Geometric morphometrics

Outline analyses

Manuel F. G. Weinkauf

26–27 August 2022

Contents

1 Introduction

Outlines are characterized by a relatively complicated method of data re-description to make the datapoints comparable with each other. Their advantage is, once these re-description has been performed, they constitute a normal multivariate dataset that can be easily analysed using standard methods.

Here, we will have a look at some useful analyses that can be performed with outlines, using our belemnite armhooks as an example to continue working with.

2 Setting up the R session

For outline analyses, we will mainly use the R-package 'Momocs'. This package is in constant development and the only R-package with significant outline analysis functionality. It includes plenty of functions, which can be overwhelming, and unfortunately the documentation is lacking details in some places and it still has quite some bugs. But overall, it is growing quickly and is very useful for both outline and landmark analyses.

Beside, we will use some of my own code included in the 'MorphoFiles_Function.r' and 'OutlineAnalysis_Functions.r' source files.

```
setwd("C:/R_Data/Erlangen_Morphometrics/Session3_OutlineDataAnalysis")
library(Momocs)
library(stringr)
library(MASS)
library(robust)
library(vegan)
```

```
source("MorphoFiles_Function.r")
source("OutlineAnalysis_Functions.r")
```

3 Reading and preparing the dataset

We will prepare the data according to the guidelines we worked out in the last exercise, using 7 harmonics for the EFA calculation.

```
#Read data
Belemnite.Full<-Read.NTS("Belemnite_SmoothedOutline.nts")

#Separate dataset into replicattions 1 and 2
Specimens<-unlist(dimnames(Belemnite.Full)[3])
Belemnite.R1<-Belemnite.Full[,,str_detect(Specimens, "R1")]
Belemnite.R2<-Belemnite.Full[,,str_detect(Specimens, "R2")]

#Calculating EFA (Replication 1 only)
EFA<-list()
Spec.Names<-strsplit(dimnames(Belemnite.R1)[[3]], split=".", fixed=TRUE)
for (i in 1:dim(Belemnite.R1)[3]) {
  EFA[[i]]<-NEF(Belemnite.R1[,,i], Harmonics=7)
  names(EFA)[i]<-Spec.Names[[i]][1]
}
```

We also have covariates for the belemnite hooks available that per hook contain data about the hook size and the distance of the hook to the mouth of the animal (as the hooks were preserved as an entire arm crown). These data are available in 'Belemnite_Metadata.txt'.

```
Belemnite.Covariates<-read.table("Belemnite_Metadata.txt", header=TRUE, sep="\t")
colnames(Belemnite.Covariates)<-c("Specimen", "Hook.name", "Hook.number",
                                  "CrossSection.R1.mm", "CrossSection.R2.mm",
                                  "Distance.to.mouth.mm")
```

4 Exploratory data analysis

Exploratory data analysis includes methods that are not strictly speaking statistics (i.e. they do not provide a p-value for evaluation of a hypothesis) but can help shed some light on the data.

4.1 Principal component analysis

Principal component analysis (PCA) helps detecting grouping structures in a dataset without prior knowledge of any logical grouping structure. **Caution**: If you detect apparent groups using a PCA, you cannot use the same data to then confirm this grouping statistically, as this would be a Texas Sharpshooter fallacy.

A PCA can be calculated on either the variance–covariance matrix or the correlation matrix of the data. The latter is useful when the different components in the multivariate dataset are measured on very different scales or with very different errors. For outline data, the PCA is nearly always calculated on the variance–covariance matrix. The 'princomp()' function in R is one potential method to calculate a PCA.

