped'. For teaching purposes we are doing the 'long way' of GMM.

If we wish to examine the relationship between different principal components we can use the `axes` argument to change our graph configuration. For example, if we wish to examine differences in shape between PC1 and PC3 we can specify the `axes` argument in the following way:

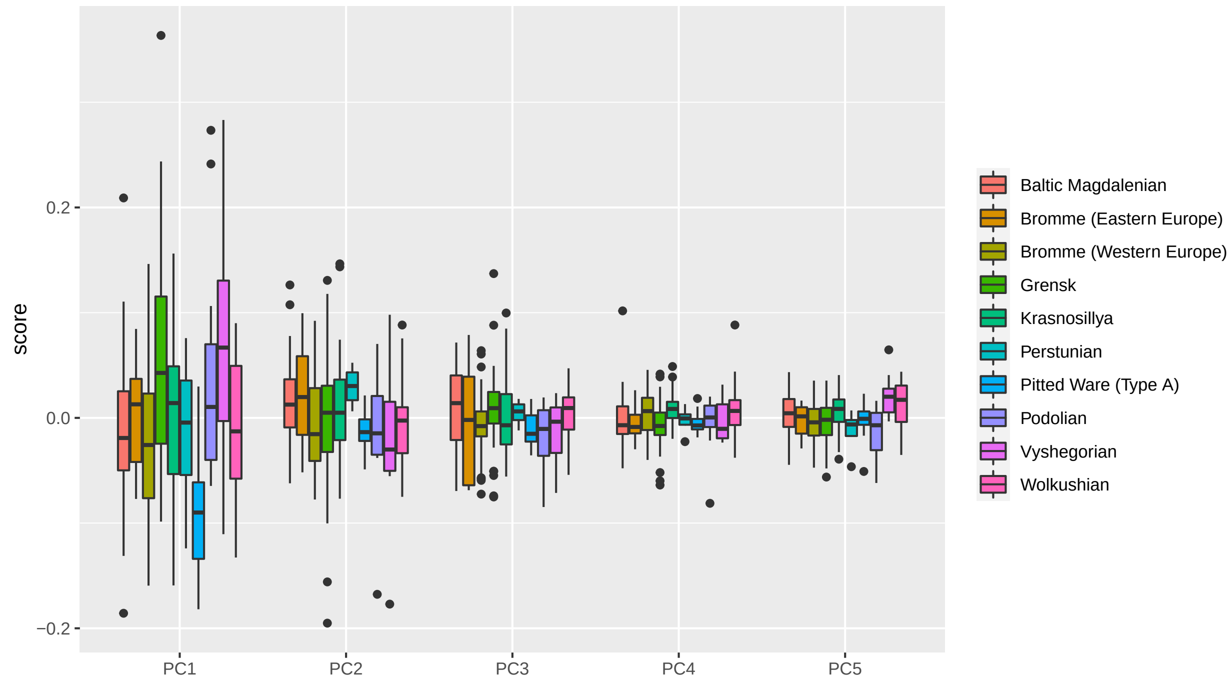
```
plot_PCA(pcashape, axes = c(1,3), ~Archaeological_Unit, morphospace_position = "full_axes", zoom = 2, c
```



There are also a number of other visualisation options including the addition of confidence axes, convex hulls, and morphospace layouts (not explored here).

We can also visualise these principal components, and the variance within different archaeological units, in an alternative way, through the `Momocs::boxplot()` function:

```
boxplot(pcashape, ~Archaeological_Unit, nax = 1:5)
```



GMM Procedure 6: Discriminant Analysis (LDA/CVA)

As we highlighted in the first workshop, PCA explores differences in shape variation irrespective of group composition (i.e. *a priori* groupings). Through a discriminant analysis we can examine differences in shape as based on their maximum group seperation (between-group variation in contrast to within-group variation). In Momocs, we use the `Momocs::LDA()` function on either the elliptic Fourier coefficients or the PCA scores to produce our class accuracy, plots and correction scores. There is no correct answer as to which to use, it depends on the data you wish to examine. In using the PCA scores it is possible to retain a number of components that are deemed important, this can be either: 1) the first nth components, 2) the number of components representing a certain level of shape variance (e.g. 95%, 99%, 99.9%), or 3) all principal components. The coefficients, in contrast would encapsulate all shape data.

With greater levels of data you may include a degree of statistical importance, with smaller unimportant variables taking precedence, and so an optimal level of data is necessary.

When we produced a scree table (the table with documented the percentage variance for each component) we observed that the first ten components defined 95% cumulative shape variance. We can produce a discriminant analysis on just these ten components if we wish.

First we create the object:

```
dashape <- LDA(pcashape, ~Archaeological_Unit, retain = 0.95)
```

We can now examine different aspects of our discriminant analysis data, including the cross-validation table (actual vs. predicted categories for artefacts) and the proportion of correctly classified individuals.

```
dashape$CV.correct
```

```
## [1] 0.324
```

```
dashape$CV.ce
```

```
##      Baltic Magdalenian Bromme (Eastern Europe) Bromme (Western Europe)
##               0.2222222               0.0000000               0.5510204
##                  Grensk            Krasnosillya              Perstunian
```

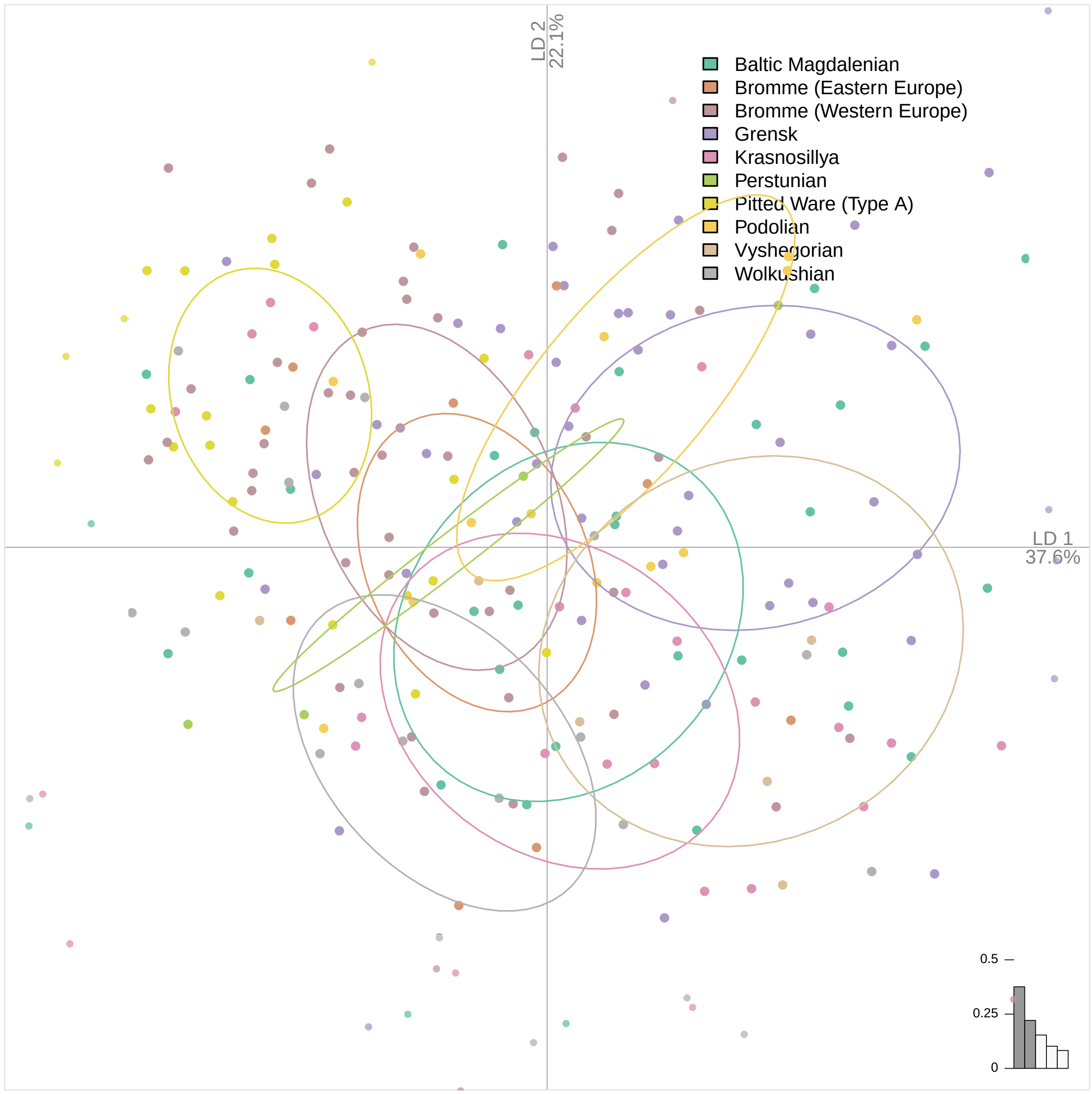
```
##                0.4727273                 0.1379310                 0.0000000
##     Pitted Ware (Type A)                  Podolian               Vyshegorian
##                0.5000000                 0.0000000                 0.0000000
##               Wolkushian
##                0.1818182
```

