Assignment 2

Wenjuan Bian

2023-07-09

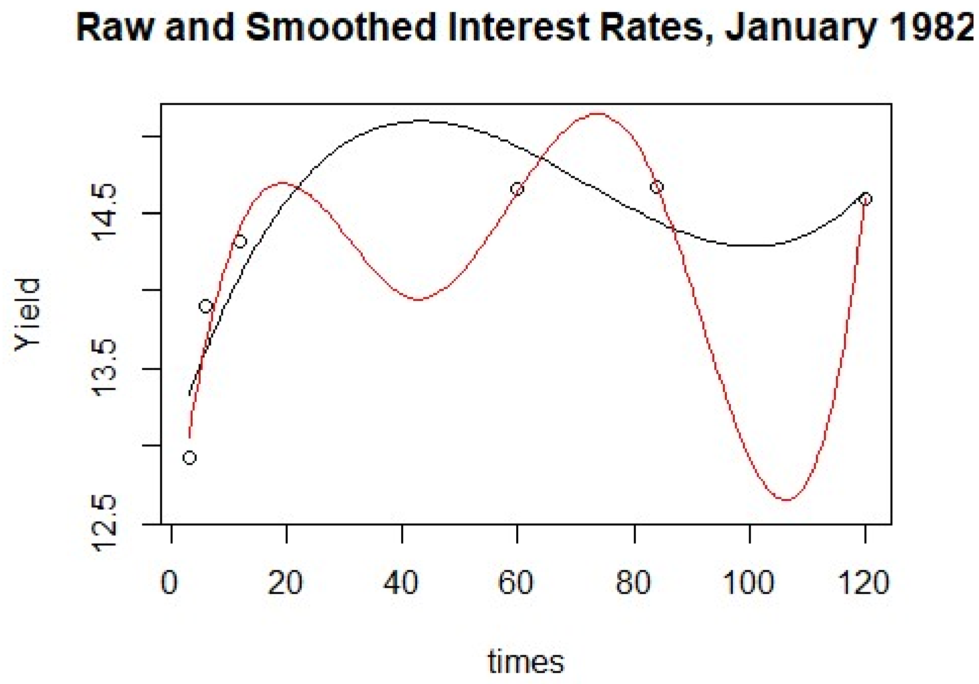
R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