For this, we first need to consolidate the data in 'EFA' into a matrix, with harmonics in columns and specimens in rows.

```
#Set up matrix representation of harmonics
EFA.Mat<-matrix(NA, length(EFA), length(EFA[[1]][[1]])*4)
colnames(EFA.Mat)<-c(paste(rep("A", length(EFA[[1]][[1]])),
```

```
                          1:length(EFA[[1]][[1]]), sep=""),
                    paste(rep("B", length(EFA[[1]][[1]])),
                          1:length(EFA[[1]][[1]]), sep=""),
                    paste(rep("C", length(EFA[[1]][[1]])),
                          1:length(EFA[[1]][[1]]), sep=""),
                    paste(rep("D", length(EFA[[1]][[1]])),
                          1:length(EFA[[1]][[1]]), sep=""))
rownames(EFA.Mat)<-names(EFA)

#Populate matrix with data
for (i in 1:length(EFA)) {
  Temp<-EFA[[i]]
  EFA.Mat[i,]<-c(Temp[[1]], Temp[[2]], Temp[[3]], Temp[[4]])
}
```

When you use normalized elliptic Fourier analyses, that orients the specimens based on the first harmonic. If you do this, you would normally want to not also use the first harmonic for any ordination solution, as it is 'consumed' by the alignment process. Practically, only the first three components of Harmonic 1 are normally removed.

```
EFA.PCA<-EFA.Mat[,-which(colnames(EFA.Mat) %in% c("A1", "B1", "C1"))]
```

With these data, we can now calculate a PCA using basic R-functionality (Fig. 1).

```
#Calculate PCA
PCA<-princomp(EFA.PCA)

#Calculate proportion of variance explained per axis
PoV<-(PCA$sdev^2/sum(PCA$sdev^2))*100

#Color-code by specimen
cols<-hcl.colors(n=length(unique(Belemnite.Covariates[,"Specimen"])),
                 palette="viridis", alpha=0.8)
Col.vec<-cols[Belemnite.Covariates[,"Specimen"]]

#Plot results
plot(PCA$scores[,1], PCA$scores[,2], type="p",
     xlab=paste("PC 1 (", round(PoV[1], digits=0), "%)", sep=""),
     ylab=paste("PC 2 (", round(PoV[2], digits=0), "%)", sep=""), pch=16,
     cex=1.5, col=Col.vec)
legend("topleft", pch=16, col=cols, legend=c("Specimen 1", "Specimen 2", "Specimen 3"))
```

The data from the different specimens are well distributed. Do you see any obvious groupings?

EXERCISE 1: Experiment with the PCA, try the other replicate, different numbers of harmonics, etc.

4.1.1 Shape reconstruction in PCA

The neat thing about a PCA is that you can reconstruct shapes in the morphospace for any point, not only for points for which you have actual data. This is because you can calculate the original harmonics based on the mean shape of the organism and the sum of the products of scores and loadings: $H = shape_{mean} + \sum_{i=1}^{n}(score_i \times loading_i)$.

You can for instance calculate how a hook at the right side of the morphospace looks like (Fig. 2):

```
#Calculate mean shape
mshcoef<-apply(EFA.Mat[,], 2, mean)
```