When we use the `CV.correct` argument we see that 32.4% of tanged points can be correctly classified. We can examine this in further detail through the `CV.ce` argument.

We can see through our classification error table that certain archaeological units are better defined in two-dimensional shape than others e.g. Pitted Ware (Type A), Grensk and Bromme (Western Europe). More detailed metrics are included in the `Momocs::classification_metrics()` function (not covered here).

If we wish to visualise our plot, as is common in exploratory procedures we can use the `Momocs::plot_LDA()` function, using similar arguments to `Momocs::plot_PCA()`:

```
plot_LDA(dashape, axes = c(1,2), zoom = 2, chull = FALSE) %>% layer_points(cex = 1) %>% layer_ellipses(
```



Here we can see how Pitted Ware (Type A), a control from a different period (but of similar technique) can be differentiated from all other archaeological units.

Note: for the elliptic Fourier coefficient discriminant analysis it is relatively straight forward to impose the shapes onto the graph using the `layer_morphospace_LDA()` argument.

GMM Procedure 7: Multiple Analysis of Variance (MANOVA)

So far we have explored the differences in shape within the whole group of artefacts and explored how well they can be seperated through their group variance. Now we need to test, within an statistical framework, whether there is a difference in the PC scores (representative of shape) within and between the different archaeological units. A MANOVA will be our required test given we have multiple groups and multiple column data (PC scores).

Once we have chosen a desired alpha level as of marker of difference (that is to say the boundary with which we are able to reject the null hypothesis of same populations) e.g. 0.05 we can use the `Momocs::MANOVA()` function, noting "Archaeological_Unit" to be our factor which we want to consider:

```
MANOVA(pcashape, ~Archaeological_Unit, retain = 0.95)
```

```
##            Df Hotelling-Lawley approx F num Df den Df    Pr(>F)
## fac         9           1.0137   2.5819     90   2063 1.593e-13 ***
## Residuals 240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note how we are still using 95% cumulative shape variance as represented by our principal component scores. Once we perform the MANOVA we can see that the null hypothesis is rejected as the p value is below the 0.05 level (or any of the other significance levels). We can examine this in finer detail through pair-wise MANOVA analyses, using the `Momocs::MANOVA_PW()` function:

```
MANOVA_PW(pcashape, ~Archaeological_Unit, retain = 0.95)
```

```
## $stars.tab
##                           Baltic Magdalenian Bromme (Eastern Europe)
## Baltic Magdalenian                          -
## Bromme (Eastern Europe)
## Bromme (Western Europe)
## Grensk
## Krasnosillya
## Perstunian
## Pitted Ware (Type A)
## Podolian
## Vyshegorian
##                           Bromme (Western Europe) Grensk Krasnosillya Perstunian
## Baltic Magdalenian        **                      *      -            -
## Bromme (Eastern Europe) -                         -      -            -
## Bromme (Western Europe)                           ***    *            -
## Grensk                                                   **           -
## Krasnosillya                                                          -
## Perstunian
## Pitted Ware (Type A)
## Podolian
## Vyshegorian
##                           Pitted Ware (Type A) Podolian Vyshegorian Wolkushian
## Baltic Magdalenian        ***                  *        *           -
## Bromme (Eastern Europe) ***                    -        -           -
## Bromme (Western Europe) **                     -        *           -
## Grensk                    ***                  -        -           **
## Krasnosillya              ***                  -        .           -
## Perstunian                ***                  -        -           -
## Pitted Ware (Type A)                           ***      ***         ***
## Podolian                                                -           -
## Vyshegorian                                                         .
##
## $summary (see also $manovas)
##                                                      Df  Pillai approx F num Df
## Baltic Magdalenian - Bromme (Eastern Europe)          1 0.02214  0.22642      4
## Baltic Magdalenian - Bromme (Western Europe)          1 0.17590  4.26888      4
## Baltic Magdalenian - Grensk                           1 0.12880  3.17862      4
## Baltic Magdalenian - Krasnosillya                     1 0.07292  1.17975      4
```

```
## Baltic Magdalenian - Perstunian                      1 0.01123  0.09936      4
## Baltic Magdalenian - Pitted Ware (Type A)            1 0.43615 10.63610      4
## Baltic Magdalenian - Podolian                        1 0.19030  2.64405      4
## Baltic Magdalenian - Vyshegorian                     1 0.22929  2.90074      4
## Baltic Magdalenian - Wolkushian                      1 0.08186  1.18136      4
## Bromme (Eastern Europe) - Bromme (Western Europe)  1 0.10733  1.59313      4
## Bromme (Eastern Europe) - Grensk                     1 0.05268  0.82029      4
## Bromme (Eastern Europe) - Krasnosillya               1 0.12193  1.14560      4
## Bromme (Eastern Europe) - Perstunian                 1 0.10234  0.22801      4
## Bromme (Eastern Europe) - Pitted Ware (Type A)       1 0.51283  7.36881      4
## Bromme (Eastern Europe) - Podolian                   1 0.20740  1.17751      4
## Bromme (Eastern Europe) - Vyshegorian                1 0.25072  1.00384      4
## Bromme (Eastern Europe) - Wolkushian                 1 0.11384  0.83503      4
## Bromme (Western Europe) - Grensk                     1 0.23821  7.73923      4
## Bromme (Western Europe) - Krasnosillya               1 0.14484  3.09097      4
## Bromme (Western Europe) - Perstunian                 1 0.09731  1.29366      4
## Bromme (Western Europe) - Pitted Ware (Type A)       1 0.21505  4.65731      4
## Bromme (Western Europe) - Podolian                   1 0.12311  2.03571      4
## Bromme (Western Europe) - Vyshegorian                1 0.19687  3.18671      4
## Bromme (Western Europe) - Wolkushian                 1 0.08053  1.44514      4
## Grensk - Krasnosillya                                1 0.17393  4.15845      4
## Grensk - Perstunian                                  1 0.04248  0.59899      4
## Grensk - Pitted Ware (Type A)                        1 0.36709 10.73009      4
## Grensk - Podolian                                    1 0.10425  1.86218      4
## Grensk - Vyshegorian                                 1 0.07279  1.13823      4
## Grensk - Wolkushian                                  1 0.18226  4.01179      4
## Krasnosillya - Perstunian                            1 0.07646  0.57950      4
## Krasnosillya - Pitted Ware (Type A)                  1 0.49924 11.96361      4
## Krasnosillya - Podolian                              1 0.17440  2.00682      4
## Krasnosillya - Vyshegorian                           1 0.23935  2.51739      4
## Krasnosillya - Wolkushian                            1 0.03428  0.40820      4
## Perstunian - Pitted Ware (Type A)                    1 0.59901  8.58949      4
## Perstunian - Podolian                                1 0.31055  1.46393      4
## Perstunian - Vyshegorian                             1 0.28762  0.70655      4
## Perstunian - Wolkushian                              1 0.11196  0.66190      4
## Pitted Ware (Type A) - Podolian                      1 0.54690  9.95782      4
## Pitted Ware (Type A) - Vyshegorian                   1 0.57670  9.19628      4
## Pitted Ware (Type A) - Wolkushian                    1 0.44407  8.18756      4
## Podolian - Vyshegorian                               1 0.19298  1.01626      4
## Podolian - Wolkushian                                1 0.18524  1.76197      4
## Vyshegorian - Wolkushian                             1 0.27536  2.37503      4
##                                                    den Df    Pr(>F)
## Baltic Magdalenian - Bromme (Eastern Europe)           40 9.220e-01
## Baltic Magdalenian - Bromme (Western Europe)           80 3.501e-03
## Baltic Magdalenian - Grensk                            86 1.740e-02
## Baltic Magdalenian - Krasnosillya                      60 3.289e-01
## Baltic Magdalenian - Perstunian                        35 9.820e-01
## Baltic Magdalenian - Pitted Ware (Type A)              55 1.865e-06
## Baltic Magdalenian - Podolian                          45 4.572e-02
## Baltic Magdalenian - Vyshegorian                       39 3.407e-02
## Baltic Magdalenian - Wolkushian                        53 3.296e-01
## Bromme (Eastern Europe) - Bromme (Western Europe)      53 1.897e-01
## Bromme (Eastern Europe) - Grensk                       59 5.174e-01
## Bromme (Eastern Europe) - Krasnosillya                 33 3.524e-01
```

