Parkinson: Decorrelation-Based Feature Discovery

Jose Tamez

2022-10-04

Table of Contents

# Outputs Analysis and Graphics: Effect of GDSTM-Based Decorrelation on Feature Discovery

This will load the results of the Parkinson Analysis and create the outputs.

## From the original

Here I showcase of to use BSWiMS feature selection/modeling function coupled with Goal Driven Sparse Transformation Matrix (GDSTM) as a pre-processing step to decorrelate highly correlated features. The aim(s) are:

1. To improve model performance by uncovering the hidden information between correlated features.
2. To simplify the interpretation of the machine learning models.

This demo will use:

* FRESA.CAD::GDSTMDecorrelation(). For Decorrelation of Multidimensional data sets
  + FRESA.CAD::getDerivedCoefficients(). For the extraction of the decorrelated features.
* FRESA.CAD::randomCV() For the cross-validation of the Machine Learning models
* FRESA.CAD::BSWiMS.model(). For the generation of bootstrapped logistic models
  + FRESA.CAD::summary(). For the summary description of the BSWiMS model
* FRESA.CAD::predictionStats\_binary(). For describing the performance of the model
* heatmap.2(). For displaying the correlation matrix
* vioplot::vioplot(). For the display of the z-distribution of significant features.

### Loading the libraries

library("FRESA.CAD")  
library(readxl)  
library(vioplot)  
library(igraph)  
  
op <- par(no.readonly = TRUE)  
pander::panderOptions('digits', 3)  
pander::panderOptions('table.split.table', 400)  
pander::panderOptions('keep.trailing.zeros',TRUE)

## Material and Methods

### Signed Log Transform

The function will be used to transform all the continuous features of the data

signedlog <- function(x) { return (sign(x)\*log(abs(x)+1.0e-12))}

## Data: The Parkinson Data-Set

The data to process is described in:

Erdogdu Sakar, Betul, Gorkem Serbes, and C. Okan Sakar. “Analyzing the effectiveness of vocal features in early telediagnosis of Parkinson’s disease.” *PloS one* 12, no. 8 (2017): e0182428.

The data was obtained from the UCI ML repository:

<https://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification>

I added a column to the data identifying the repeated experiments.

load(file="~/GitHub/FCA/ParkinsonDemo.RData")  
  
namecode <- read.csv("~/GitHub/FCA/Data/Parkinson\_names.csv")

### The Average of the Three Repetitions

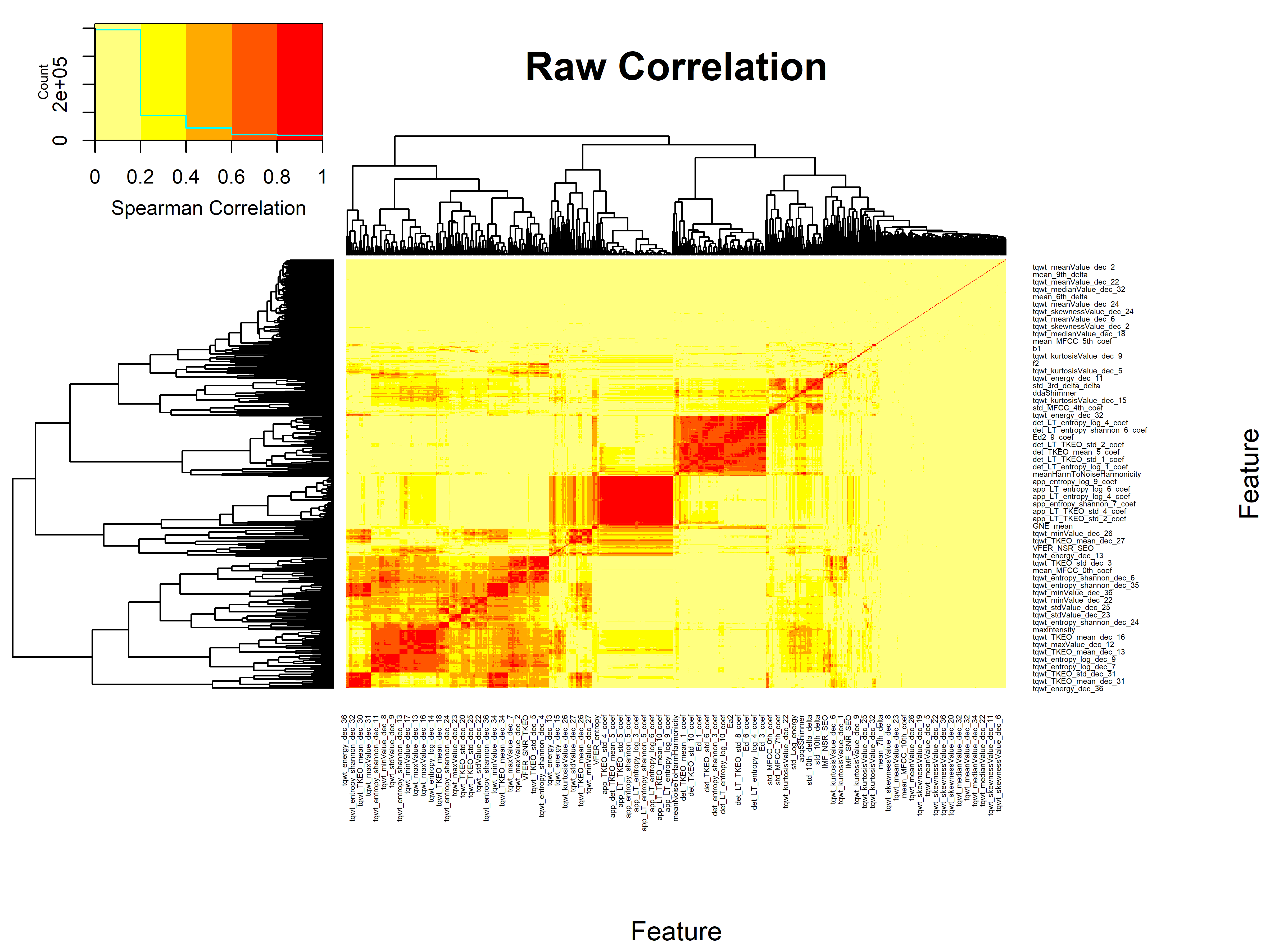
Each subject had three repeated observations. Here I’ll use the average of the three experiments per subject.

# rep1Parkison <- subset(pd\_speech\_features,RID==1)  
# rownames(rep1Parkison) <- rep1Parkison$id  
# rep1Parkison$id <- NULL  
# rep1Parkison$RID <- NULL  
# rep1Parkison[,1:ncol(rep1Parkison)] <- sapply(rep1Parkison,as.numeric)  
#   
# rep2Parkison <- subset(pd\_speech\_features,RID==2)  
# rownames(rep2Parkison) <- rep2Parkison$id  
# rep2Parkison$id <- NULL  
# rep2Parkison$RID <- NULL  
# rep2Parkison[,1:ncol(rep2Parkison)] <- sapply(rep2Parkison,as.numeric)  
#   
# rep3Parkison <- subset(pd\_speech\_features,RID==3)  
# rownames(rep3Parkison) <- rep3Parkison$id  
# rep3Parkison$id <- NULL  
# rep3Parkison$RID <- NULL  
# rep3Parkison[,1:ncol(rep3Parkison)] <- sapply(rep3Parkison,as.numeric)  
#   
# whof <- !(colnames(rep1Parkison) %in% c("gender","class"));  
# avgParkison <- rep1Parkison;  
# avgParkison[,whof] <- (rep1Parkison[,whof] + rep2Parkison[,whof] + rep3Parkison[,whof])/3  
# ## I apply the log transform to the data  
# avgParkison[,whof] <- signedlog(avgParkison[,whof])  
# pander::pander(table(avgParkison$class))

### Correlation Matrix of the Parkinson Data

The heat-map of the correlation:

cormat <- cor(avgParkison,method="spearman")  
gplots::heatmap.2(abs(cormat),  
 trace = "none",  
 scale = "none",  
 mar = c(10,10),  
 col=rev(heat.colors(5)),  
 main = "Raw Correlation",  
 cexRow = 0.35,  
 cexCol = 0.35,  
 key.title=NA,  
 key.xlab="Spearman Correlation",  
 xlab="Feature", ylab="Feature",  
# srtRow = 45,  
# srtCol = 45  
)



### Training and Testing Sets

We divided the data into training and testing sets.

# set.seed(2)  
# caseSet <- subset(avgParkison, class == 1)  
# controlSet <- subset(avgParkison, class == 0)  
# caseTrainSize <- nrow(caseSet)\*trainFraction;  
# controlTrainSize <- nrow(controlSet)\*trainFraction;  
# sampleCaseTrain <- sample(nrow(caseSet),caseTrainSize)  
# sampleControlTrain <- sample(nrow(controlSet),controlTrainSize)  
# trainSet <- rbind(caseSet[sampleCaseTrain,], controlSet[sampleControlTrain,])  
# testSet <- rbind(caseSet[-sampleCaseTrain,],controlSet[-sampleControlTrain,])  
pander::pander(table(trainSet$class))

| 0 | 1 |
| --- | --- |
| 41 | 122 |

pander::pander(table(testSet$class))

| 0 | 1 |
| --- | --- |
| 23 | 66 |

#### Decorrelation: Training and Testing Sets Creation

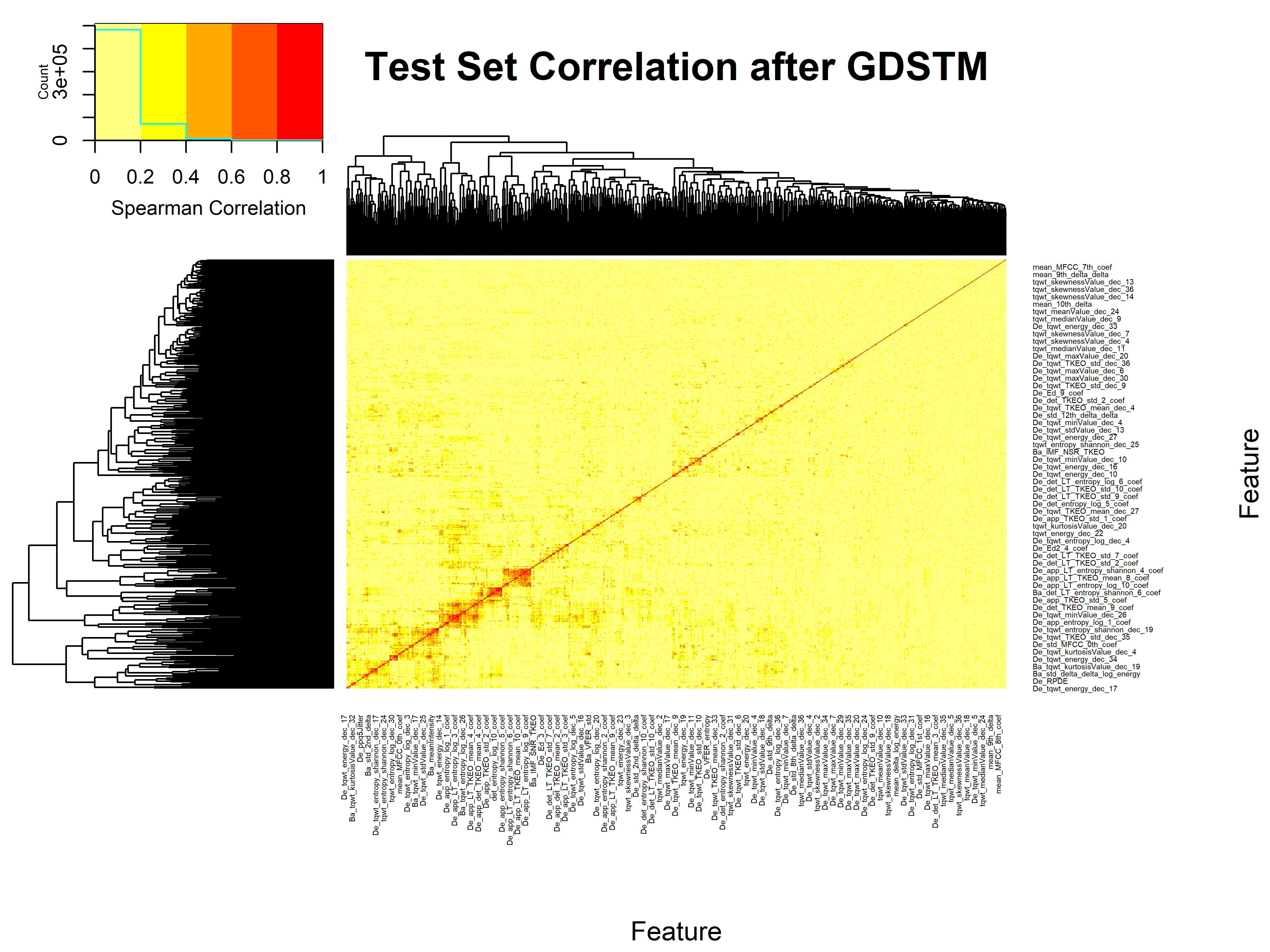
I compute a decorrelated version of the training and testing sets using the *GDSTMDecorrelation()* function of FRESA.CAD. The first decorrelation will be driven by features associated with the outcome. The second decorrelation will find the GDSTM without the outcome restriction.

