Congestive Heart Failure

Mahnoor Ali

12/29/2021

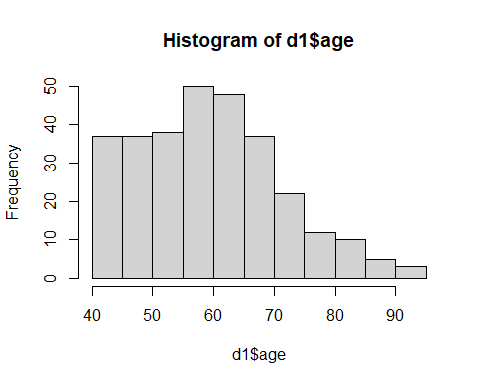
Import Data set:

library(readxl)  
d1<-read.csv(file = "Project.csv",header = TRUE,sep = ",")  
summary(d1)

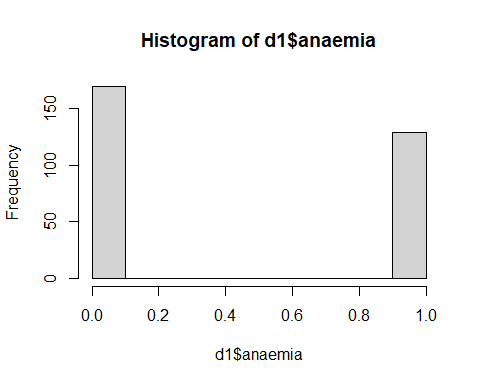
## age anaemia creatinine\_phosphokinase diabetes   
## Min. :40.00 Min. :0.0000 Min. : 23.0 Min. :0.0000   
## 1st Qu.:51.00 1st Qu.:0.0000 1st Qu.: 116.5 1st Qu.:0.0000   
## Median :60.00 Median :0.0000 Median : 250.0 Median :0.0000   
## Mean :60.83 Mean :0.4314 Mean : 581.8 Mean :0.4181   
## 3rd Qu.:70.00 3rd Qu.:1.0000 3rd Qu.: 582.0 3rd Qu.:1.0000   
## Max. :95.00 Max. :1.0000 Max. :7861.0 Max. :1.0000   
## ejection\_fraction high\_blood\_pressure platelets serum\_creatinine  
## Min. :14.00 Min. :0.0000 Min. : 25100 Min. :0.500   
## 1st Qu.:30.00 1st Qu.:0.0000 1st Qu.:212500 1st Qu.:0.900   
## Median :38.00 Median :0.0000 Median :262000 Median :1.100   
## Mean :38.08 Mean :0.3512 Mean :263358 Mean :1.394   
## 3rd Qu.:45.00 3rd Qu.:1.0000 3rd Qu.:303500 3rd Qu.:1.400   
## Max. :80.00 Max. :1.0000 Max. :850000 Max. :9.400   
## serum\_sodium sex smoking time   
## Min. :113.0 Min. :0.0000 Min. :0.0000 Min. : 4.0   
## 1st Qu.:134.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 73.0   
## Median :137.0 Median :1.0000 Median :0.0000 Median :115.0   
## Mean :136.6 Mean :0.6488 Mean :0.3211 Mean :130.3   
## 3rd Qu.:140.0 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:203.0   
## Max. :148.0 Max. :1.0000 Max. :1.0000 Max. :285.0   
## DEATH\_EVENT   
## Min. :0.0000   
## 1st Qu.:0.0000   
## Median :0.0000   
## Mean :0.3211   
## 3rd Qu.:1.0000   
## Max. :1.0000

Histogram of Data set:

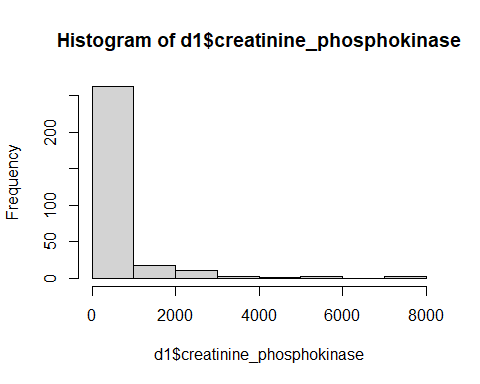
hist(d1$age)



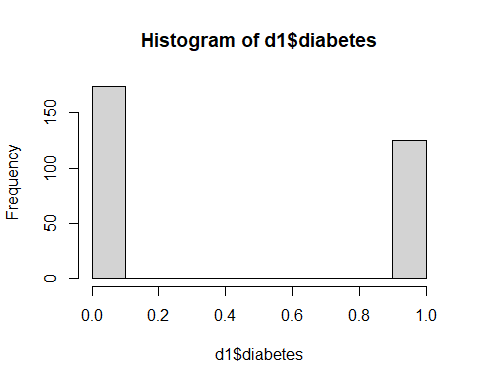
hist(d1$anaemia)



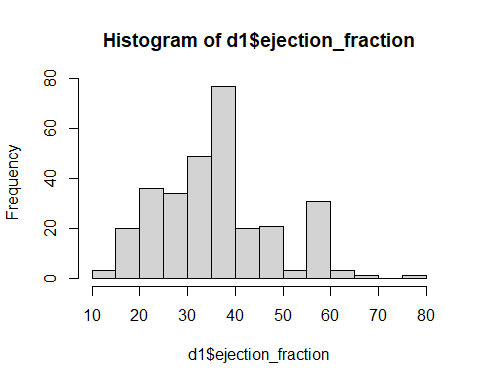
hist(d1$creatinine\_phosphokinase)



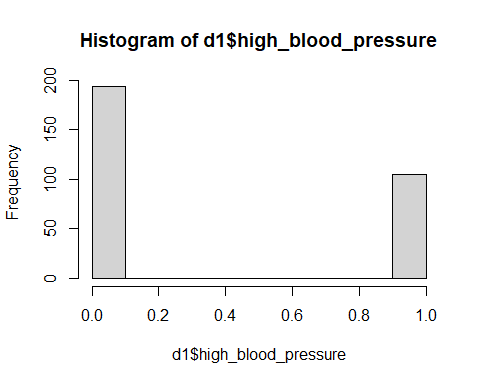
hist(d1$diabetes)



