

HeartFailureProject.R

danielpeslherbe

2020-07-30

```
##Heart Failure Clinical Records Dataset

##First we load our csv dataset into R
dataf <- read.csv("~/Downloads/heart_failure_clinical_records_dataset.csv")

##Packages used:
##install.packages('MASS')
library(MASS)
##install.packages('mgcv')
library(mgcv)

## Loading required package: nlme
## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
##install.packages('mda')
library(mda)

## Loading required package: class
## Loaded mda 0.5-2
##install.packages('tree')
library(tree)
##install.packages('matrixStats')
library(matrixStats)
##install.packages('caret')
library(caret)

## Loading required package: lattice
## Loading required package: ggplot2
##install.packages('pROC')
library(pROC)

## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##      cov, smooth, var
##install.packages('ggplot2')
library(ggplot2)
```

```

##Commentary: from an article link on the dataset repository
##we are given the following logical indicators and/or units of measurement of the data
##anemia: 0 is false & 1 is true
##high blood pressure: follows anemia logic
##diabetes: follows anemia logic
##sex: 0 is woman & 1 is man
##smoking: follows anemia logic
##CPK: mcg/L units
##Ejection fraction: Percentage
##Platelets: kiloplatelets/mL
##Serum creatine: mg/dL
##Serum sodium: mEq/L
##Time: number of days
##DEATH_EVENT: 0 is false & 1 is true

```

```

##In this specific script, we explore different types of regression on the data
##Note that since DEATH_EVENT is either 0 or 1, our regression estimates will
##our response variables will be equivalent to percentage of DEATH_EVENT happening

```

```

##Let us start by splitting the data into 10 groups

```

```

set.seed(5)

```

```

groups <- split(dataf, sample(1:10, nrow(dataf), replace = T))

```

```

##we will perform cross-validation for each model on the 8 first groups

```

```

##the remaining two groups will consist of our test sets

```

```

##let us create our error metrics

```

```

##our first model is a simple linear regression on all predictors

```

```

simplelinear.RMSE <- c()

```

```

##our second model is also a simple linear regression, but,

```

```

##this time, we only include the predictors with high significance

```

```

summary(lm(DEATH_EVENT~., data = dataf))

```

```

##

```

```

## Call:

```

```

## lm(formula = DEATH_EVENT ~ ., data = dataf)

```

```

##

```

```

## Residuals:

```

```

##      Min       1Q   Median       3Q      Max

```

```

## -0.80866 -0.28041 -0.04205  0.24742  0.96983

```

```

##

```

```

## Coefficients:

```

```

##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.664e+00  6.954e-01   2.392  0.01738 *
## age            5.767e-03  1.867e-03   3.088  0.00221 **
## anaemia       -2.766e-03  4.438e-02  -0.062  0.95035
## creatinine_phosphokinase 3.427e-05  2.247e-05   1.525  0.12840
## diabetes       1.928e-02  4.410e-02   0.437  0.66236
## ejection_fraction -9.834e-03  1.844e-03  -5.333 1.96e-07 ***
## high_blood_pressure -1.430e-02  4.565e-02  -0.313  0.75438
## platelets      -8.370e-08  2.208e-07  -0.379  0.70492
## serum_creatinine  8.527e-02  2.123e-02   4.017 7.54e-05 ***
## serum_sodium    -7.599e-03  5.024e-03  -1.513  0.13149
## sex            -6.369e-02  5.108e-02  -1.247  0.21353
## smoking        -5.733e-03  5.119e-02  -0.112  0.91091

```

```
## time -2.733e-03 2.903e-04 -9.415 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3646 on 286 degrees of freedom
## Multiple R-squared: 0.4168, Adjusted R-squared: 0.3924
## F-statistic: 17.04 on 12 and 286 DF, p-value: < 2.2e-16
```

```
anova(lm(DEATH_EVENT~., data = dataf))
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## age	1	4.196	4.1960	31.5729	4.557e-08 ***
## anaemia	1	0.127	0.1268	0.9543	0.32946
## creatinine_phosphokinase	1	0.569	0.5695	4.2851	0.03934 *
## diabetes	1	0.043	0.0433	0.3256	0.56869
## ejection_fraction	1	5.203	5.2030	39.1504	1.429e-09 ***
## high_blood_pressure	1	0.275	0.2746	2.0664	0.15167
## platelets	1	0.025	0.0254	0.1908	0.66255
## serum_creatinine	1	4.107	4.1070	30.9035	6.218e-08 ***
## serum_sodium	1	0.631	0.6310	4.7479	0.03015 *
## sex	1	0.201	0.2015	1.5160	0.21923
## smoking	1	0.009	0.0091	0.0687	0.79346
## time	1	11.781	11.7810	88.6465	< 2.2e-16 ***
## Residuals	286	38.009	0.1329		

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##Note those predictors are age, ejection_fraction, serum_creatinine,
```

```
## time
```

```
revisedlinear.RMSE <- c()
```

```
##our third model is generalized linear model, and since we are working
```

```
##with binary {0,1} results we must choose the bernouilli distribution
```

```
generlinear.model <- glm(DEATH_EVENT~., family = binomial(link = "logit"), data = dataf)
stepAIC(generlinear.model, direction = 'both')
```

```
## Start: AIC=245.55
```

```
## DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes +
```

```
## ejection_fraction + high_blood_pressure + platelets + serum_creatinine +
```

```
## serum_sodium + sex + smoking + time
```

```
##
```