## The GDSTM transformation driven by the Outcome  
  
# deTrain <- GDSTMDecorrelation(trainSet,Outcome="class",thr=0.8,verbose = TRUE)  
# deTest <- predictDecorrelate(deTrain,testSet)  
  
## The GDSTM transformation without outcome  
  
# deTrainU <- GDSTMDecorrelation(trainSet,thr=0.8,verbose = TRUE)  
# deTestU <- predictDecorrelate(deTrainU,testSet)

#### Correlation Matrix of the Decorrelated Test Data

The heat map of the testing set.

cormat <- cor(deTest,method="spearman")  
gplots::heatmap.2(abs(cormat),  
 trace = "none",  
 scale = "none",  
 mar = c(10,10),  
 col=rev(heat.colors(5)),  
 main = "Test Set Correlation after GDSTM",  
 cexRow = 0.35,  
 cexCol = 0.35,  
 key.title=NA,  
 key.xlab="Spearman Correlation",  
 xlab="Feature", ylab="Feature")



### Holdout Cross-Validation

Before doing the feature analysis. I’ll explore BSWiMS modeling using the Holdout cross validation method of FRESA.CAD. The purpose of the cross-validation is to observe and estimate the performance gain of decorrelation.

par(op)  
par(mfrow=c(1,3))  
  
## The Raw validation  
# cvBSWiMSRaw <- randomCV(avgParkison,  
# "class",  
# fittingFunction= BSWiMS.model,  
# classSamplingType = "Pro",  
# trainFraction = trainFraction,  
# repetitions = 150  
# )  
  
bpraw <- predictionStats\_binary(cvBSWiMSRaw$medianTest,"BSWiMS RAW",cex=0.60)

BSWiMS RAW

pander::pander(bpraw$CM.analysis$tab)

|  | Outcome + | Outcome - | Total |
| --- | --- | --- | --- |
| **Test +** | 147 | 19 | 166 |
| **Test -** | 41 | 45 | 86 |
| **Total** | 188 | 64 | 252 |

pander::pander(bpraw$accc)

|  | est | lower | upper |
| --- | --- | --- | --- |
| **5** | 0.762 | 0.704 | 0.813 |

pander::pander(bpraw$aucs)

| est | lower | upper |
| --- | --- | --- |
| 0.84 | 0.784 | 0.895 |

pander::pander(bpraw$berror)

| 50% | 2.5% | 97.5% |
| --- | --- | --- |
| 0.257 | 0.197 | 0.321 |

## The validation with Outcome-driven Decorrelation  
# cvBSWiMSDeCor <- randomCV(avgParkison,  
# "class",  
# trainSampleSets= cvBSWiMSRaw$trainSamplesSets,  
# fittingFunction= filteredFit,  
# fitmethod=BSWiMS.model,  
# filtermethod=NULL,  
# DECOR = TRUE,  
# DECOR.control=list(Outcome="class",thr=0.8)  
# )  
  
bpDecor <- predictionStats\_binary(cvBSWiMSDeCor$medianTest,"BSWiMS Outcome-Driven GDSTM",cex=0.60)

BSWiMS Outcome-Driven GDSTM

pander::pander(bpDecor$CM.analysis$tab)

|  | Outcome + | Outcome - | Total |
| --- | --- | --- | --- |
| **Test +** | 167 | 14 | 181 |
| **Test -** | 21 | 50 | 71 |
| **Total** | 188 | 64 | 252 |

pander::pander(bpDecor$accc)

|  | est | lower | upper |
| --- | --- | --- | --- |
| **5** | 0.861 | 0.812 | 0.901 |

pander::pander(bpDecor$aucs)

| est | lower | upper |
| --- | --- | --- |
| 0.873 | 0.816 | 0.931 |

pander::pander(bpDecor$berror)

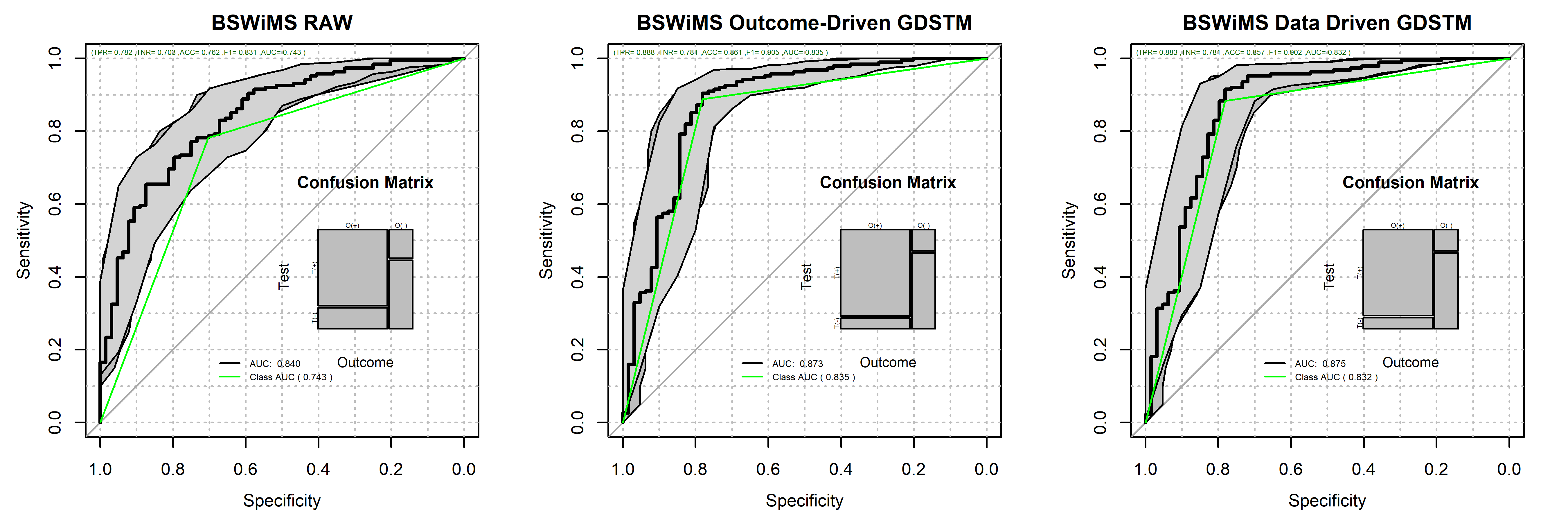
| 50% | 2.5% | 97.5% |
| --- | --- | --- |
| 0.166 | 0.113 | 0.22 |

### Here we compute the probability that the outcome-driven decorrelation ROC is superior to the RAW ROC.   
pander::pander(roc.test(bpDecor$ROC.analysis$roc.predictor,bpraw$ROC.analysis$roc.predictor))

DeLong’s test for two correlated ROC curves: bpDecor$ROC.analysis$roc.predictor and bpraw$ROC.analysis$roc.predictor

| Test statistic | P value | Alternative hypothesis | AUC of roc1 | AUC of roc2 |
| --- | --- | --- | --- | --- |
| 2.37 | 0.0179 \* | two.sided | 0.873 | 0.84 |

## The validation of Decorrelation without the outcome restriction  
# cvBSWiMSDeCorU <- randomCV(avgParkison,  
# "class",  
# trainSampleSets= cvBSWiMSRaw$trainSamplesSets,  
# fittingFunction= filteredFit,  
# fitmethod=BSWiMS.model,  
# filtermethod=NULL,  
# DECOR = TRUE,  
# DECOR.control=list(thr=0.8)  
# )  
  
bpDecorU <- predictionStats\_binary(cvBSWiMSDeCorU$medianTest,"BSWiMS Data Driven GDSTM",cex=0.60)

BSWiMS Data Driven GDSTM 

pander::pander(bpDecorU$CM.analysis$tab)

|  | Outcome + | Outcome - | Total |
| --- | --- | --- | --- |
| **Test +** | 166 | 14 | 180 |
| **Test -** | 22 | 50 | 72 |
| **Total** | 188 | 64 | 252 |

pander::pander(bpDecorU$accc)

|  | est | lower | upper |
| --- | --- | --- | --- |
| **5** | 0.857 | 0.808 | 0.898 |

pander::pander(bpDecorU$aucs)

| est | lower | upper |
| --- | --- | --- |
| 0.875 | 0.816 | 0.933 |

pander::pander(bpDecorU$berror)

| 50% | 2.5% | 97.5% |
| --- | --- | --- |
| 0.167 | 0.113 | 0.224 |

### Here we compute the probability that the blind decorrelation ROC is superior to the RAW ROC.   
pander::pander(roc.test(bpDecorU$ROC.analysis$roc.predictor,bpraw$ROC.analysis$roc.predictor))

DeLong’s test for two correlated ROC curves: bpDecorU$ROC.analysis$roc.predictor and bpraw$ROC.analysis$roc.predictor

| Test statistic | P value | Alternative hypothesis | AUC of roc1 | AUC of roc2 |
| --- | --- | --- | --- | --- |
| 2.27 | 0.0235 \* | two.sided | 0.875 | 0.84 |

par(op)

## The Raw Model *vs.* the Decorrelated-Based Model

After demonstrating that decorrelation is able to improve BSWiMS model performance, I’ll focus is showcasing the ability to discover new features associated with the outcome.

First, I’ll compute the BSWiMS models for the original data, and for the decorrelated data-set. The model estimation will be done using the training set and tested on the holdout test set, and repeated 10 times. After that, I’ll compare the statistical difference of both ROC curves.

par(op)  
par(mfrow=c(1,3))  
  
# bm <- BSWiMS.model(class~.,trainSet,NumberofRepeats = 20)  
bpraw <- predictionStats\_binary(cbind(testSet$class,predict(bm,testSet)),"BSWiMS RAW",cex=0.60)

BSWiMS RAW

# bmd <- BSWiMS.model(class~.,deTrain,NumberofRepeats = 20)  
bpdecor <- predictionStats\_binary(cbind(deTest$class,predict(bmd,deTest)),"Outcome-Driven Decorrelation",cex=0.60)

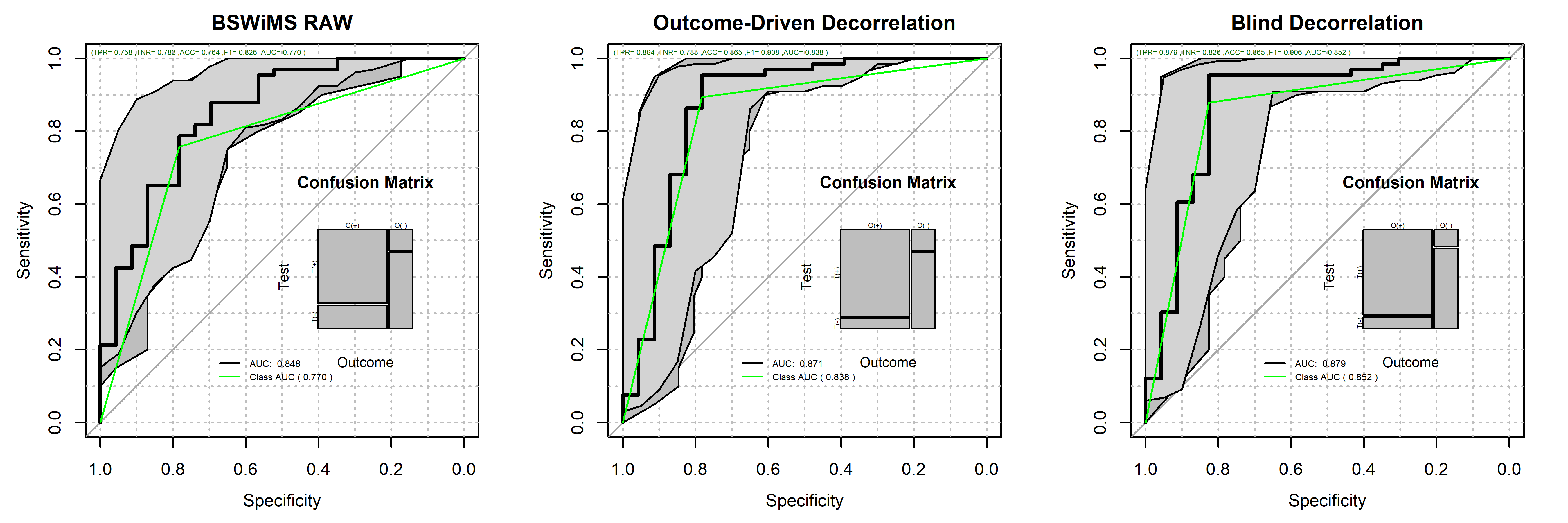
Outcome-Driven Decorrelation

pander::pander(roc.test(bpraw$ROC.analysis$roc.predictor,bpdecor$ROC.analysis$roc.predictor))

DeLong’s test for two correlated ROC curves: bpraw$ROC.analysis$roc.predictor and bpdecor$ROC.analysis$roc.predictor

| Test statistic | P value | Alternative hypothesis | AUC of roc1 | AUC of roc2 |
| --- | --- | --- | --- | --- |
| -0.97 | 0.332 | two.sided | 0.848 | 0.871 |

# bmdU <- BSWiMS.model(class~.,deTrainU,NumberofRepeats = 20)  
bpdecorU <- predictionStats\_binary(cbind(deTest$class,predict(bmdU,deTestU)),"Blind Decorrelation",cex=0.60)

Blind Decorrelation 

pander::pander(roc.test(bpraw$ROC.analysis$roc.predictor,bpdecorU$ROC.analysis$roc.predictor))

DeLong’s test for two correlated ROC curves: bpraw$ROC.analysis$roc.predictor and bpdecorU$ROC.analysis$roc.predictor

| Test statistic | P value | Alternative hypothesis | AUC of roc1 | AUC of roc2 |
| --- | --- | --- | --- | --- |
| -1.29 | 0.196 | two.sided | 0.848 | 0.879 |

par(op)

## The feature associations

par(op)  
par(mfrow=c(1,3))  
### The raw model  
  
pander::pander(nrow(bm$bagging$formulaNetwork))