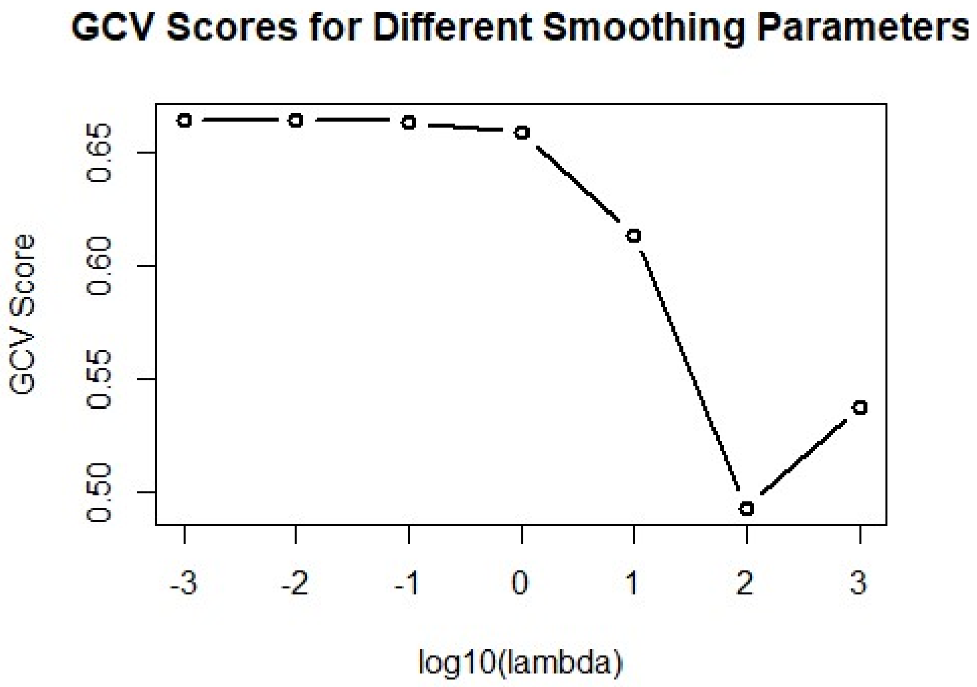
```r
library(fds)

## Warning: package 'fds' was built under R version 4.2.3

## Loading required package: rainbow

## Warning: package 'rainbow' was built under R version 4.2.3

## Loading required package: MASS

## Loading required package: pcaPP

## Warning: package 'pcaPP' was built under R version 4.2.3

## Loading required package: RCurl

## Warning: package 'RCurl' was built under R version 4.2.3

#####
# Download the R package fds and use the data set FedYieldcurve, which contains the
 monthly Federal Reserve interest rates, cf. Problem 1.2.
# (a) Smooth the interest rates (yields) in January 1982 using a B-spline basis wit
h four basis functions. Plot the raw and smoothed interest rates on one graph.
#####

library(fda)

## Warning: package 'fda' was built under R version 4.2.3

## Loading required package: splines

## Loading required package: deSolve

## Warning: package 'deSolve' was built under R version 4.2.3
```

```
##
## Attaching package: 'fda'

## The following object is masked from 'package:graphics':
##
##     matplot

# Load the FedYieldcurve data
data(FedYieldcurve)


# Define the independent variable (time in years) and dependent variable (yields)
times <- FedYieldcurve$x
yield <- FedYieldcurve$y[,1] # yields for January 1982

# Create a B-spline basis with 4 basis functions
xrange<- range(times) # get the range of times
my_basis <- create.bspline.basis(xrange, nbasis=4)

# Fit a smooth curve to the data using the B-spline basis
smoothed1 <- smooth.basis(times, yield, my_basis)


plot(times, yield, ylim = c(12.6, 15.1),ylab="Yield", main="Raw and Smoothed Intere
st Rates, January 1982")

# Add the smoothed curve to the plot
lines(smoothed1$fd)

#####
# (b) Re-fit the January 1982 yields using a penalized smoothing based on six basis
 functions (as many as data points) with with the smoothing parameter λ = 1, and th
e second derivative as the penalty operator. Add the smooth in red to the graph you
 obtained in part (a) and comment on the result.
#####

# Penalized smoothing - R

my_basis2<-create.bspline.basis(xrange, nbasis=6,norder = 4)
my_par<-fdPar(my_basis2,Lfdobj=2,lambda=1)
smoothed2 <- smooth.basis(times, yield, my_par)

# Add the penalized smoothed curve to the plot in red
lines(smoothed2$fd, col="red")
```