	Df	Deviance	AIC
## - anaemia	1	219.56	243.55
## - smoking	1	219.56	243.56
## - high_blood_pressure	1	219.64	243.64
## - diabetes	1	219.72	243.72
## - platelets	1	219.97	243.97
## - sex	1	221.24	245.24
## - creatinine_phosphokinase	1	221.33	245.33
## <none>		219.55	245.55
## - serum_sodium	1	222.40	246.40
## - age	1	229.37	253.37
## - serum_creatinine	1	233.58	257.58

```

## - ejection_fraction          1   246.70 270.70
## - time                       1   294.28 318.28
##
## Step: AIC=243.55
## DEATH_EVENT ~ age + creatinine_phosphokinase + diabetes + ejection_fraction +
##   high_blood_pressure + platelets + serum_creatinine + serum_sodium +
##   sex + smoking + time
##
##               Df Deviance    AIC
## - smoking          1   219.56 241.56
## - high_blood_pressure 1   219.64 241.64
## - diabetes          1   219.73 241.73
## - platelets         1   219.97 241.97
## - sex               1   221.25 243.25
## - creatinine_phosphokinase 1 221.41 243.41
## <none>              219.56 243.55
## - serum_sodium      1   222.43 244.43
## + anaemia           1   219.55 245.55
## - age               1   229.38 251.38
## - serum_creatinine  1   233.58 255.58
## - ejection_fraction 1   246.71 268.71
## - time              1   296.22 318.22
##
## Step: AIC=241.56
## DEATH_EVENT ~ age + creatinine_phosphokinase + diabetes + ejection_fraction +
##   high_blood_pressure + platelets + serum_creatinine + serum_sodium +
##   sex + time
##
##               Df Deviance    AIC
## - high_blood_pressure 1   219.64 239.64
## - diabetes            1   219.73 239.73
## - platelets           1   219.98 239.98
## - creatinine_phosphokinase 1 221.42 241.42
## <none>                219.56 241.56
## - sex                1   221.62 241.62
## - serum_sodium        1   222.43 242.43
## + smoking             1   219.56 243.55
## + anaemia             1   219.56 243.56
## - age                 1   229.38 249.38
## - serum_creatinine    1   233.78 253.78
## - ejection_fraction   1   246.71 266.71
## - time                1   296.28 316.28
##
## Step: AIC=239.64
## DEATH_EVENT ~ age + creatinine_phosphokinase + diabetes + ejection_fraction +
##   platelets + serum_creatinine + serum_sodium + sex + time
##
##               Df Deviance    AIC
## - diabetes            1   219.81 237.81
## - platelets           1   220.08 238.08
## - creatinine_phosphokinase 1 221.54 239.54
## - sex                1   221.62 239.62
## <none>                219.64 239.64
## - serum_sodium        1   222.51 240.51

```

```

## + high_blood_pressure      1   219.56 241.56
## + smoking                  1   219.64 241.64
## + anaemia                  1   219.64 241.64
## - age                      1   229.41 247.41
## - serum_creatinine         1   234.31 252.31
## - ejection_fraction        1   246.71 264.71
## - time                     1   298.25 316.25
##
## Step: AIC=237.81
## DEATH_EVENT ~ age + creatinine_phosphokinase + ejection_fraction +
##   platelets + serum_creatinine + serum_sodium + sex + time
##
##           Df Deviance    AIC
## - platelets      1   220.22 236.22
## - creatinine_phosphokinase 1   221.71 237.71
## <none>              219.81 237.81
## - sex            1   221.93 237.93
## - serum_sodium   1   222.89 238.89
## + diabetes       1   219.64 239.64
## + high_blood_pressure 1   219.73 239.73
## + smoking        1   219.81 239.81
## + anaemia        1   219.81 239.81
## - age            1   229.42 245.42
## - serum_creatinine 1   234.39 250.39
## - ejection_fraction 1   246.78 262.78
## - time           1   298.46 314.46
##
## Step: AIC=236.22
## DEATH_EVENT ~ age + creatinine_phosphokinase + ejection_fraction +
##   serum_creatinine + serum_sodium + sex + time
##
##           Df Deviance    AIC
## - creatinine_phosphokinase 1   222.04 236.04
## - sex                      1   222.18 236.18
## <none>                      220.22 236.22
## - serum_sodium            1   223.46 237.46
## + platelets               1   219.81 237.81
## + diabetes                1   220.08 238.08
## + high_blood_pressure     1   220.12 238.12
## + smoking                 1   220.21 238.21
## + anaemia                 1   220.22 238.22
## - age                     1   229.90 243.90
## - serum_creatinine        1   235.03 249.03
## - ejection_fraction       1   247.35 261.35
## - time                    1   298.56 312.56
##
## Step: AIC=236.04
## DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium +
##   sex + time
##
##           Df Deviance    AIC
## - sex            1   223.49 235.49
## <none>            222.04 236.04
## + creatinine_phosphokinase 1   220.22 236.22

```

```

## - serum_sodium          1  225.08 237.08
## + platelets             1  221.71 237.71
## + high_blood_pressure   1  221.90 237.90
## + diabetes              1  221.91 237.91
## + anaemia               1  221.98 237.98
## + smoking               1  222.02 238.02
## - age                   1  231.03 243.03
## - serum_creatinine      1  237.43 249.43
## - ejection_fraction     1  249.72 261.72
## - time                  1  301.44 313.44
##
## Step: AIC=235.49
## DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium +
##   time
##
##               Df Deviance    AIC
## <none>                223.49 235.49
## + sex                 1  222.04 236.04
## + creatinine_phosphokinase 1  222.18 236.18
## - serum_sodium        1  226.30 236.30
## + smoking             1  223.09 237.09
## + diabetes            1  223.25 237.25
## + platelets           1  223.26 237.26
## + high_blood_pressure 1  223.46 237.46
## + anaemia             1  223.48 237.48
## - age                 1  232.02 242.02
## - serum_creatinine    1  239.56 249.56
## - ejection_fraction   1  249.83 259.83
## - time                1  303.09 313.09
##
## Call: glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##   serum_sodium + time, family = binomial(link = "logit"), data = dataf)
##
## Coefficients:
##   (Intercept)          age ejection_fraction  serum_creatinine
##      9.49303       0.04247       -0.07343         0.68599
##   serum_sodium          time
##    -0.06456       -0.02089
##
## Degrees of Freedom: 298 Total (i.e. Null);  293 Residual
## Null Deviance:      375.3
## Residual Deviance: 223.5    AIC: 235.5
##
##Note the predictors used for the generalized linear model are age,
##ejection_fraction, serum_creatinine, serum_sodium, & time
generlinear.RMSE <- c()
##our fourth model is a generalized additive model
generaddit.model <- gam(DEATH_EVENT ~ s(age) + anaemia + s(creatinine_phosphokinase) + diabetes + s(eje
summary(generaddit.model)
##
## Family: gaussian
## Link function: identity
##