*71*

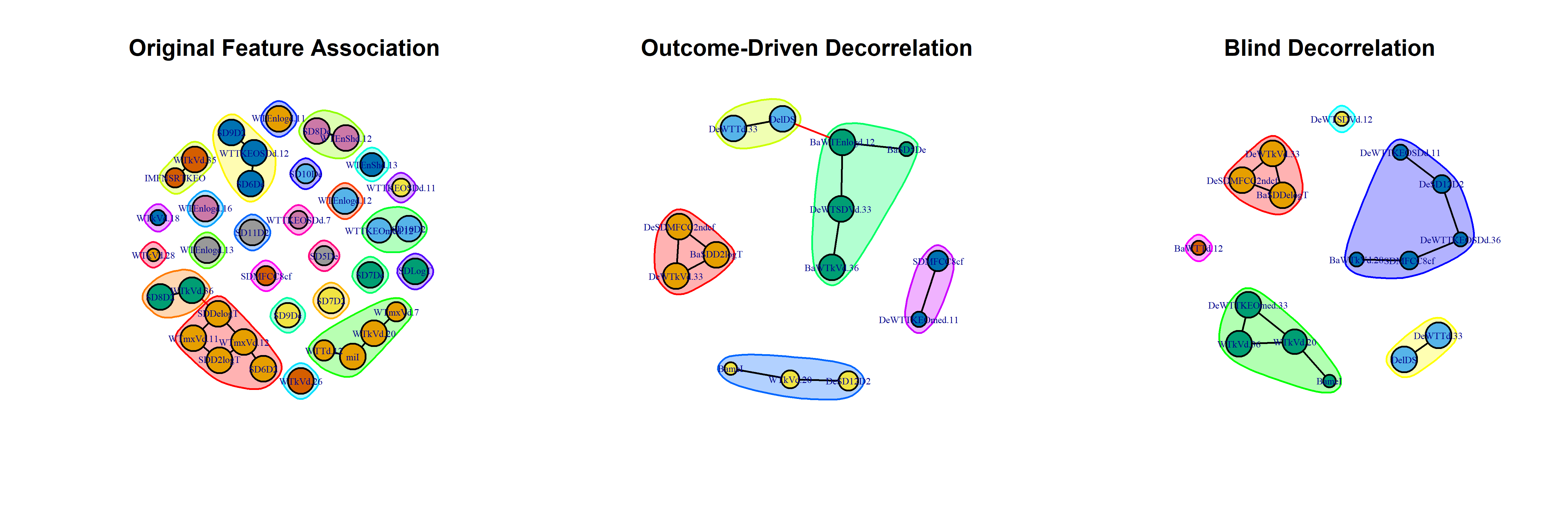
cmax <- apply(bm$bagging$formulaNetwork,2,max)  
cnames <- names(cmax[cmax>=0.5])  
cmax <- cmax[cmax>=0.5]  
adma <- bm$bagging$formulaNetwork[cnames,cnames]  
  
for (cx in c(1:nrow(namecode)))  
{  
 cnames <- str\_replace\_all(cnames,namecode[cx,1],namecode[cx,2])  
}  
cnames <- str\_replace\_all(cnames,"\_","")  
cnames <- str\_replace\_all(cnames,"th","")  
rownames(adma) <- cnames  
colnames(adma) <- cnames  
names(cmax) <- cnames  
adma[adma<0.25] <- 0;  
gr <- graph\_from\_adjacency\_matrix(adma,mode = "undirected",diag = FALSE,weighted=TRUE)  
gr$layout <- layout\_with\_fr  
  
fc <- cluster\_optimal(gr)  
plot(fc, gr,  
 vertex.size=20\*cmax,  
 vertex.label.cex=0.5,  
 vertex.label.dist=0,  
 main="Original Feature Association")  
  
  
  
### The Outcome Driven Model  
  
pander::pander(nrow(bmd$bagging$formulaNetwork))

*44*

cmax <- apply(bmd$bagging$formulaNetwork,2,max)  
cnames <- names(cmax[cmax>=0.5])  
cmax <- cmax[cmax>=0.5]  
outcomeNames <- cnames  
adma <- bmd$bagging$formulaNetwork[cnames,cnames]  
  
for (cx in c(1:nrow(namecode)))  
{  
 cnames <- str\_replace\_all(cnames,namecode[cx,1],namecode[cx,2])  
}  
cnames <- str\_replace\_all(cnames,"\_","")  
cnames <- str\_replace\_all(cnames,"th","")  
rownames(adma) <- cnames  
colnames(adma) <- cnames  
names(cmax) <- cnames  
adma[adma<0.25] <- 0;  
gr <- graph\_from\_adjacency\_matrix(adma,mode = "undirected",diag = FALSE,weighted=TRUE)  
gr$layout <- layout\_with\_fr  
  
fc <- cluster\_optimal(gr)  
clusterOutcome <- fc  
clusterOutcome$names <- outcomeNames  
plot(fc, gr,  
 vertex.size=20\*cmax,  
 vertex.label.cex=0.5,  
 vertex.label.dist=0,  
 main="Outcome-Driven Decorrelation")  
  
  
### The Blind Decorrelation  
  
pander::pander(nrow(bmdU$bagging$formulaNetwork))

*46*

cmax <- apply(bmdU$bagging$formulaNetwork,2,max)  
cnames <- names(cmax[cmax>=0.5])  
cmax <- cmax[cmax>=0.5]  
adma <- bmdU$bagging$formulaNetwork[cnames,cnames]  
  
for (cx in c(1:nrow(namecode)))  
{  
 cnames <- str\_replace\_all(cnames,namecode[cx,1],namecode[cx,2])  
}  
cnames <- str\_replace\_all(cnames,"\_","")  
cnames <- str\_replace\_all(cnames,"th","")  
rownames(adma) <- cnames  
colnames(adma) <- cnames  
names(cmax) <- cnames  
adma[adma<0.25] <- 0;  
gr <- graph\_from\_adjacency\_matrix(adma,mode = "undirected",diag = FALSE,weighted=TRUE)  
gr$layout <- layout\_with\_fr  
  
fc <- cluster\_optimal(gr)  
plot(fc, gr,  
 vertex.size=20\*cmax,  
 vertex.label.cex=0.5,  
 vertex.label.dist=0,  
 main="Blind Decorrelation")



### Feature Analysis of Models

The analysis of the features required to predict the outcome will use the following:

1. Analysis of the BSWiMS bagged model using the summary function.
2. Analysis of the sparse GDSMT
3. Analysis of the univariate association of the model features of both models
4. Report the new features not found by the Original data analysis

par(op)  
par(mfrow=c(1,1))  
## 1 Get the Model Features  
smOriginal <- summary(bm)  
rawnames <- rownames(smOriginal$coefficients)  
  
### From Drived Decorrelation  
smDecor <- summary(bmd)  
decornames <- rownames(smDecor$coefficients)  
  
### From Blind Decorrelation  
smDecorU <- summary(bmdU)  
decornamesU <- rownames(smDecorU$coefficients)  
  
  
  
## 2 Get the decorrelation matrix formulas  
dc <- getDerivedCoefficients(deTrain)  
### 2a Get only the ones that were decorrelated by the decorrelation-based model  
deNames\_in\_dc <- decornames[decornames %in% names(dc)]  
selectedlist <- dc[deNames\_in\_dc]  
theDeFormulas <- selectedlist  
  
pander::pander(selectedlist)

* **De\_tqwt\_energy\_dec\_33**:

| * tqwt\_energy\_dec\_31 | * tqwt\_energy\_dec\_33 |
| --- | --- |
| * -0.907 | * 1 |

* **De\_std\_MFCC\_2nd\_coef**:

| * std\_MFCC\_2nd\_coef | * std\_2nd\_delta |
| --- | --- |
| * 1 | * -0.828 |

* **De\_tqwt\_stdValue\_dec\_33**:

| * tqwt\_TKEO\_mean\_dec\_35 | * tqwt\_stdValue\_dec\_32 | * tqwt\_stdValue\_dec\_33 |
| --- | --- | --- |
| * -0.131 | * -0.715 | * 1 |

* **De\_tqwt\_TKEO\_std\_dec\_11**:

| * tqwt\_TKEO\_mean\_dec\_10 | * tqwt\_TKEO\_mean\_dec\_11 | * tqwt\_TKEO\_std\_dec\_11 | * tqwt\_stdValue\_dec\_10 | * tqwt\_minValue\_dec\_17 |
| --- | --- | --- | --- | --- |
| * 0.713 | * -0.644 | * 1 | * -1.64 | * 0.314 |

* **De\_std\_12th\_delta\_delta**:

| * std\_MFCC\_12th\_coef | * std\_12th\_delta\_delta |
| --- | --- |
| * -0.914 | * 1 |

* **De\_tqwt\_TKEO\_std\_dec\_36**:

| * tqwt\_TKEO\_mean\_dec\_35 | * tqwt\_TKEO\_std\_dec\_36 |
| --- | --- |
| * -0.926 | * 1 |

* **De\_locDbShimmer**:

| * locDbShimmer | * ddaShimmer |
| --- | --- |
| * 1 | * -0.963 |

* **De\_tqwt\_kurtosisValue\_dec\_33**:

| * tqwt\_kurtosisValue\_dec\_32 | * tqwt\_kurtosisValue\_dec\_33 |
| --- | --- |
| * -0.883 | * 1 |

* **De\_tqwt\_TKEO\_mean\_dec\_17**:

| * tqwt\_TKEO\_mean\_dec\_17 | * tqwt\_minValue\_dec\_17 |
| --- | --- |
| * 1 | * 2.32 |

* **De\_tqwt\_entropy\_log\_dec\_1**:

| * tqwt\_entropy\_log\_dec\_1 | * tqwt\_entropy\_log\_dec\_2 |
| --- | --- |
| * 1 | * -0.849 |

* **De\_app\_LT\_TKEO\_std\_9\_coef**:

| * Ea | * app\_det\_TKEO\_mean\_9\_coef | * app\_LT\_entropy\_log\_8\_coef | * app\_LT\_TKEO\_std\_9\_coef |
| --- | --- | --- | --- |
| * -1.94 | * -0.0885 | * -4.8 | * 1 |

* **De\_tqwt\_minValue\_dec\_7**:

| * tqwt\_minValue\_dec\_7 | * tqwt\_minValue\_dec\_17 | * tqwt\_maxValue\_dec\_8 |
| --- | --- | --- |
| * 1 | * -0.0186 | * 0.967 |

* **De\_det\_LT\_TKEO\_mean\_3\_coef**:

| * Ed2\_3\_coef | * det\_LT\_TKEO\_mean\_1\_coef | * det\_LT\_TKEO\_mean\_3\_coef |
| --- | --- | --- |
| * -0.89 | * -0.0834 | * 1 |

* **De\_tqwt\_TKEO\_std\_dec\_7**:

| * tqwt\_TKEO\_std\_dec\_7 | * tqwt\_stdValue\_dec\_7 | * tqwt\_minValue\_dec\_17 |
| --- | --- | --- |
| * 1 | * -1.66 | * 0.145 |

names(selectedlist) <- NULL  
### 2b Get the the names of the original features  
  
allDevar <- unique(c(names(unlist(selectedlist)),decornames))  
allDevar <- allDevar[!str\_detect(allDevar,"De\_")]  
allDevar <- str\_remove(allDevar,"Ba\_")  
allDevar <- unique(allDevar)  
  
  
# The analysis of the blind decorrelation  
  
dcU <- getDerivedCoefficients(deTrainU)  
### 2a Get only the ones that were decorrelated by the decorrelation-based model  
deNames\_in\_dcU <- decornamesU[decornamesU %in% names(dcU)]  
selectedlistU <- dcU[deNames\_in\_dcU]  
pander::pander(selectedlistU)

* **De\_tqwt\_TKEO\_mean\_dec\_17**:

| * tqwt\_TKEO\_mean\_dec\_17 | * tqwt\_minValue\_dec\_17 |
| --- | --- |
| * 1 | * 2.32 |

* **De\_tqwt\_energy\_dec\_33**:

| * tqwt\_energy\_dec\_31 | * tqwt\_energy\_dec\_33 |
| --- | --- |
| * -0.907 | * 1 |

* **De\_locDbShimmer**:

| * locDbShimmer | * ddaShimmer |
| --- | --- |
| * 1 | * -0.963 |

* **De\_std\_MFCC\_2nd\_coef**:

| * std\_MFCC\_2nd\_coef | * std\_2nd\_delta |
| --- | --- |
| * 1 | * -0.828 |

* **De\_tqwt\_maxValue\_dec\_28**:

| * tqwt\_stdValue\_dec\_28 | * tqwt\_maxValue\_dec\_28 |
| --- | --- |
| * -0.703 | * 1 |

* **De\_tqwt\_entropy\_log\_dec\_1**:

| * tqwt\_entropy\_log\_dec\_1 | * tqwt\_entropy\_log\_dec\_2 |
| --- | --- |
| * 1 | * -0.849 |

* **De\_tqwt\_kurtosisValue\_dec\_33**:

| * tqwt\_kurtosisValue\_dec\_32 | * tqwt\_kurtosisValue\_dec\_33 |
| --- | --- |
| * -0.883 | * 1 |

* **De\_tqwt\_TKEO\_std\_dec\_11**:

| * tqwt\_TKEO\_std\_dec\_11 | * tqwt\_stdValue\_dec\_10 | * tqwt\_minValue\_dec\_17 |
| --- | --- | --- |
| * 1 | * -1.64 | * 0.314 |

* **De\_tqwt\_TKEO\_std\_dec\_36**:

| * tqwt\_TKEO\_mean\_dec\_35 | * tqwt\_TKEO\_std\_dec\_36 |
| --- | --- |
| * -0.926 | * 1 |