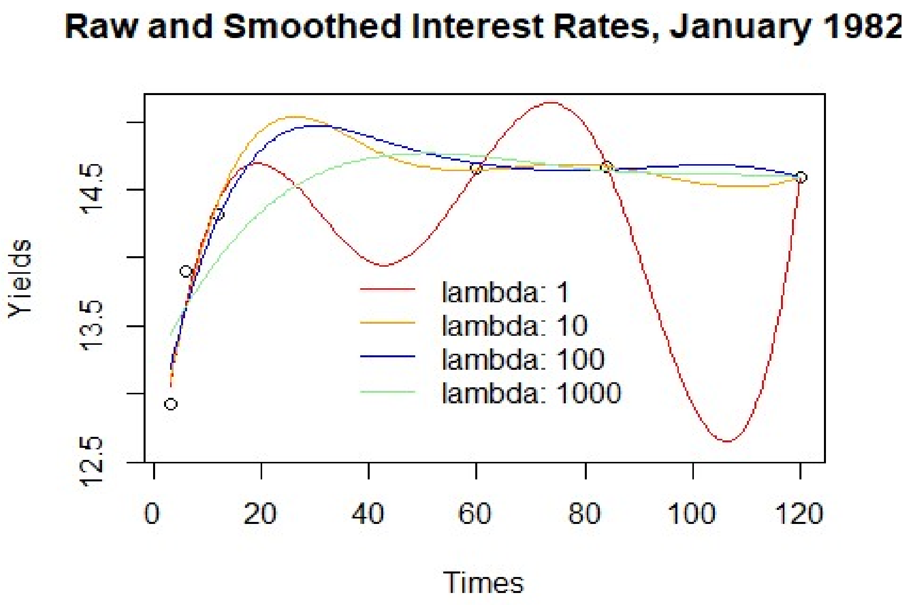


By the plot, it appears that the red curve, generated by penalized smoothing based on six basis functions with *λ = 1*, doesn't fit the data very well. Despite using as many basis functions as there are data points, the resulting curve fails to capture the underlying trend of the data effectively. On the other hand, the curve generated in part (a) seems to provide a much better fit to the data.

```
#####
# (c) Repeat part (b) with several other smoothing parameters λ. Which λ gives the
most informative smooth curve?
#####
loglam = -3:3 # This gives a range of lambda from 10^-3 to 10^3. Adjust this to you
r needs.
nlam = length(loglam)
gcvsave = rep(NA,nlam)
names(gcvsave) = loglam

for (ilam in 1:nlam) {
  lambda = 10^loglam[ilam]
  my_par <- fdPar(my_basis2,Lfdobj=2,lambda=lambda)
  smooth_result <- smooth.basis(times, yield, my_par)
  gcvsave[ilam] = sum(smooth_result$gcv)
}

# Plot GCV scores against log(lambda)
plot(loglam, gcvsave, type='b', lwd=2, xlab="log10(lambda)", ylab="GCV Score",
     main="GCV Scores for Different Smoothing Parameters")
```

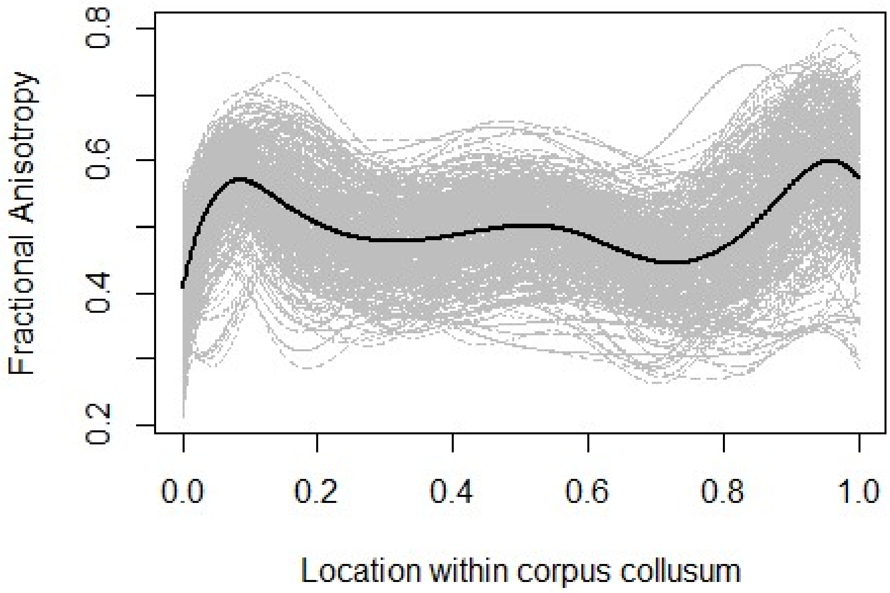


```
my_par2<-fdPar(my_basis2,Lfdobj=2,lambda=10)
my_par3<-fdPar(my_basis2,Lfdobj=2,lambda=100)
my_par4<-fdPar(my_basis2,Lfdobj=2,lambda=1000)

smoothed3 <- smooth.basis(times, yield, my_par2)
smoothed4 <- smooth.basis(times, yield, my_par3)
smoothed5 <- smooth.basis(times, yield, my_par4)

# Add the penalized smoothed curve to the plot in red
plot(times, yield, ylim = c(12.6, 15.1),xlab = "Times",ylab="Yields", main="Raw and
 Smoothed Interest Rates, January 1982")
lines(smoothed2$fd, col="red")
lines(smoothed3$fd, col="orange")
lines(smoothed4$fd, col="blue")
lines(smoothed5$fd, col="lightgreen")

legend("bottom", inset=c(0,0.1), legend=c("lambda: 1", "lambda: 10", "lambda: 100",
 "lambda: 1000"),
       col=c("red", "orange", "blue", "lightgreen"), lty=1, bty="n")
```

