

# 12aug2020

Zach Wang

8/12/2020

## 1. progress summary

- 1. Replacing the original data of each node with the corresponding 2 PCs after transformations (transformed to sinusoidal cycle)
- 2. tried to perform fPCA again on the PCs data

## 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. Getting Harmonics curves based on Eigenfunctions

$$Eigenfunction(t) = \sum_{k=1}^K c_k \Phi_k(t)$$

and

$$\Phi_k(t) = c_1 + c_2 \sin(t) + c_3 \cos(t) + c_4 \sin(t) + c_5 \cos(t) + \dots = c_1 + (c_2 + c_4 + \dots) \sin(t) + (c_3 + c_5 + \dots) \cos(t)$$

```

# find y value based on PC coef and sin/cos functions

## FPCA function
fPCA_subset <- function(data_mat, k, nharm, plt){
  basis <- create.fourier.basis(c(1,nrow(data_mat)), k)
  smoothfd <- smooth.basis(1:nrow(data_mat), data_mat, basis)$fd
  pcalist = pca.fd(smoothfd, nharm, harmfdPar=fdPar(smoothfd))
  rotpcalist = varmx.pca.fd(pcalist)
  par(mfrow=c(nharm,1))
  if(plt==1){
    plot.pca.fd(rotpcalist)
  }
  return(rotpcalist)
}

## PC1 PC2 for all 32 nodes
PC1_df = data.frame(matrix(nrow=80))
PC2_df = data.frame(matrix(nrow=80))
mean_df = data.frame(matrix(nrow=80))
for(node in 1:32){
  result_obj <- f_fourier_smooth(time_subset=c(1:600), data_mat, node_subset=c(node), k=
32)
  smoothed_curve = eval.fd(c(1:600),result_obj$fd)
  transformed_node = transform.Cycle(smoothed_curve, register=1)
  df_tmp = data.frame(cycle=integer(), time=integer(), y_value=integer())
  for(i in 1:length(unique(transformed_node$cycle))){
    tmp=subset(transformed_node, cycle==i)
    tmp$time=(tmp$time)/max(tmp$time)
    df_tmp=rbind(df_tmp,tmp)
  }
  df_new = data.frame(matrix(nrow=80))
  for(i in 1:length(unique(df_tmp$cycle))){
    xx=seq(0,1,length.out=80)
    tmp=subset(df_tmp, cycle==i)
    s=smooth.spline(x=tmp$time, y=tmp$y_value, df = 10)
    df_new[,ncol(df_new)+1]=predict(s,xx)$y
  }
  df_new=df_new[,2:(length(unique(df_tmp$cycle))+1)]
  df_new <- as.matrix(df_new)
  rotpcalist = fPCA_subset(df_new, k=11, nharm=2, plt=0)
  PCA_df <- data.frame(matrix(nrow=rotpcalist$harmonics$basis$rangeval[2]))
  # PC1 & PC2
  harmfd <- rotpcalist[[1]]
  basisfd <- harmfd$basis
  rangex <- basisfd$rangeval
  x <- seq(rangex[1], rangex[2], length = harmfd$basis$rangeval[2])
  fdmat <- eval.fd(x, harmfd)
  meanmat <- eval.fd(x, rotpcalist$meanfd)

  PC1_df[,ncol(PC1_df)+1]=fdmat[,1]
  PC2_df[,ncol(PC2_df)+1]=fdmat[,2]
  mean_df[,ncol(mean_df)+1]=meanmat
}