* **De\_tqwt\_maxValue\_dec\_29**:

| * tqwt\_stdValue\_dec\_28 | * tqwt\_maxValue\_dec\_29 |
| --- | --- |
| * -0.555 | * 1 |

* **De\_tqwt\_entropy\_log\_dec\_35**:

| * tqwt\_entropy\_log\_dec\_35 | * tqwt\_TKEO\_mean\_dec\_35 |
| --- | --- |
| * 1 | * -0.106 |

* **De\_tqwt\_TKEO\_mean\_dec\_33**:

| * tqwt\_TKEO\_mean\_dec\_33 | * tqwt\_TKEO\_mean\_dec\_35 | * tqwt\_stdValue\_dec\_32 |
| --- | --- | --- |
| * 1 | * -0.248 | * -1.49 |

* **De\_tqwt\_TKEO\_std\_dec\_7**:

| * tqwt\_TKEO\_std\_dec\_7 | * tqwt\_stdValue\_dec\_7 | * tqwt\_minValue\_dec\_17 |
| --- | --- | --- |
| * 1 | * -1.66 | * 0.145 |

* **De\_det\_LT\_TKEO\_mean\_3\_coef**:

| * Ed2\_3\_coef | * det\_LT\_TKEO\_mean\_1\_coef | * det\_LT\_TKEO\_mean\_3\_coef |
| --- | --- | --- |
| * -0.89 | * -0.0834 | * 1 |

* **De\_tqwt\_minValue\_dec\_7**:

| * tqwt\_minValue\_dec\_7 | * tqwt\_minValue\_dec\_17 | * tqwt\_maxValue\_dec\_8 |
| --- | --- | --- |
| * 1 | * -0.0186 | * 0.967 |

* **De\_tqwt\_TKEO\_std\_dec\_17**:

| * tqwt\_TKEO\_std\_dec\_17 | * tqwt\_minValue\_dec\_17 |
| --- | --- |
| * 1 | * 2.13 |

names(selectedlistU) <- NULL  
### 2b Get the the names of the original features  
  
allDevarU <- unique(c(names(unlist(selectedlistU)),decornamesU))  
allDevarU <- allDevarU[!str\_detect(allDevarU,"De\_")]  
allDevarU <- str\_remove(allDevarU,"Ba\_")  
allDevarU <- unique(allDevarU)  
  
pander::pander(c(length(rawnames),length(decornames),length(decornamesU)))

*63*, *32* and *33*

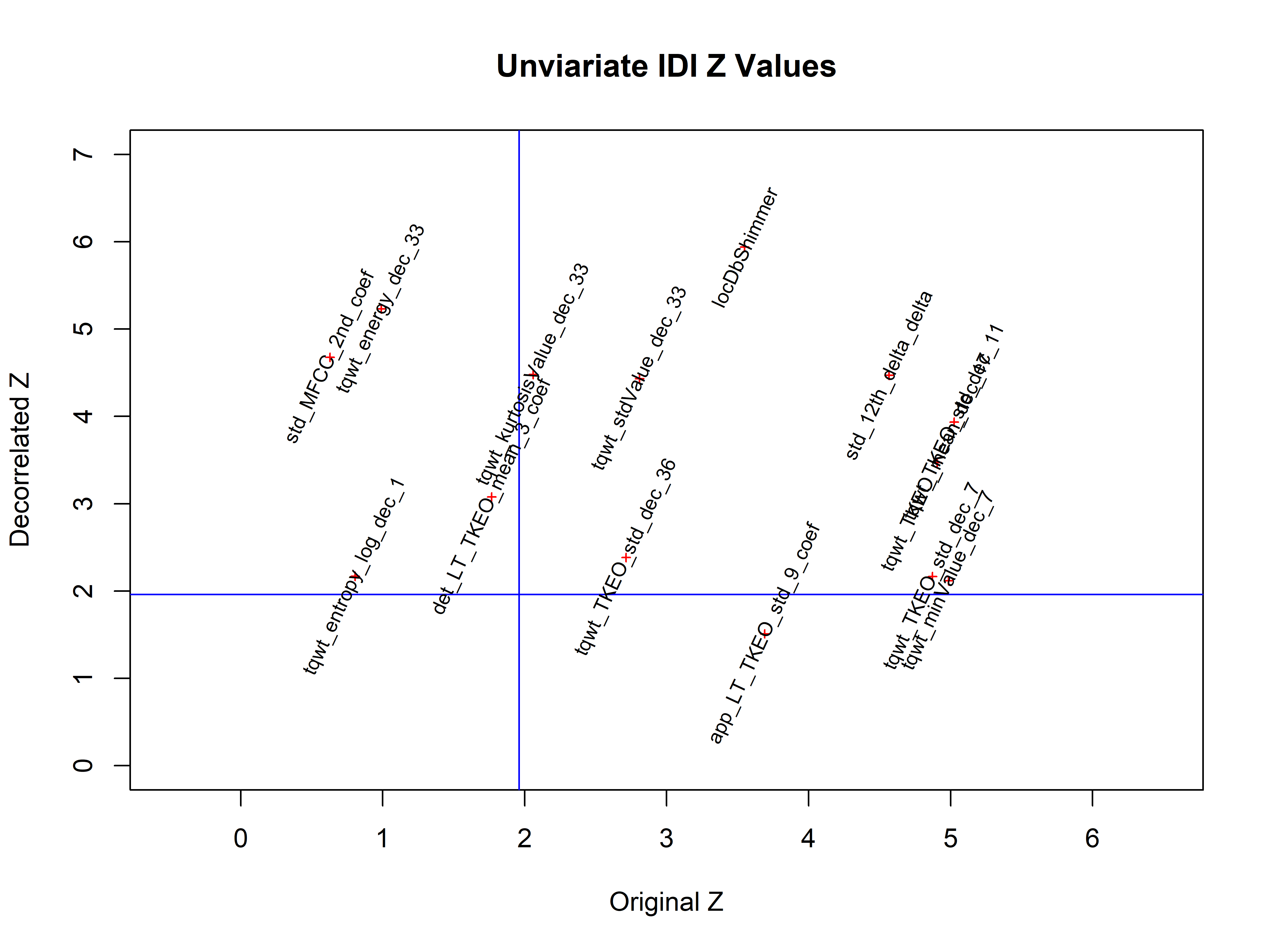
pander::pander(c(length(rawnames),length(allDevar),length(allDevarU)))

*63*, *48* and *44*

### 2c Get only the new feautres not found in the original analysis  
dvar <- allDevar[!(allDevar %in% rawnames)]   
  
### 2d Get the decorrelated variables that have new features  
newvars <- character();  
for (cvar in deNames\_in\_dc)  
{  
 lvar <- dc[cvar]  
 names(lvar) <- NULL  
 lvar <- names(unlist(lvar))  
 if (length(lvar[lvar %in% dvar]) > 0)  
 {  
 newvars <- append(newvars,cvar)  
 }  
}  
  
## 3 Here is the univariate z values of the orignal set  
#pander::pander(bm$univariate[dvar,])  
## 4 Here is the univariate z values of the decorrelated set  
#pander::pander(bmd$univariate[newvars,])  
  
## 4a The scater plot of the decorrelated vs original Univariate values  
  
zvalueNew <- bmd$univariate[newvars,]  
rownames(zvalueNew) <- str\_remove(rownames(zvalueNew),"De\_")  
rownames(zvalueNew) <- str\_remove(rownames(zvalueNew),"Ba\_")  
  
zvaluePrePost <- bm$univariate[rownames(zvalueNew),c(1,3)]  
zvaluePrePost$Name <- NULL  
zvaluePrePost$NewZ <- zvalueNew[rownames(zvaluePrePost),"ZUni"]  
pander::pander(zvaluePrePost)

|  | ZUni | NewZ |
| --- | --- | --- |
| **tqwt\_energy\_dec\_33** | 0.990 | 5.23 |
| **std\_MFCC\_2nd\_coef** | 0.628 | 4.68 |
| **tqwt\_stdValue\_dec\_33** | 2.810 | 4.43 |
| **tqwt\_TKEO\_std\_dec\_11** | 5.025 | 3.94 |
| **std\_12th\_delta\_delta** | 4.566 | 4.47 |
| **tqwt\_TKEO\_std\_dec\_36** | 2.714 | 2.38 |
| **locDbShimmer** | 3.548 | 5.94 |
| **tqwt\_kurtosisValue\_dec\_33** | 2.060 | 4.48 |
| **tqwt\_TKEO\_mean\_dec\_17** | 4.905 | 3.47 |
| **tqwt\_entropy\_log\_dec\_1** | 0.805 | 2.17 |
| **app\_LT\_TKEO\_std\_9\_coef** | 3.690 | 1.51 |
| **tqwt\_minValue\_dec\_7** | 4.986 | 2.12 |
| **det\_LT\_TKEO\_mean\_3\_coef** | 1.767 | 3.08 |
| **tqwt\_TKEO\_std\_dec\_7** | 4.872 | 2.17 |

plot(zvaluePrePost,  
 xlim=c(-0.5,6.5),  
 ylim=c(0,7),  
 xlab="Original Z",  
 ylab="Decorrelated Z",  
 main="Unviariate IDI Z Values",  
 pch=3,cex=0.5,  
 col="red")  
abline(v=1.96,col="blue")  
abline(h=1.96,col="blue")  
text(zvaluePrePost$ZUni,zvaluePrePost$NewZ,rownames(zvaluePrePost),srt=65,cex=0.75)



### The Summary of the Decorrelated-based Model

Here I will print the summary statistics of the Logistic models found by BSWiMS, using the original and transformed dataset. After that, I will show the characteristics of the features not found by the original analysis.

pander::pander(smOriginal$coefficients)

