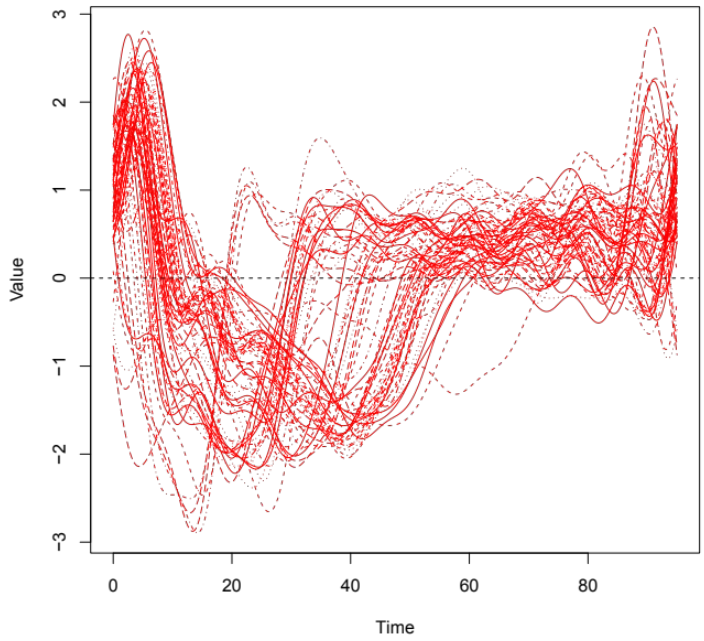
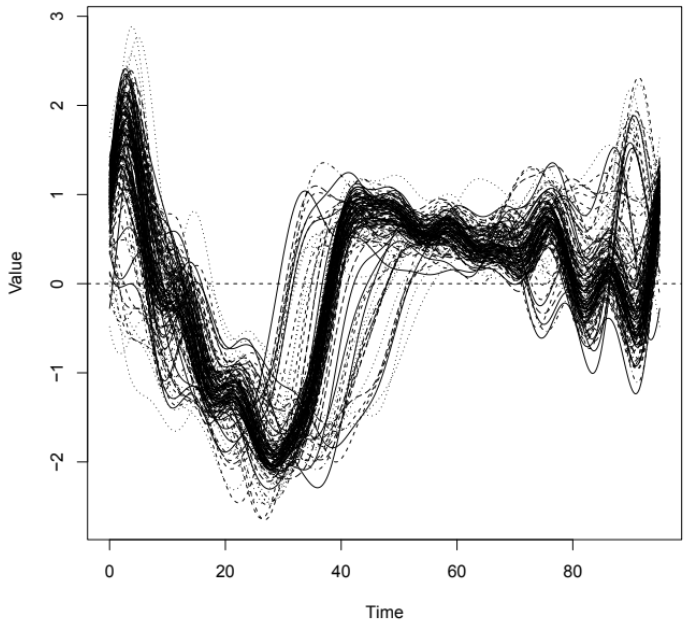
REPORT 2

Converting Data into Functional Object

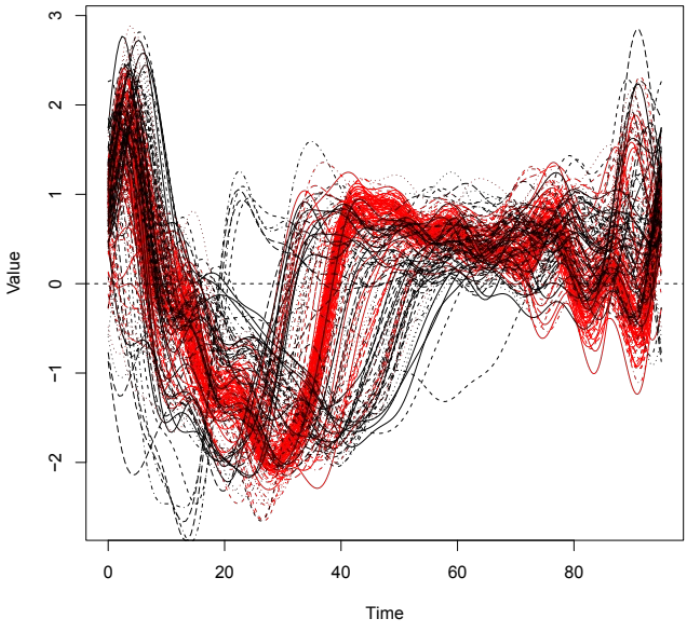
```
functional_data <- function(data, nsplines=NSPLINES) {
    if (FOURIER_BASIS) {
        basis <- fda::create.fourier.basis(
            rangeval = c(0, 95),
            nbasis = nsplines
        )
        ecg_fdata <- smooth.basis(
                                    argvals = seq(0, 95, length.out = 96),
                                    y = t(data.matrix(data)),
                                    fdParobj = basis
        )$fd
        return(ecg_fdata)
    } else {
        return(fda.usc::fdata(data.matrix(data)))
    }
}
```