```
## Bromme (Eastern Europe) - Perstunian                     8 9.151e-01
## Bromme (Eastern Europe) - Pitted Ware (Type A)        28 3.469e-04
## Bromme (Eastern Europe) - Podolian                    18 3.539e-01
## Bromme (Eastern Europe) - Vyshegorian                 12 4.432e-01
## Bromme (Eastern Europe) - Wolkushian                  26 5.153e-01
## Bromme (Western Europe) - Grensk                      99 1.810e-05
## Bromme (Western Europe) - Krasnosillya                73 2.081e-02
## Bromme (Western Europe) - Perstunian                  48 2.858e-01
## Bromme (Western Europe) - Pitted Ware (Type A)        68 2.210e-03
## Bromme (Western Europe) - Podolian                    58 1.012e-01
## Bromme (Western Europe) - Vyshegorian                 52 2.047e-02
## Bromme (Western Europe) - Wolkushian                  66 2.290e-01
## Grensk - Krasnosillya                                 79 4.151e-03
## Grensk - Perstunian                                   54 6.649e-01
## Grensk - Pitted Ware (Type A)                         74 6.508e-07
## Grensk - Podolian                                     64 1.279e-01
## Grensk - Vyshegorian                                  58 3.476e-01
## Grensk - Wolkushian                                   72 5.405e-03
## Krasnosillya - Perstunian                             28 6.799e-01
## Krasnosillya - Pitted Ware (Type A)                   48 8.025e-07
## Krasnosillya - Podolian                               38 1.131e-01
## Krasnosillya - Vyshegorian                            32 6.065e-02
## Krasnosillya - Wolkushian                             46 8.018e-01
## Perstunian - Pitted Ware (Type A)                     23 2.153e-04
## Perstunian - Podolian                                 13 2.692e-01
## Perstunian - Vyshegorian                               7 6.123e-01
## Perstunian - Wolkushian                               21 6.253e-01
## Pitted Ware (Type A) - Podolian                       33 2.129e-05
## Pitted Ware (Type A) - Vyshegorian                    27 8.007e-05
## Pitted Ware (Type A) - Wolkushian                     41 5.989e-05
## Podolian - Vyshegorian                                17 4.268e-01
## Podolian - Wolkushian                                 31 1.617e-01
## Vyshegorian - Wolkushian                              25 7.926e-02
```

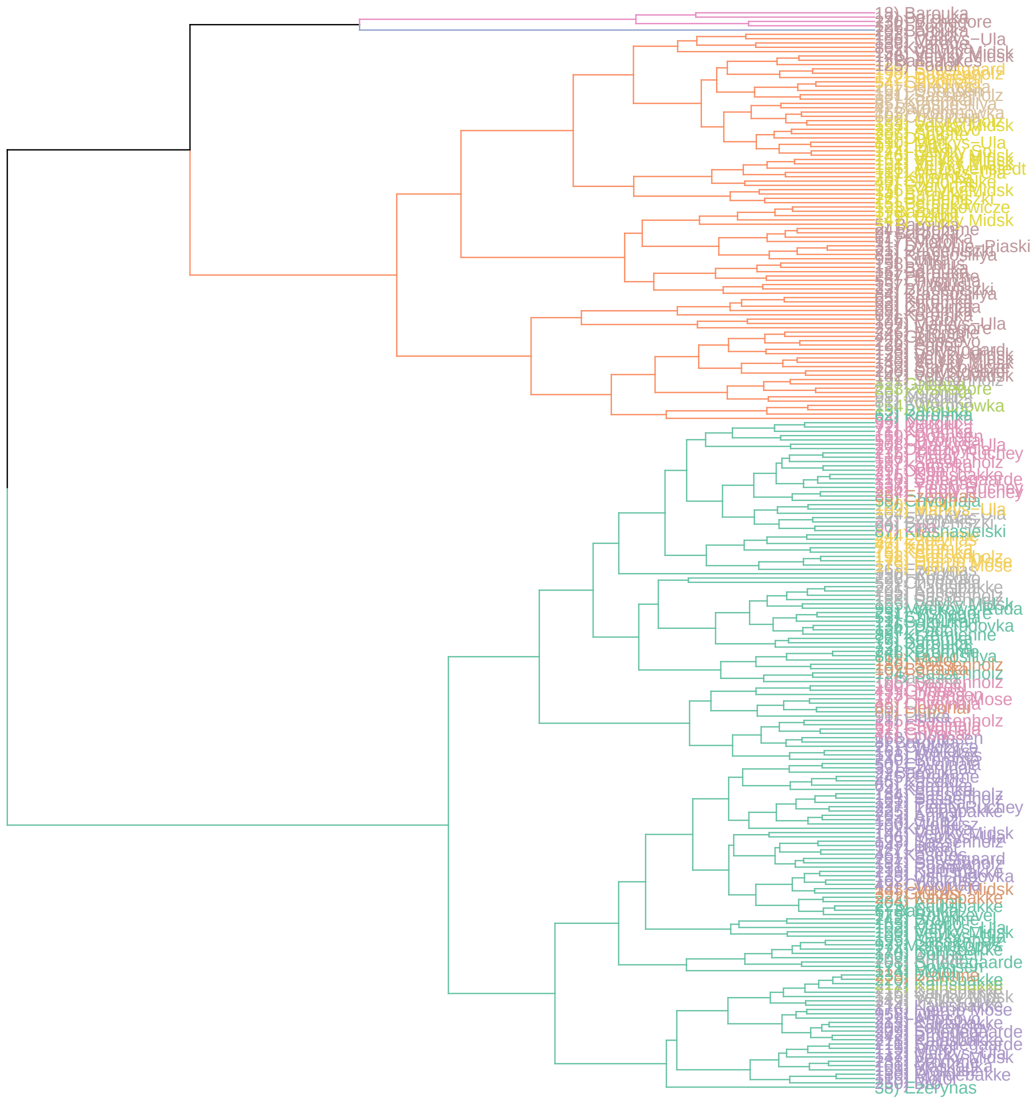
This rather large amount of information provides the p values for each combination of archaeological units and depicts level of significance in star form. In terms of analysis this data highlights, as previously the degree to which specific archaeological units can be distinguished from others in terms of their two-dimensional outline shape.

GMM Procedure 8: Hierarchical Cluster Analysis and K-Means

We can now use the elliptic Fourier coefficients and PCA data to examine, irrespective of previous groupings, how similar objects relate to one another within the overall set of examples. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar i.e. of similar shape. This can be done through two different methods in Momocs: Hierarchical Cluster Analysis (through its various subcategories), where the structure is provided, or through a K-Means analysis where partitions the shapes into k groups.

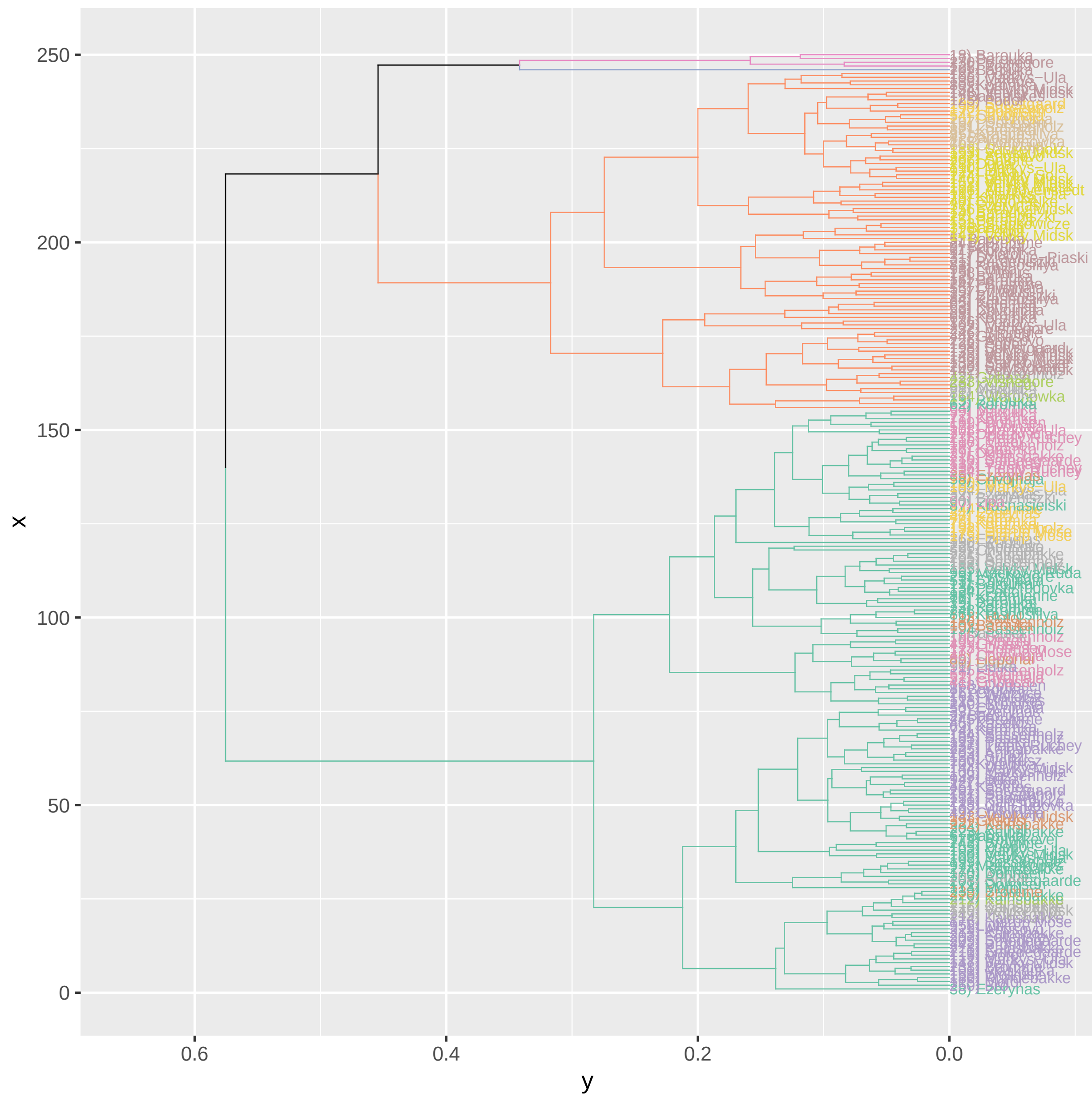
To perform a Hierarchical Cluster Analysis we can use the `Momocs::CLUST()` function, a wrapper of `stats::dist()` and `stats::hclust()`. We can specify what type of shape we wish for our tree to be using the `type` argument (horizontal as default), and the specific `hclust` (complete as default) and `dist_method` (euclidean as default). Again, we can retain the number of PCA scores we find suitable or use the elliptic Fourier coefficients.