|  | Estimate | lower | OR | upper | u.Accuracy | r.Accuracy | full.Accuracy | u.AUC | r.AUC | full.AUC | IDI | NRI | z.IDI | z.NRI | Frequency |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **minIntensity** | -0.782786 | 0.368 | 0.457 | 0.569 | 0.610 | 0.694 | 0.697 | 0.681 | 0.672 | 0.734 | 0.1421 | 0.680 | 6.29 | 5.98 | 1.00 |
| **tqwt\_entropy\_shannon\_dec\_16** | -0.018914 | 0.976 | 0.981 | 0.986 | 0.676 | 0.687 | 0.688 | 0.733 | 0.590 | 0.736 | 0.1364 | 0.770 | 6.15 | 6.92 | 0.45 |
| **std\_6th\_delta\_delta** | 0.330533 | 1.258 | 1.392 | 1.540 | 0.671 | 0.715 | 0.756 | 0.702 | 0.727 | 0.776 | 0.1375 | 0.707 | 6.11 | 6.00 | 1.00 |
| **tqwt\_stdValue\_dec\_11** | -0.005482 | 0.993 | 0.995 | 0.996 | 0.672 | 0.693 | 0.707 | 0.679 | 0.622 | 0.723 | 0.1307 | 0.625 | 6.07 | 5.19 | 0.15 |
| **std\_delta\_delta\_log\_energy** | 0.111852 | 1.080 | 1.118 | 1.159 | 0.716 | 0.712 | 0.753 | 0.725 | 0.725 | 0.768 | 0.1334 | 0.694 | 6.03 | 5.89 | 1.00 |
| **tqwt\_kurtosisValue\_dec\_28** | -0.018964 | 0.975 | 0.981 | 0.987 | 0.679 | 0.676 | 0.719 | 0.672 | 0.676 | 0.730 | 0.1320 | 0.706 | 5.99 | 5.96 | 0.50 |
| **std\_8th\_delta\_delta** | 0.312373 | 1.248 | 1.367 | 1.497 | 0.690 | 0.689 | 0.725 | 0.713 | 0.705 | 0.745 | 0.1300 | 0.739 | 5.84 | 6.33 | 1.00 |
| **tqwt\_entropy\_log\_dec\_11** | -0.284056 | 0.694 | 0.753 | 0.817 | 0.696 | 0.642 | 0.729 | 0.694 | 0.668 | 0.746 | 0.1209 | 0.758 | 5.83 | 6.52 | 1.00 |
| **std\_11th\_delta\_delta** | 0.298860 | 1.244 | 1.348 | 1.461 | 0.687 | 0.642 | 0.730 | 0.707 | 0.668 | 0.745 | 0.1258 | 0.621 | 5.76 | 5.26 | 1.00 |
| **tqwt\_maxValue\_dec\_10** | -0.008753 | 0.988 | 0.991 | 0.994 | 0.661 | 0.699 | 0.715 | 0.691 | 0.665 | 0.723 | 0.1180 | 0.731 | 5.67 | 6.31 | 0.15 |
| **std\_delta\_log\_energy** | 0.097553 | 1.066 | 1.102 | 1.140 | 0.708 | 0.706 | 0.746 | 0.715 | 0.719 | 0.766 | 0.1189 | 0.610 | 5.66 | 5.07 | 1.00 |
| **IMF\_SNR\_entropy** | 0.015935 | 1.011 | 1.016 | 1.021 | 0.623 | 0.714 | 0.757 | 0.612 | 0.729 | 0.773 | 0.1214 | 0.721 | 5.65 | 6.17 | 0.15 |
| **tqwt\_TKEO\_mean\_dec\_11** | -0.002147 | 0.997 | 0.998 | 0.999 | 0.636 | 0.637 | 0.730 | 0.644 | 0.635 | 0.738 | 0.1152 | 0.543 | 5.61 | 4.44 | 0.10 |
| **tqwt\_TKEO\_std\_dec\_6** | -0.006453 | 0.991 | 0.994 | 0.996 | 0.654 | 0.676 | 0.708 | 0.664 | 0.700 | 0.725 | 0.1183 | 0.745 | 5.61 | 6.39 | 0.15 |
| **std\_6th\_delta** | 0.228776 | 1.161 | 1.257 | 1.361 | 0.636 | 0.712 | 0.732 | 0.664 | 0.713 | 0.754 | 0.1181 | 0.611 | 5.57 | 5.08 | 1.00 |
| **std\_Log\_energy** | 0.107579 | 1.076 | 1.114 | 1.153 | 0.666 | 0.702 | 0.710 | 0.689 | 0.670 | 0.723 | 0.1196 | 0.552 | 5.53 | 4.65 | 1.00 |
| **tqwt\_TKEO\_std\_dec\_12** | -0.023172 | 0.970 | 0.977 | 0.985 | 0.726 | 0.669 | 0.727 | 0.720 | 0.696 | 0.747 | 0.1113 | 0.714 | 5.51 | 6.12 | 1.00 |
| **std\_7th\_delta\_delta** | 0.227305 | 1.166 | 1.255 | 1.352 | 0.689 | 0.716 | 0.722 | 0.716 | 0.712 | 0.734 | 0.1187 | 0.719 | 5.49 | 6.16 | 1.00 |
| **std\_8th\_delta** | 0.236333 | 1.169 | 1.267 | 1.372 | 0.663 | 0.717 | 0.722 | 0.685 | 0.702 | 0.735 | 0.1187 | 0.687 | 5.47 | 5.78 | 1.00 |
| **tqwt\_TKEO\_mean\_dec\_12** | -0.016535 | 0.978 | 0.984 | 0.989 | 0.728 | 0.678 | 0.721 | 0.708 | 0.669 | 0.730 | 0.1119 | 0.675 | 5.47 | 5.69 | 0.95 |
| **std\_7th\_delta** | 0.206918 | 1.146 | 1.230 | 1.320 | 0.675 | 0.685 | 0.721 | 0.709 | 0.694 | 0.742 | 0.1133 | 0.760 | 5.39 | 6.59 | 1.00 |
| **tqwt\_TKEO\_std\_dec\_7** | -0.019201 | 0.975 | 0.981 | 0.987 | 0.643 | 0.657 | 0.702 | 0.676 | 0.682 | 0.730 | 0.1078 | 0.614 | 5.37 | 5.13 | 0.70 |
| **tqwt\_TKEO\_std\_dec\_11** | -0.016172 | 0.979 | 0.984 | 0.989 | 0.661 | 0.678 | 0.717 | 0.678 | 0.668 | 0.743 | 0.1061 | 0.550 | 5.37 | 4.55 | 0.70 |
| **tqwt\_maxValue\_dec\_11** | -0.064874 | 0.916 | 0.937 | 0.958 | 0.693 | 0.698 | 0.742 | 0.706 | 0.714 | 0.767 | 0.1008 | 0.521 | 5.37 | 4.30 | 1.00 |
| **tqwt\_maxValue\_dec\_12** | -0.061126 | 0.922 | 0.941 | 0.960 | 0.713 | 0.701 | 0.745 | 0.733 | 0.715 | 0.768 | 0.0986 | 0.613 | 5.30 | 5.18 | 1.00 |
| **std\_12th\_delta\_delta** | 0.015046 | 1.010 | 1.015 | 1.021 | 0.661 | 0.662 | 0.690 | 0.669 | 0.676 | 0.699 | 0.1028 | 0.630 | 5.29 | 5.22 | 0.10 |
| **tqwt\_maxValue\_dec\_7** | -0.044861 | 0.941 | 0.956 | 0.971 | 0.641 | 0.671 | 0.710 | 0.680 | 0.686 | 0.736 | 0.1055 | 0.689 | 5.26 | 5.86 | 0.75 |
| **tqwt\_energy\_dec\_6** | -0.013198 | 0.982 | 0.987 | 0.992 | 0.604 | 0.687 | 0.747 | 0.632 | 0.711 | 0.764 | 0.1039 | 0.691 | 5.25 | 5.84 | 0.25 |
| **tqwt\_entropy\_shannon\_dec\_13** | -0.022752 | 0.970 | 0.978 | 0.985 | 0.689 | 0.664 | 0.702 | 0.713 | 0.677 | 0.729 | 0.1005 | 0.679 | 5.21 | 5.77 | 0.90 |
| **tqwt\_entropy\_log\_dec\_16** | -0.100677 | 0.874 | 0.904 | 0.935 | 0.689 | 0.679 | 0.713 | 0.697 | 0.665 | 0.735 | 0.1011 | 0.620 | 5.18 | 5.17 | 1.00 |
| **tqwt\_entropy\_log\_dec\_12** | -0.151213 | 0.813 | 0.860 | 0.909 | 0.740 | 0.691 | 0.738 | 0.731 | 0.712 | 0.749 | 0.0967 | 0.720 | 5.17 | 6.13 | 1.00 |
| **std\_9th\_delta** | 0.176158 | 1.119 | 1.193 | 1.271 | 0.667 | 0.693 | 0.714 | 0.704 | 0.698 | 0.731 | 0.1051 | 0.665 | 5.16 | 5.65 | 0.95 |
| **std\_9th\_delta\_delta** | 0.203441 | 1.139 | 1.226 | 1.319 | 0.691 | 0.716 | 0.729 | 0.720 | 0.717 | 0.744 | 0.1013 | 0.612 | 5.10 | 5.11 | 1.00 |
| **tqwt\_kurtosisValue\_dec\_18** | 0.134140 | 1.090 | 1.144 | 1.200 | 0.652 | 0.666 | 0.707 | 0.676 | 0.687 | 0.720 | 0.1016 | 0.671 | 5.08 | 5.66 | 0.60 |
| **tqwt\_kurtosisValue\_dec\_27** | -0.007659 | 0.989 | 0.992 | 0.995 | 0.691 | 0.649 | 0.718 | 0.667 | 0.675 | 0.719 | 0.0969 | 0.591 | 5.06 | 4.93 | 0.30 |
| **tqwt\_kurtosisValue\_dec\_34** | 0.009973 | 1.006 | 1.010 | 1.014 | 0.632 | 0.657 | 0.716 | 0.646 | 0.684 | 0.725 | 0.0923 | 0.654 | 5.02 | 5.48 | 0.15 |
| **tqwt\_entropy\_shannon\_dec\_12** | -0.019618 | 0.974 | 0.981 | 0.988 | 0.720 | 0.667 | 0.713 | 0.718 | 0.696 | 0.728 | 0.0959 | 0.652 | 5.02 | 5.49 | 1.00 |
| **std\_MFCC\_8th\_coef** | 0.142999 | 1.092 | 1.154 | 1.219 | 0.654 | 0.679 | 0.708 | 0.673 | 0.686 | 0.732 | 0.0968 | 0.659 | 4.97 | 5.52 | 0.75 |
| **std\_10th\_delta\_delta** | 0.173992 | 1.115 | 1.190 | 1.270 | 0.679 | 0.707 | 0.719 | 0.691 | 0.697 | 0.730 | 0.0992 | 0.592 | 4.95 | 4.93 | 1.00 |
| **std\_5th\_delta** | 0.169082 | 1.109 | 1.184 | 1.265 | 0.621 | 0.692 | 0.722 | 0.655 | 0.707 | 0.751 | 0.0942 | 0.510 | 4.93 | 4.20 | 0.75 |
| **IMF\_NSR\_TKEO** | -0.092081 | 0.880 | 0.912 | 0.945 | 0.621 | 0.725 | 0.766 | 0.591 | 0.741 | 0.775 | 0.0872 | 0.539 | 4.89 | 4.52 | 0.75 |
| **IMF\_SNR\_SEO** | 0.003881 | 1.002 | 1.004 | 1.005 | 0.606 | 0.728 | 0.765 | 0.603 | 0.735 | 0.775 | 0.0932 | 0.564 | 4.89 | 4.63 | 0.10 |
| **tqwt\_entropy\_log\_dec\_13** | -0.111220 | 0.857 | 0.895 | 0.934 | 0.710 | 0.668 | 0.717 | 0.712 | 0.697 | 0.735 | 0.0888 | 0.664 | 4.88 | 5.60 | 1.00 |
| **tqwt\_energy\_dec\_5** | -0.004147 | 0.994 | 0.996 | 0.998 | 0.577 | 0.700 | 0.754 | 0.584 | 0.720 | 0.758 | 0.0915 | 0.695 | 4.88 | 5.86 | 0.10 |
| **tqwt\_kurtosisValue\_dec\_17** | 0.059488 | 1.039 | 1.061 | 1.084 | 0.638 | 0.665 | 0.689 | 0.676 | 0.649 | 0.712 | 0.0924 | 0.550 | 4.87 | 4.61 | 0.30 |
| **tqwt\_maxValue\_dec\_13** | -0.012443 | 0.983 | 0.988 | 0.992 | 0.679 | 0.683 | 0.693 | 0.713 | 0.646 | 0.721 | 0.0855 | 0.564 | 4.84 | 4.68 | 0.45 |
| **tqwt\_entropy\_log\_dec\_33** | -0.018697 | 0.974 | 0.981 | 0.989 | 0.637 | 0.685 | 0.743 | 0.619 | 0.693 | 0.750 | 0.0866 | 0.444 | 4.83 | 3.62 | 0.10 |
| **tqwt\_kurtosisValue\_dec\_36** | 0.042288 | 1.027 | 1.043 | 1.059 | 0.678 | 0.708 | 0.757 | 0.710 | 0.723 | 0.766 | 0.0907 | 0.740 | 4.81 | 6.34 | 1.00 |
| **std\_MFCC\_6th\_coef** | 0.119264 | 1.074 | 1.127 | 1.182 | 0.580 | 0.685 | 0.733 | 0.603 | 0.695 | 0.755 | 0.0924 | 0.467 | 4.79 | 3.81 | 0.45 |
| **tqwt\_energy\_dec\_12** | -0.018251 | 0.975 | 0.982 | 0.989 | 0.675 | 0.647 | 0.712 | 0.674 | 0.688 | 0.730 | 0.0818 | 0.592 | 4.78 | 4.90 | 0.80 |
| **tqwt\_kurtosisValue\_dec\_26** | -0.031938 | 0.956 | 0.969 | 0.981 | 0.762 | 0.657 | 0.719 | 0.662 | 0.689 | 0.715 | 0.0883 | 0.574 | 4.76 | 5.12 | 1.00 |
| **app\_TKEO\_std\_1\_coef** | -0.010207 | 0.986 | 0.990 | 0.994 | 0.629 | 0.706 | 0.736 | 0.624 | 0.726 | 0.751 | 0.0776 | 0.633 | 4.76 | 5.26 | 0.10 |
| **std\_10th\_delta** | 0.108438 | 1.067 | 1.115 | 1.164 | 0.656 | 0.692 | 0.711 | 0.679 | 0.698 | 0.732 | 0.0886 | 0.426 | 4.74 | 3.48 | 0.75 |
| **tqwt\_kurtosisValue\_dec\_20** | 0.201329 | 1.128 | 1.223 | 1.326 | 0.673 | 0.658 | 0.716 | 0.690 | 0.696 | 0.740 | 0.0860 | 0.660 | 4.74 | 5.56 | 1.00 |
| **tqwt\_maxValue\_dec\_1** | -0.007378 | 0.990 | 0.993 | 0.996 | 0.649 | 0.675 | 0.695 | 0.681 | 0.670 | 0.703 | 0.0849 | 0.549 | 4.65 | 4.58 | 0.20 |
| **tqwt\_entropy\_log\_dec\_34** | -0.050034 | 0.932 | 0.951 | 0.971 | 0.675 | 0.692 | 0.723 | 0.663 | 0.693 | 0.737 | 0.0828 | 0.646 | 4.62 | 5.39 | 0.30 |
| **tqwt\_kurtosisValue\_dec\_35** | 0.066920 | 1.040 | 1.069 | 1.100 | 0.642 | 0.707 | 0.756 | 0.668 | 0.721 | 0.770 | 0.0777 | 0.612 | 4.55 | 5.10 | 1.00 |
| **tqwt\_entropy\_log\_dec\_35** | -0.015453 | 0.978 | 0.985 | 0.991 | 0.661 | 0.673 | 0.705 | 0.646 | 0.688 | 0.717 | 0.0785 | 0.575 | 4.53 | 4.74 | 0.15 |
| **tqwt\_meanValue\_dec\_11** | 0.000372 | 1.000 | 1.000 | 1.001 | 0.613 | 0.692 | 0.722 | 0.653 | 0.715 | 0.739 | 0.0639 | 0.609 | 4.28 | 5.09 | 0.20 |
| **std\_MFCC\_7th\_coef** | 0.015390 | 1.008 | 1.016 | 1.023 | 0.585 | 0.723 | 0.745 | 0.616 | 0.723 | 0.753 | 0.0686 | 0.469 | 4.01 | 3.85 | 0.10 |
| **tqwt\_kurtosisValue\_dec\_33** | 0.024874 | 1.013 | 1.025 | 1.037 | 0.533 | 0.693 | 0.728 | 0.582 | 0.708 | 0.730 | 0.0607 | 0.532 | 3.99 | 4.43 | 0.35 |
| **app\_LT\_TKEO\_mean\_7\_coef** | -0.024754 | 0.964 | 0.976 | 0.987 | 0.660 | 0.644 | 0.695 | 0.675 | 0.680 | 0.720 | 0.0598 | 0.503 | 3.95 | 4.11 | 0.10 |
| **tqwt\_kurtosisValue\_dec\_1** | -0.003684 | 0.995 | 0.996 | 0.998 | 0.623 | 0.759 | 0.787 | 0.620 | 0.771 | 0.789 | 0.0525 | 0.490 | 3.94 | 3.98 | 0.20 |