Class 1



Class 2

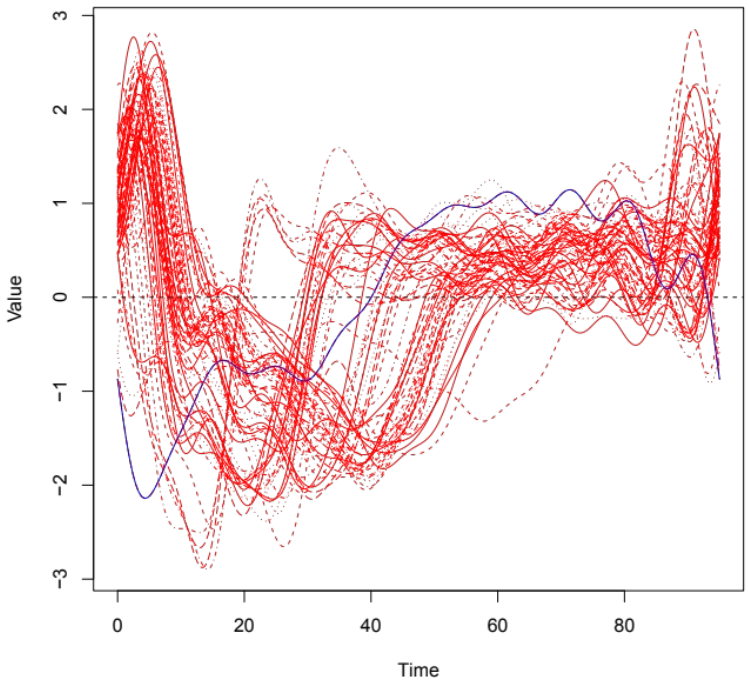


All Classes

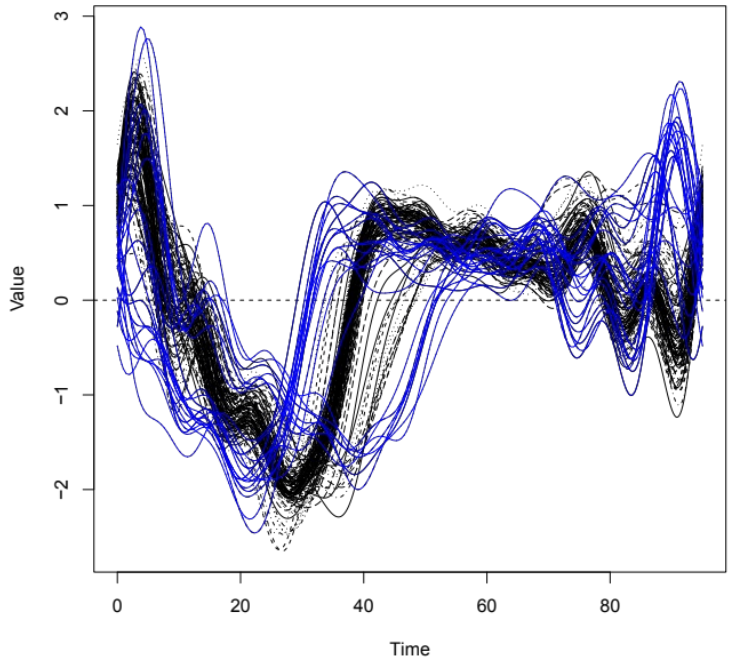
The ECG data was converted into a functional data object by smoothing it with the “smooth.basis()” function from the “fda” library. A fourier basis was used as the basis for the smoothing function. The sequence length of the data was set to 96 (number of features) and the number of basis (nbasis) was set to 20 (the same as the paper).

Detecting Outliers in the Functional Data

```
print("Detecting outliers in Class 1.....")
drops <- c("X1")
ecg_df <- df_class_1[, !(names(df_class_1) %in% drops)]
ecg_fdata <- functional_data(ecg_df)
ecg_outliers <- outliers.depth.trim(ecg_fdata, trim = OUTLIER_TRIM)
num_of_outliers <- 0
cat("Number of outliers in Class 1:",
    length(ecg_outliers$outliers), "\n")
pdf("class_1_outliers.pdf")
plot.fd(
           ecg_fdata,
           col = "red",
           main = "Outliers in Class 1",
           xlab = "Time",
           ylab = "Value"
)
for (otlr in ecg_outliers$outliers) {
    lines(
          ecg_fdata[ecg_fdata$fdnames$reps == otlr],
          col = "blue"
    )
}
```