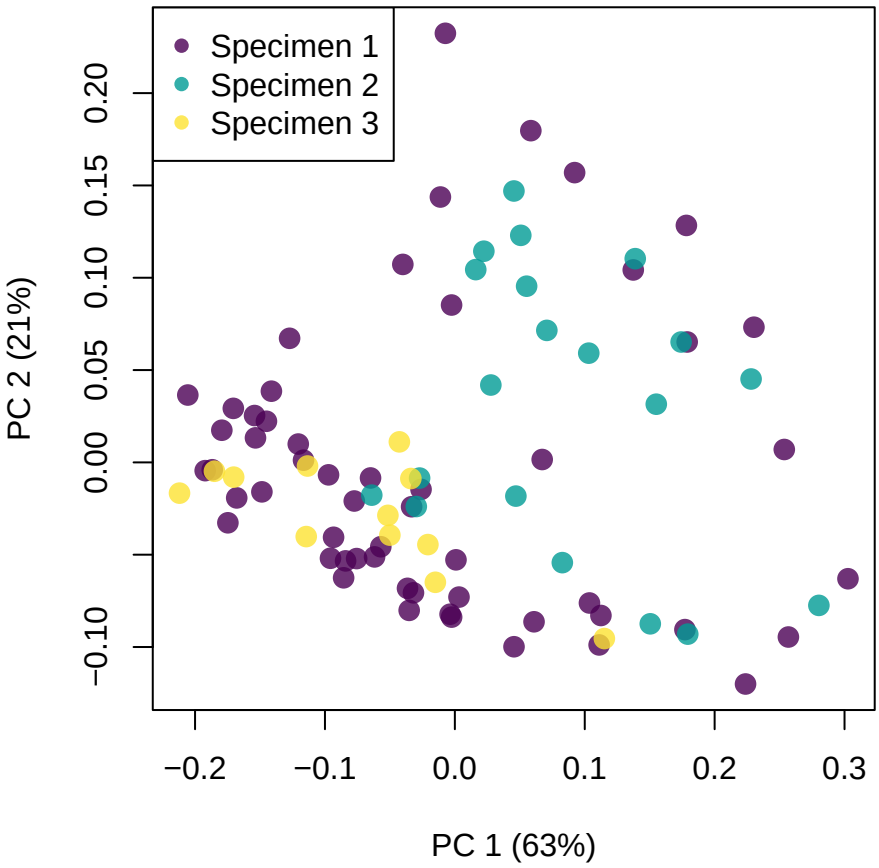

Figure 1: PCA of belemnite arm hooks.

```
#Extract eigenvalues
ev<-PCA$loadings

#Calculate shape at the right extreme of the PCA morphospace
Mx1<-mshcoef+max(PCA$score[,1])*
  c(0, ev[1:6,1], 0, ev[7:12,1], 0, ev[13:18,1], ev[19:25,1])
#Note the insertion of 0 to compensate that we eliminated the 'A1', 'B1',
#and 'C1' components for the PCA calculation

#Reconstruct the shape
PC1Max<-iefourier(an=Mx1[1:7], bn=Mx1[8:14], cn=Mx1[15:21], dn=Mx1[22:28],
                  Harmonics=7, Points=70)
plot(PC1Max$x, PC1Max$y, pch=16, type="b", xlab="X", ylab="Y", asp=1)
```