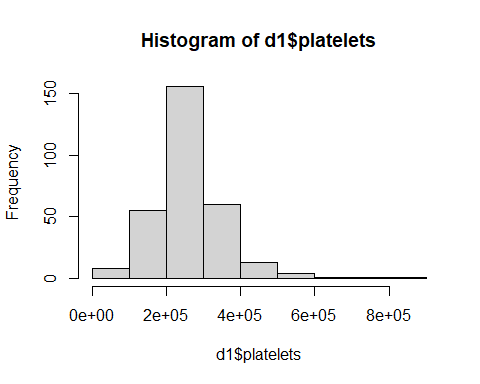
hist(d1$ejection\_fraction)



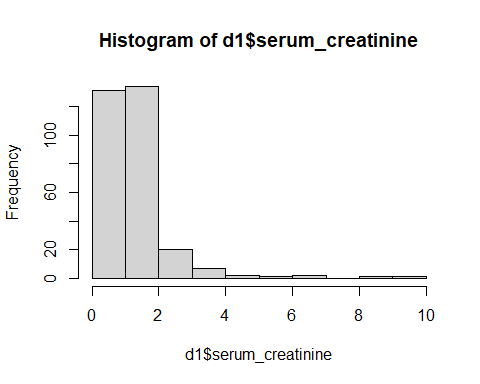
hist(d1$high\_blood\_pressure)



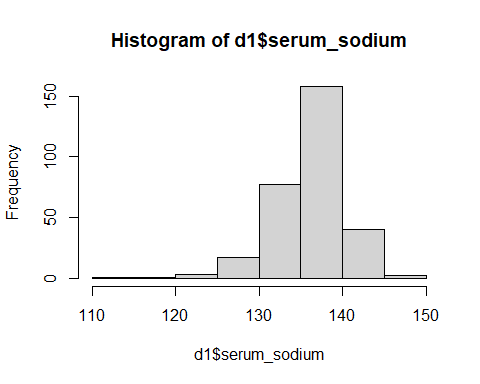
hist(d1$platelets)



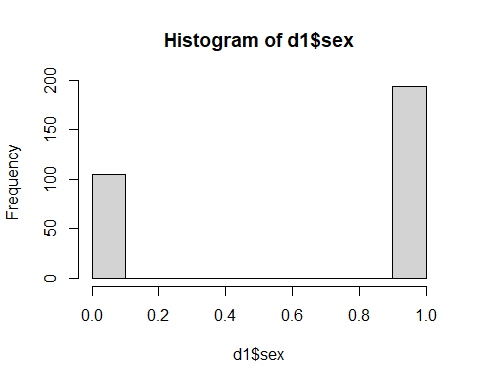
hist(d1$serum\_creatinine)



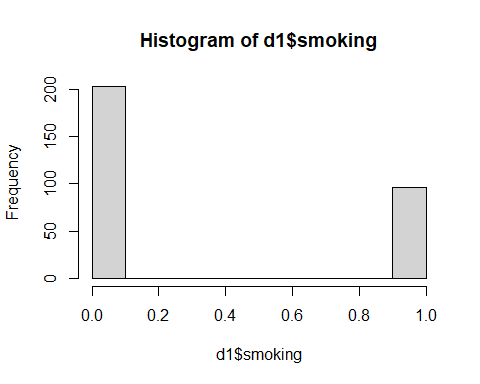
hist(d1$serum\_sodium)



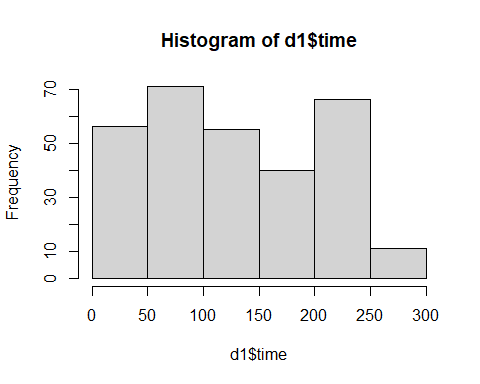
hist(d1$sex)



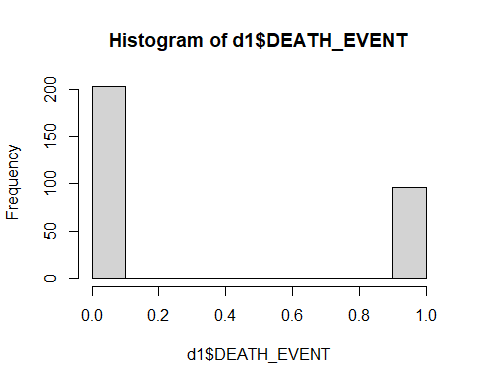
hist(d1$smoking)



hist(d1$time)



hist(d1$DEATH\_EVENT)



Split data into test and train:

set.seed(123)  
ind <- sample(2, nrow(d1),replace = TRUE,prob = c(0.8,0.2))  
train <- d1[ind == 1,]  
test <- d1[ind == 2,]  
train$DEATH\_EVENT <- as.character(train$DEATH\_EVENT)  
train$DEATH\_EVENT <- as.factor(train$DEATH\_EVENT)  
test$DEATH\_EVENT <- as.character(test$DEATH\_EVENT)  
test$DEATH\_EVENT <- as.factor(test$DEATH\_EVENT)  
summary(train)

## age anaemia creatinine\_phosphokinase diabetes   
## Min. :40.00 Min. :0.0000 Min. : 30.0 Min. :0.0000   
## 1st Qu.:51.50 1st Qu.:0.0000 1st Qu.: 117.0 1st Qu.:0.0000   
## Median :60.00 Median :0.0000 Median : 258.0 Median :0.0000   
## Mean :61.04 Mean :0.4156 Mean : 610.6 Mean :0.4198   
## 3rd Qu.:70.00 3rd Qu.:1.0000 3rd Qu.: 582.0 3rd Qu.:1.0000   
## Max. :95.00 Max. :1.0000 Max. :7861.0 Max. :1.0000   
## ejection\_fraction high\_blood\_pressure platelets serum\_creatinine  
## Min. :14.00 Min. :0.0000 Min. : 25100 Min. :0.500   
## 1st Qu.:30.00 1st Qu.:0.0000 1st Qu.:211500 1st Qu.:0.900   
## Median :38.00 Median :0.0000 Median :255000 Median :1.100   
## Mean :37.28 Mean :0.3621 Mean :258669 Mean :1.388   
## 3rd Qu.:40.00 3rd Qu.:1.0000 3rd Qu.:297500 3rd Qu.:1.400   
## Max. :70.00 Max. :1.0000 Max. :850000 Max. :9.400   
## serum\_sodium sex smoking time DEATH\_EVENT  
## Min. :113.0 Min. :0.0000 Min. :0.0000 Min. : 4.0 0:165   
## 1st Qu.:134.0 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 73.5 1: 78   
## Median :137.0 Median :1.0000 Median :0.0000 Median :117.0   
## Mean :136.8 Mean :0.6626 Mean :0.3333 Mean :130.4   
## 3rd Qu.:140.0 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:203.0   
## Max. :148.0 Max. :1.0000 Max. :1.0000 Max. :285.0

summary(test)

