Data Simulation

Simulation-based settings

Article: Lizbeth Naranjo, Carlos J. Perez, Daniel F. Merino (2025). A data ensemble-based approach for detecting vocal disorders using replicated acoustic biomarkers from electroglottography.
Sensing and Bio-Sensing Research Journal, vol, num, pages.

Simulation-based settings

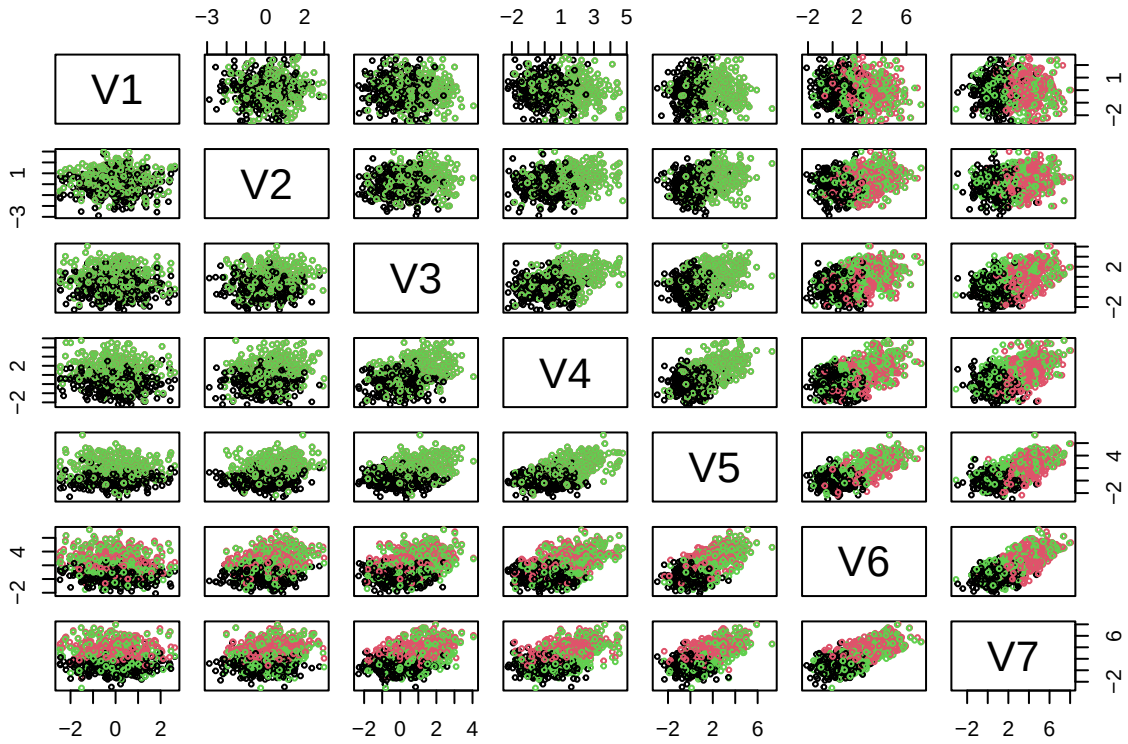
The generation process for this three-class classification problem is adapted to introduce replications in the explanatory variables. The covariates x_{ikj} are generated from the algorithm 3, considering $n = 300$ subjects, $i = 1, \ldots, n$, $K = 21$ predictor variables, $k = 1, \ldots, K$, and $J = 3$ replicated measures, $j = 1, \ldots, J$. Note that Algorithm 3 simulates the same number of subjects for each class or category C, this number is fixed for each class. This procedure enables the simulation of a dataset with many covariates and replicated measures.

```
## change the address where the file will be saved
address = "~/Documents/GitHub/"
```

