

24july2020_p2

Zach Wang

7/24/2020

1. progress summary

This week, as we discussed from last meeting, I have: - 1. selected the 1st complete sin curve of every node and align them to register at the origin. - 2. Scale the the time frame of each node to $[-1, 1]$ while time=0 remain unchanged. I also make sure that each node have the same number of rows so that fPCA will be able to apply. - 3. Apply fPCA to the new dataset of curves (1st cycle of each node), and the result is that the first two Principle Component functions covers 91.5% of total variance. - 4. repeat the process above for 2nd cycle of each node, and first two Principle component functions covered in total 93.7% of total variance.

2. read data and attach packages

3. Defined fourier smoothing functions

To study a single brain node response, specify the node number in the *node_subset* list.

```
f_fourier_smooth <- function(time_subset, data_mat, node_subset, k){  
  basis <- create.fourier.basis(c(time_subset[1],time_subset[length(time_subset)]), k)  
  fd_obj <- smooth.basis(time_subset, data_mat[time_subset,node_subset], basis)  
  smoothfd <- fd_obj$fd  
  #plot(smoothfd)  
  #title(main=paste("Fourier Basis Smoothing of node:", node_subset, ", Basis_number:",k  
  ))  
  return(fd_obj)  
}
```

4. define the function to extract periodic cycle of a single node response

5.1 single node (node=1), of the entire time frame

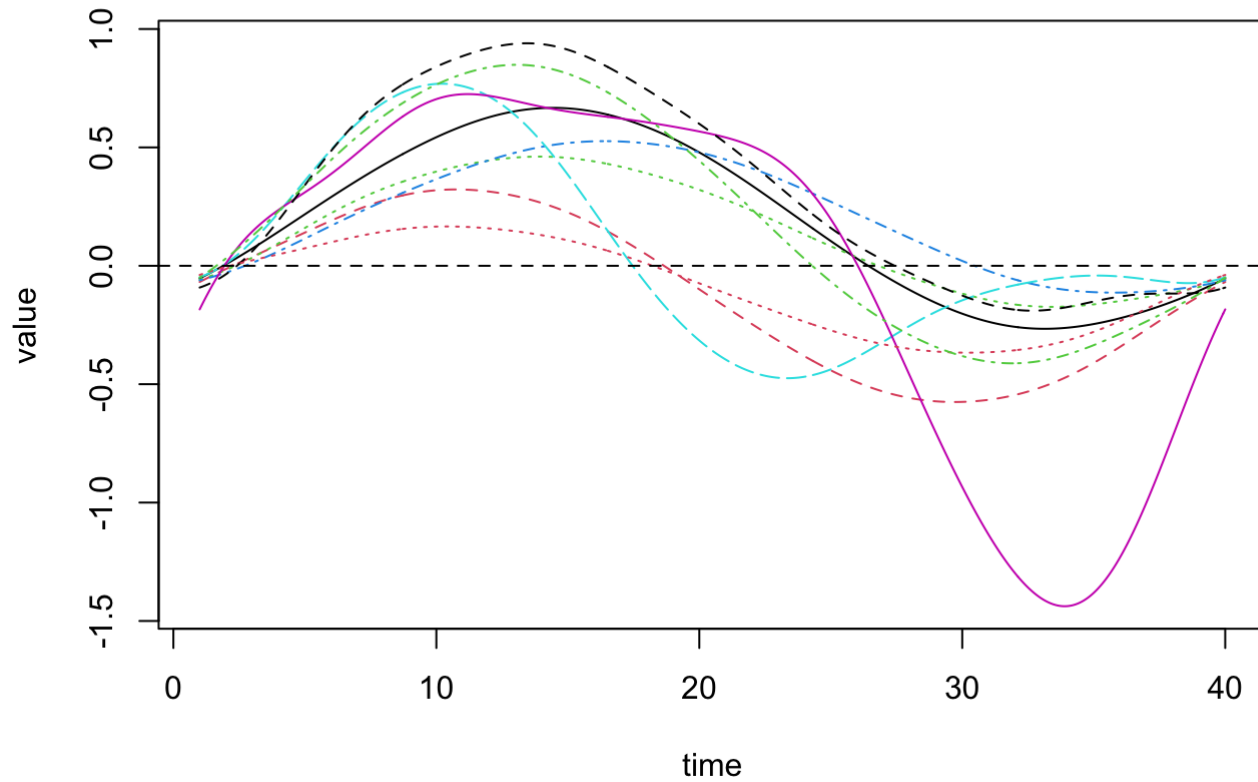
5.2 Scale the timeframe of each node

5.3 pivot the new dataframe

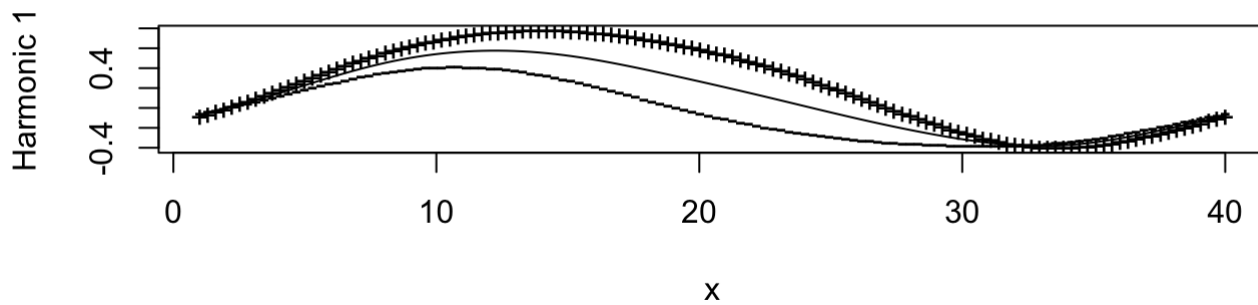
5.4 fPCA on new dataframe

```
fPCA_subset <- function(data_mat, k, nharm){  
  basis <- create.fourier.basis(c(1,nrow(data_mat)), k)  
  smoothfd <- smooth.basis(1:nrow(data_mat), data_mat, basis)$fd  
  plot(smoothfd)  
  title(main="smoothed curves")  
  pcalist = pca.fd(smoothfd, nharm, harmfdPar=fdPar(smoothfd))  
  rotpcalist = varmx.pca.fd(pcalist)  
  par(mfrow=c(nharm,1))  
  plot.pca.fd(rotpcalist)  
  return(rotpcalist)  
}  
df_new <- as.matrix(df_new)  
rotpcalist = fPCA_subset(df_new, k=11, nharm=2)
```

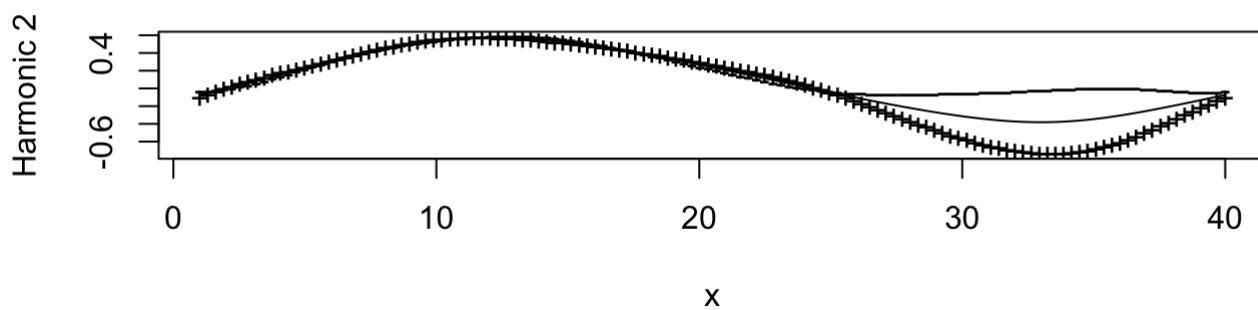
smoothed curves



PCA function 1 (Percentage of variability 43.4)



PCA function 2 (Percentage of variability 41.6)