## age anaemia creatinine\_phosphokinase diabetes   
## Min. :40.00 Min. :0.0 Min. : 23.0 Min. :0.0000   
## 1st Qu.:49.75 1st Qu.:0.0 1st Qu.: 116.2 1st Qu.:0.0000   
## Median :60.33 Median :0.5 Median : 221.5 Median :0.0000   
## Mean :59.94 Mean :0.5 Mean : 457.0 Mean :0.4107   
## 3rd Qu.:70.00 3rd Qu.:1.0 3rd Qu.: 582.0 3rd Qu.:1.0000   
## Max. :94.00 Max. :1.0 Max. :5209.0 Max. :1.0000   
## ejection\_fraction high\_blood\_pressure platelets serum\_creatinine  
## Min. :15.00 Min. :0.0000 Min. : 47000 Min. :0.700   
## 1st Qu.:30.00 1st Qu.:0.0000 1st Qu.:219000 1st Qu.:1.000   
## Median :40.00 Median :0.0000 Median :268500 Median :1.100   
## Mean :41.59 Mean :0.3036 Mean :283704 Mean :1.419   
## 3rd Qu.:51.25 3rd Qu.:1.0000 3rd Qu.:327000 3rd Qu.:1.375   
## Max. :80.00 Max. :1.0000 Max. :742000 Max. :6.100   
## serum\_sodium sex smoking time DEATH\_EVENT  
## Min. :116.0 Min. :0.0000 Min. :0.0000 Min. : 7.0 0:38   
## 1st Qu.:134.8 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 65.0 1:18   
## Median :136.0 Median :1.0000 Median :0.0000 Median :109.5   
## Mean :136.1 Mean :0.5893 Mean :0.2679 Mean :129.9   
## 3rd Qu.:139.0 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:201.8   
## Max. :145.0 Max. :1.0000 Max. :1.0000 Max. :278.0

Random Forest:

library(caret)

## Warning: package 'caret' was built under R version 4.1.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.2

## Loading required package: lattice

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.1.2

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

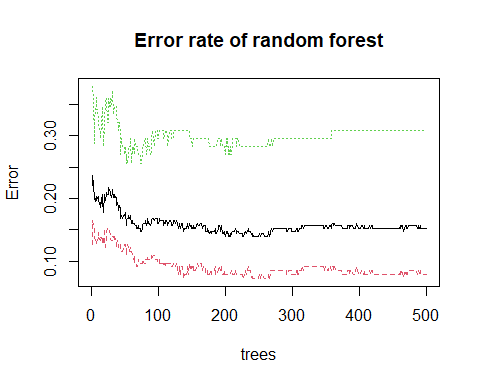
set.seed(222)  
rf<-randomForest::randomForest(DEATH\_EVENT~.,data=train)  
print(rf)

##   
## Call:  
## randomForest(formula = DEATH\_EVENT ~ ., data = train)   
## Type of random forest: classification  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 15.23%  
## Confusion matrix:  
## 0 1 class.error  
## 0 152 13 0.07878788  
## 1 24 54 0.30769231

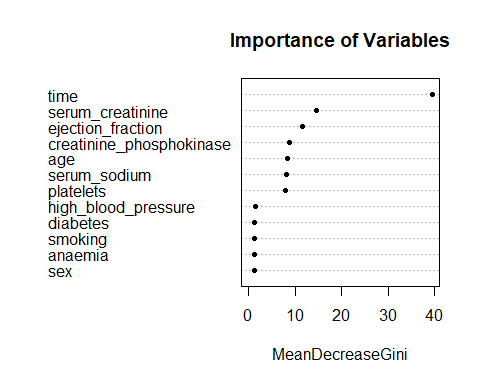
p1 <- predict(rf,train)  
confusionMatrix(p1,train$DEATH\_EVENT)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 165 0  
## 1 0 78  
##   
## Accuracy : 1   
## 95% CI : (0.9849, 1)  
## No Information Rate : 0.679   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 1   
##   
## Mcnemar's Test P-Value : NA   
##   
## Sensitivity : 1.000   
## Specificity : 1.000   
## Pos Pred Value : 1.000   
## Neg Pred Value : 1.000   
## Prevalence : 0.679   
## Detection Rate : 0.679   
## Detection Prevalence : 0.679   
## Balanced Accuracy : 1.000   
##   
## 'Positive' Class : 0   
##

plot(rf, main = "Error rate of random forest")



varImpPlot(rf, pch = 20, main = "Importance of Variables")

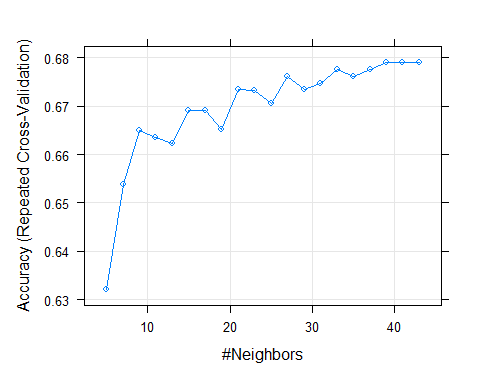


K- Nearest Neighbors:

trControl <- trainControl(method = "repeatedcv",  
 number = 10,  
 repeats = 3)  
set.seed(221)  
fit1<- train(DEATH\_EVENT ~.,  
 data = train,  
 method = 'knn',  
 tuneLength = 20,  
 trControl=trControl)  
fit1

## k-Nearest Neighbors   
##   
## 243 samples  
## 12 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 3 times)   
## Summary of sample sizes: 218, 219, 219, 218, 219, 219, ...   
## Resampling results across tuning parameters:  
##   
## k Accuracy Kappa   
## 5 0.6321232 0.036336378  
## 7 0.6537899 0.050917785  
## 9 0.6649614 0.055019765  
## 11 0.6636232 0.028776718  
## 13 0.6622295 0.012196010  
## 15 0.6691739 0.015725048  
## 17 0.6691739 0.015340557  
## 19 0.6651787 -0.006377673  
## 21 0.6735169 -0.002470489  
## 23 0.6732850 -0.004735198  
## 25 0.6706232 -0.015721043  
## 27 0.6761836 -0.005411765  
## 29 0.6734614 -0.010585322  
## 31 0.6747947 -0.008033488  
## 33 0.6776329 -0.002666667  
## 35 0.6761232 -0.005214153  
## 37 0.6775725 -0.002745098  
## 39 0.6790217 0.000000000  
## 41 0.6790217 0.000000000  
## 43 0.6790217 0.000000000  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was k = 43.