```

```

## Formula:
## DEATH_EVENT ~ s(age) + anaemia + s(creatinine_phosphokinase) +
##     diabetes + s(ejection_fraction) + high_blood_pressure + s(platelets) +
##     s(serum_creatinine) + s(serum_sodium) + sex + smoking + s(time)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.369576   0.046626   7.926 5.68e-14 ***
## anaemia         0.007868   0.039865    0.197   0.844
## diabetes       -0.007025   0.040455   -0.174   0.862
## high_blood_pressure -0.004575  0.041349   -0.111   0.912
## sex            -0.073657   0.045351   -1.624   0.105
## smoking         0.001350   0.046190    0.029   0.977
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)         1.941  2.447  3.394  0.02426 *
## s(creatinine_phosphokinase) 1.000  1.000  4.955  0.02681 *
## s(ejection_fraction)      3.100  3.915  9.164 8.31e-07 ***
## s(platelets)              1.348  1.624  0.195  0.67710
## s(serum_creatinine)       3.517  4.304  5.209  0.00033 ***
## s(serum_sodium)           1.001  1.001  0.445  0.50536
## s(time)                  6.371  7.532 18.699 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.531   Deviance explained = 56.8%
## -REML = 126.74   Scale est. = 0.10261   n = 299

##Note we use only the predictors that are significant; for our
##generalized additive model, they are s(age), s(creatinine_phosphokinase),
##s(ejection_fraction), s(serum_creatinine), & s(time)
generaddit.RMSE <- c()
##our fifth model is a linear discriminant analysis model
lindiscr.RMSE <- c()
##our sixth model is a quadratic discriminant analysis model
quaddiscr.RMSE <- c()
##our seventh model is a mixed discriminant analysis model
mixeddiscr.RMSE <- c()
##our eighth model is a flexible discriminant analysis model
flexdiscr.RMSE <- c()
##our ninth model is a simple decision tree model
decisiontree.RMSE <- c()
##our tenth model is a pruned version of our ninth models
prunedectree.RMSE <- c()
##our last model is a K nearest neighbours model
knearest.RMSE <- c()

##1st fold-out training
trainset <- data.frame()
for (i in 2:8){

```

```

trainset <- rbind(trainset, groups[[i]])
}

simplelinear.model <- lm(DEATH_EVENT ~ ., data = trainset)
simplelinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Coefficients:
##              (Intercept)              age              anaemia
##              1.514e+00              7.791e-03              -3.184e-02
## creatinine_phosphokinase              diabetes              ejection_fraction
##              3.655e-05              6.290e-02              -7.477e-03
##              high_blood_pressure              platelets              serum_creatinine
##              3.698e-02              4.251e-09              6.676e-02
##              serum_sodium              sex              smoking
##              -8.616e-03              -5.773e-02              1.197e-02
##              time
##              -2.527e-03
summary(simplelinear.model)

```

```

##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75829 -0.26686 -0.05271  0.22311  0.99729
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.514e+00  9.059e-01   1.672 0.096229 .
## age              7.791e-03  2.300e-03   3.388 0.000856 ***
## anaemia          -3.184e-02  5.543e-02  -0.574 0.566392
## creatinine_phosphokinase 3.655e-05  2.663e-05   1.372 0.171591
## diabetes          6.290e-02  5.460e-02   1.152 0.250715
## ejection_fraction  -7.477e-03  2.418e-03  -3.092 0.002285 **
## high_blood_pressure  3.698e-02  5.530e-02   0.669 0.504442
## platelets          4.251e-09  2.675e-07   0.016 0.987337
## serum_creatinine    6.676e-02  2.956e-02   2.259 0.025025 *
## serum_sodium       -8.616e-03  6.504e-03  -1.325 0.186822
## sex                -5.773e-02  6.299e-02  -0.917 0.360518
## smoking            1.197e-02  6.531e-02   0.183 0.854826
## time              -2.527e-03  3.543e-04  -7.134 1.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3711 on 192 degrees of freedom
## Multiple R-squared:  0.383, Adjusted R-squared:  0.3444
## F-statistic: 9.93 on 12 and 192 DF, p-value: 5.272e-15