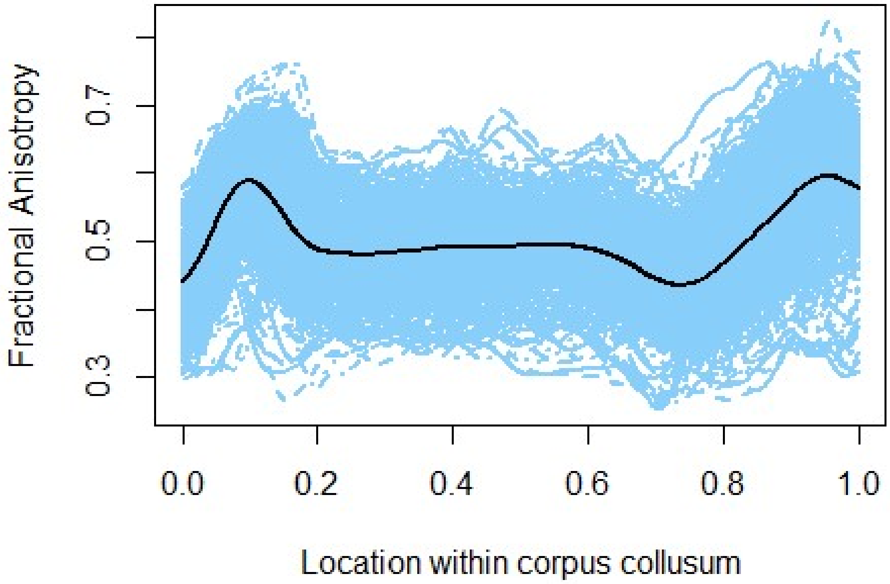


By GCV score, λ = 100 gives the most informative smooth curve.

```
#####
# 2.4 Consider the DTI data in Section 1.5.
# (a) Use penalized smoothing with 100 basis functions and GCV to convert the data to functional objects. Plot the data and the mean function. Comment on any differences with the direct splines expansion plotted in Figure 1.12.
#####

library(refund)
data("DTI", package = "refund")
data1 <- DTI
Corp<-data1$cca
drop<-unique(which(is.na(Corp),arr.ind=TRUE)[,1])
Corp<-Corp[-drop,] # Missing value
pts<-seq(0,1,length=93)
basis4<-create.bspline.basis(c(0,1),nbasis=20)
Corp.F<-Data2fd(pts,t(Corp),basis4)

# DTI - mean
plot(Corp.F,col="gray",xlab="Location within corpus collusum",ylab="Fractional Anisotropy")

## [1] "done"

Corp.F.mean<-mean.fd(Corp.F)
plot(Corp.F.mean,add=TRUE,lwd=2)
```



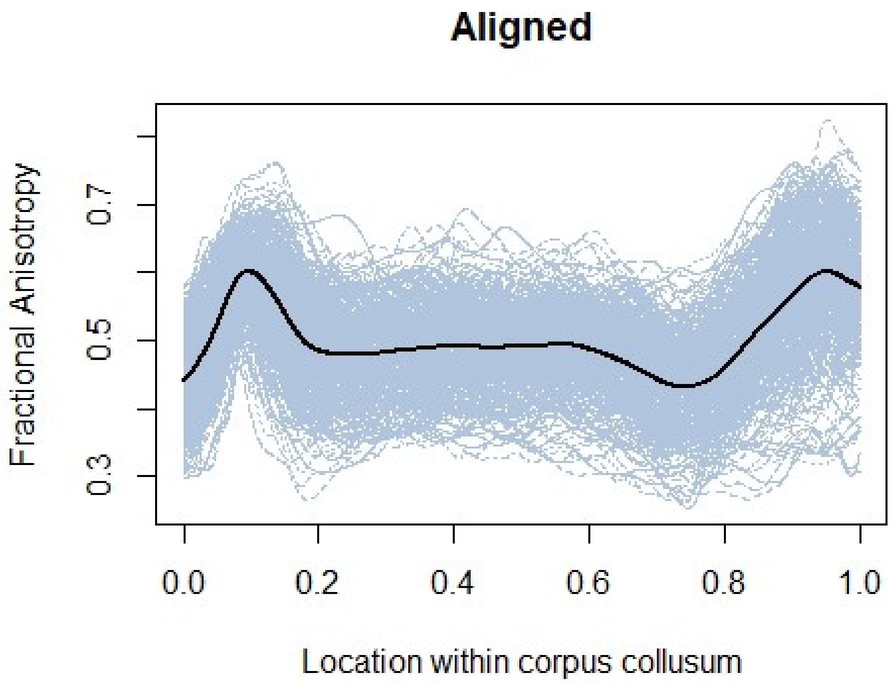
```
## [1] "done"

basis4_a <- create.bspline.basis(c(0,1),nbasis=100,norder = 6)

lambda_all<-10^(-(10:20)/2)
gcv_all<-numeric(0)
for(lambda in lambda_all){
  myPar<-fdPar(basis4_a,2,lambda)
  Corp.F<-smooth.basis(pts,t(Corp),myPar)
  gcv_all<-c(gcv_all,mean(Corp.F$gcv))
}
lambda_all[which.min(gcv_all)]

## [1] 1e-07

mypar = fdPar(basis4_a,Lfdobj=2,lambda=lambda_all[which.min(gcv_all)])
X2fd <- smooth.basis(pts,t(Corp),mypar)
plot(X2fd,col='lightskyblue',lwd=2, xlab="Location within corpus collusum",ylab="Fr
actional Anisotropy")

## [1] "done"

a_fd <- X2fd$fd

a_fd.mean<-mean.fd(a_fd)
plot(a_fd.mean,add=TRUE,lwd=2)
```