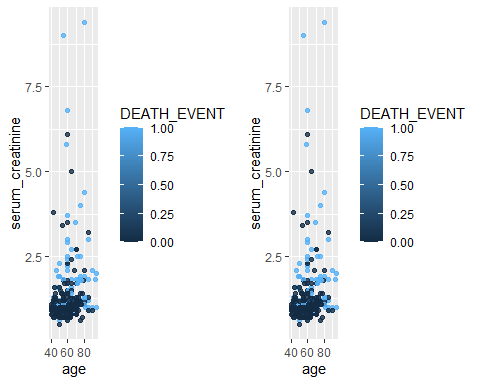
plot(fit1)



predknn <- predict(fit1,newdata = test)  
confusionMatrix(predknn,test$DEATH\_EVENT)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 38 18  
## 1 0 0  
##   
## Accuracy : 0.6786   
## 95% CI : (0.5404, 0.7971)  
## No Information Rate : 0.6786   
## P-Value [Acc > NIR] : 0.5634   
##   
## Kappa : 0   
##   
## Mcnemar's Test P-Value : 6.151e-05   
##   
## Sensitivity : 1.0000   
## Specificity : 0.0000   
## Pos Pred Value : 0.6786   
## Neg Pred Value : NaN   
## Prevalence : 0.6786   
## Detection Rate : 0.6786   
## Detection Prevalence : 1.0000   
## Balanced Accuracy : 0.5000   
##   
## 'Positive' Class : 0   
##

p1 <- ggplot(d1, aes(x= age, y=serum\_creatinine, col=DEATH\_EVENT)) +  
 geom\_point(alpha=0.8)  
  
p2 <- ggplot(d1, aes(x= age, y=serum\_creatinine, col=DEATH\_EVENT)) +  
 geom\_point(alpha=0.8)  
  
gridExtra::grid.arrange(p1, p2, ncol=2)



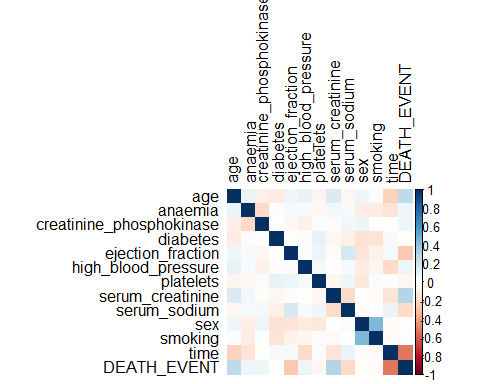
Heat map Of Data set:

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.1.2

## corrplot 0.92 loaded

C<-cor(d1)  
corrplot(C, method = "color",  
   
 tl.col = "black")



head(test)

## age anaemia creatinine\_phosphokinase diabetes ejection\_fraction  
## 4 50 1 111 0 20  
## 5 65 1 160 1 20  
## 8 60 1 315 1 60  
## 11 75 1 81 0 38  
## 16 82 1 379 0 50  
## 20 48 1 582 1 55  
## high\_blood\_pressure platelets serum\_creatinine serum\_sodium sex smoking time  
## 4 0 210000 1.9 137 1 0 7  
## 5 0 327000 2.7 116 0 0 8  
## 8 0 454000 1.1 131 1 1 10  
## 11 1 368000 4.0 131 1 1 10  
## 16 0 47000 1.3 136 1 0 13  
## 20 0 87000 1.9 121 0 0 15  
## DEATH\_EVENT  
## 4 1  
## 5 1  
## 8 1  
## 11 1  
## 16 1  
## 20 1

levels(train$DEATH\_EVENT)

## [1] "0" "1"

Logistic Regression:

mymodel <- glm(DEATH\_EVENT ~ ejection\_fraction +age+serum\_creatinine +time,data = train ,family = 'binomial')  
summary(mymodel)

##   
## Call:  
## glm(formula = DEATH\_EVENT ~ ejection\_fraction + age + serum\_creatinine +   
## time, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0819 -0.6290 -0.2472 0.5134 2.9108   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.696706 1.186177 0.587 0.5570   
## ejection\_fraction -0.081556 0.018777 -4.343 1.40e-05 \*\*\*  
## age 0.040843 0.016635 2.455 0.0141 \*   
## serum\_creatinine 0.841623 0.202542 4.155 3.25e-05 \*\*\*  
## time -0.019697 0.003137 -6.279 3.40e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 305.02 on 242 degrees of freedom  
## Residual deviance: 187.32 on 238 degrees of freedom  
## AIC: 197.32  
##   
## Number of Fisher Scoring iterations: 5

predlog <- predict(mymodel,train,type = 'response')  
head(predlog)

## 1 2 3 6 7 9   
## 0.9746451 0.6573601 0.9356323 0.9382063 0.9660906 0.2923164

predlog1 <-ifelse(predlog>0.5,1,0)  
tab <- table(Predicted = predlog1 , Actual= train$DEATH\_EVENT)  
tab

## Actual  
## Predicted 0 1  
## 0 148 24  
## 1 17 54

1-sum(diag(tab))/sum(tab)

## [1] 0.1687243

predlog2 <- predict(mymodel,test,type = 'response')  
head(predlog2)

## 4 5 8 11 16 20   
## 0.9288358 0.9788593 0.2655644 0.9787585 0.6912753 0.3717385

predlog2 <-ifelse(predlog2>0.5,1,0)  
tab2 <- table(Predicted = predlog2 , Actual= test$DEATH\_EVENT)  
tab2

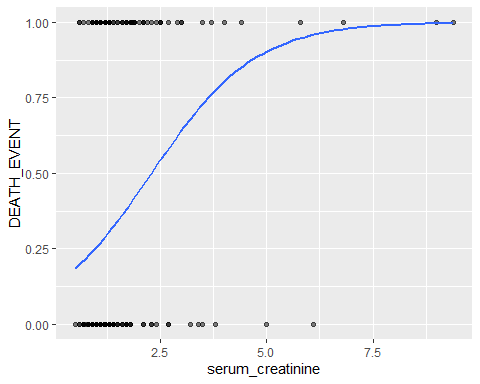
## Actual  
## Predicted 0 1  
## 0 35 6  
## 1 3 12