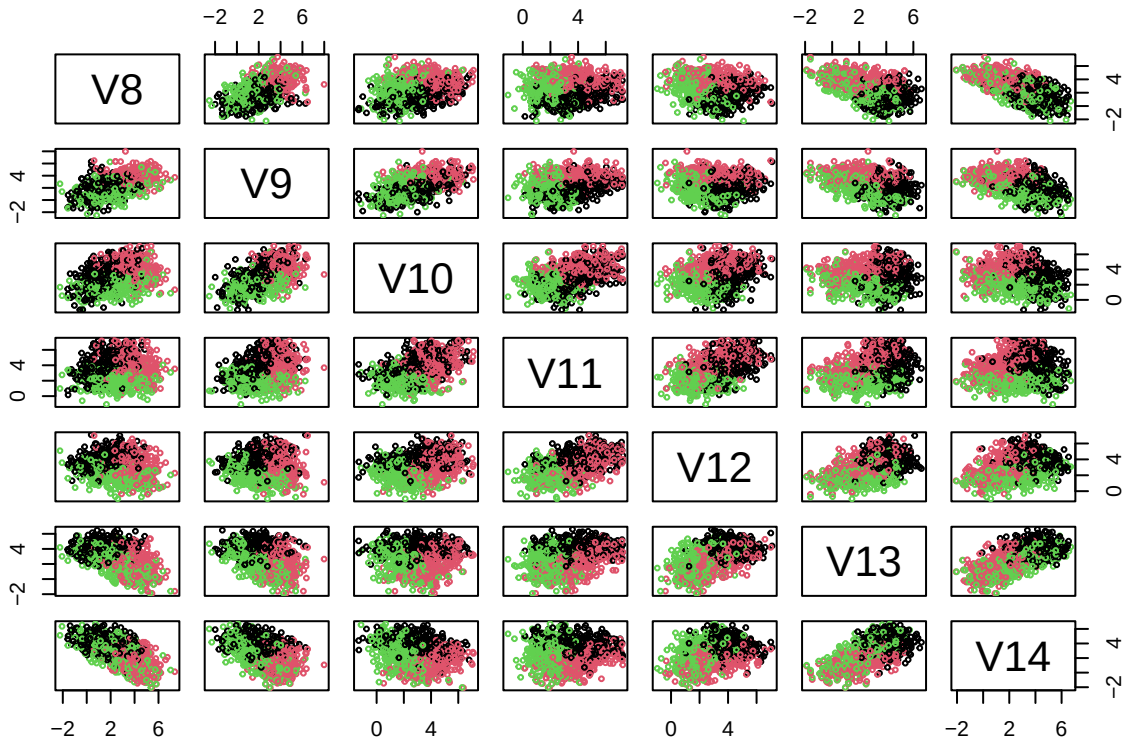
```
set.seed(12345) ## seed
G = 3   ## number of classes or groups
n = 300   ## sample size
m = n/G   ## n/G = sample size / number of classes
K = 21   ## predictor variables
J = 3   ## number of replicated measures

## shifted triangular waveforms functions:
v1 <- function(k){ max(6-abs(k-11),0) }
v2 <- function(k){ v1(k-4) }
v3 <- function(k){ v1(k+4) }

## function for predictors variables with replications
predictors <- function(K,G){
  u = runif(1)
  eps = rnorm(K)
  x = matrix(NA,G,K)
  for(k in 1:K){
    x[1,k] = u*v1(k) + (1-u)*v2(k) + eps[k]   ## predictors for class 1
    x[2,k] = u*v1(k) + (1-u)*v3(k) + eps[k]   ## predictors for class 2
    x[3,k] = u*v2(k) + (1-u)*v3(k) + eps[k]   ## predictors for class 3
  }
  y = c(1,2,3)   ## response variable
  return(list(x=x,y=y))
}

Ytrue = matrix(NA,n)   ## response variable
```

```
X = array(NA,dim=c(n*J,K))   ## K explanatory variables with J replications
colnames(X) = paste0("V",c(1:K))
ID = rep(NA,n*J)   ## ID of the subject
Rep = rep(NA,n*J)   ## ID for the replication

aux = 0
for(i in 1:m){
  for(j in 1:J){
    train <- predictors(K,G)   ## generate variables with replications
    for(h in 1:G){
      aux = aux+1
      Ytrue[aux] = train$y[h]
      X[aux,] = train$x[h,]
      ID[aux] = G*(i-1)+h
      Rep[aux] = j
} } }
Y = factor(Ytrue)   ## categorical response variable
```