```

```

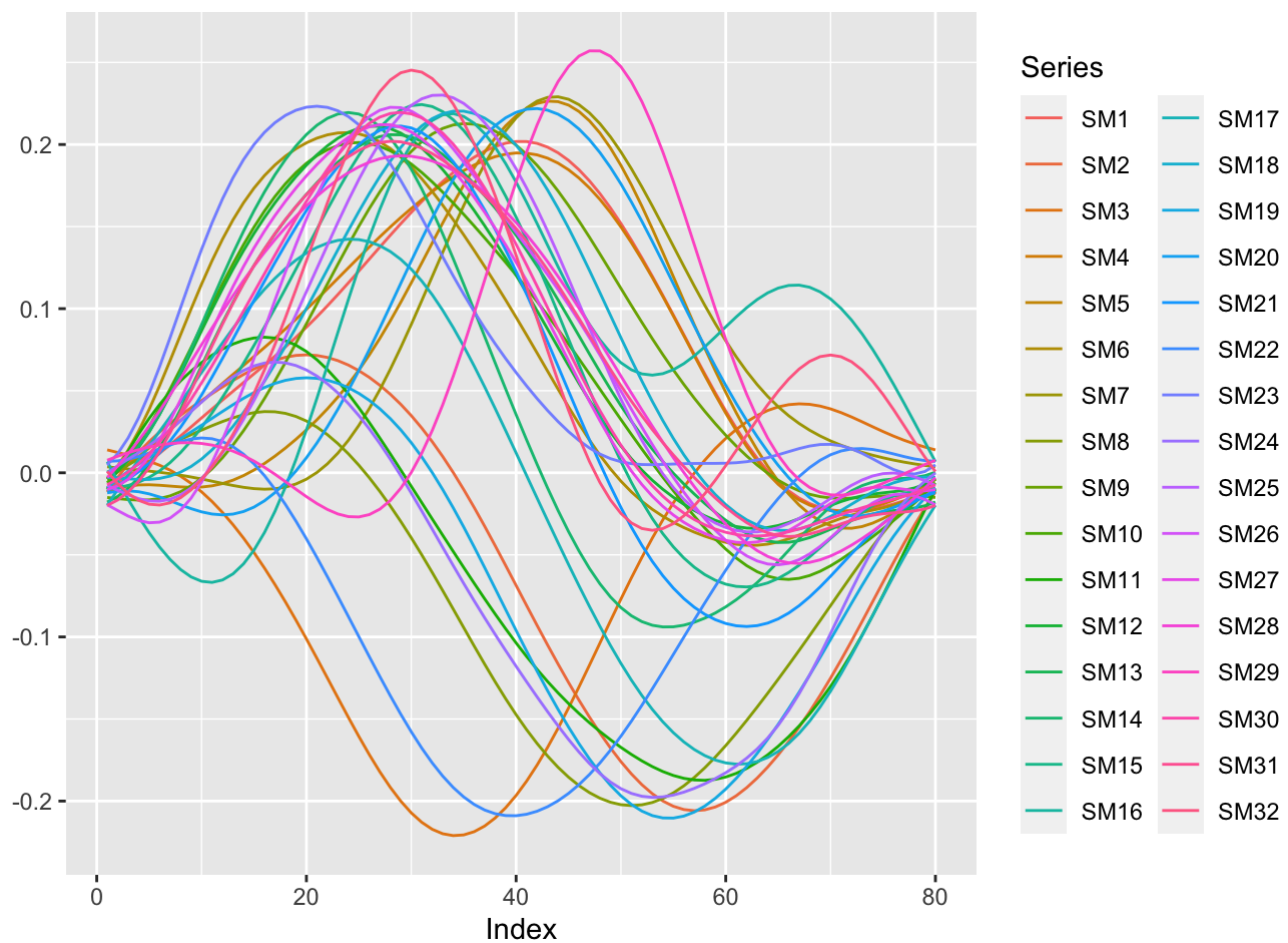
PC1_df = data.frame(PC1_df[,2:(ncol(PC1_df))])
names(PC1_df)=colnames(data_mat)

PC2_df = data.frame(PC2_df[,2:(ncol(PC2_df))])
names(PC2_df)=colnames(data_mat)

mean_df = data.frame(mean_df[,2:(ncol(mean_df))])