```



```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1  3.9845   3.9845  28.9348 2.162e-07 ***
## anaemia     1  0.0079   0.0079   0.0571 0.8113746
## creatinine_phosphokinase 1  0.3334   0.3334   2.4210 0.1213685
## diabetes    1  0.1782   0.1782   1.2938 0.2567712
## ejection_fraction 1  2.1380   2.1380  15.5254 0.0001139 ***
## high_blood_pressure 1  0.4755   0.4755   3.4529 0.0646728 .
## platelets    1  0.0640   0.0640   0.4649 0.4961843
## serum_creatinine 1  1.6134   1.6134  11.7161 0.0007573 ***
## serum_sodium 1  0.5186   0.5186   3.7663 0.0537590 .
## sex         1  0.0599   0.0599   0.4353 0.5101714
## smoking     1  0.0280   0.0280   0.2034 0.6525191
## time        1  7.0077   7.0077  50.8884 1.945e-11 ***
## Residuals   192 26.4397   0.1377
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[1]])
```

```
simplelinear.valid
```

```
##           22           46           49           52           66           68
## 0.75485423 0.53463934 1.00043060 0.57506353 0.72704046 0.51212838
##           76           77           126           129           132           134
## 0.42178640 0.32483417 0.13616199 0.37874728 0.66731875 0.12305869
##           149           158           170           173           176           177
## 0.48899211 0.27308743 0.40777409 0.03253430 -0.03081954 0.29140654
##           191           211           215           223           239           259
## 0.44969646 0.41591104 0.12114531 -0.12528937 0.14433360 -0.03635845
##           260           268
## -0.18929500 -0.06413539
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
```

```
revisedlinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##           (Intercept)           age ejection_fraction serum_creatinine
##           0.416492           0.006931           -0.007641           0.071791
##           time
##           -0.002603
```

```
summary(revisedlinear.model)
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7277 -0.2925 -0.0499  0.2429  1.0237
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.4164919  0.1675215   2.486  0.01373 *
## age            0.0069313  0.0022353   3.101  0.00221 **
## ejection_fraction -0.0076412  0.0023449  -3.259  0.00132 **
## serum_creatinine  0.0717913  0.0288038   2.492  0.01350 *
## time           -0.0026033  0.0003448  -7.551 1.51e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.37 on 200 degrees of freedom
## Multiple R-squared:  0.3612, Adjusted R-squared:  0.3484
## F-statistic: 28.27 on 4 and 200 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1  3.9845   3.9845  29.113 1.920e-07 ***
## ejection_fraction  1  2.1391   2.1391  15.629 0.0001069 ***
## serum_creatinine  1  1.5496   1.5496  11.322 0.0009182 ***
## time         1  7.8028   7.8028  57.011 1.508e-12 ***
## Residuals    200 27.3728   0.1369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[1]])
revisedlinear.valid
```

```
##           22           46           49           52           66           68
## 0.700586573 0.517975436 1.027313840 0.619589194 0.721125321 0.627087421
##           76           77           126           129           132           134
## 0.539754522 0.307993614 0.173281273 0.389570576 0.647884064 0.209460950
##           149           158           170           173           176           177
## 0.544302551 0.331417188 0.314188821 0.142092078 0.065600348 0.278203026
##           191           211           215           223           239           259
## 0.401333912 0.354146583 0.151969913 -0.004134889 0.086468918 -0.003962419
##           260           268
## -0.149325617 0.001111149
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
```

```
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)          age  ejection_fraction  serum_creatinine
##      7.85408        0.04864        -0.05768          0.56010
##      serum_sodium          time
##      -0.05989        -0.01943
##
## Degrees of Freedom: 204 Total (i.e. Null);  199 Residual
## Null Deviance:      249.6
## Residual Deviance: 158.7    AIC: 170.7
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9704  -0.6167  -0.2596   0.4320   2.7036
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    7.854078   7.129229   1.102  0.27060
## age             0.048641   0.017903   2.717  0.00659 **
## ejection_fraction -0.057678   0.018983  -3.038  0.00238 **
## serum_creatinine  0.560104   0.246482   2.272  0.02306 *
## serum_sodium    -0.059889   0.051009  -1.174  0.24037
## time            -0.019429   0.003413  -5.692 1.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 249.60  on 204  degrees of freedom
## Residual deviance: 158.74  on 199  degrees of freedom
## AIC: 170.74
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL              204      249.60
```

```
## age          1    19.300      203      230.30
## ejection_fraction 1    11.989      202      218.31
## serum_creatinine 1     8.537      201      209.77
## serum_sodium     1     2.272      200      207.50
## time            1    48.761      199      158.74
```

```
generlinear.valid <- predict(generlinear.model, groups[[1]], type = 'response')
generlinear.valid
```

```
##          22          46          49          52          66          68
## 0.838638417 0.597142231 0.986540017 0.711497768 0.916366271 0.695739880
##          76          77          126          129          132          134
## 0.566655045 0.171388741 0.101586895 0.335690558 0.841487819 0.070643120
##          149          158          170          173          176          177
## 0.640204634 0.255013169 0.250848788 0.042855777 0.038077314 0.185769252
##          191          211          215          223          239          259
## 0.376865971 0.277084755 0.087129071 0.023308190 0.049443547 0.029182357
##          260          268
## 0.009633338 0.022167821
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - generlinear.valid)^2))/nrow
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 2.24 2.77 2.47 2.16 5.87 total = 16.51
##
## REML score: 85.60611
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29756    0.02263   13.15  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F    p-value
```

```
## s(age) 2.239 2.811 4.342 0.00580 **
## s(creatinine_phosphokinase) 2.774 3.436 1.890 0.11210
## s(ejection_fraction) 2.467 3.062 4.657 0.00345 **
## s(serum_creatinine) 2.158 2.652 3.310 0.03393 *
## s(time) 5.867 7.046 13.241 1.47e-14 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.5 Deviance explained = 53.8%
## -REML = 85.606 Scale est. = 0.10495 n = 205
```

```
generaddit.valid <- predict(generaddit.model, groups[[1]], type = 'response')
generaddit.valid
```

```
##      22      46      49      52      66      68
## 0.804859901 0.717722489 1.268319478 0.804965915 0.719584808 0.503739051
##      76      77      126      129      132      134
## 0.378667243 0.160464029 0.039465593 0.074161680 0.381716041 0.072506129
##      149      158      170      173      176      177
## 0.420395949 0.252539191 0.341084905 0.267471540 0.180255165 0.372867632
##      191      211      215      223      239      259
## 0.562168896 0.475260362 0.035993518 -0.031585616 0.028527375 0.006531876
##      260      268
## -0.177209167 0.009426024
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~ ., data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.702439 0.297561
##
## Group means:
##      age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 0 58.37963 0.3819444      594.0833 0.4236111      39.34722
## 1 66.58470 0.4098361      734.3443 0.4426230      34.86885
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3263889 265322.2      1.187847      137.4514 0.6180556
## 1      0.4590164 267614.9      1.741803      135.8361 0.6065574
##      smoking      time
## 0 0.2916667 159.66667
## 1 0.2950820 71.19672
##
## Coefficients of linear discriminants:
##      LD1
## age      3.488701e-02
## anaemia  -1.425656e-01
## creatinine_phosphokinase 1.636465e-04
## diabetes  2.816367e-01
## ejection_fraction  -3.347912e-02
## high_blood_pressure  1.655944e-01
```

```
## platelets          1.903384e-08
## serum_creatinine   2.989227e-01
## serum_sodium       -3.857932e-02
## sex                -2.585079e-01
## smoking             5.357903e-02
## time               -1.131608e-02
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[1]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##          22          46          49          52          66          68
## 0.886639419 0.590513672 0.980966944 0.662948122 0.863340282 0.548163607
##          76          77          126          129          132          134
## 0.377451868 0.223608203 0.063366466 0.303470551 0.799757750 0.057651311
##          149          158          170          173          176          177
## 0.503904187 0.162184243 0.352524199 0.029627001 0.018425627 0.182214728
##          191          211          215          223          239          259
## 0.428961653 0.366911861 0.056858371 0.009006693 0.067193772 0.017672028
##          260          268
## 0.005529214 0.014326571
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[1]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model
```

```
## Call:
```

```
## qda(DEATH_EVENT ~ ., data = trainset)
```

```
##
```

```
## Prior probabilities of groups:
```

```
##          0          1
## 0.702439 0.297561
```

```
##
```

```
## Group means:
```

```
##          age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 0 58.37963 0.3819444          594.0833 0.4236111          39.34722
## 1 66.58470 0.4098361          734.3443 0.4426230          34.86885
## high_blood_pressure platelets serum_creatinine serum_sodium sex
## 0          0.3263889 265322.2          1.187847          137.4514 0.6180556
## 1          0.4590164 267614.9          1.741803          135.8361 0.6065574
## smoking time
## 0 0.2916667 159.66667
## 1 0.2950820 71.19672
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[1]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid
```

```
##          22          46          49          52          66          68
## 0.6484236689 0.1290294053 0.9999816675 0.3761835987 0.8106612578 0.2520952658
##          76          77          126          129          132          134
## 0.0982664847 0.2675913765 0.0226982908 0.4489631267 1.0000000000 0.0408428027
##          149          158          170          173          176          177
## 0.2569575854 0.0457233324 0.1842998717 0.0195610940 0.0015495226 0.0352212930
##          191          211          215          223          239          259
```

```

## 0.0845602123 0.5429912042 0.0262879246 0.0009710506 0.0071049115 0.0072301920
##          260          268
## 0.0006609945 0.0042201697

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[1]]))

mixeddiscr.model <- mda(DEATH_EVENT ~ ., data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 59.69 83.47 93.50 98.83 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.16098 ( N = 205 )
##
## Deviance: 149.802

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[1]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 0.696093381 0.471082704 0.975311528 0.290101288 0.754988399 0.549108384
## [7] 0.354458247 0.348772613 0.055941427 0.365775385 0.999996602 0.056776180
## [13] 0.485125484 0.185719227 0.423129012 0.016131173 0.016554205 0.089199465
## [19] 0.203447862 0.339505452 0.073658673 0.006854146 0.036312682 0.017340550
## [25] 0.007960823 0.014809824

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[1]]))

flexdiscr.model <- fda(DEATH_EVENT ~ ., data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.16585 ( N = 205 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[1]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

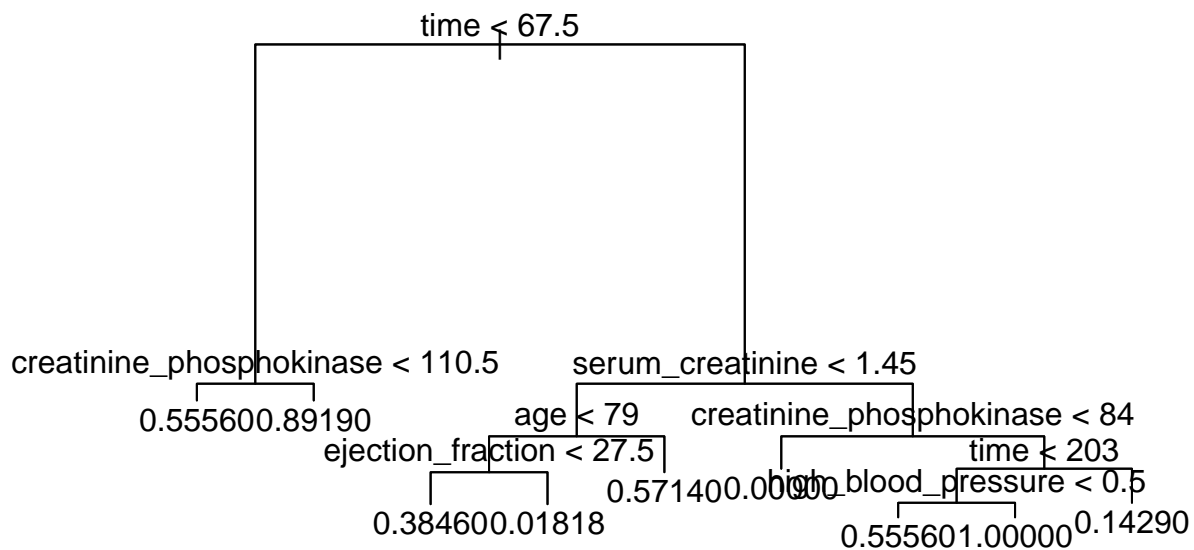
##          22          46          49          52          66          68
## 0.889494870 0.593428884 0.981830333 0.666319863 0.866451113 0.550729864