```
CLUST(pcashape, ~Archaeological_Unit, dist_method = "euclidean", hclust_method = "complete", k = 4, reta
```



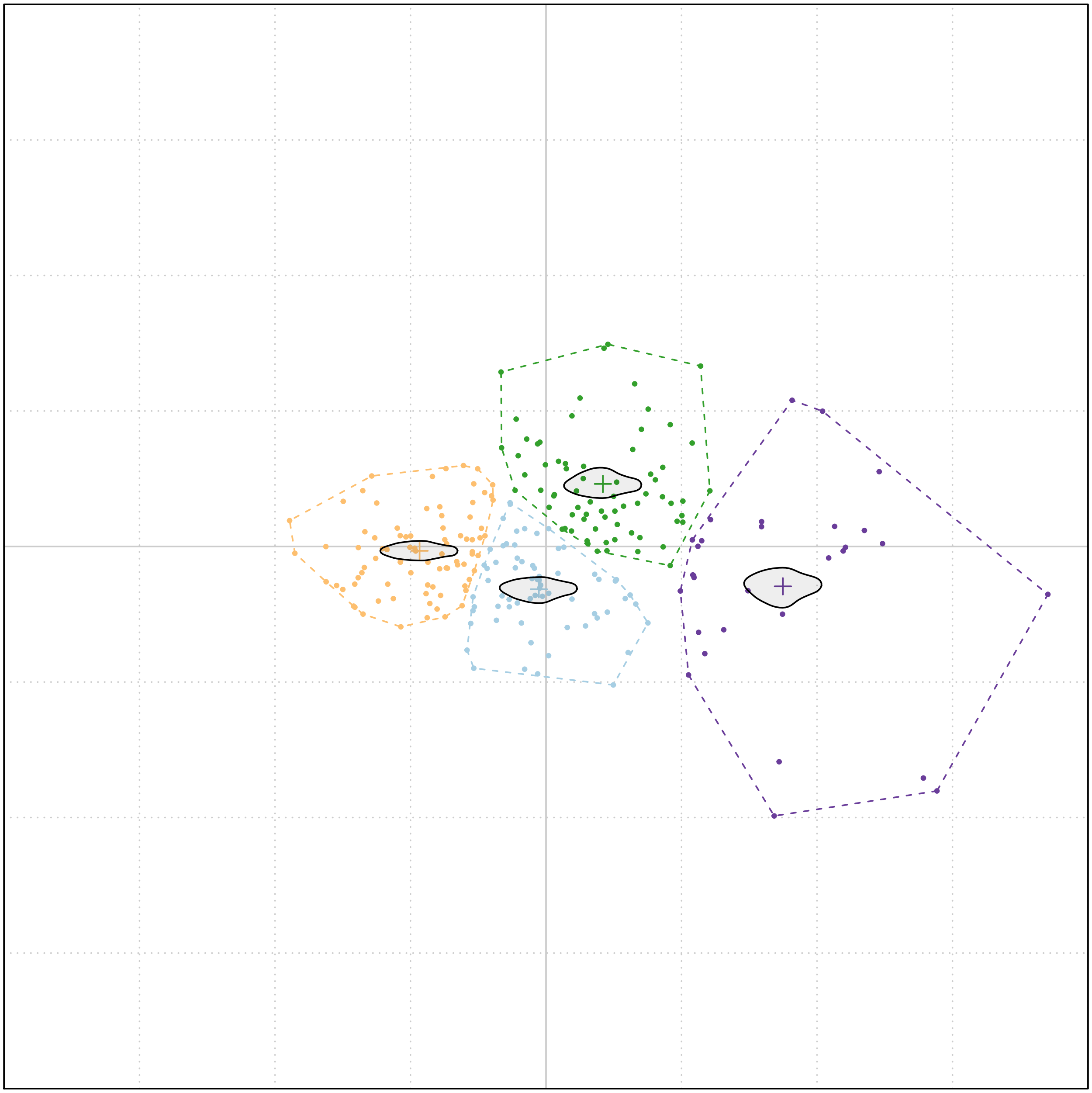
Using the `k` argument, I've also specified what the best four groupings would be. We can also modify the aesthetic further through arguments in the tidyverse.