names(mean_df)=colnames(data_mat)

z_1 = read.zoo(PC1_df, index='index')
z_2 = read.zoo(PC2_df, index='index')
z_3 = read.zoo(mean_df, index='index')
autoplot(z_1, facet = NULL)

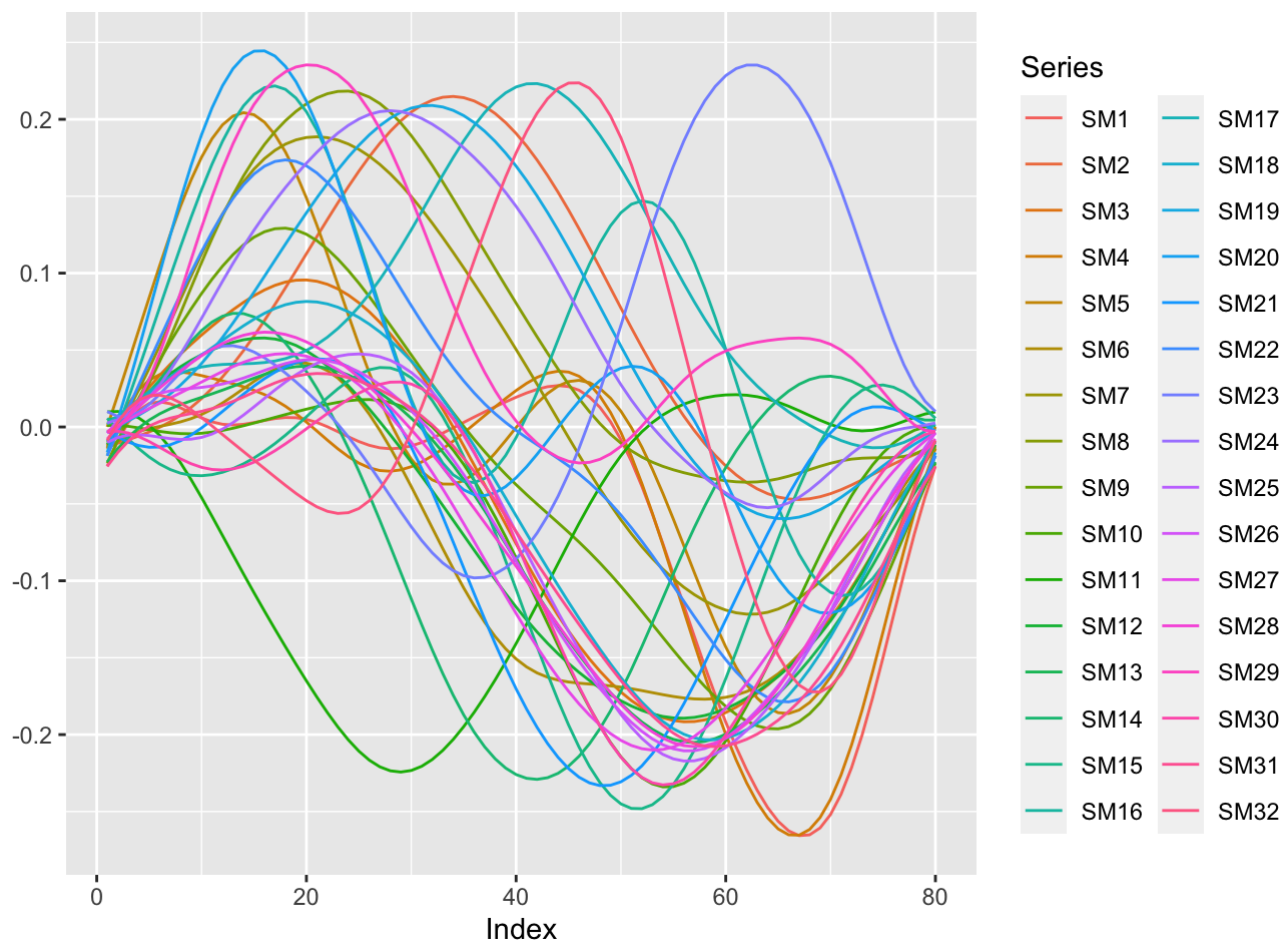
```