pander::pander(smDecor$coefficients)

|  | Estimate | lower | OR | upper | u.Accuracy | r.Accuracy | full.Accuracy | u.AUC | r.AUC | full.AUC | IDI | NRI | z.IDI | z.NRI | Frequency |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ba\_tqwt\_kurtosisValue\_dec\_36** | 0.163608 | 1.130 | 1.178 | 1.23e+00 | 0.679 | 0.683 | 0.763 | 0.708 | 0.692 | 0.766 | 0.1547 | 0.850 | 6.78 | 7.53 | 1.00 |
| **Ba\_tqwt\_kurtosisValue\_dec\_17** | 0.093066 | 1.071 | 1.098 | 1.12e+00 | 0.626 | 0.678 | 0.741 | 0.675 | 0.712 | 0.764 | 0.1502 | 0.743 | 6.71 | 6.39 | 0.10 |
| **Ba\_std\_delta\_delta\_log\_energy** | 0.447243 | 1.376 | 1.564 | 1.78e+00 | 0.719 | 0.758 | 0.814 | 0.725 | 0.752 | 0.815 | 0.1500 | 0.776 | 6.59 | 6.66 | 1.00 |
| **Ba\_std\_5th\_delta** | 0.398259 | 1.319 | 1.489 | 1.68e+00 | 0.619 | 0.729 | 0.744 | 0.650 | 0.697 | 0.756 | 0.1321 | 0.646 | 6.01 | 5.46 | 0.55 |
| **Ba\_meanIntensity** | -1.440831 | 0.159 | 0.237 | 3.52e-01 | 0.610 | 0.714 | 0.709 | 0.690 | 0.671 | 0.748 | 0.1279 | 0.714 | 5.88 | 6.37 | 0.50 |
| **De\_tqwt\_energy\_dec\_33** | -0.214863 | 0.756 | 0.807 | 8.60e-01 | 0.748 | 0.721 | 0.779 | 0.734 | 0.739 | 0.777 | 0.1268 | 0.757 | 5.82 | 6.51 | 1.00 |
| **Ba\_locAbsJitter** | 0.040106 | 1.027 | 1.041 | 1.06e+00 | 0.639 | 0.680 | 0.730 | 0.652 | 0.708 | 0.745 | 0.1182 | 0.654 | 5.74 | 5.52 | 0.20 |
| **Ba\_tqwt\_kurtosisValue\_dec\_28** | -0.047188 | 0.939 | 0.954 | 9.69e-01 | 0.680 | 0.708 | 0.748 | 0.673 | 0.706 | 0.756 | 0.1205 | 0.718 | 5.61 | 6.09 | 0.40 |
| **De\_std\_MFCC\_2nd\_coef** | -1.473868 | 0.139 | 0.229 | 3.78e-01 | 0.702 | 0.782 | 0.820 | 0.687 | 0.775 | 0.820 | 0.1123 | 0.707 | 5.58 | 6.02 | 1.00 |
| **tqwt\_kurtosisValue\_dec\_20** | 0.530680 | 1.426 | 1.700 | 2.03e+00 | 0.663 | 0.699 | 0.755 | 0.685 | 0.714 | 0.770 | 0.1108 | 0.738 | 5.51 | 6.30 | 0.70 |
| **Ba\_tqwt\_energy\_dec\_12** | -0.029173 | 0.962 | 0.971 | 9.80e-01 | 0.672 | 0.690 | 0.747 | 0.674 | 0.698 | 0.758 | 0.1075 | 0.690 | 5.46 | 5.83 | 0.30 |
| **De\_tqwt\_stdValue\_dec\_33** | -0.899159 | 0.297 | 0.407 | 5.58e-01 | 0.685 | 0.721 | 0.782 | 0.686 | 0.733 | 0.774 | 0.1040 | 0.656 | 5.43 | 5.53 | 1.00 |
| **De\_tqwt\_TKEO\_std\_dec\_11** | -0.127279 | 0.842 | 0.880 | 9.21e-01 | 0.700 | 0.708 | 0.754 | 0.701 | 0.711 | 0.760 | 0.1079 | 0.860 | 5.41 | 7.53 | 0.25 |
| **Ba\_tqwt\_kurtosisValue\_dec\_26** | -0.019475 | 0.974 | 0.981 | 9.88e-01 | 0.766 | 0.695 | 0.752 | 0.665 | 0.708 | 0.764 | 0.1027 | 0.616 | 5.32 | 5.62 | 0.15 |
| **De\_std\_12th\_delta\_delta** | 0.733599 | 1.596 | 2.083 | 2.72e+00 | 0.694 | 0.727 | 0.764 | 0.675 | 0.729 | 0.768 | 0.1063 | 0.621 | 5.31 | 5.20 | 0.75 |
| **Ba\_tqwt\_entropy\_log\_dec\_12** | -0.499152 | 0.519 | 0.607 | 7.11e-01 | 0.741 | 0.720 | 0.776 | 0.731 | 0.723 | 0.774 | 0.1092 | 0.762 | 5.27 | 6.58 | 1.00 |
| **De\_tqwt\_TKEO\_mean\_dec\_11** | -0.031689 | 0.958 | 0.969 | 9.79e-01 | 0.639 | 0.655 | 0.727 | 0.646 | 0.668 | 0.750 | 0.1031 | 0.549 | 5.26 | 4.52 | 0.60 |
| **De\_tqwt\_TKEO\_std\_dec\_36** | 0.101535 | 1.067 | 1.107 | 1.15e+00 | 0.639 | 0.714 | 0.761 | 0.609 | 0.720 | 0.767 | 0.0938 | 0.701 | 5.22 | 6.00 | 0.40 |
| **De\_std\_MFCC\_6th\_coef** | 0.219560 | 1.148 | 1.246 | 1.35e+00 | 0.578 | 0.677 | 0.739 | 0.601 | 0.682 | 0.756 | 0.1002 | 0.600 | 5.16 | 5.02 | 0.30 |
| **De\_tqwt\_kurtosisValue\_dec\_27** | -0.007444 | 0.990 | 0.993 | 9.95e-01 | 0.690 | 0.664 | 0.760 | 0.666 | 0.671 | 0.752 | 0.0968 | 0.585 | 5.13 | 4.87 | 0.10 |
| **De\_locDbShimmer** | 1.091823 | 2.097 | 2.980 | 4.23e+00 | 0.712 | 0.758 | 0.785 | 0.738 | 0.745 | 0.780 | 0.0834 | 0.744 | 5.08 | 6.50 | 1.00 |
| **De\_tqwt\_kurtosisValue\_dec\_33** | 0.643638 | 1.490 | 1.903 | 2.43e+00 | 0.694 | 0.774 | 0.819 | 0.680 | 0.771 | 0.819 | 0.0847 | 0.536 | 5.06 | 4.41 | 1.00 |
| **std\_MFCC\_8th\_coef** | 0.480794 | 1.352 | 1.617 | 1.93e+00 | 0.654 | 0.683 | 0.742 | 0.675 | 0.693 | 0.763 | 0.0965 | 0.632 | 4.99 | 5.27 | 0.80 |
| **De\_tqwt\_TKEO\_mean\_dec\_17** | -0.052526 | 0.929 | 0.949 | 9.69e-01 | 0.641 | 0.648 | 0.722 | 0.668 | 0.702 | 0.737 | 0.0833 | 0.716 | 4.75 | 6.06 | 0.10 |
| **tqwt\_meanValue\_dec\_11** | 0.000963 | 1.001 | 1.001 | 1.00e+00 | 0.613 | 0.724 | 0.745 | 0.653 | 0.721 | 0.757 | 0.0860 | 0.654 | 4.67 | 5.54 | 0.20 |
| **De\_tqwt\_entropy\_log\_dec\_1** | 0.851185 | 1.650 | 2.342 | 3.32e+00 | 0.606 | 0.725 | 0.767 | 0.633 | 0.737 | 0.776 | 0.0809 | 0.580 | 4.60 | 4.85 | 0.20 |
| **De\_app\_LT\_TKEO\_std\_9\_coef** | 7.198084 | 57.084 | 1336.866 | 3.13e+04 | 0.649 | 0.736 | 0.762 | 0.578 | 0.766 | 0.781 | 0.0670 | 0.567 | 4.44 | 4.78 | 0.10 |
| **De\_tqwt\_minValue\_dec\_7** | 0.087481 | 1.051 | 1.091 | 1.13e+00 | 0.560 | 0.726 | 0.762 | 0.580 | 0.758 | 0.781 | 0.0754 | 0.715 | 4.41 | 6.03 | 0.10 |
| **De\_det\_LT\_TKEO\_mean\_3\_coef** | -0.151126 | 0.805 | 0.860 | 9.18e-01 | 0.610 | 0.697 | 0.735 | 0.624 | 0.712 | 0.750 | 0.0726 | 0.614 | 4.39 | 5.09 | 0.30 |
| **Ba\_IMF\_NSR\_TKEO** | -0.021154 | 0.970 | 0.979 | 9.88e-01 | 0.621 | 0.743 | 0.777 | 0.593 | 0.736 | 0.787 | 0.0683 | 0.495 | 4.32 | 4.09 | 0.10 |
| **De\_tqwt\_TKEO\_std\_dec\_7** | -0.090031 | 0.878 | 0.914 | 9.52e-01 | 0.599 | 0.704 | 0.738 | 0.593 | 0.732 | 0.760 | 0.0726 | 0.574 | 4.27 | 4.71 | 0.25 |
| **mean\_MFCC\_2nd\_coef** | 0.013013 | 1.007 | 1.013 | 1.02e+00 | 0.676 | 0.725 | 0.762 | 0.648 | 0.736 | 0.760 | 0.0650 | 0.533 | 3.88 | 4.38 | 0.10 |

pander::pander(smDecorU$coefficients)