```

```
##          76          77          126          129          132          134
## 0.378282331 0.222948978 0.062302235 0.303529102 0.803274415 0.056624287
##          149          158          170          173          176          177
## 0.506058050 0.161138584 0.353088796 0.028891065 0.017878441 0.181273406
##          191          211          215          223          239          259
## 0.430344685 0.367629504 0.055837004 0.008675061 0.066107947 0.017139771
##          260          268
## 0.005299354 0.013864892
```

```
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[1]]))
```

```
decisiontree.model <- tree(DEATH_EVENT ~., data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



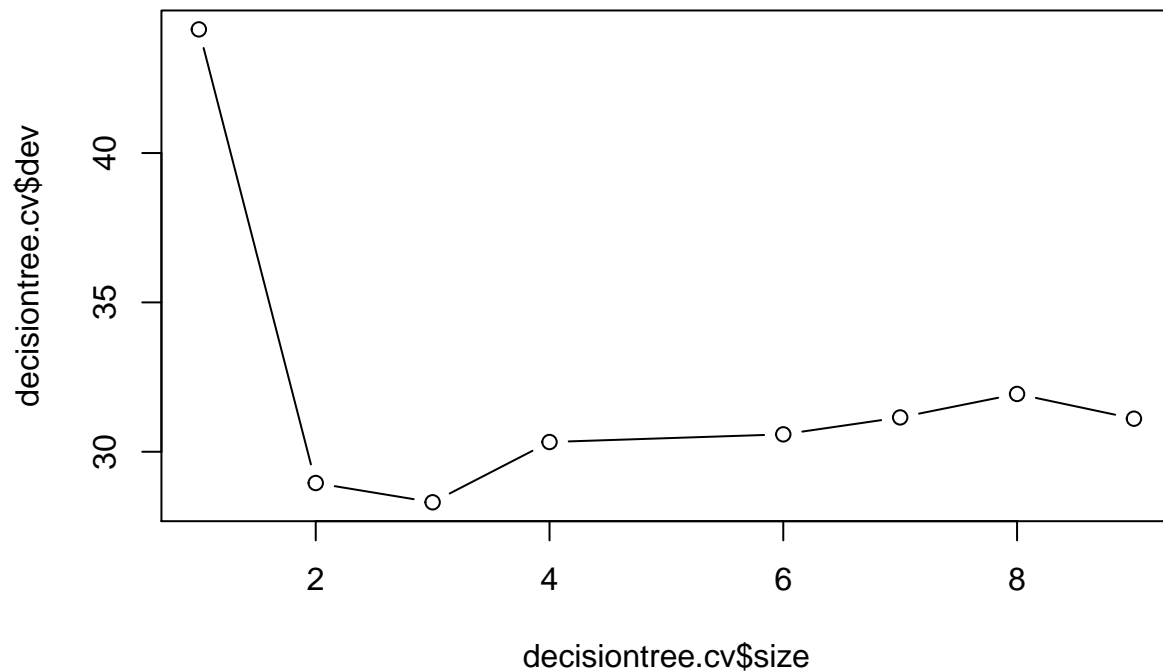
```
decisiontree.valid <- predict(decisiontree.model, groups[[1]], type = 'vector')
decisiontree.valid
```