6.1 single node (node=2), of the entire time frame

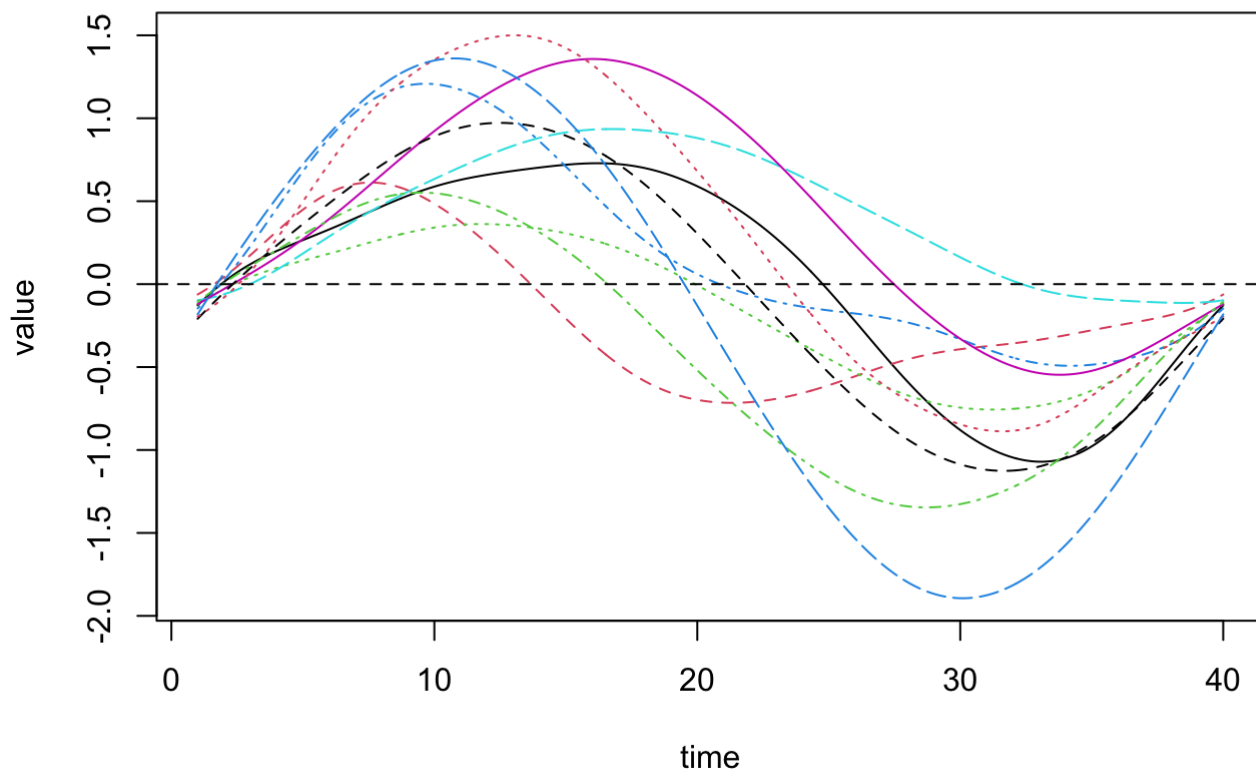
6.2 Scale the timeframe of each node

6.3 pivot the new dataframe

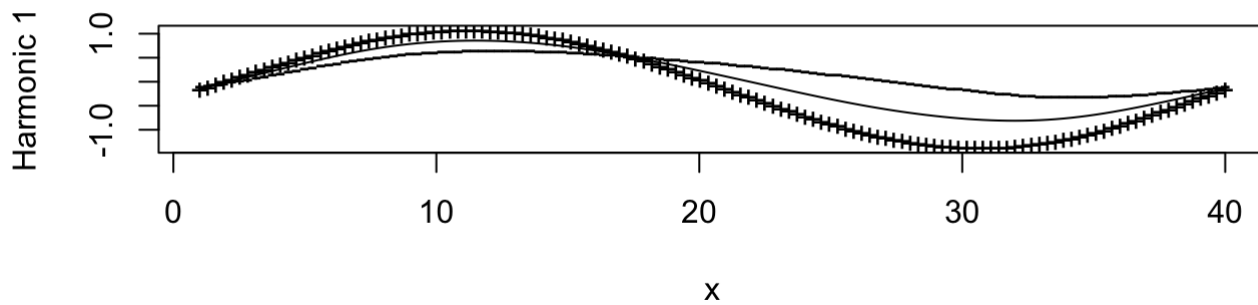
6.4 fPCA on new dataframe

```
fPCA_subset <- function(data_mat, k, nharm){  
  basis <- create.fourier.basis(c(1,nrow(data_mat)), k)  
  smoothfd <- smooth.basis(1:nrow(data_mat), data_mat, basis)$fd  
  plot(smoothfd)  
  title(main="smoothed curves")  