|  | Estimate | lower | OR | upper | u.Accuracy | r.Accuracy | full.Accuracy | u.AUC | r.AUC | full.AUC | IDI | NRI | z.IDI | z.NRI | Frequency |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **tqwt\_kurtosisValue\_dec\_36** | 0.1793 | 1.149 | 1.196 | 1.245 | 0.678 | 0.680 | 0.757 | 0.709 | 0.692 | 0.764 | 0.1628 | 0.858 | 6.96 | 7.60 | 1.00 |
| **Ba\_std\_5th\_delta** | 0.3915 | 1.348 | 1.479 | 1.623 | 0.622 | 0.697 | 0.755 | 0.658 | 0.709 | 0.762 | 0.1538 | 0.759 | 6.63 | 6.56 | 0.35 |
| **Ba\_std\_delta\_log\_energy** | 0.4431 | 1.374 | 1.558 | 1.766 | 0.709 | 0.756 | 0.815 | 0.716 | 0.750 | 0.812 | 0.1425 | 0.707 | 6.40 | 6.01 | 1.00 |
| **tqwt\_kurtosisValue\_dec\_26** | -0.0223 | 0.971 | 0.978 | 0.985 | 0.758 | 0.703 | 0.731 | 0.660 | 0.714 | 0.736 | 0.1370 | 0.659 | 6.16 | 6.10 | 0.15 |
| **De\_tqwt\_TKEO\_mean\_dec\_17** | -0.0625 | 0.921 | 0.939 | 0.958 | 0.642 | 0.675 | 0.740 | 0.669 | 0.705 | 0.746 | 0.1311 | 0.826 | 6.11 | 7.18 | 0.10 |
| **Ba\_tqwt\_kurtosisValue\_dec\_17** | 0.1080 | 1.076 | 1.114 | 1.154 | 0.626 | 0.757 | 0.730 | 0.674 | 0.673 | 0.753 | 0.1258 | 0.732 | 5.95 | 6.25 | 0.15 |
| **De\_tqwt\_stdValue\_dec\_12** | -0.0670 | 0.919 | 0.935 | 0.952 | 0.725 | 0.744 | 0.757 | 0.721 | 0.658 | 0.756 | 0.1309 | 0.654 | 5.88 | 5.55 | 0.55 |
| **De\_tqwt\_energy\_dec\_33** | -0.1996 | 0.768 | 0.819 | 0.873 | 0.748 | 0.725 | 0.786 | 0.734 | 0.738 | 0.782 | 0.1252 | 0.733 | 5.79 | 6.27 | 1.00 |
| **De\_std\_MFCC\_6th\_coef** | 0.1717 | 1.120 | 1.187 | 1.258 | 0.582 | 0.675 | 0.732 | 0.603 | 0.667 | 0.735 | 0.1201 | 0.634 | 5.69 | 5.33 | 0.25 |
| **De\_std\_12th\_delta\_delta** | 0.5131 | 1.421 | 1.670 | 1.963 | 0.659 | 0.708 | 0.746 | 0.671 | 0.713 | 0.756 | 0.1216 | 0.754 | 5.64 | 6.51 | 0.70 |
| **De\_locDbShimmer** | 1.1817 | 2.238 | 3.260 | 4.747 | 0.717 | 0.755 | 0.790 | 0.737 | 0.745 | 0.785 | 0.0982 | 0.802 | 5.63 | 7.03 | 1.00 |
| **tqwt\_kurtosisValue\_dec\_27** | -0.0172 | 0.977 | 0.983 | 0.989 | 0.692 | 0.674 | 0.743 | 0.667 | 0.690 | 0.746 | 0.1128 | 0.689 | 5.53 | 5.80 | 0.20 |
| **Ba\_tqwt\_energy\_dec\_12** | -0.0575 | 0.927 | 0.944 | 0.962 | 0.670 | 0.688 | 0.752 | 0.672 | 0.707 | 0.766 | 0.1101 | 0.611 | 5.48 | 5.13 | 0.55 |
| **Ba\_tqwt\_kurtosisValue\_dec\_28** | -0.0560 | 0.929 | 0.946 | 0.962 | 0.681 | 0.710 | 0.742 | 0.674 | 0.696 | 0.750 | 0.1140 | 0.685 | 5.48 | 5.77 | 0.55 |
| **De\_std\_MFCC\_2nd\_coef** | -1.3311 | 0.165 | 0.264 | 0.423 | 0.703 | 0.765 | 0.820 | 0.688 | 0.760 | 0.816 | 0.1110 | 0.686 | 5.46 | 5.78 | 1.00 |
| **std\_MFCC\_8th\_coef** | 0.5012 | 1.379 | 1.651 | 1.976 | 0.656 | 0.720 | 0.744 | 0.673 | 0.705 | 0.753 | 0.1077 | 0.619 | 5.36 | 5.15 | 0.70 |
| **Ba\_meanIntensity** | -0.9124 | 0.291 | 0.402 | 0.554 | 0.609 | 0.686 | 0.721 | 0.690 | 0.696 | 0.755 | 0.0994 | 0.640 | 5.21 | 5.60 | 0.50 |
| **tqwt\_kurtosisValue\_dec\_20** | 0.5410 | 1.415 | 1.718 | 2.085 | 0.668 | 0.687 | 0.744 | 0.688 | 0.721 | 0.764 | 0.0994 | 0.701 | 5.20 | 5.94 | 0.95 |
| **tqwt\_meanValue\_dec\_11** | 0.0011 | 1.001 | 1.001 | 1.002 | 0.613 | 0.721 | 0.746 | 0.653 | 0.711 | 0.756 | 0.0970 | 0.623 | 5.00 | 5.24 | 0.20 |
| **De\_tqwt\_maxValue\_dec\_28** | -0.0278 | 0.962 | 0.973 | 0.983 | 0.644 | 0.680 | 0.726 | 0.647 | 0.684 | 0.742 | 0.0941 | 0.561 | 4.98 | 4.61 | 0.10 |
| **Ba\_IMF\_SNR\_SEO** | 0.0112 | 1.007 | 1.011 | 1.016 | 0.604 | 0.731 | 0.783 | 0.604 | 0.730 | 0.778 | 0.0866 | 0.469 | 4.92 | 3.85 | 0.10 |
| **De\_tqwt\_entropy\_log\_dec\_1** | 1.2906 | 2.203 | 3.635 | 5.997 | 0.605 | 0.726 | 0.765 | 0.630 | 0.732 | 0.770 | 0.0888 | 0.653 | 4.92 | 5.49 | 0.25 |
| **De\_tqwt\_kurtosisValue\_dec\_33** | 0.5700 | 1.415 | 1.768 | 2.210 | 0.689 | 0.763 | 0.820 | 0.678 | 0.766 | 0.816 | 0.0799 | 0.510 | 4.89 | 4.24 | 1.00 |
| **Ba\_tqwt\_stdValue\_dec\_5** | -0.0202 | 0.972 | 0.980 | 0.988 | 0.615 | 0.700 | 0.740 | 0.645 | 0.706 | 0.746 | 0.0921 | 0.622 | 4.82 | 5.24 | 0.20 |
| **De\_tqwt\_TKEO\_std\_dec\_11** | -0.1199 | 0.846 | 0.887 | 0.930 | 0.657 | 0.687 | 0.746 | 0.661 | 0.700 | 0.756 | 0.0895 | 0.727 | 4.81 | 6.19 | 0.60 |
| **De\_tqwt\_TKEO\_std\_dec\_36** | 0.1218 | 1.075 | 1.130 | 1.186 | 0.642 | 0.713 | 0.760 | 0.615 | 0.724 | 0.766 | 0.0785 | 0.615 | 4.73 | 5.15 | 0.55 |
| **De\_tqwt\_maxValue\_dec\_29** | -0.0440 | 0.940 | 0.957 | 0.974 | 0.633 | 0.697 | 0.739 | 0.657 | 0.715 | 0.748 | 0.0843 | 0.628 | 4.72 | 5.29 | 0.15 |
| **De\_tqwt\_entropy\_log\_dec\_35** | -0.3768 | 0.589 | 0.686 | 0.799 | 0.553 | 0.722 | 0.762 | 0.557 | 0.730 | 0.775 | 0.0782 | 0.670 | 4.72 | 5.69 | 0.15 |
| **De\_tqwt\_TKEO\_mean\_dec\_33** | -0.3776 | 0.591 | 0.685 | 0.795 | 0.655 | 0.707 | 0.759 | 0.663 | 0.729 | 0.766 | 0.0753 | 0.558 | 4.62 | 4.65 | 1.00 |
| **De\_tqwt\_TKEO\_std\_dec\_7** | -0.1112 | 0.853 | 0.895 | 0.938 | 0.596 | 0.712 | 0.746 | 0.597 | 0.729 | 0.760 | 0.0763 | 0.638 | 4.47 | 5.34 | 0.30 |
| **De\_det\_LT\_TKEO\_mean\_3\_coef** | -0.1952 | 0.757 | 0.823 | 0.894 | 0.614 | 0.701 | 0.741 | 0.624 | 0.718 | 0.749 | 0.0788 | 0.567 | 4.46 | 4.69 | 0.25 |
| **De\_tqwt\_minValue\_dec\_7** | 0.2083 | 1.127 | 1.232 | 1.346 | 0.568 | 0.727 | 0.758 | 0.585 | 0.740 | 0.769 | 0.0777 | 0.689 | 4.45 | 5.88 | 0.35 |
| **De\_tqwt\_TKEO\_std\_dec\_17** | -0.0549 | 0.924 | 0.947 | 0.970 | 0.622 | 0.698 | 0.729 | 0.624 | 0.714 | 0.734 | 0.0753 | 0.579 | 4.37 | 4.78 | 0.10 |

## Let focus on the new features  
  
decorCoeff <- smDecor$coefficients[newvars,];  
ncoef <- dc[newvars]  
cnames <- lapply(ncoef,names)  
names(cnames) <- NULL;  
decorCoeff$Elements <- lapply(cnames,paste,collapse="+")  
pander::pander(decorCoeff)

|  | Estimate | lower | OR | upper | u.Accuracy | r.Accuracy | full.Accuracy | u.AUC | r.AUC | full.AUC | IDI | NRI | z.IDI | z.NRI | Frequency | Elements |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **De\_tqwt\_energy\_dec\_33** | -0.2149 | 0.756 | 0.807 | 8.60e-01 | 0.748 | 0.721 | 0.779 | 0.734 | 0.739 | 0.777 | 0.1268 | 0.757 | 5.82 | 6.51 | 1.00 | tqwt\_energy\_dec\_31+tqwt\_energy\_dec\_33 |
| **De\_std\_MFCC\_2nd\_coef** | -1.4739 | 0.139 | 0.229 | 3.78e-01 | 0.702 | 0.782 | 0.820 | 0.687 | 0.775 | 0.820 | 0.1123 | 0.707 | 5.58 | 6.02 | 1.00 | std\_MFCC\_2nd\_coef+std\_2nd\_delta |
| **De\_tqwt\_stdValue\_dec\_33** | -0.8992 | 0.297 | 0.407 | 5.58e-01 | 0.685 | 0.721 | 0.782 | 0.686 | 0.733 | 0.774 | 0.1040 | 0.656 | 5.43 | 5.53 | 1.00 | tqwt\_TKEO\_mean\_dec\_35+tqwt\_stdValue\_dec\_32+tqwt\_stdValue\_dec\_33 |
| **De\_tqwt\_TKEO\_std\_dec\_11** | -0.1273 | 0.842 | 0.880 | 9.21e-01 | 0.700 | 0.708 | 0.754 | 0.701 | 0.711 | 0.760 | 0.1079 | 0.860 | 5.41 | 7.53 | 0.25 | tqwt\_TKEO\_mean\_dec\_10+tqwt\_TKEO\_mean\_dec\_11+tqwt\_TKEO\_std\_dec\_11+tqwt\_stdValue\_dec\_10+tqwt\_minValue\_dec\_17 |
| **De\_std\_12th\_delta\_delta** | 0.7336 | 1.596 | 2.083 | 2.72e+00 | 0.694 | 0.727 | 0.764 | 0.675 | 0.729 | 0.768 | 0.1063 | 0.621 | 5.31 | 5.20 | 0.75 | std\_MFCC\_12th\_coef+std\_12th\_delta\_delta |
| **De\_tqwt\_TKEO\_std\_dec\_36** | 0.1015 | 1.067 | 1.107 | 1.15e+00 | 0.639 | 0.714 | 0.761 | 0.609 | 0.720 | 0.767 | 0.0938 | 0.701 | 5.22 | 6.00 | 0.40 | tqwt\_TKEO\_mean\_dec\_35+tqwt\_TKEO\_std\_dec\_36 |
| **De\_locDbShimmer** | 1.0918 | 2.097 | 2.980 | 4.23e+00 | 0.712 | 0.758 | 0.785 | 0.738 | 0.745 | 0.780 | 0.0834 | 0.744 | 5.08 | 6.50 | 1.00 | locDbShimmer+ddaShimmer |
| **De\_tqwt\_kurtosisValue\_dec\_33** | 0.6436 | 1.490 | 1.903 | 2.43e+00 | 0.694 | 0.774 | 0.819 | 0.680 | 0.771 | 0.819 | 0.0847 | 0.536 | 5.06 | 4.41 | 1.00 | tqwt\_kurtosisValue\_dec\_32+tqwt\_kurtosisValue\_dec\_33 |
| **De\_tqwt\_TKEO\_mean\_dec\_17** | -0.0525 | 0.929 | 0.949 | 9.69e-01 | 0.641 | 0.648 | 0.722 | 0.668 | 0.702 | 0.737 | 0.0833 | 0.716 | 4.75 | 6.06 | 0.10 | tqwt\_TKEO\_mean\_dec\_17+tqwt\_minValue\_dec\_17 |
| **De\_tqwt\_entropy\_log\_dec\_1** | 0.8512 | 1.650 | 2.342 | 3.32e+00 | 0.606 | 0.725 | 0.767 | 0.633 | 0.737 | 0.776 | 0.0809 | 0.580 | 4.60 | 4.85 | 0.20 | tqwt\_entropy\_log\_dec\_1+tqwt\_entropy\_log\_dec\_2 |
| **De\_app\_LT\_TKEO\_std\_9\_coef** | 7.1981 | 57.084 | 1336.866 | 3.13e+04 | 0.649 | 0.736 | 0.762 | 0.578 | 0.766 | 0.781 | 0.0670 | 0.567 | 4.44 | 4.78 | 0.10 | Ea+app\_det\_TKEO\_mean\_9\_coef+app\_LT\_entropy\_log\_8\_coef+app\_LT\_TKEO\_std\_9\_coef |
| **De\_tqwt\_minValue\_dec\_7** | 0.0875 | 1.051 | 1.091 | 1.13e+00 | 0.560 | 0.726 | 0.762 | 0.580 | 0.758 | 0.781 | 0.0754 | 0.715 | 4.41 | 6.03 | 0.10 | tqwt\_minValue\_dec\_7+tqwt\_minValue\_dec\_17+tqwt\_maxValue\_dec\_8 |
| **De\_det\_LT\_TKEO\_mean\_3\_coef** | -0.1511 | 0.805 | 0.860 | 9.18e-01 | 0.610 | 0.697 | 0.735 | 0.624 | 0.712 | 0.750 | 0.0726 | 0.614 | 4.39 | 5.09 | 0.30 | Ed2\_3\_coef+det\_LT\_TKEO\_mean\_1\_coef+det\_LT\_TKEO\_mean\_3\_coef |
| **De\_tqwt\_TKEO\_std\_dec\_7** | -0.0900 | 0.878 | 0.914 | 9.52e-01 | 0.599 | 0.704 | 0.738 | 0.593 | 0.732 | 0.760 | 0.0726 | 0.574 | 4.27 | 4.71 | 0.25 | tqwt\_TKEO\_std\_dec\_7+tqwt\_stdValue\_dec\_7+tqwt\_minValue\_dec\_17 |

## Differences Between Blind *vs.* Outcome-Driven Decorrelation

In this section I will show the differences in unaltered basis vectors between the Outcome driven Transformation vs the blind decorrelated transformation

par(op)  
par(mfrow=c(1,1))  
  
  
smDecorU <- summary(bmdU)  
decornamesU <- rownames(smDecorU$coefficients)  
  
get\_De\_names <- decornames[!str\_detect(decornames,"De\_")]  
get\_De\_namesU <- decornamesU[!str\_detect(decornamesU,"De\_")]  
  
unn <- bmd$univariate[,3]  
names(unn) <- rownames(bmd$univariate)  
pander::pander(as.matrix(unn[get\_De\_names]))

|  |  |
| --- | --- |
| **Ba\_tqwt\_kurtosisValue\_dec\_36** | 5.83 |
| **Ba\_tqwt\_kurtosisValue\_dec\_17** | 4.27 |
| **Ba\_std\_delta\_delta\_log\_energy** | 6.15 |
| **Ba\_std\_5th\_delta** | 4.28 |
| **Ba\_meanIntensity** | 5.34 |
| **Ba\_locAbsJitter** | 4.11 |
| **Ba\_tqwt\_kurtosisValue\_dec\_28** | 4.51 |
| **tqwt\_kurtosisValue\_dec\_20** | 4.85 |
| **Ba\_tqwt\_energy\_dec\_12** | 4.45 |
| **Ba\_tqwt\_kurtosisValue\_dec\_26** | 3.88 |
| **Ba\_tqwt\_entropy\_log\_dec\_12** | 5.38 |
| **std\_MFCC\_8th\_coef** | 4.40 |
| **tqwt\_meanValue\_dec\_11** | 3.67 |
| **Ba\_IMF\_NSR\_TKEO** | 2.92 |
| **mean\_MFCC\_2nd\_coef** | 3.54 |

pander::pander(summary(unn[get\_De\_names]))