```
CLUST(pcashape, ~Archaeological_Unit, dist_method = "euclidean", hclust_method = "complete", k = 4, reta
```



Alternatively we can use the `Momocs::KMEANS()` function to derive four groups from the data.

```
KMEANS(pcashape, centers = 4)
```



```
## K-means clustering with 4 clusters of sizes 66, 70, 84, 30
##
## Cluster means:
##            PC1          PC2
## 1 -0.005172848 -0.030952613
## 2  0.041217708  0.045308604
## 3 -0.091564235 -0.003133373
## 4  0.171585474 -0.028850883
##
## Clustering vector:
##      1) Baltasiskes          2) Barouka          3) Barouka          4) Barouka
##                   2                   3                   2                   2
##          5) Barouka          6) Barouka          7) Barouka          8) Barouka
##                   4                   3                   2                   2
```

```
##          9) Barouka         10) Barouka         11) Barouka         12) Barouka 
##                   2                   1                   1                   4 
##         13) Barouka         14) Barouka         15) Barouka         16) Barouka 
##                   2                   2                   2                   4 
##         17) Barouka         18) Barouka         19) Barouka         20) Barouka 
##                   4                   4                   1                   4 
##     21) Burdeniszki     22) Burdeniszki     23) Burdeniszki     24) Burdeniszki 
##                   2                   2                   1                   1 
##       25) Chilczyce      26) Dereznycia      27) Dereznycia            28) Duba 
##                   1                   2                   1                   2 
##            29) Duba            30) Duba 31) Dziewule-Piaski        32) Ezerynas 
##                   3                   3                   1                   1 
##        33) Ezerynas        34) Ezerynas        35) Ezerynas        36) Ezerynas 
##                   3                   1                   1                   1 
##        37) Ezerynas        38) Ezerynas          39) Glukas     40) Glyno Pelke 
##                   2                   3                   1                   2 
##         41) Gribasa         42) Gribasa         43) Gribasa         44) Gribasa 
##                   3                   2                   3                   4 
##         45) Kasetos         46) Kasetos           47) Katra       48) Chvojnaja 
##                   3                   1                   1                   3 
##       49) Chvojnaja       50) Chvojnaja       51) Chvojnaja       52) Chvojnaja 
##                   1                   3                   3                   3 
##       53) Chvojnaja       54) Chvojnaja       55) Chvojnaja       56) Chvojnaja 
##                   1                   2                   4                   4 
##       57) Chvojnaja       58) Chvojnaja       59) Chvojnaja       60) Chvojnaja 
##                   3                   1                   1                   2 
##         61) Koromka         62) Koromka         63) Koromka         64) Koromka 
##                   2                   4                   4                   4 
##         65) Koromka         66) Koromka         67) Koromka         68) Koromka 
##                   4                   2                   4                   4 
##         69) Koromka         70) Koromka         71) Koromka         72) Koromka 
##                   3                   3                   1                   2 
##         73) Koromka         74) Koromka         75) Koromka         76) Koromka 
##                   1                   3                   1                   3 
##         77) Koromka         78) Koromka         79) Koromka         80) Koromka 
##                   1                   1                   2                   2 
##         81) Koromka         82) Koromka    83) Krasnosillya    84) Krasnosillya 
##                   2                   2                   1                   1 
##    85) Krasnosillya    86) Krasnosillya   87) Krasnasielski      88) Krzemienne 
##                   2                   1                   2                   1 
##       89) Lieporiai            90) Lipa          91) Liutka          92) Liutka 
##                   3                   2                   3                   2 
##          93) Liutka          94) Liutka          95) Liutka    96) Mackowa Ruda 
##                   2                   3                   3                   1 
##     97) Marcinkonys          98) Margiu          99) Margiu         100) Margiu 
##                   3                   2                   2                   2 
##       101) Maskauka     102) Markys-Ula     103) Markys-Ula     104) Markys-Ula 
##                   3                   3                   3                   1 
##     105) Markys-Ula     106) Markys-Ula     107) Markys-Ula     108) Markys-Ula 
##                   2                   3                   4                   2 
##     109) Markys-Ula     110) Markys-Ula     111) Markys-Ula     112) Markys-Ula 
##                   1                   2                   2                   3 
##      113) Mitriskes          114) Motol          115) Motol          116) Motol 
##                   2                   1                   3                   1 
```

```
##         117) Motol          118) Motol          119) Motol          120) Motol
##                  2                   1                   3                   1
##        121) Plaska          122) Podol          123) Podol          124) Podol
##                  3                   4                   2                   1
##         125) Podol          126) Podol          127) Podol          128) Podol
##                  2                   4                   1                   4
##         129) Podol         130) Rudnia         131) Rudnya    132) Stankowicze
##                  2                   1                   3                   2
##   133) Stankowicze          134) Suraz    135) Ust-Tudovka    136) Ust-Tudovka
##                  2                   3                   1                   1
##        137) Varena         138) Varene   139) Velyky Midsk   140) Velyky Midsk
##                  3                   2                   4                   2
##  141) Velyky Midsk   142) Velyky Midsk   143) Velyky Midsk   144) Velyky Midsk
##                  2                   4                   3                   3
##  145) Velyky Midsk   146) Velyky Midsk   147) Velyky Midsk   148) Velyky Midsk
##                  4                   1                   3                   2
##  149) Velyky Midsk   150) Velyky Midsk   151) Velyky Midsk   152) Velyky Midsk
##                  3                   3                   2                   2
##  153) Velyky Midsk   154) Velyky Midsk   155) Velyky Midsk   156) Velyky Midsk
##                  2                   2                   1                   2
##       157) Vilnius        158) Vilnius        159) Wolkusz        160) Wolkusz
##                  4                   4                   3                   3
##       161) Wolkusz        162) Wolkusz      163) Woronowka      164) Woronowka
##                  2                   1                   2                   2
##         165) Zusno 166) Alt Duvenstedt        167) Dohnsen        168) Dohnsen
##                  3                   2                   2                   1
##       169) Dohnsen        170) Dohnsen        171) Dohnsen        172) Dohnsen
##                  1                   3                   3                   2
##       173) Dohnsen       174) Elemly So    175) Hjarup Mose    176) Hjarup Mose
##                  3                   2                   1                   3
##   177) Hjarup Mose    178) Hjarup Mose     179) Rolykkevej     180) Rundebakke
##                  3                   1                   3                   3
##    181) Sassenholz     182) Sassenholz     183) Sassenholz     184) Sassenholz
##                  2                   1                   3                   3
##    185) Sassenholz     186) Sassenholz     187) Sassenholz     188) Sassenholz
##                  2                   1                   3                   1
##    189) Sassenholz     190) Sassenholz     191) Sassenholz     192) Sassenholz
##                  2                   1                   2                   1
##    193) Sassenholz     194) Sassenholz     195) Sassenholz     196) Sassenholz
##                  1                   1                   3                   1
##    197) Sassenholz    198) Solystgaard    199) Solystgaard    200) Solystgaard
##                  3                   4                   2                   2
##   201) Solystgaard         202) Anholt         203) Anholt         204) Anholt
##                  3                   3                   3                   3
##        205) Anholt        206) Sotofte     207) Follenslev    208) Smedegaarde
##                  1                   2                   3                   3
##   209) Smedegaarde    210) Smedegaarde    211) Smedegaarde     212) Kainsbakke
##                  3                   3                   3                   3
##    213) Kainsbakke     214) Kainsbakke     215) Kainsbakke     216) Kainsbakke
##                  3                   3                   3                   3
##    217) Kainsbakke     218) Kainsbakke     219) Kainsbakke     220) Kainsbakke
##                  3                   3                   1                   3
##    221) Kainsbakke     222) Kainsbakke     223) Kainsbakke     224) Kainsbakke
##                  1                   3                   3                   3
```