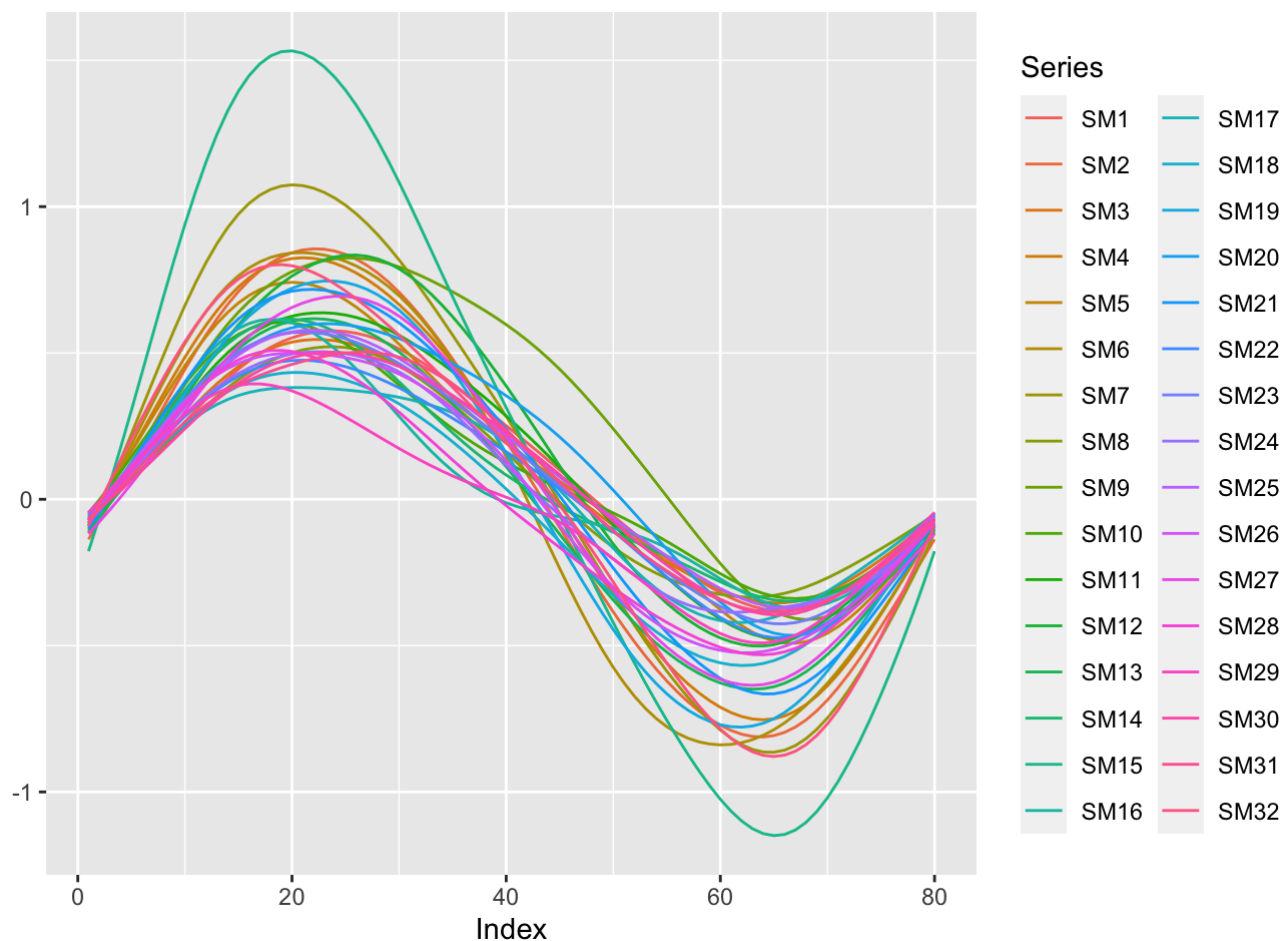
```

autoplot(z_2, facet = NULL)