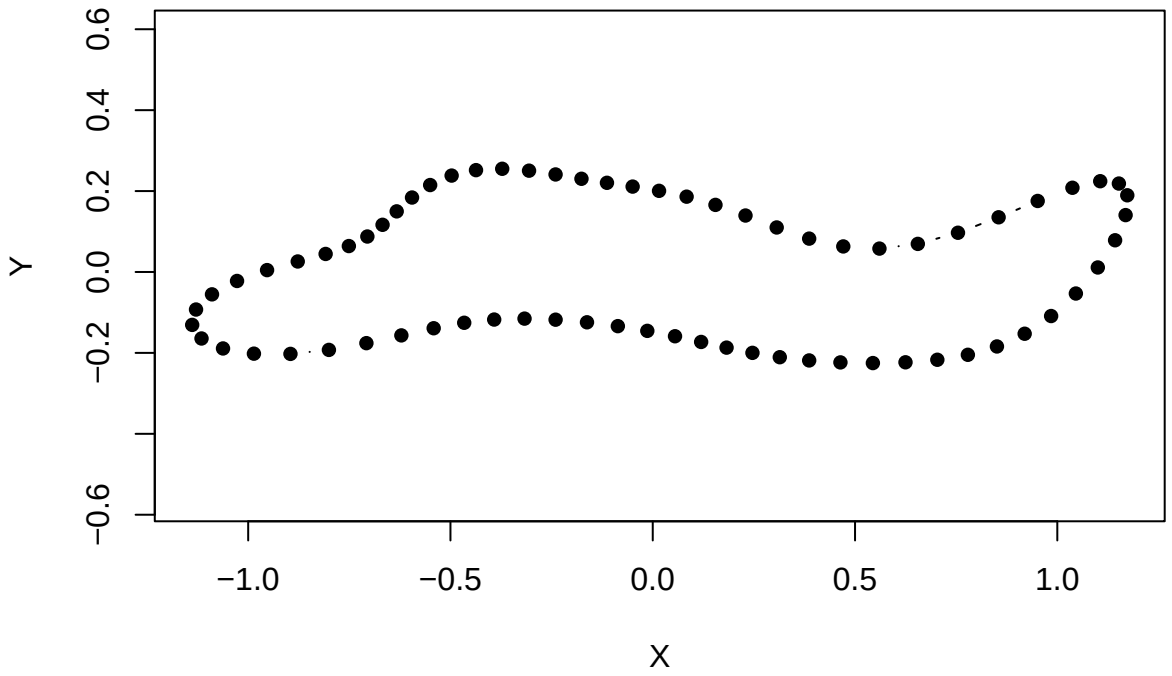

Figure 2: Reconstruction of the hook shape at the right extreme of the PCA morphospace.

With these possibilities, it is comparatively easy to reconstruct the entire morphospace, for instance in a grid as shown in Fig. 3.

EXERCISE 2: Try to reconstruct the shapes at the other extremes of the morphospace.

Thanks to the development of 'Momocs' this can now all be done a little easier using the 'PCA()'/'plot.PCA()' functions. Unfortunately, these functions clash with some in 'OutlineAnalysis_Functions.r' and as they are used internally within other 'Momocs'-functions, you cannot specify the use of either one version in the code. A solution is to load a new R-session without the 'OutlineAnalysis_Functions.r'. Check out the functions 'efourier(), 'efourier_i()', 'PCA()', and 'plot.PCA()'. Internally, 'plot.PCA()' correctly reassembles the normalized harmonic coefficients to be compatible with 'efourier_i()'.

EXERCISE 3: Another form or exploratory data analysis would be the calculation of a cluster analysis. Give it a try and don't forget that StackOverflow is an invaluable resource for programming at all stages.