  pcalist = pca.fd(smoothfd, nharm, harmfdPar=fdPar(smoothfd))  
  rotpcalist = varmx.pca.fd(pcalist)  
  par(mfrow=c(nharm,1))  
  plot.pca.fd(rotpcalist)  
  return(rotpcalist)  
}  
df_new <- as.matrix(df_new)  
rotpcalist = fPCA_subset(df_new, k=11, nharm=2)
```

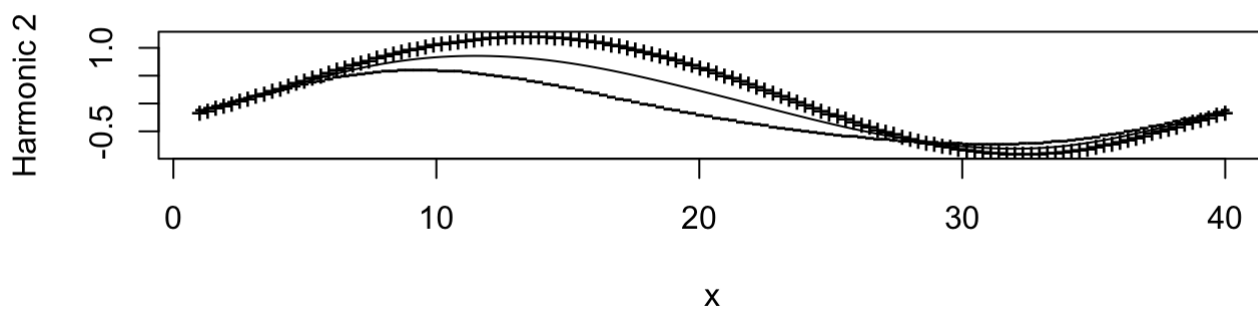
smoothed curves



PCA function 1 (Percentage of variability 51)



PCA function 2 (Percentage of variability 40.7)