```



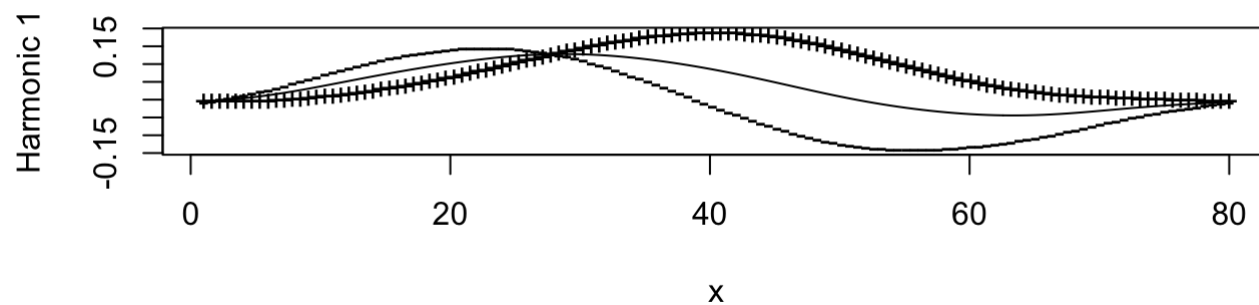
```
autoplot(z_3, facet = NULL)
```



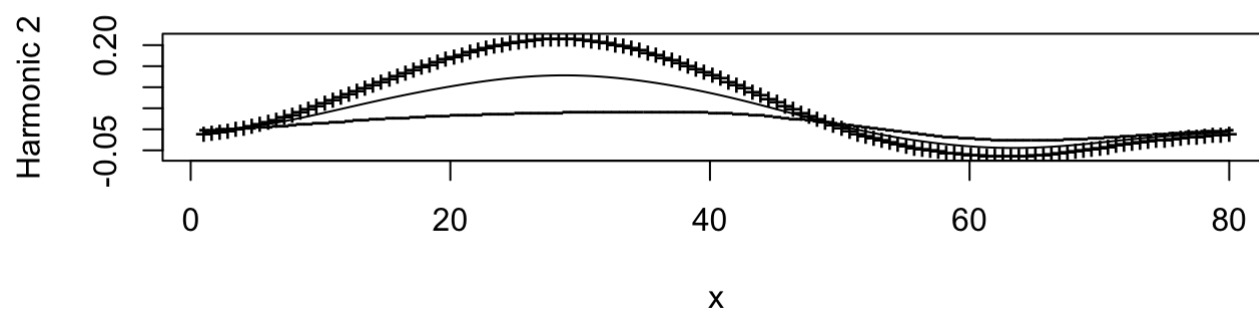
## 6. fPCA after replacing data with PC1 and PC2

```
par(mfrow=c(2,1))
PC1_df <- as.matrix(PC1_df)
PC2_df <- as.matrix(PC2_df)
mean_df <- as.matrix(mean_df)
rotpcalist1 = fPCA_subset(PC1_df, k=11, nharm=2, plt=0)
plot.pca.fd(rotpcalist1)
```

### PCA function 1 (Percentage of variability 55.3 )

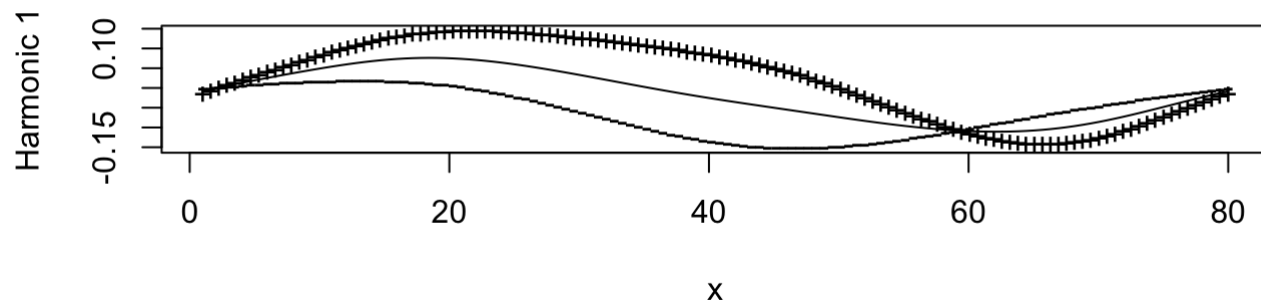


### PCA function 2 (Percentage of variability 36.3 )



```
rotpcalist2 = fPCA_subset(PC2_df, k=11, nharm=2, plt=0)
plot.pca.fd(rotpcalist2)
```

### PCA function 1 (Percentage of variability 41.3 )



### PCA function 2 (Percentage of variability 34.1 )