1-sum(diag(tab2))/sum(tab2)

## [1] 0.1607143

ggplot(d1, aes(x=serum\_creatinine, y=DEATH\_EVENT)) +   
 geom\_point(alpha=.5) +  
 stat\_smooth(method="glm", se=FALSE, method.args = list(family=binomial))

## `geom\_smooth()` using formula 'y ~ x'



with(mymodel,pchisq(null.deviance - deviance,df.null-df.residual,lower.tail = FALSE))

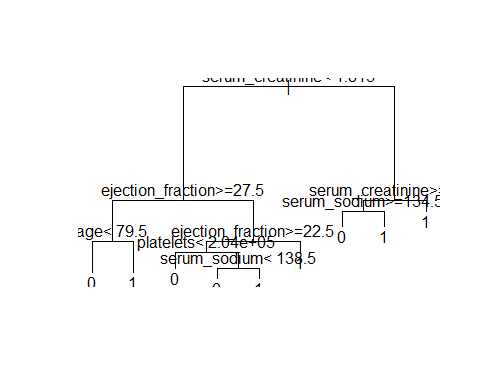
## [1] 1.651878e-24

Decision Tree:

library(rpart)

## Warning: package 'rpart' was built under R version 4.1.2

fit<-rpart(DEATH\_EVENT~sex+age+smoking+high\_blood\_pressure+anaemia+diabetes+creatinine\_phosphokinase+ejection\_fraction+platelets+serum\_creatinine+serum\_sodium,data=d1,method="class")  
plot(fit)  
text(fit)



predDT <- predict(fit,test,type = 'class')  
confusionMatrix(predDT,data = test$DEATH\_EVENT)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 37 1  
## 1 5 13  
##   
## Accuracy : 0.8929   
## 95% CI : (0.7812, 0.9597)  
## No Information Rate : 0.75   
## P-Value [Acc > NIR] : 0.006653   
##   
## Kappa : 0.7391   
##   
## Mcnemar's Test P-Value : 0.220671   
##   
## Sensitivity : 0.8810   
## Specificity : 0.9286   
## Pos Pred Value : 0.9737   
## Neg Pred Value : 0.7222   
## Prevalence : 0.7500   
## Detection Rate : 0.6607   
## Detection Prevalence : 0.6786   
## Balanced Accuracy : 0.9048   
##   
## 'Positive' Class : 0   
##

library(rattle)

## Warning: package 'rattle' was built under R version 4.1.2

## Loading required package: tibble

## Loading required package: bitops

## Rattle: A free graphical interface for data science with R.  
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

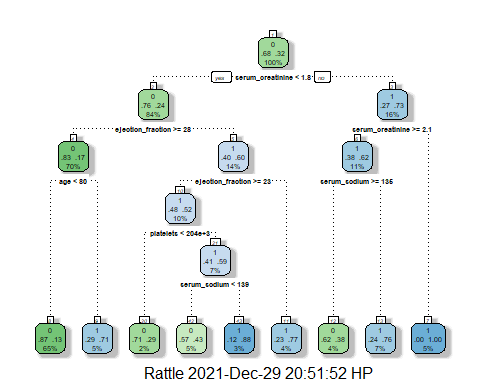
##   
## Attaching package: 'rattle'

## The following object is masked from 'package:randomForest':  
##   
## importance

library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.1.2

library(RColorBrewer)  
fancyRpartPlot(fit)



Fuzzy C Mean:

library(ppclust)

## Warning: package 'ppclust' was built under R version 4.1.2

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.1.2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(cluster)  
library(fclust)

## Warning: package 'fclust' was built under R version 4.1.2

## Registered S3 method overwritten by 'fclust':  
## method from   
## print.fclust e1071

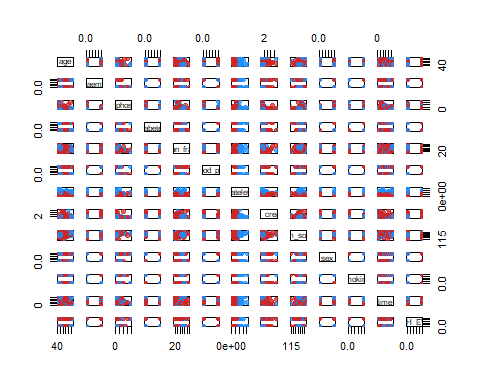
res.fcm <- fcm(d1, centers=2)  
u0<- inaparc::imembrand(nrow(d1), k=2)$u  
res.fcm <- fcm(d1, centers=2, memberships=u0)  
res.fcm <- fcm(d1, centers=2, dmetric="correlation")  
res.fcm <- fcm(d1, centers=2)  
as.data.frame(res.fcm$u)