Outliers in Class 1

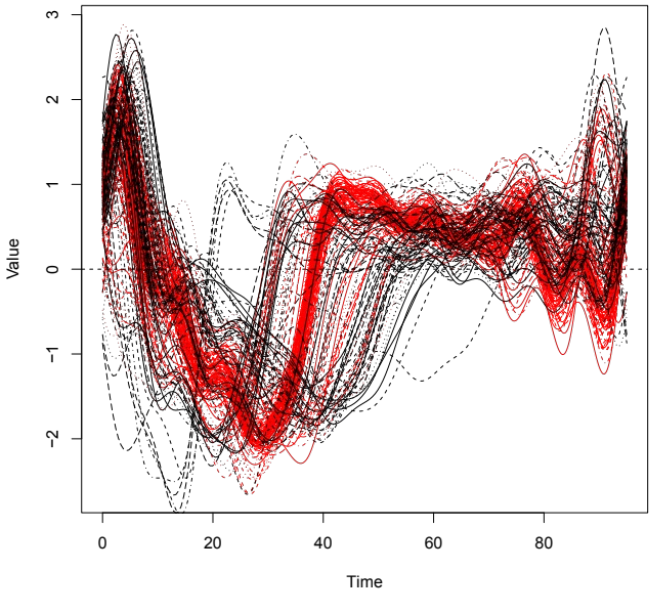


Outliers in Class 2

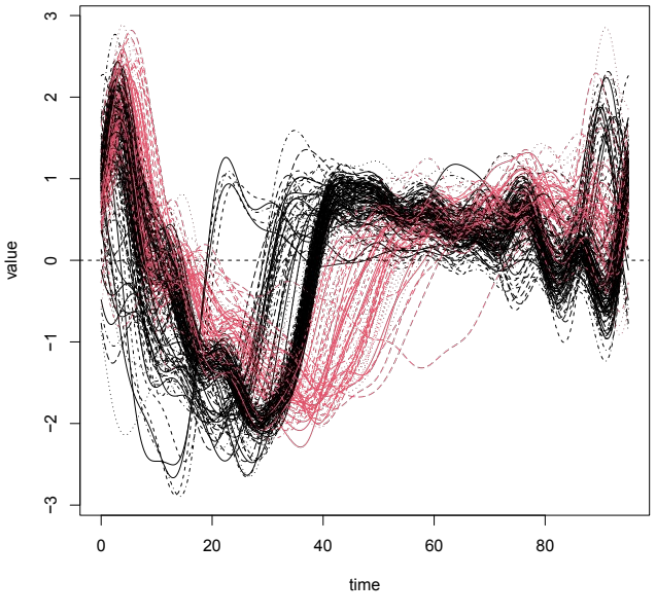
The outliers in both classes was identified using the “outliers.depth.trim()” function from the “fda.usc” library. The “trim” attribute was set to 0.1. The outliers are then plotted along with the original data. An interesting fact to note is that there are more outliers in Class 1 than in Class 2. The number of outliers identified in class 1 was 1 and the number of outliers identified in Class 2 was 21.

Clustering using funHDDC

```
drops <- c("X1")
ecg_df <- df[, !(names(df) %in% drops)]
ecg_fdata <- functional_data(ecg_df)
for (model in FUNHDDC_MODELS) {
    result <- funHDDC(
                      ecg_fdata,
                      K = 2,
                      init = "kmeans",
                      threshold = FUNHDDC_THRESHOLD,
                      model = model,
                      itermax = FUNHDDC_ITER_MAX,
                      nb.rep = 50
    )
    pdf("funHDDC_clusters.pdf")
    plot.fd(ecg_fdata, col = result$class, lwd = 2, lty = 1)
    cf_matrix <- table(labels, result$class)
    print(cf_matrix)
    ccr <- (cf_matrix[1, 1] + cf_matrix[2, 2]) / sum(cf_matrix)
    cat("The correct classification rate:", ccr * 100, "%\n")
}
```



Original Clusters



funHDDC Clusters

The funHDDC algorithm was run on the functional data across the models "AkjBkQkDk", "AkjBQkDk", "AkBkQkDk", "AkBQkDk", "AbkQkDk" and "ABQkDk". The number of clusters was set to 2 and the threshold was set to 0.1. The maximum number of iterations was set to 200 which was ample for the algorithm to converge with the “init” parameter set to “kmeans”. The maximum CCR (correct classification rate) obtained for the models was 76.5%.

Model Name	Correct Classification Rate
AkjBkQkDk	76.0 %
AkjBQkDk	76.5 %
AkBkQkDk	76.5 %
AkBQkDk	23.5 % ~ 76.5 %
ABkQkDk	23.5 % ~ 76.5 %
ABQkDk	23.5 % ~ 76.5 %