```
##     225) Kainsbakke          226) Anosovo          227) Anosovo          228) Anosovo
##                   3                     4                     1                     1
##        229) Anosovo       230) Vishegore       231) Vishegore       232) Vishegore
##                   3                     4                     1                     4
##      233) Vishegore   234) Tieply Ruchey   235) Tieply Ruchey   236) Tieply Ruchey
##                   2                     1                     3                     1
##  237) Tieply Ruchey          238) Bromme          239) Bromme          240) Bromme
##                   3                     3                     3                     2
##         241) Bromme          242) Bromme          243) Bromme          244) Bromme
##                   2                     3                     1                     1
##         245) Bromme          246) Bromme          247) Bromme          248) Bromme
##                   3                     4                     4                     1
##    249) Trollesgave             250) Bro
##                   3                     3
##
## Within cluster sum of squares by cluster:
## [1] 0.1297335 0.1942442 0.1765680 0.2787359
##  (between_SS / total_SS =  71.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

If more computationally-intensive tree-building exercises were investigated we could explore the principal components through **maxiumum likelihood** in the `RPhylip` package (this requires the Phylip software to be installed on a computer already). All these trees can also be imported into the `ggtree` for full customisation, or analysed for their structures (phylogenetic or otherwise) in the `ape` package.

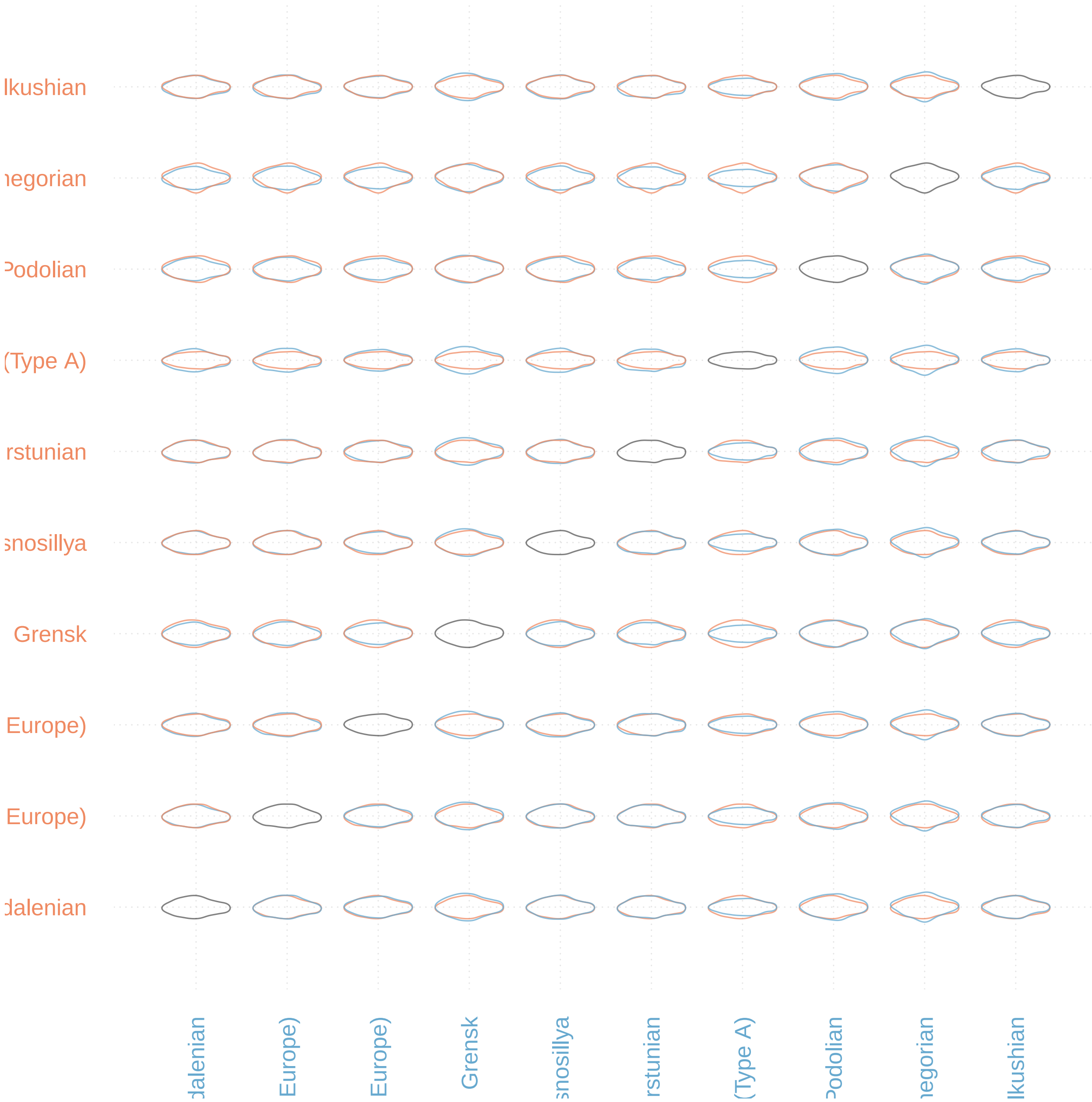
GMM Procedure 9: Constructing Mean Shapes

If we wish, we can retrieve mean shapes for a provided factor (e.g. "Archaeological_Unit"), using the elliptic Fourier coefficients or PCA scores. This is done through the `Momocs::MSHAPES()` function with the object first being made.

```
meanshapes <- MSHAPES(efashape, ~Archaeological_Unit)
```

The `Momocs::plot_MSHAPES()` function is particularly useful for displaying the mean shapes for all the archaeological units and the visualisation of different configurations of mean shapes.

```
plot_MSHAPES(meanshapes, size = 0.75)
```



GMM Procedure 10: Further Work: Incorporating Size...

In this example we have so far only examined shape, however we still have size data (as all images have a scale). From this we can extract various different measures including length or symmetry through the `Momocs::coo_length()` option (noting that the converted value is pixels!).