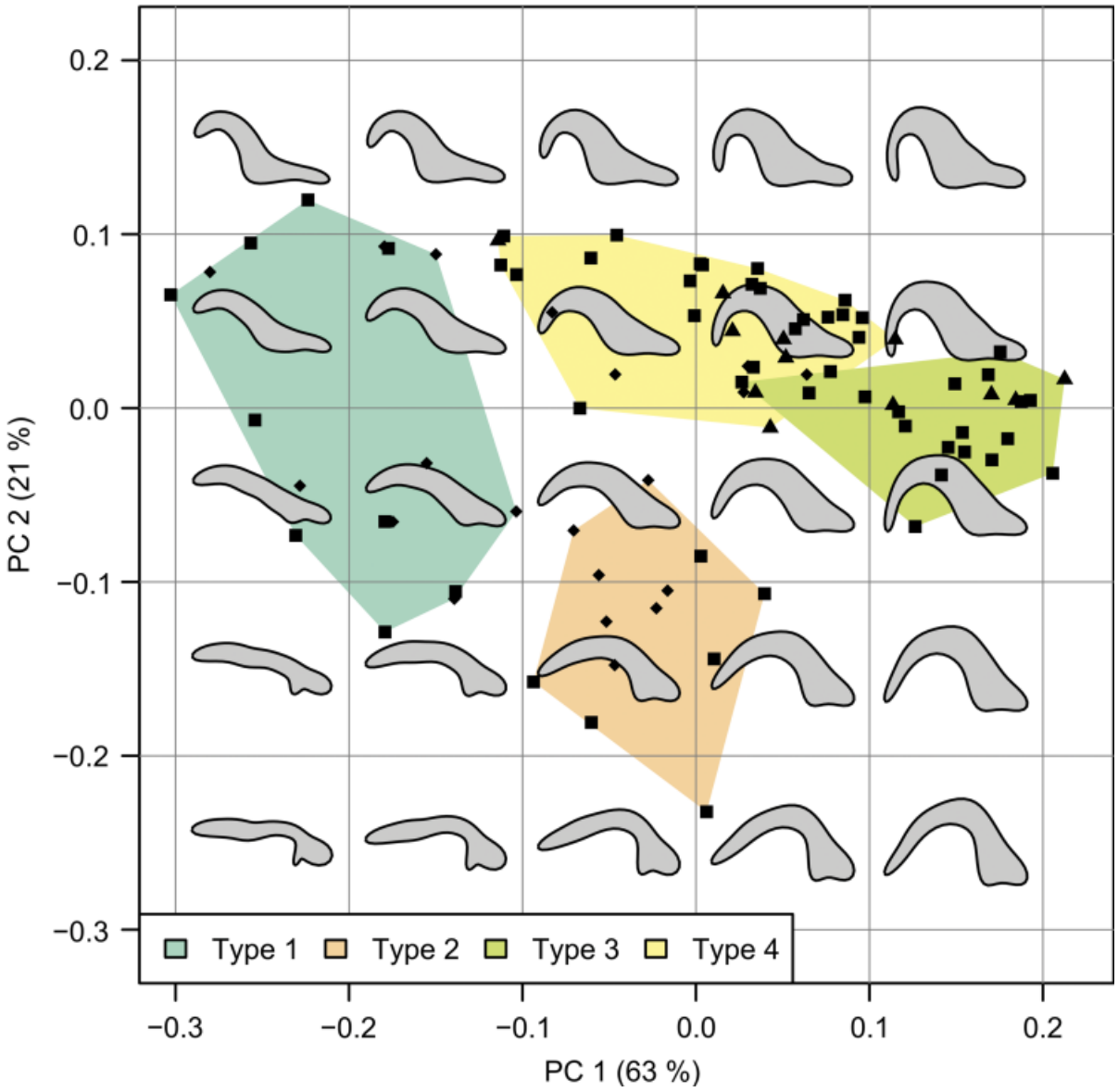

Figure 3: Reconstruction of the morphospace of belemnite hooks. From Hoffmann et al. (2017). Grasping the shape of belemnoid arm hooks—a quantitative approach. *Paleobiology* 43 (2): 304–20.

5 Hypothesis-driven data analysis

In contrast to exploratory data analysis, in hypothesis-driven data analysis you have a hypothesis and want to test it using probabilistic or other methods. Here, we will go through two examples of such analyses in morphometrics.

5.1 Linear discriminant analysis/canonical variates analysis

If groups are known *a priori* (e.g. in our case that hooks belong to three different specimens) the data can also be explicitly tested for a difference between these groups using a linear discriminant function (for two groups) or a canonical variates analysis (for more than two groups). This uses the fact that the principal component scores can be used as low-dimensional descriptive variables for the morphology (Fig. 4).

```
#Calculate CVA
Specimen<-as.factor(Belemnite.Covariates[,"Specimen"])
LDA.Data<-as.data.frame(cbind(Specimen, PCA$scores))
LDA<-lda(Specimen~., data=LDA.Data)
LDA.predict<-predict(LDA, LDA.Data)

#Calculate correct classification
LDA.crossval<-lda(Specimen~., data=LDA.Data, CV=TRUE)
Correct<-((sum(LDA.Data[,"Specimen"]==LDA.crossval$class)/(nrow(LDA.Data)))*100)

#Plot results
cols<-hcl.colors(n=length(unique(Belemnite.Covariates[,"Specimen"])),
                 palette="viridis", alpha=0.8)
Col.vec<-cols[Belemnite.Covariates[,"Specimen"]]

plot(LDA.predict$x[,1], LDA.predict$x[,2], type="p",
     xlab=paste("CV 1 (", round(PoV[1], digits=0), "%)", sep=""),
     ylab=paste("CV 2 (", round(PoV[2], digits=0), "%)", sep=""), pch=16,
     cex=1.5, col=Col.vec)
legend("topright", pch=16, col=cols, legend=c("Specimen 1", "Specimen 2", "Specimen 3"))
```