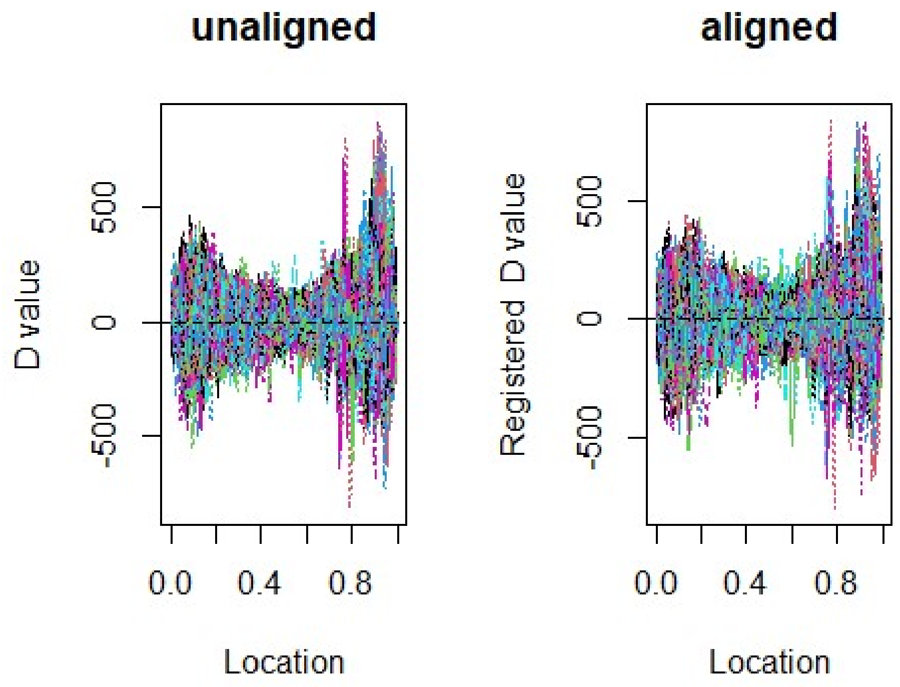
```
## [1] "done"
```

By comparing the results of the two methods, no noticeable difference has been detected.

```
#####
# (b) Use continuous registration to align the curves. Plot the resulting curves an
d mean function. Comment on any differences from the plot in (a).
#####

X.reg <- register.fd(a_fd)

X.reg.mean<-mean.fd(X.reg$regfd)

plot(X.reg$regfd,col='lightsteelblue',main='Aligned', xlab="Location within corpus
collusum",ylab="Fractional Anisotropy") # the component "regfd" contains the regist
ered functions.

## [1] "done"

plot(X.reg.mean,add=TRUE,lwd=2)
```



```
## [1] "done"

X.f.2nd.deriv <- deriv.fd(a_fd,2)
X.f.2nd.deriv.reg <- register.fd(X.f.2nd.deriv)

par(mfrow=c(1,2))
plot(X.f.2nd.deriv,main='unaligned',xlab = "Location")

## [1] "done"

plot(X.f.2nd.deriv.reg$regfd,main='aligned',xlab = "Location") # the component "reg
fd" contains the registered functions.
```



```
## [1] "done"
```

```
par(mfrow=c(1,1))

AmpPhaseDecomp(X.f.2nd.deriv,X.f.2nd.deriv.reg$regfd,X.f.2nd.deriv.reg$warpfd)$MS.p
ha # MSE_pha
```