## Cluster 1 Cluster 2  
## 1 1.632645e-01 8.367355e-01  
## 2 1.523673e-01 8.476327e-01  
## 3 7.039624e-02 9.296038e-01  
## 4 3.196944e-03 9.968031e-01  
## 5 8.678710e-01 1.321290e-01  
## 6 8.193279e-03 9.918067e-01  
## 7 1.261508e-01 8.738492e-01  
## 8 8.846383e-01 1.153617e-01  
## 9 1.495606e-01 8.504394e-01  
## 10 9.876883e-01 1.231174e-02  
## 11 9.999288e-01 7.124180e-05  
## 12 7.894807e-02 9.210519e-01  
## 13 1.124896e-01 8.875104e-01  
## 14 2.724903e-01 7.275097e-01  
## 15 9.281674e-01 7.183257e-02  
## 16 2.218407e-01 7.781593e-01  
## 17 1.387463e-01 8.612537e-01  
## 18 6.373360e-02 9.362664e-01  
## 19 1.822991e-02 9.817701e-01  
## 20 1.795808e-01 8.204192e-01  
## 21 2.724941e-01 7.275059e-01  
## 22 5.389832e-01 4.610168e-01  
## 23 4.327841e-01 5.672159e-01  
## 24 9.999280e-01 7.198979e-05  
## 25 1.495517e-01 8.504483e-01  
## 26 9.183031e-02 9.081697e-01  
## 27 1.734774e-02 9.826523e-01  
## 28 3.682459e-01 6.317541e-01  
## 29 8.529594e-02 9.147041e-01  
## 30 1.247029e-02 9.875297e-01  
## 31 1.495517e-01 8.504483e-01  
## 32 9.958002e-01 4.199768e-03  
## 33 7.991106e-01 2.008894e-01  
## 34 6.044798e-01 3.955202e-01  
## 35 2.846184e-02 9.715382e-01  
## 36 4.050478e-03 9.959495e-01  
## 37 2.399097e-03 9.976009e-01  
## 38 8.178404e-01 1.821596e-01  
## 39 6.424904e-01 3.575096e-01  
## 40 8.824916e-01 1.175084e-01  
## 41 1.495516e-01 8.504484e-01  
## 42 8.529494e-02 9.147051e-01  
## 43 3.296938e-02 9.670306e-01  
## 44 5.275126e-05 9.999472e-01  
## 45 1.996919e-02 9.800308e-01  
## 46 7.030732e-01 2.969268e-01  
## 47 2.192450e-01 7.807550e-01  
## 48 8.892796e-01 1.107204e-01  
## 49 1.062382e-01 8.937618e-01  
## 50 9.788570e-01 2.114302e-02  
## 51 6.373347e-02 9.362665e-01  
## 52 9.431335e-01 5.686647e-02  
## 53 1.501670e-01 8.498330e-01  
## 54 9.814529e-01 1.854711e-02  
## 55 9.048738e-02 9.095126e-01  
## 56 8.740546e-01 1.259454e-01  
## 57 7.603846e-04 9.992396e-01  
## 58 3.838972e-04 9.996161e-01  
## 59 7.991157e-01 2.008843e-01  
## 60 8.459263e-02 9.154074e-01  
## 61 9.836914e-01 1.630856e-02  
## 62 3.864932e-04 9.996135e-01  
## 63 8.460224e-02 9.153978e-01  
## 64 9.909518e-01 9.048231e-03  
## 65 1.495514e-01 8.504486e-01  
## 66 1.378020e-01 8.621980e-01  
## 67 1.478791e-03 9.985212e-01  
## 68 2.505015e-01 7.494985e-01  
## 69 3.839017e-02 9.616098e-01  
## 70 8.253938e-01 1.746062e-01  
## 71 9.990083e-01 9.917461e-04  
## 72 1.334868e-01 8.665132e-01  
## 73 3.652551e-02 9.634745e-01  
## 74 9.182929e-02 9.081707e-01  
## 75 1.719555e-01 8.280445e-01  
## 76 8.192841e-03 9.918072e-01  
## 77 7.793838e-01 2.206162e-01  
## 78 1.823050e-02 9.817695e-01  
## 79 3.556908e-01 6.443092e-01  
## 80 8.440197e-01 1.559803e-01  
## 81 4.857633e-01 5.142367e-01  
## 82 1.495513e-01 8.504487e-01  
## 83 1.734852e-02 9.826515e-01  
## 84 5.378852e-02 9.462115e-01  
## 85 6.044819e-01 3.955181e-01  
## 86 9.625888e-01 3.741121e-02  
## 87 5.214176e-02 9.478582e-01  
## 88 6.300098e-01 3.699902e-01  
## 89 1.404114e-02 9.859589e-01  
## 90 3.920152e-02 9.607985e-01  
## 91 5.871166e-02 9.412883e-01  
## 92 5.389863e-01 4.610137e-01  
## 93 1.495513e-01 8.504487e-01  
## 94 3.194959e-03 9.968050e-01  
## 95 8.678710e-01 1.321290e-01  
## 96 8.566318e-06 9.999914e-01  
## 97 8.459027e-02 9.154097e-01  
## 98 9.049644e-02 9.095036e-01  
## 99 7.893696e-01 2.106304e-01  
## 100 1.941277e-04 9.998059e-01  
## 101 5.522218e-01 4.477782e-01  
## 102 1.495512e-01 8.504488e-01  
## 103 9.182919e-02 9.081708e-01  
## 104 3.427181e-03 9.965728e-01  
## 105 3.937452e-01 6.062548e-01  
## 106 7.181171e-01 2.818829e-01  
## 107 1.466555e-01 8.533445e-01  
## 108 2.468846e-03 9.975312e-01  
## 109 6.300157e-01 3.699843e-01  
## 110 6.326149e-01 3.673851e-01  
## 111 6.550123e-01 3.449877e-01  
## 112 4.062901e-03 9.959371e-01  
## 113 7.353296e-02 9.264670e-01  
## 114 9.814572e-01 1.854282e-02  
## 115 8.753185e-01 1.246815e-01  
## 116 6.706357e-02 9.329364e-01  
## 117 2.192234e-01 7.807766e-01  
## 118 8.135743e-01 1.864257e-01  
## 119 9.196721e-03 9.908033e-01  
## 120 1.495512e-01 8.504488e-01  
## 121 3.189291e-03 9.968107e-01  
## 122 7.039731e-02 9.296027e-01  
## 123 4.061127e-03 9.959389e-01  
## 124 1.261495e-01 8.738505e-01  
## 125 1.728176e-04 9.998272e-01  
## 126 1.822100e-02 9.817790e-01  
## 127 2.192206e-01 7.807794e-01  
## 128 5.785154e-01 4.214846e-01  
## 129 1.809082e-01 8.190918e-01  
## 130 3.162370e-03 9.968376e-01  
## 131 5.878576e-02 9.412142e-01  
## 132 6.342801e-02 9.365720e-01  
## 133 1.495524e-01 8.504476e-01  
## 134 5.124000e-01 4.876000e-01  
## 135 8.296668e-03 9.917033e-01  
## 136 1.495511e-01 8.504489e-01  
## 137 5.378839e-02 9.462116e-01  
## 138 6.425907e-01 3.574093e-01  
## 139 1.860797e-04 9.998139e-01  
## 140 2.590629e-03 9.974094e-01  
## 141 1.495543e-01 8.504457e-01  
## 142 9.738466e-01 2.615343e-02  
## 143 8.824990e-01 1.175010e-01  
## 144 5.078194e-03 9.949218e-01  
## 145 9.359133e-01 6.408674e-02  
## 146 1.719625e-01 8.280375e-01  
## 147 5.215062e-05 9.999478e-01  
## 148 3.175121e-02 9.682488e-01  
## 149 1.729924e-03 9.982701e-01  
## 150 4.191898e-03 9.958081e-01  
## 151 1.403449e-02 