With a correct classification of only 61%, the CVA is not very convincing. We can also test the difference explicitly. An MANOVA (**M**ultivariate **AN**alysis **O**f **VA**rianaces) is a common method for that, but morphometric data are so aberrant that when possible, permuational methods should be used. This is because these methods are not that sensible to non-Gaussian data distributions and, thus, perform more reliably. The package 'vegan' offers an implementation of **PER**mutational **M**ultivariate **AN**alysis **O**f **VA**rianaces (PERMANOVA; also called NPMANOVA, **N**on-**P**arametric **M**ultivariate **AN**alysis **O**f **VA**rianaces).

```
PERM.Data<-as.data.frame(PCA$scores)
PERM<-adonis2(PERM.Data~Specimen, data=PERM.Data, method="euclidean")
```

With a p-value of $p = round(PERM[[5]][1], digits = 3)$, the morphological differences between the hooks of different specimens are nevertheless significant. It is not likely that hooks look similar across specimens.

EXERCISE 4: Try what happens when you want to conduct these analyses on other data, e.g. on the harmonics instead of the PCs.

5.2 Regression analyses

Sometimes, you are not interested in the difference between distinct (nominal) groups but in the relationship between morphology and a continuous covariate. For these purposes, regression analyses are optimal. As with MANOVA, the nature of morphometric data makes robust linear regression methods, as they were implemented in the R-package 'robust', the better choice.

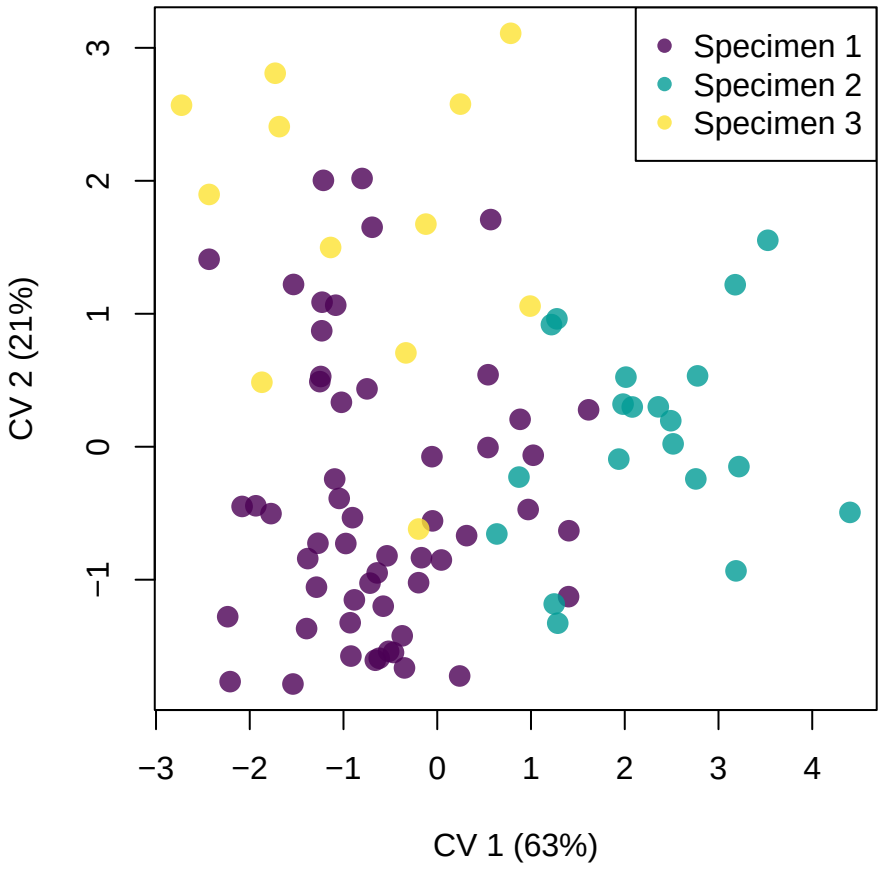

Figure 4: CVA of belemnite arm hooks separated by specimens.