```
##          22          46          49          52          66          68          76
## 0.89189189 0.89189189 0.89189189 0.55555556 0.55555556 0.55555556 0.38461538
##          77          126          129          132          134          149          158
## 0.01818182 0.01818182 0.01818182 0.55555556 0.01818182 0.55555556 0.38461538
##          170          173          176          177          191          211          215
## 0.01818182 0.01818182 0.01818182 0.01818182 0.55555556 0.38461538 0.01818182
##          223          239          259          260          268
## 0.01818182 0.01818182 0.38461538 0.01818182 0.14285714
```

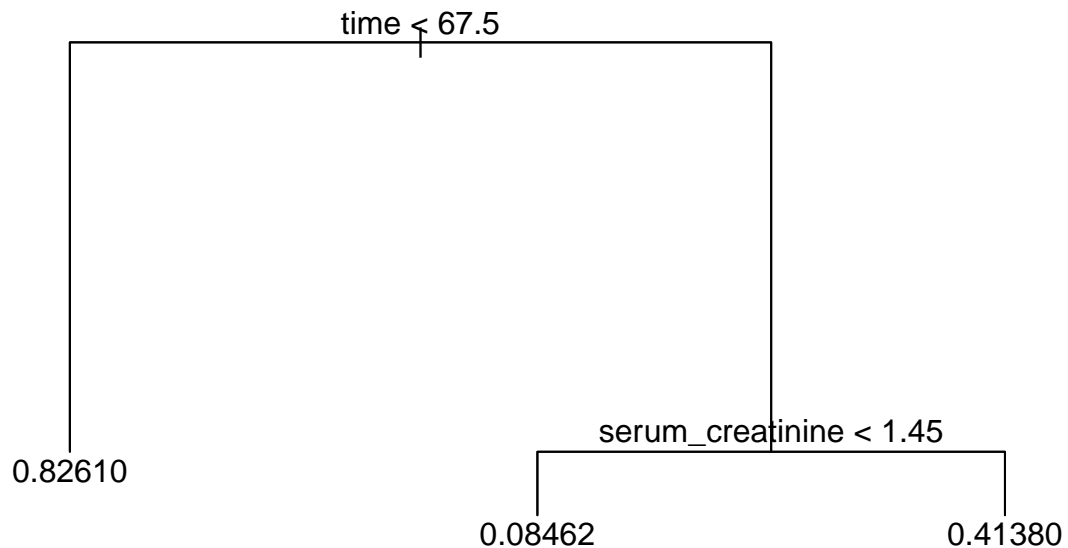


```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - decisiontree.valid)^2))/n)

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree.cv$dev)])
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)
```



```
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.434e+00                6.516e-03                2.920e-02
## creatinine_phosphokinase                diabetes                ejection_fraction
##             5.299e-05                3.743e-02                -9.554e-03
##             high_blood_pressure                platelets                serum_creatinine
##             -2.424e-02                -1.635e-08                6.116e-02
##             serum_sodium                sex                smoking
##             -6.752e-03                -6.124e-02                2.090e-02
##             time
##             -2.581e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7885 -0.2653 -0.0584  0.2521  1.0016
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.434e+00  9.236e-01   1.553  0.12218
## age             6.516e-03  2.258e-03   2.886  0.00436 **
## anaemia         2.920e-02  5.439e-02   0.537  0.59202
## creatinine_phosphokinase 5.299e-05  2.856e-05   1.855  0.06508 .
## diabetes        3.743e-02  5.559e-02   0.673  0.50154
## ejection_fraction -9.554e-03  2.274e-03  -4.202 4.06e-05 ***
## high_blood_pressure -2.424e-02  5.510e-02  -0.440  0.66044
## platelets        -1.635e-08  2.614e-07  -0.063  0.95020
## serum_creatinine   6.116e-02  2.761e-02   2.215  0.02793 *
## serum_sodium      -6.752e-03  6.573e-03  -1.027  0.30559
## sex              -6.124e-02  6.301e-02  -0.972  0.33239
## smoking           2.090e-02  6.532e-02   0.320  0.74932
## time             -2.581e-03  3.533e-04  -7.305 7.43e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3651 on 190 degrees of freedom
## Multiple R-squared:  0.4008, Adjusted R-squared:  0.363
## F-statistic: 10.59 on 12 and 190 DF,  p-value: 6.222e-16
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.5301   3.5301  26.4844 6.597e-07 ***
## anaemia       1  0.2334   0.2334   1.7509 0.187349
## creatinine_phosphokinase 1  0.5213   0.5213   3.9108 0.049423 *
## diabetes      1  0.0566   0.0566   0.4247 0.515402
```