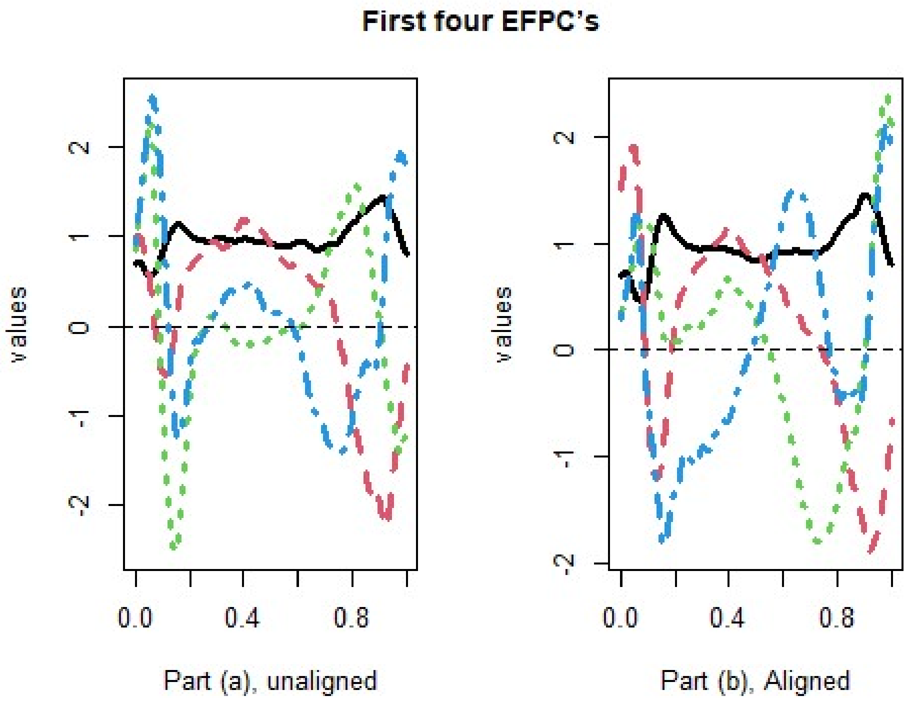
```
## [1] 354.783
```

When comparing the plot resulting from the alignment of the curves in part (b) with the original plot in part (a), there appears to be no obvious difference. This observation holds true even for the plots of the second derivatives. This suggests that the visual patterns and structures of the curves were preserved through the registration process.

However, it is noteworthy that the Mean Squared Error (MSE) of phase, MSE_pha, is calculated to be 354,783. The MSE_pha represents the part of the total variance that has been removed by the alignment process. In this case, despite the visual similarities between the aligned and unaligned plots, the sizable MSE_pha value indicates that the alignment process has indeed made substantial adjustments to the curves' phase variation.

```
#####
# (c) Carry out a PCA for both (a) and (b). Comment on any differences between FPCs
 and explained variance.
#####
par(mfrow=c(1,2), cex=0.8)
```

```
pca1 = pca.fd(X2fd$fd, nharm=4)
plot(pca1$harmonics, lwd=3, xlab="Part (a), unaligned")

## [1] "done"

pca2 = pca.fd(X.reg$regfd, nharm=4)
plot(pca2$harmonics, lwd=3, xlab="Part (b), Aligned")

## [1] "done"

par(mfrow=c(1,1))
title("First four EFPC’s", outer = TRUE, line = -2, cex.main = 0.9)
```



```
pca1$varprop

## [1] 0.63690954 0.08138867 0.06748268 0.06031304

pca2$varprop

## [1] 0.68846574 0.07162954 0.05677696 0.04197376
```

From the variance proportions (varprop) outputted from the two PCAs (principal component analysis), we can see that the explained variance for the first four principal components differs before and after the continuous registration.

For the unregistered data (W.pca1), the first four principal components explain approximately 63.7%, 8.1%, 6.7%, and 6.0% of the variance respectively.

For the registered data (W.pca2), the first four principal components explain about 68.8%, 7.2%, 5.7%, and 4.2% of the variance respectively.

We can see that after registration, PC1 explains a larger portion of the variance (68.8% vs. 63.7%). This indicates that the registration process may have reduced some of the variability in the data that was related to the phase variability among the curves.

On the other hand, the proportions of variance explained by the second, third, and fourth principal components are smaller in the registered data compared to the unregistered data. This further supports the idea that the registration process has reduced some of the phase variability in the data.

```
#####
# (d) In your opinion, was curve alignment necessary for this data? Explain.
#####
```

From the results, we can see that the explained variance of the first functional principal component increased after curve alignment, indicating that a larger proportion of the variability in the data is explained by the first FPC after alignment. This could mean that some of the variation that was previously "hidden" in phase differences has been brought into the amplitude (shape) of the curves by the alignment, making it easier to explain with a single FPC.

This result might suggest that alignment was beneficial for this dataset.