We may for instance be interested if shape and size in the belemnite hooks are correlated, as this may imply an ontogenetic effect on shape.

```
#Calculate robust regression
LM.Data<-as.data.frame(cbind(Belemnite.Covariates[,"CrossSection.R1.mm"], PCA$scores))
colnames(LM.Data)[1]<-"Size"
LM<-lmRob(Comp.1~log(Size), data=LM.Data)
pval<-summary(LM)$coefficients["log(Size)","Pr(>|t|)"]

#Plot results
plot(Comp.1~log(Size), data=LM.Data, xlab="log-transformed hook size (mm)",
     ylab="Hook shape (PC1)", pch=16, col="grey50")
curve(coef(LM)[2]*x+coef(LM)[2], lwd=2, col="darkred", add=TRUE)
```

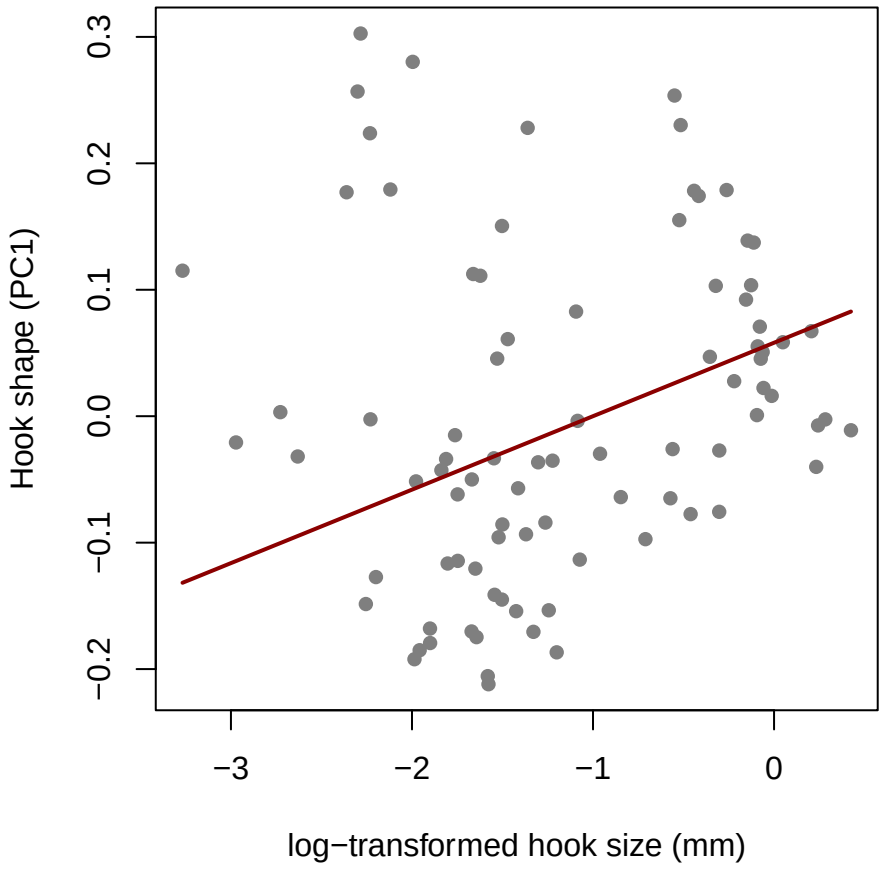

Figure 5: Robust linear regression of belemnite hook shape in dependence on size.

With a p-value of $p = 0.001$, the relationship between hook size and shape is significant. We therefore cannot rule out an ontogenetic effect on hook shape.

EXERCISE 5: Find another relationship that you can test. You can have a look at the CRAN website of 'robust' if you want to try other methods or parameters.