9.859655e-01  
## 152 3.839697e-02 9.616030e-01  
## 153 3.451108e-02 9.654889e-01  
## 154 1.496389e-01 8.503611e-01  
## 155 1.403138e-02 9.859686e-01  
## 156 1.997238e-02 9.800276e-01  
## 157 2.838103e-01 7.161897e-01  
## 158 1.387415e-01 8.612585e-01  
## 159 1.403471e-02 9.859653e-01  
## 160 9.975037e-01 2.496284e-03  
## 161 3.170408e-02 9.682959e-01  
## 162 5.049789e-02 9.495021e-01  
## 163 8.939790e-01 1.060210e-01  
## 164 1.933547e-01 8.066453e-01  
## 165 9.144196e-01 8.558043e-02  
## 166 2.270645e-02 9.772935e-01  
## 167 5.126100e-05 9.999487e-01  
## 168 1.987841e-01 8.012159e-01  
## 169 2.092641e-01 7.907359e-01  
## 170 6.425898e-01 3.574102e-01  
## 171 1.495511e-01 8.504489e-01  
## 172 8.516345e-01 1.483655e-01  
## 173 4.723039e-02 9.527696e-01  
## 174 2.699232e-02 9.730077e-01  
## 175 3.310328e-01 6.689672e-01  
## 176 9.309300e-01 6.907003e-02  
## 177 1.570289e-01 8.429711e-01  
## 178 1.186326e-01 8.813674e-01  
## 179 1.809129e-01 8.190871e-01  
## 180 3.070442e-01 6.929558e-01  
## 181 6.173109e-01 3.826891e-01  
## 182 1.893391e-04 9.998107e-01  
## 183 1.632651e-01 8.367349e-01  
## 184 1.195987e-03 9.988040e-01  
## 185 8.060102e-06 9.999919e-01  
## 186 9.865250e-01 1.347505e-02  
## 187 8.529164e-02 9.147084e-01  
## 188 9.991190e-01 8.809738e-04  
## 189 1.133672e-02 9.886633e-01  
## 190 2.613829e-01 7.386171e-01  
## 191 9.791033e-01 2.089675e-02  
## 192 6.913187e-01 3.086813e-01  
## 193 1.236867e-01 8.763133e-01  
## 194 7.372095e-02 9.262790e-01  
## 195 1.276318e-01 8.723682e-01  
## 196 7.498941e-04 9.992501e-01  
## 197 1.495513e-01 8.504487e-01  
## 198 1.165399e-01 8.834601e-01  
## 199 3.070415e-01 6.929585e-01  
## 200 1.495737e-01 8.504263e-01  
## 201 1.955429e-01 8.044571e-01  
## 202 9.975265e-01 2.473532e-03  
## 203 5.516039e-05 9.999448e-01  
## 204 1.991009e-03 9.980090e-01  
## 205 2.838102e-01 7.161898e-01  
## 206 9.975052e-01 2.494767e-03  
## 207 2.396793e-03 9.976032e-01  
## 208 3.145130e-02 9.685487e-01  
## 209 3.557651e-01 6.442349e-01  
## 210 1.901142e-01 8.098858e-01  
## 211 9.865281e-01 1.347192e-02  
## 212 9.506788e-02 9.049321e-01  
## 213 8.458150e-01 1.541850e-01  
## 214 3.839159e-02 9.616084e-01  
## 215 4.459575e-01 5.540425e-01  
## 216 9.189488e-03 9.908105e-01  
## 217 9.936019e-01 6.398146e-03  
## 218 8.856851e-02 9.114315e-01  
## 219 2.192237e-01 7.807763e-01  
## 220 9.998497e-01 1.502659e-04  
## 221 1.495514e-01 8.504486e-01  
## 222 1.997525e-02 9.800247e-01  
## 223 9.991846e-01 8.153825e-04  
## 224 1.216591e-01 8.783409e-01  
## 225 8.170481e-01 1.829519e-01  
## 226 1.632646e-01 8.367354e-01  
## 227 2.699405e-02 9.730059e-01  
## 228 1.047276e-01 8.952724e-01  
## 229 1.823307e-02 9.817669e-01  
## 230 2.504983e-01 7.495017e-01  
## 231 2.071650e-01 7.928350e-01  
## 232 3.297529e-02 9.670247e-01  
## 233 9.049289e-02 9.095071e-01  
## 234 8.894238e-01 1.105762e-01  
## 235 6.425905e-01 3.574095e-01  
## 236 9.625895e-01 3.741051e-02  
## 237 5.421671e-02 9.457833e-01  
## 238 5.213835e-02 9.478617e-01  
## 239 1.030062e-01 8.969938e-01  
## 240 1.495592e-01 8.504408e-01  
## 241 7.859577e-01 2.140423e-01  
## 242 5.869745e-02 9.413025e-01  
## 243 9.049495e-02 9.095050e-01  
## 244 6.232926e-05 9.999377e-01  
## 245 1.548298e-01 8.451702e-01  
## 246 3.432859e-01 6.567141e-01  
## 247 7.479397e-01 2.520603e-01  
## 248 4.588351e-02 9.541165e-01  
## 249 5.915493e-01 4.084507e-01  
## 250 7.551536e-04 9.992448e-01  
## 251 9.656100e-01 3.438997e-02  
## 252 7.490397e-03 9.925096e-01  
## 253 2.504974e-01 7.495026e-01  
## 254 1.605555e-02 9.839444e-01  
## 255 1.495522e-01 8.504478e-01  
## 256 9.146024e-01 8.539757e-02  
## 257 4.990816e-01 5.009184e-01  
## 258 7.895217e-02 9.210478e-01  
## 259 1.047875e-02 9.895212e-01  
## 260 6.793885e-01 3.206115e-01  
## 261 9.198613e-03 9.908014e-01  
## 262 3.556887e-01 6.443113e-01  
## 263 1.483718e-02 9.851628e-01  
## 264 4.645184e-03 9.953548e-01  
## 265 9.506799e-02 9.049320e-01  
## 266 9.975095e-01 2.490488e-03  
## 267 1.495733e-01 8.504267e-01  
## 268 1.171067e-01 8.828933e-01  
## 269 6.044843e-01 3.955157e-01  
## 270 4.162076e-04 9.995838e-01  
## 271 1.495518e-01 8.504482e-01  
## 272 1.828695e-04 9.998171e-01  
## 273 6.788139e-04 9.993212e-01  
## 274 2.699407e-02 9.730059e-01  
## 275 9.020174e-02 9.097983e-01  
## 276 9.364957e-01 6.350430e-02  
## 277 8.678822e-01 1.321178e-01  
## 278 2.410704e-01 7.589296e-01  
## 279 8.919025e-03 9.910810e-01  
## 280 8.892760e-01 1.107240e-01  
## 281 2.888672e-02 9.711133e-01  
## 282 2.180539e-01 7.819461e-01  
## 283 6.842735e-04 9.993157e-01  
## 284 1.496193e-01 8.503807e-01  
## 285 3.070497e-01 6.929503e-01  
## 286 9.257263e-01 7.427371e-02  
## 287 3.070445e-01 6.929555e-01  
## 288 7.764268e-01 2.235732e-01  
## 289 1.495577e-01 8.504423e-01  
## 290 9.853314e-01 1.466864e-02  
## 291 4.164452e-04 9.995836e-01  
## 292 1.171053e-01 8.828947e-01  
## 293 9.938051e-01 6.194892e-03  
## 294 4.238673e-02 9.576133e-01  
## 295 8.200446e-02 9.179955e-01  
## 296 2.093396e-01 7.906604e-01  
## 297 6.630106e-01 3.369894e-01  
## 298 1.062842e-01 8.937158e-01  
## 299 9.788583e-01 2.114174e-02