Centroid size is perhaps the best measure of size, incorporating the distance from all points of interest in relation to the shape. This can be extracted from the original shape data, using the `Momocs::coo_centsize()` function. We can then take this data and the principal component scores, and merge them into one database. There are a variety of ways this can be done, this is just one example.

```
centroidsize <- as_tibble(coo_centsize(shape))
centroidsize <- rename(centroidsize, cs = "value")
pcascores <- as_tibble(pcashape$x)
databasedata <- cbind(database,centroidsize, pcascores)
```

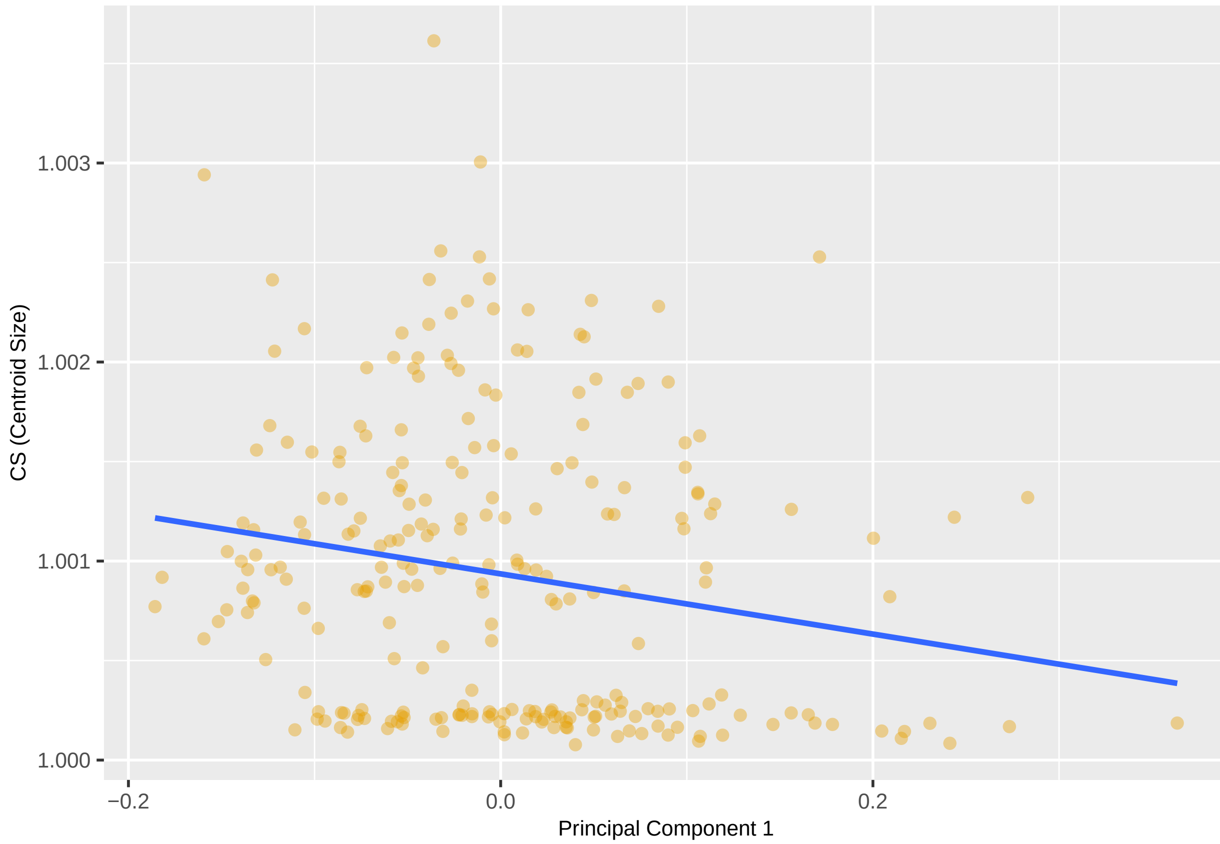
```r
head(databasedata)
```

```
##   ID        Site     Context Longitude Latitude   Country Archaeological_Unit
## 1  1 Baltašiškes Baltašiškes     23.99    54.03 Lithuania  Baltic Magdalenian
## 2  2     Barouka     Barouka     30.26    53.51   Belarus              Grensk
## 3  3     Barouka     Barouka     30.26    53.51   Belarus              Grensk
## 4  4     Barouka     Barouka     30.26    53.51   Belarus              Grensk
## 5  5     Barouka     Barouka     30.26    53.51   Belarus              Grensk
## 6  6     Barouka     Barouka     30.26    53.51   Belarus              Grensk
##   File_Name             Reference                                Notes
## 1     Bal.1 Rimantiene 1971; 1974                                 <NA>
## 2     Bor.1          Kolasau 2018 (Krasnosillya by Zalyzniak (Borovka))
## 3     Bor.3          Kolasau 2018 (Krasnosillya by Zalyzniak (Borovka))
## 4     Bor.4          Kolasau 2018 (Krasnosillya by Zalyzniak (Borovka))
## 5     Bor.5          Kolasau 2018 (Krasnosillya by Zalyzniak (Borovka))
## 6     Bor.6          Kolasau 2018 (Krasnosillya by Zalyzniak (Borovka))
##         cs         PC1           PC2          PC3         PC4           PC5
## 1 1.000807  0.02723891  0.0580419170 -0.018821385  0.01075311 -0.0362432286
## 2 1.000206 -0.09857714 -0.0005637827  0.048457316 -0.00511936  0.0053152139
## 3 1.000258  0.07921519  0.0482961705  0.035729991  0.01293937 -0.0002026685
## 4 1.000276  0.05615152  0.0292043739 -0.006578647  0.01480096  0.0095477862
## 5 1.000237  0.15608400  0.0143529019  0.040597347 -0.02173636  0.0343686118
## 6 1.000253 -0.07456266  0.0133997919 -0.028260524 -0.02272342  0.0165848492
##           PC6          PC7          PC8           PC9         PC10         PC11
## 1  0.04039065  0.002221293 -0.003887411  0.0141624637  0.011570382 -0.001929899
## 2 -0.01360586 -0.001471208  0.006836667 -0.0219516587 -0.008226835 -0.001947317
## 3 -0.02995656  0.027029617  0.029296167  0.0112590246  0.004525287 -0.006822262
## 4  0.03045869  0.030115081  0.001359413 -0.0004625243  0.022532871  0.013166032
## 5 -0.05445377  0.006164025  0.007580372 -0.0002689392 -0.012458161  0.001668338
## 6 -0.01298230 -0.008691961  0.009880353  0.0067560334  0.003571735  0.020928347
##           PC12         PC13         PC14         PC15          PC16
## 1 -0.012675160  0.008215481 -0.018238173 -0.004578046 -0.0046666536
## 2  0.006375108 -0.004841490  0.001690275  0.007453436 -0.0025290329
## 3  0.010398928  0.005558982  0.009781070 -0.004088408  0.0001736873
## 4  0.001044667 -0.007329051  0.009168578 -0.009269989  0.0032797327
## 5  0.009341902  0.010488130  0.006770371 -0.004617734  0.0034287629
## 6 -0.005352459 -0.004258540 -0.009235560 -0.005278896  0.0035957133
##           PC17          PC18         PC19          PC20          PC21
## 1 -0.001487122 -8.519330e-04 -0.006141183  0.0025277633  0.0057051817
## 2 -0.004681512 -1.978794e-03 -0.005500646  0.0032431352 -0.0026323367
## 3 -0.003117830 -1.777459e-03  0.001190171 -0.0003838019 -0.0002687512
## 4  0.003038680 -8.907413e-05  0.001670598 -0.0048705880 -0.0007977494
## 5 -0.003071278 -6.090379e-03 -0.002698053  0.0005167888 -0.0038296692
## 6  0.001855152  2.056026e-03  0.003133939  0.0082911548  0.0029166073
##           PC22          PC23          PC24         PC25          PC26
## 1 -0.002199005  0.0031339125  0.0005578041 0.0057015579 -0.0000918038
## 2  0.002709378  0.0006669588  0.0011875057 0.0015199619 -0.0016203800
## 3 -0.003902373  0.0066457039 -0.0007067034 0.0015648038 -0.0023431863
## 4  0.001760109 -0.0005884933  0.0045755061 0.0030173512 -0.0034482293
## 5 -0.011159395  0.0008422671  0.0021378488 0.0046062839 -0.0024967472
## 6  0.002358739  0.0051735558  0.0042347471 0.0008474177  0.0031718262
##            PC27         PC28          PC29          PC30         PC31
## 1 -0.0071442912 0.0001140081 -1.975473e-03  0.0028551797 -0.002676492
```

```
## 2 -0.0008167792 0.0008976529  3.809636e-05  0.0010231558 -0.001679553
## 3 -0.0011043431 0.0006605308 -4.871148e-03  0.0038723982  0.006917407
## 4  0.0039970805 0.0021537871 -2.660179e-03  0.0009938206  0.001471461
## 5 -0.0072786490 0.0010303600 -4.185127e-04 -0.0013743188  0.003132933
## 6  0.0006651295 0.0032590344 -1.864137e-04 -0.0008202582  0.001806728
##            PC32          PC33          PC34          PC35          PC36
## 1 -0.0003583197 -0.0013943507 -0.0035560834  0.0011660315  1.413841e-03
## 2  0.0007690033 -0.0017011470  0.0014070157  0.0013656916  6.617025e-05
## 3 -0.0005061710  0.0006146251  0.0011405345  0.0005078773  7.766481e-04
## 4  0.0024797067 -0.0003838746  0.0013412707 -0.0010007059  2.011609e-03
## 5 -0.0049452251  0.0014702001 -0.0032072038  0.0004185099  3.483295e-04
## 6 -0.0014203187  0.0024718622  0.0009630608  0.0013234195 -3.258149e-04
##            PC37          PC38          PC39          PC40          PC41
## 1  7.546544e-04 -0.0031047438 -1.617456e-03  2.178502e-03  1.554697e-03
## 2  1.987449e-04  0.0011389019  1.250215e-03  1.028497e-03  5.193192e-06
## 3  2.977223e-04  0.0004859952  6.423081e-04 -1.728807e-03 -6.067506e-04
## 4  5.702769e-04  0.0006614252  5.309941e-05 -2.063167e-04 -6.627334e-04
## 5  8.492301e-05  0.0013030088  6.746118e-04 -3.132884e-05 -7.200236e-04
## 6 -3.118316e-04 -0.0021313201 -1.226630e-03  1.109502e-03 -3.353373e-04
##            PC42          PC43          PC44
## 1 -7.011628e-17 -5.429479e-18  1.541621e-17
## 2 -1.060826e-16  1.264940e-17 -1.332717e-17
## 3  1.778452e-16 -1.660799e-17  1.308076e-17
## 4 -2.005661e-16 -1.539559e-17  1.441835e-18
## 5  9.469562e-17 -2.072481e-17  1.801843e-17
## 6 -8.611479e-17  3.961242e-18 -1.640800e-17
```