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| --- | --- | --- | --- | --- | --- |
| 2.92 | 4 | 4.4 | 4.5 | 5.09 | 6.15 |

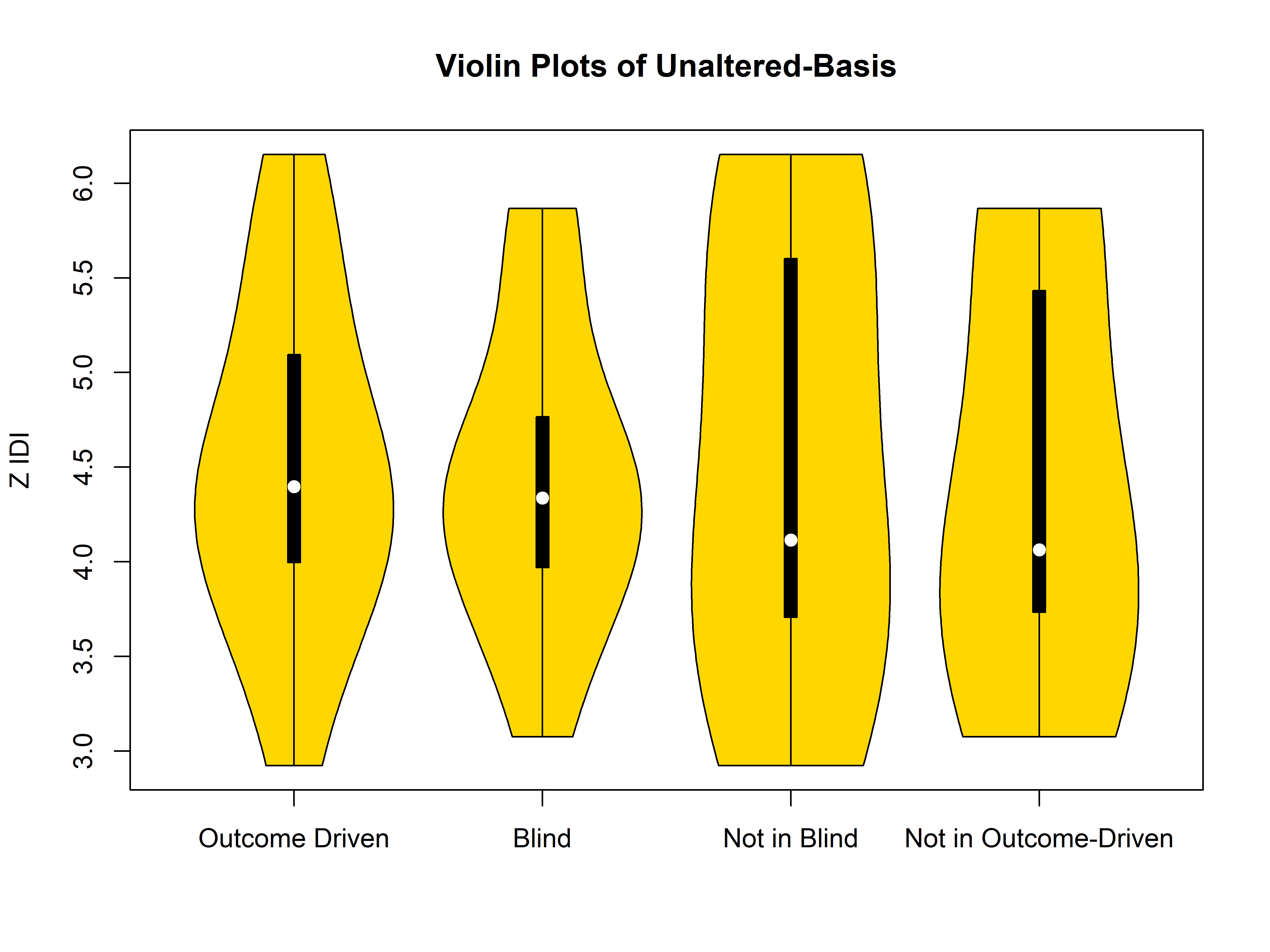
unnU <- bmdU$univariate[,3]  
names(unnU) <- rownames(bmdU$univariate)  
pander::pander(as.matrix(unnU[get\_De\_namesU]))

|  |  |
| --- | --- |
| **tqwt\_kurtosisValue\_dec\_36** | 5.83 |
| **Ba\_std\_5th\_delta** | 4.28 |
| **Ba\_std\_delta\_log\_energy** | 5.87 |
| **tqwt\_kurtosisValue\_dec\_26** | 3.88 |
| **Ba\_tqwt\_kurtosisValue\_dec\_17** | 4.27 |
| **tqwt\_kurtosisValue\_dec\_27** | 4.25 |
| **Ba\_tqwt\_energy\_dec\_12** | 4.45 |
| **Ba\_tqwt\_kurtosisValue\_dec\_28** | 4.51 |
| **std\_MFCC\_8th\_coef** | 4.40 |
| **Ba\_meanIntensity** | 5.34 |
| **tqwt\_kurtosisValue\_dec\_20** | 4.85 |
| **tqwt\_meanValue\_dec\_11** | 3.67 |
| **Ba\_IMF\_SNR\_SEO** | 3.08 |
| **Ba\_tqwt\_stdValue\_dec\_5** | 3.69 |

pander::pander(summary(unnU[get\_De\_namesU]))

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| --- | --- | --- | --- | --- | --- |
| 3.08 | 3.97 | 4.34 | 4.45 | 4.77 | 5.87 |

#boxplot(unn[get\_De\_names],unnU[get\_De\_namesU],xlab=c("Method"),ylab="Z",main="Z Values of Basis Features")  
  
x1 <- unn[get\_De\_names]  
x2 <- unnU[get\_De\_namesU]  
X3 <- x1[!(get\_De\_names %in% get\_De\_namesU)]  
X4 <- x2[!(get\_De\_namesU %in% get\_De\_names)]  
vioplot(x1, x2, X3,X4, names=c("Outcome Driven", "Blind","Not in Blind","Not in Outcome-Driven"),ylab="Z IDI",  
 col="gold")  
title("Violin Plots of Unaltered-Basis")



sameFeatures <- get\_De\_names[get\_De\_names %in% get\_De\_namesU]  
pander::pander(as.matrix(unn[sameFeatures]))

|  |  |
| --- | --- |
| **Ba\_tqwt\_kurtosisValue\_dec\_17** | 4.27 |
| **Ba\_std\_5th\_delta** | 4.28 |
| **Ba\_meanIntensity** | 5.34 |
| **Ba\_tqwt\_kurtosisValue\_dec\_28** | 4.51 |
| **tqwt\_kurtosisValue\_dec\_20** | 4.85 |
| **Ba\_tqwt\_energy\_dec\_12** | 4.45 |
| **std\_MFCC\_8th\_coef** | 4.40 |
| **tqwt\_meanValue\_dec\_11** | 3.67 |

## The features by Outcome Drive not in blind  
pander::pander(as.matrix(x1[!(get\_De\_names %in% get\_De\_namesU)]))

|  |  |
| --- | --- |
| **Ba\_tqwt\_kurtosisValue\_dec\_36** | 5.83 |
| **Ba\_std\_delta\_delta\_log\_energy** | 6.15 |
| **Ba\_locAbsJitter** | 4.11 |
| **Ba\_tqwt\_kurtosisValue\_dec\_26** | 3.88 |
| **Ba\_tqwt\_entropy\_log\_dec\_12** | 5.38 |
| **Ba\_IMF\_NSR\_TKEO** | 2.92 |
| **mean\_MFCC\_2nd\_coef** | 3.54 |

## The features not in outcome driven  
pander::pander(as.matrix(x2[!(get\_De\_namesU %in% get\_De\_names)]))

|  |  |
| --- | --- |
| **tqwt\_kurtosisValue\_dec\_36** | 5.83 |
| **Ba\_std\_delta\_log\_energy** | 5.87 |
| **tqwt\_kurtosisValue\_dec\_26** | 3.88 |
| **tqwt\_kurtosisValue\_dec\_27** | 4.25 |
| **Ba\_IMF\_SNR\_SEO** | 3.08 |
| **Ba\_tqwt\_stdValue\_dec\_5** | 3.69 |

### The Final Table

I’ll create a table subset of the logistic model from the Outcome-Driven decorrelated data.

The table will have:

1. The top associated features described by the feature network, as well as, and the new features.
   1. For Decorrelated features it will provide the decorrelation formula
2. Nugget labels
   1. The label of nugget as found by the clustering procedure
3. The feature coefficient
4. The feature Odd ratios and their corresponding 95%CI

## The features in top nugget  
clusterFeatures <- clusterOutcome$names  
## The new features   
discoveredFeatures <- newvars[zvaluePrePost$ZUni<1.96]  
  
tablefinal <- smDecor$coefficients[unique(c(clusterFeatures,discoveredFeatures)),  
 c("Estimate","lower","OR","upper","z.IDI")]  
  
nugget <- clusterOutcome$membership  
names(nugget) <- clusterOutcome$names  
tablefinal$Nugget <- nugget[rownames(tablefinal)]  
tablefinal$Nugget[is.na(tablefinal$Nugget)] <- "D"  
deFromula <- character(length(theDeFormulas))  
names(deFromula) <- names(theDeFormulas)  
for (dx in names(deFromula))  
{  
 coef <- theDeFormulas[[dx]]  
 cname <- names(theDeFormulas[[dx]])  
 names(cname) <- cname  
 for (cf in names(coef))  
 {  
 if (cf != dx)  
 {  
 if (coef[cf]>0)  
 {  
 deFromula[dx] <- paste(deFromula[dx],  
 sprintf("+ %5.3f\*%s",coef[cf],cname[cf]))  
 }  
 else  
 {  
 deFromula[dx] <- paste(deFromula[dx],  
 sprintf("%5.3f\*%s",coef[cf],cname[cf]))  
 }  
 }  
 }  
}  
tablefinal$DecorFormula <- deFromula[rownames(tablefinal)]  
pander::pander(tablefinal)

|  | Estimate | lower | OR | upper | z.IDI | Nugget | DecorFormula |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Ba\_std\_delta\_delta\_log\_energy** | 0.4472 | 1.376 | 1.564 | 1.778 | 6.59 | 1 | NA |
| **De\_std\_MFCC\_2nd\_coef** | -1.4739 | 0.139 | 0.229 | 0.378 | 5.58 | 1 | + 1.000*std\_MFCC\_2nd\_coef -0.828*std\_2nd\_delta |
| **De\_tqwt\_kurtosisValue\_dec\_33** | 0.6436 | 1.490 | 1.903 | 2.431 | 5.06 | 1 | -0.883*tqwt\_kurtosisValue\_dec\_32 + 1.000*tqwt\_kurtosisValue\_dec\_33 |
| **De\_tqwt\_energy\_dec\_33** | -0.2149 | 0.756 | 0.807 | 0.860 | 5.82 | 2 | -0.907*tqwt\_energy\_dec\_31 + 1.000*tqwt\_energy\_dec\_33 |
| **De\_locDbShimmer** | 1.0918 | 2.097 | 2.980 | 4.234 | 5.08 | 2 | + 1.000*locDbShimmer -0.963*ddaShimmer |
| **Ba\_tqwt\_entropy\_log\_dec\_12** | -0.4992 | 0.519 | 0.607 | 0.711 | 5.27 | 3 | NA |
| **De\_tqwt\_stdValue\_dec\_33** | -0.8992 | 0.297 | 0.407 | 0.558 | 5.43 | 3 | -0.131*tqwt\_TKEO\_mean\_dec\_35 -0.715*tqwt\_stdValue\_dec\_32 + 1.000\*tqwt\_stdValue\_dec\_33 |
| **Ba\_tqwt\_kurtosisValue\_dec\_36** | 0.1636 | 1.130 | 1.178 | 1.227 | 6.78 | 3 | NA |
| **De\_std\_12th\_delta\_delta** | 0.7336 | 1.596 | 2.083 | 2.717 | 5.31 | 4 | -0.914*std\_MFCC\_12th\_coef + 1.000*std\_12th\_delta\_delta |
| **tqwt\_kurtosisValue\_dec\_20** | 0.5307 | 1.426 | 1.700 | 2.027 | 5.51 | 4 | NA |
| **Ba\_std\_5th\_delta** | 0.3983 | 1.319 | 1.489 | 1.681 | 6.01 | 3 | NA |
| **std\_MFCC\_8th\_coef** | 0.4808 | 1.352 | 1.617 | 1.935 | 4.99 | 5 | NA |
| **De\_tqwt\_TKEO\_mean\_dec\_11** | -0.0317 | 0.958 | 0.969 | 0.979 | 5.26 | 5 | NA |
| **Ba\_meanIntensity** | -1.4408 | 0.159 | 0.237 | 0.352 | 5.88 | 4 | NA |
| **De\_tqwt\_entropy\_log\_dec\_1** | 0.8512 | 1.650 | 2.342 | 3.325 | 4.60 | D | + 1.000*tqwt\_entropy\_log\_dec\_1 -0.849*tqwt\_entropy\_log\_dec\_2 |
| **De\_det\_LT\_TKEO\_mean\_3\_coef** | -0.1511 | 0.805 | 0.860 | 0.918 | 4.39 | D | -0.890*Ed2\_3\_coef -0.083*det\_LT\_TKEO\_mean\_1\_coef + 1.000\*det\_LT\_TKEO\_mean\_3\_coef |