7.1 single node (node=3), of the entire time frame

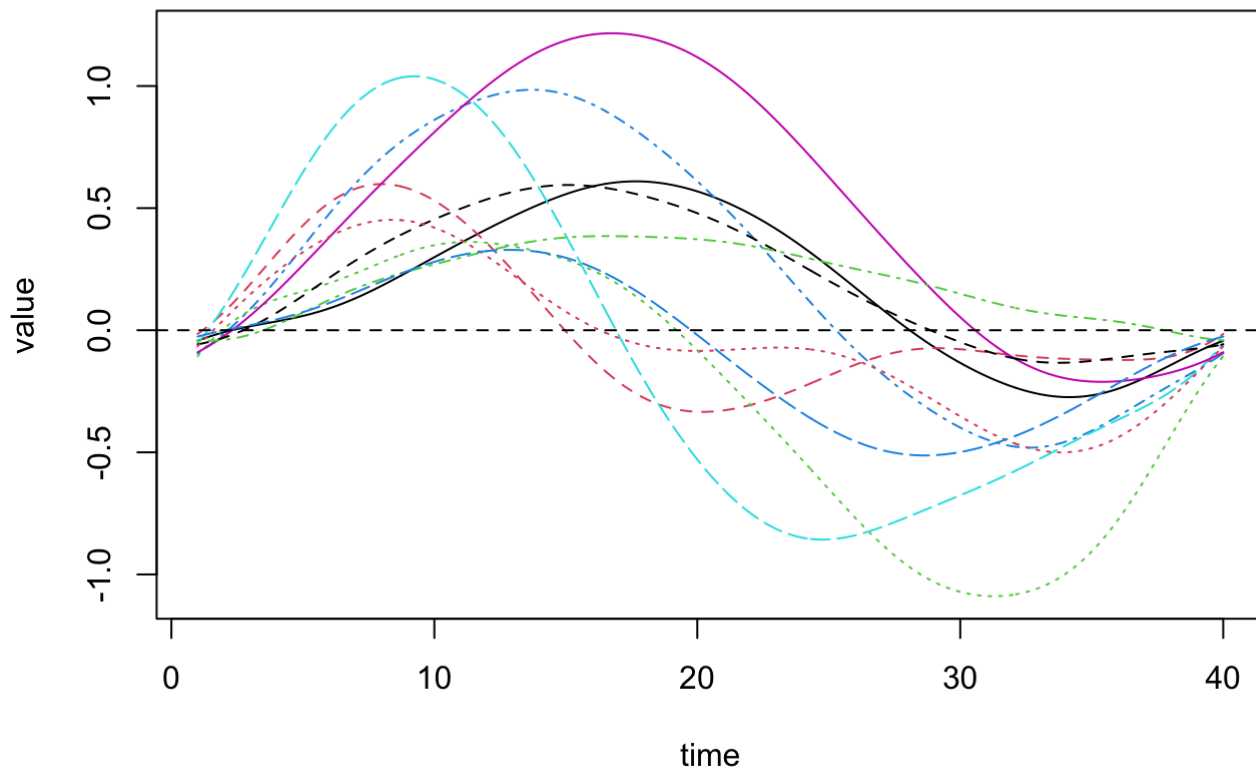
7.2 Scale the timeframe of each node

7.3 pivot the new dataframe

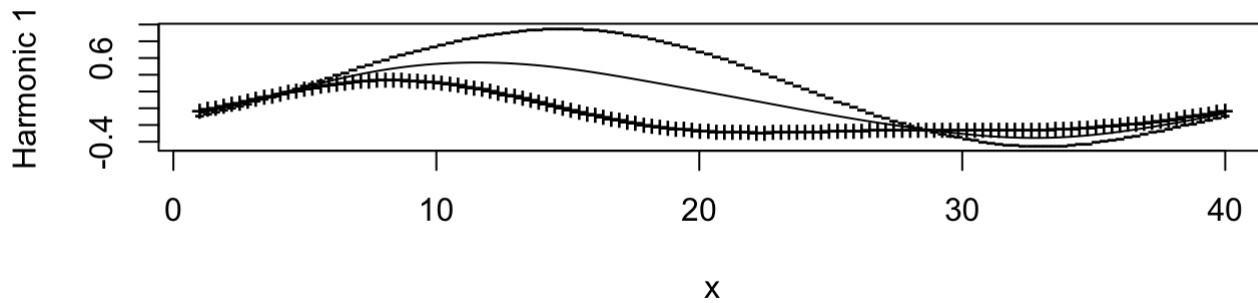
7.4 fPCA on new dataframe

```
fPCA_subset <- function(data_mat, k, nharm){  
  basis <- create.fourier.basis(c(1,nrow(data_mat)), k)  
  smoothfd <- smooth.basis(1:nrow(data_mat), data_mat, basis)$fd  
  plot(smoothfd)  
  title(main="smoothed curves")  
  pcalist = pca.fd(smoothfd, nharm, harmfdPar=fdPar(smoothfd))  
  rotpcalist = varmx.pca.fd(pcalist)  
  par(mfrow=c(nharm,1))  
  plot.pca.fd(rotpcalist)  
  return(rotpcalist)  
}  
df_new <- as.matrix(df_new)  
rotpcalist = fPCA_subset(df_new, k=11, nharm=2)
```

smoothed curves



PCA function 1 (Percentage of variability 43.8)



PCA function 2 (Percentage of variability 42.8)