summary(res.fcm)

## Summary for 'res.fcm'  
##   
## Number of data objects: 299   
##   
## Number of clusters: 2   
##   
## Crisp clustering vector:  
## [1] 2 2 2 2 1 2 2 1 2 1 1 2 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 1 1 2 2 2  
## [38] 1 1 1 2 2 2 2 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 1 2 2 1 2 2 2 2 2 1 1 2 2 2  
## [75] 2 2 1 2 2 1 2 2 2 2 1 1 2 1 2 2 2 1 2 2 1 2 2 2 1 2 1 2 2 2 2 1 2 2 1 1 1  
## [112] 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 1 1 2 1 2 2 2  
## [149] 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 1 2 2 2 2  
## [186] 1 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 1 2 1 2 2 2 1 2 2 1 2 2  
## [223] 1 2 1 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 1 2 2 2 2 2 1 2 1 2 1 2 2 2 2 1 2 2 2  
## [260] 1 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 1 1 2 2 1 2 2 2 2 2 1 2 1 2 1 2 2 1 2 2 2  
## [297] 1 2 1  
##   
## Initial cluster prototypes:  
## age anaemia creatinine\_phosphokinase diabetes ejection\_fraction  
## Cluster 1 58 0 144 1 38  
## Cluster 2 60 1 257 1 30  
## high\_blood\_pressure platelets serum\_creatinine serum\_sodium sex  
## Cluster 1 1 327000 0.7 142 0  
## Cluster 2 0 150000 1.0 137 1  
## smoking time DEATH\_EVENT  
## Cluster 1 0 83 0  
## Cluster 2 1 245 0  
##   
## Final cluster prototypes:  
## age anaemia creatinine\_phosphokinase diabetes  
## Cluster 1 59.94217 0.4675379 602.8002 0.4546466  
## Cluster 2 61.21116 0.4256562 569.2299 0.4011170  
## ejection\_fraction high\_blood\_pressure platelets serum\_creatinine  
## Cluster 1 38.84609 0.3914613 369137.9 1.391867  
## Cluster 2 37.76043 0.3372006 218999.8 1.414250  
## serum\_sodium sex smoking time DEATH\_EVENT  
## Cluster 1 137.1772 0.5839675 0.3379140 129.6027 0.3431873  
## Cluster 2 136.4659 0.6821061 0.3129952 130.7181 0.3182897  
##   
## Distance between the final cluster prototypes  
## Cluster 1  
## Cluster 2 22541453774  
##   
## Difference between the initial and final cluster prototypes  
## age anaemia creatinine\_phosphokinase diabetes  
## Cluster 1 1.942170 0.4675379 458.8002 -0.5453534  
## Cluster 2 1.211157 -0.5743438 312.2299 -0.5988830  
## ejection\_fraction high\_blood\_pressure platelets serum\_creatinine  
## Cluster 1 0.8460913 -0.6085387 42137.95 0.6918675  
## Cluster 2 7.7604340 0.3372006 68999.83 0.4142498  
## serum\_sodium sex smoking time DEATH\_EVENT  
## Cluster 1 -4.8228294 0.5839675 0.3379140 46.60272 0.3431873  
## Cluster 2 -0.5341029 -0.3178939 -0.6870048 -114.28195 0.3182897  
##   
## Root Mean Squared Deviations (RMSD): 57170.39   
## Mean Absolute Deviation (MAD): 728608.6   
##   
## Membership degrees matrix (top and bottom 5 rows):   
## Cluster 1 Cluster 2  
## 1 0.163264502 0.8367355  
## 2 0.152367260 0.8476327  
## 3 0.070396243 0.9296038  
## 4 0.003196944 0.9968031  
## 5 0.867871045 0.1321290  
## ...  
## Cluster 1 Cluster 2  
## 295 0.08200446 0.91799554  
## 296 0.20933962 0.79066038  
## 297 0.66301055 0.33698945  
## 298 0.10628416 0.89371584  
## 299 0.97885826 0.02114173  
##   
## Descriptive statistics for the membership degrees by clusters  
## Size Min Q1 Mean Median Q3 Max  
## Cluster 1 85 0.5124000 0.7030732 0.8383316 0.8824916 0.9791033 0.9999288  
## Cluster 2 214 0.5009184 0.8504397 0.8956336 0.9154085 0.9859637 0.9999919  
##   
## Dunn's Fuzziness Coefficients:  
## dunn\_coeff normalized   
## 0.8176925 0.6353849   
##   
## Within cluster sum of squares by cluster:  
## 1 2   
## 732105367502 626943609639   
## (between\_SS / total\_SS = 50.22%)   
##   
## Available components:   
## [1] "u" "v" "v0" "d" "x"   
## [6] "cluster" "csize" "sumsqrs" "k" "m"   
## [11] "iter" "best.start" "func.val" "comp.time" "inpargs"   
## [16] "algorithm" "call"

plotcluster(res.fcm, cp=1, trans=TRUE)



res.fcm2<- ppclust2(res.fcm, "kmeans")  
factoextra::fviz\_cluster(res.fcm2, data = d1,   
 ellipse.type = "convex",  
 palette = "jco",  
 repel = TRUE)

## Warning: ggrepel: 262 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps