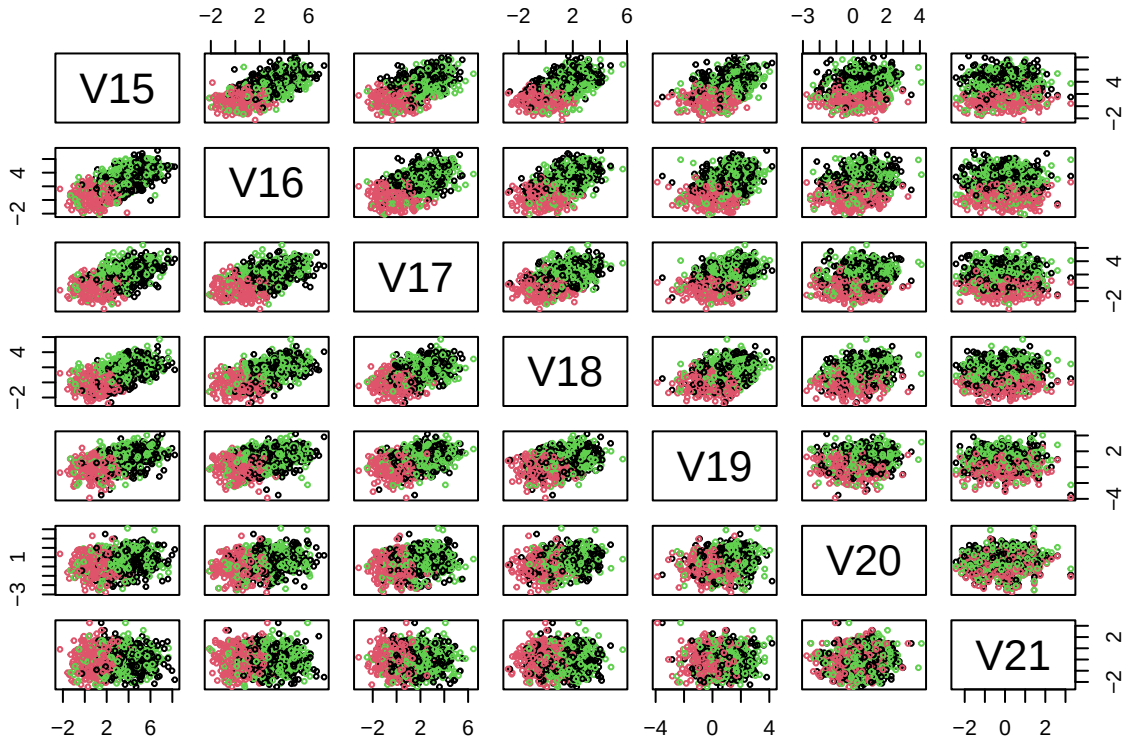
```
pairs(X[,1:7], col=Ytrue,cex=0.5)
```




```
pairs(X[,8:14], col=Ytrue,cex=0.5)
```




```
pairs(X[,15:21], col=Ytrue,cex=0.5)
```




```
## Save the data
data = data.frame(X,ID,rep=Rep, status=Y)
head(data)
```

```
##           V1        V2         V3         V4         V5         V6          V7
## 1  1.1541136  1.206173 -0.9686025 0.02312405  2.3165891 -0.3054029 -1.61019789
## 2  1.1541136  1.485269 -0.4104103 0.86041236  3.4329735  1.0900776  0.06437873
## 3  1.1541136  1.485269 -0.4104103 0.86041236  3.4329735  0.3691737 -1.37742906
## 4 -0.6443284 -1.553137 -1.5977095 1.80509752 -0.4816474  1.3936230  2.15860994
## 5 -0.6443284 -1.326381 -1.1441960 2.48536784  0.4253797  2.5274069  3.51915059
## 6 -0.6443284 -1.326381 -1.1441960 2.48536784  0.4253797  1.7541637  1.97266414
##          V8        V9      V10      V11      V12      V13      V14      V15
## 1 2.0685608 2.6366958 5.544770 4.434717 4.987851 3.720234 3.668854 3.077667
## 2 3.4640413 3.7530802 6.102962 4.434717 4.429659 2.603849 2.273374 1.403090
## 3 1.3013296 0.8694646 3.219346 1.551101 2.987851 2.603849 3.715182 4.286706
## 4 2.1574187 3.9048461 6.289806 7.142163 6.178932 4.254271 3.944702 2.582941
## 5 3.2912026 4.8118732 6.743320 7.142163 5.725419 3.347244 2.810918 1.222400
## 6 0.9714729 1.7189003 3.650347 4.049190 4.178932 3.347244 4.357404 4.315373
##          V16       V17         V18      V19        V20        V21 ID rep status
## 1 -0.3964881 1.5892168  0.48317124 1.863786 -0.8284375 -0.1784143  1   1      1
## 2 -1.7919686 0.4728324 -0.35411707 1.305594 -1.1075336 -0.1784143  2   1      2
## 3  1.0916470 3.3564480  1.80859462 2.747402 -0.3866297 -0.1784143  3   1      3
## 4  0.2449769 2.6747610  0.70607137 1.582024 -2.1536013 -1.0602656  1   2      1
## 5 -0.8888070 1.7677339  0.02580105 1.128511 -2.3803581 -1.0602656  2   2      2
## 6  2.2041659 4.8607067  2.34553072 2.674997 -1.6071148 -1.0602656  3   2      3
```

```
write.table(data,paste0(address,"data_simulated.csv"),row.names=FALSE,sep=";",dec=".")
```