```
## ejection_fraction      1  3.3384  3.3384 25.0461 1.271e-06 ***
## high_blood_pressure    1  0.0195  0.0195  0.1461  0.702756
## platelets              1  0.0179  0.0179  0.1341  0.714587
## serum_creatinine       1  1.6795  1.6795 12.6006  0.000486 ***
## serum_sodium           1  0.1879  0.1879  1.4093  0.236647
## sex                   1  0.1046  0.1046  0.7848  0.376804
## smoking                1  0.1391  0.1391  1.0438  0.308230
## time                   1  7.1129  7.1129 53.3647 7.428e-12 ***
## Residuals              190 25.3249  0.1333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[2]])
simplelinear.valid
```

```
##          1          28          43          47          54          62
## 0.900759606 0.528975941 0.509949009 0.589882957 0.495530980 0.468006230
##          70          97          104          138          140          143
## 0.620216262 0.570403702 0.544385497 0.607369439 0.304680354 0.413218502
##          151          153          154          169          171          172
## 0.402038799 0.114084304 0.356022846 0.307270030 0.283974512 0.288119072
##          179          183          194          197          205          229
## -0.006312004 0.390026413 0.307764864 0.073549416 0.182940124 0.511568948
##          237          256          261          294
## -0.109690407 0.037804115 -0.122832019 -0.021157401
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##      0.538749          0.006220          -0.009699          0.066634
##              time
##      -0.002578
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74594 -0.29396 -0.04133  0.25367  1.03859
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.538748   0.165932   3.247  0.00137 **
```

```
## age                0.006220    0.002193    2.836  0.00505 **
## ejection_fraction -0.009699    0.002207   -4.394  1.81e-05 ***
## serum_creatinine   0.066634    0.026925    2.475  0.01417 *
## time               -0.002578    0.000344   -7.493  2.19e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3634 on 198 degrees of freedom
## Multiple R-squared:  0.3814, Adjusted R-squared:  0.3689
## F-statistic: 30.52 on 4 and 198 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.5301   3.5301  26.733 5.693e-07 ***
## ejection_fraction 1  3.4449   3.4449  26.088 7.643e-07 ***
## serum_creatinine 1  1.7321   1.7321  13.117 0.0003718 ***
## time          1  7.4134   7.4134  56.142 2.193e-12 ***
## Residuals     198 26.1456   0.1320
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[2]])
revisedlinear.valid
```

```
##           1           28           43           47           54           62
## 0.92754327 0.55728471 0.53257751 0.57550112 0.44240205 0.46037779
##           70           97          104          138          140          143
## 0.64979376 0.56079535 0.35138667 0.58075903 0.33368606 0.33777178
##          151          153          154          169          171          172
## 0.42027120 0.16909028 0.28215838 0.26083420 0.24268122 0.15784865
##          179          183          194          197          205          229
## 0.04970138 0.38355581 0.31646680 0.05183633 0.20054889 0.50014820
##          237          256          261          294
## 0.05484617 0.08106246 -0.04108731 -0.04487333
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##         serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##           7.11442             0.04766            -0.07442             0.50869
##      serum_sodium              time
##        -0.04907            -0.01975
##
## Degrees of Freedom: 202 Total (i.e. Null);  197 Residual
## Null Deviance:      246.5
```

```
## Residual Deviance: 151.6      AIC: 163.6
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0516  -0.6173  -0.2405   0.4199   2.7969
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    7.11442    7.59775   0.936 0.349075
## age             0.04766    0.01796   2.654 0.007966 **
## ejection_fraction -0.07442    0.01940  -3.835 0.000126 ***
## serum_creatinine  0.50869    0.22869   2.224 0.026121 *
## serum_sodium    -0.04907    0.05384  -0.911 0.362114
## time           -0.01975    0.00352  -5.612 2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.47  on 202  degrees of freedom
## Residual deviance: 151.56  on 197  degrees of freedom
## AIC: 163.56
##
## Number of Fisher Scoring iterations: 6
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			202	246.47
## age	1	17.137	201	229.33
## ejection_fraction	1	19.592	200	209.74
## serum_creatinine	1	9.998	199	199.74
## serum_sodium	1	0.521	198	199.22
## time	1	47.659	197	151.56

```
generlinear.valid <- predict(generlinear.model, groups[[2]], type = 'response')
generlinear.valid
```

```
##          1          28          43          47          54          62          70
## 0.97608357 0.64015873 0.55964030 0.73355440 0.44867092 0.51933078 0.79165076
```

```

##           97           104           138           140           143           151           153
## 0.66885340 0.23218275 0.74091698 0.22563393 0.28749394 0.39482029 0.09114333
##           154           169           171           172           179           183           194
## 0.17077878 0.14274056 0.13774242 0.06417322 0.02283414 0.32014316 0.17334990
##           197           205           229           237           256           261           294
## 0.03404792 0.09930128 0.60526817 0.02057220 0.03338716 0.01620150 0.01735260

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - generlinear.valid)^2))/nrow

generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 2.60 1.00 2.49 2.25 4.56 total = 13.89
##
## REML score: 81.01911

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29557    0.02269   13.03  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)          2.600   3.257   3.825  0.00819 **
## s(creatinine_phosphokinase) 1.000   1.000   5.386  0.02134 *
## s(ejection_fraction)      2.488   3.080   8.341 2.66e-05 ***
## s(serum_creatinine)       2.249   2.784   2.731  0.06243 .
## s(time)            4.555   5.604  13.865 9.46e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.501 Deviance explained = 53.2%
## -REML = 81.019 Scale est. = 0.1045 n = 203

generaddit.valid <- predict(generaddit.model, groups[[2]], type = 'response')
generaddit.valid