We can now explore these through regression and correlation based analyses. For example, using the ggplot functions we can create a scatter plot and add a regression line (these functions will be detailed by Prof. Ben Marwick in a forthcoming workshop).

```
ggplot(databasedata, aes(PC1, cs)) + geom_point(size = 2, pch = 16, alpha = 0.4, colour = "#E69F00", fil
```



We can then perform a correlation (and test) using the `cor` and `cor.test` functions:

```
cor(databasedata$PC1, databasedata$cs)
```

```
## [1] -0.1913964
```

```
cor.test(databasedata$PC1, databasedata$cs)
```

```
##
##  Pearson's product-moment correlation
##
## data:  databasedata$PC1 and databasedata$cs
## t = -3.0709, df = 248, p-value = 0.002372
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.30814620 -0.06896712
## sample estimates:
##        cor
## -0.1913964
```

Concluding Remarks

This workshop was designed to highlight how geometric morphometrics (outline analysis) can be examined for archaeological material in the R Environment, from data importing to visualisation and analysis. It is worth stressing that Momocs is only one of a number of packages in the R Environment, and the methods showcased here are only one way (and one style) of conducting GMM. Landmark analysis can be incorporated into Momocs but there are a number of functions in `Geomorph` which are particularly impressive and powerful. Similarly, a number of other packages have been referenced throughout this workflow. Only through exploring the packages and their functions will you be able understand what workflow works best for your research question and process.

If there are any questions please feel free to contact me: C.S.Hoggard@soton.ac.uk

References

For literature pertaining to outline analysis see:
* Claude, J. (2008). *Morphometrics with R.* Springer Publishing.
* Bonhomme, V., Picq, S., Gaucherel, C., & Claude, J. (2014). Momocs: Outline analysis using R. *Journal of Statistical Software*, 56: 1–24.
* Caple, J., Byrd, J., & Stephan, C. N. (2017). Elliptical Fourier analysis: Fundamentals, applications, and value for forensic anthropology. *International Journal of Legal Medicine*, 131 (6): 1675–1690.
* Ferson, S., Rohlf, F. J., & Koehn, R. K. (1985). Measuring shape variation of two-dimensional outlines. *Systematic Zoology*, 34 (1): 59–68.
* Kuhl, F. P., & Giardina, C. R. (1982). Elliptic Fourier features of a closed contour. *Computer Graphics and Image Processing*, 18 (3): 236–258.
* Yoshioka, Y. (2004). Analysis of petal shape variation of Primula sieboldii by elliptic fourier descriptors and principal component analysis. *Annals of Botany*, 94 (5), 657–664.
* Zahn, C. T., & Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on Computers*, C-21 (3): 269–281.

For literature pertaining to GMM by the author (including code and data) see:
* Hoggard, C.S., Lauridsen, L. and Witte, K.B. (2019). The Potential of Geometric Morphometrics for Danish Archaeology: Two Case Studies. *Arkaeologisk Forum*, 40: 30-42. (http://www.archaeology.dk/16738/Nr.%2040%20-%202019). OSF: https://osf.io/en5d2/./
* Hoggard, C.S., McNabb, J. and Cole, J.N. (2019). The application of elliptic Fourier analysis in understanding biface shape and symmetry through the British Acheulean. *Journal of Paleolithic Archaeology*, 2 (2): 115-133. (https://doi.org/10.1007/s41982-019-00024-6). OSF: https://osf.io/td92j/./
* Ivanovaite, L., Swertka, K., Hoggard, C.S., Sauer, F. and Riede, F. (2020). All these fantastic cultures? Research history and regionalisation in the Late Palaeolithic tanged point cultures of Eastern Europe. *European Journal of Archaeology.* (https://doi.org/10.1017/eaa.2019.59). OSF: https://osf.io/agrwb/./
* Vestergaard, C. and Hoggard, C.S. (2019). A Novel Geometric Morphometric (GMM) Application to the Study of Bronze Age Tutuli. *Danish Journal of Archaeology*, 8: 5-28. (https://tidsskrift.dk/dja/article/view/112494/164318). OSF: https://osf.io/fcp43/./
* Riede, F., Hoggard, C.S. and Shennan, S. (2019). Reconciling material cultures in archaeology with genetic data requires robust cultural evolutionary taxonomies. *Nature: Palgrave Communications*, 5 (1): 55. (https://doi.org/10.1057/s41599-019-0260-7). OSF: https://osf.io/vtdf2/.