```

```
##           1           28           43           47           54           62
## 1.29763809 0.61315319 0.57168877 0.75141733 0.41492901 0.42155127
##           70           97          104          138          140          143
## 0.55002237 0.39612203 0.48700743 0.45653934 0.22762873 0.17995991
##          151          153          154          169          171          172
## 0.25080886 0.01920570 0.23439624 0.13387577 0.25302576 0.32289902
##          179          183          194          197          205          229
## 0.11599623 0.39411137 0.30284145 0.10807864 0.14636362 0.39863361
##          237          256          261          294
## 0.04762893 0.07550049 -0.08403167 -0.08812700
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.66201 0.3916084                526.2867 0.3986014          40.02098
## 1 66.47778 0.5000000                719.5500 0.4000000          33.50000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3356643 267715.5          1.192378      137.3566 0.6153846
## 1          0.4000000 262375.1          1.797500      136.0500 0.6166667
##      smoking      time
## 0 0.3006993 159.86014
## 1 0.3166667 70.93333
##
## Coefficients of linear discriminants:
##                                LD1
## age                2.899437e-02
## anaemia            1.299287e-01
## creatinine_phosphokinase 2.358096e-04
## diabetes           1.665669e-01
## ejection_fraction   -4.251588e-02
## high_blood_pressure -1.078733e-01
## platelets           -7.275186e-08
## serum_creatinine     2.721739e-01
## serum_sodium         -3.004561e-02
## sex                 -2.724938e-01
## smoking              9.300765e-02
## time                -1.148492e-02
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[2]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##           1           28           43           47           54           62
## 0.963892205 0.582654594 0.545540066 0.693618295 0.517051584 0.462513735
##           70           97          104          138          140          143
```



```
## 0.742277721 0.659818655 0.612060958 0.722295676 0.190537788 0.357767098
##      151      153      154      169      171      172
## 0.337646390 0.049301083 0.261338741 0.193728055 0.166469433 0.171084034
##      179      183      194      197      205      229
## 0.019554176 0.316664980 0.194342246 0.036229810 0.082205356 0.548725795
##      237      256      261      294
## 0.008703252 0.027526983 0.007847980 0.017418607
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[2]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model
```

```
## Call:
## qda(DEATH_EVENT ~., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 0 58.66201 0.3916084      526.2867 0.3986014      40.02098
## 1 66.47778 0.5000000      719.5500 0.4000000      33.50000
## high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3356643 267715.5      1.192378      137.3566 0.6153846
## 1      0.4000000 262375.1      1.797500      136.0500 0.6166667
##      smoking      time
## 0 0.3006993 159.86014
## 1 0.3166667 70.93333
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[2]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid
```

```
##      1      28      43      47      54      62
## 0.9417488351 0.1706904163 0.2064048702 0.8059759506 0.2409361650 0.5240567486
##      70      97      104      138      140      143
## 0.0198298726 0.1891507503 0.9999997868 0.7708776316 0.1553733759 0.1616776307
##      151      153      154      169      171      172
## 0.1550369957 0.0061451964 0.2023192585 0.0480780543 0.0213390821 0.9957637551
##      179      183      194      197      205      229
## 0.1006288138 0.0216360390 0.0251208126 0.0159386097 0.1048403242 0.9999462027
##      237      256      261      294
## 0.0588188868 0.0157742118 0.0061514465 0.0003341749
```

```
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[2]]))
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~., data = trainset)
##
## Dimension: 5
##
```

```

## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 64.48  91.32  95.19  98.50 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.133 ( N = 203 )
##
## Deviance: 148.029
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[2]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

##      [1] 0.96400888 0.59087946 0.51040707 0.70792686 0.30066512 0.38435095
##      [7] 0.89920959 0.62754106 0.45724616 0.57756295 0.26156144 0.14402651
##     [13] 0.28180661 0.06264071 0.23380673 0.12676962 0.13479776 0.12865196
##     [19] 0.01832666 0.35597304 0.13143515 0.02247750 0.10725213 0.42117248
##     [25] 0.01208973 0.01877596 0.01710670 0.01666880

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[2]]))

flexdiscr.model <- fda(DEATH_EVENT ~., data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.14778 ( N = 203 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[2]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

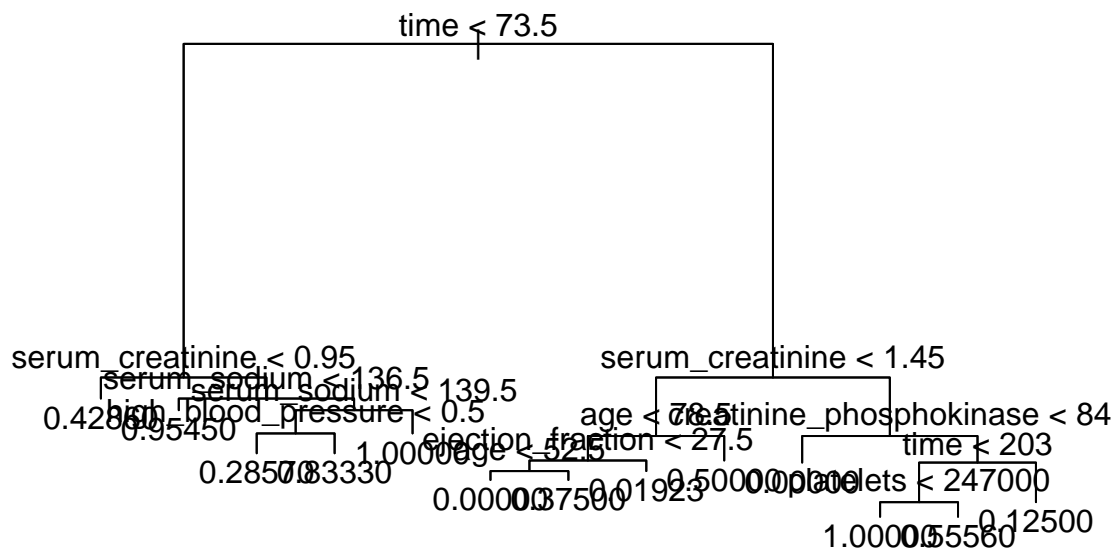
##           1           28           43           47           54           62
## 0.965303166 0.585560463 0.548131958 0.697170934 0.519378669 0.464290909
##           70           97          104          138          140          143
## 0.745927446 0.663229609 0.615185537 0.725922158 0.189652305 0.358415357
##          151          153          154          169          171          172
## 0.338079789 0.048335105 0.261011334 0.192863111 0.165447005 0.170085241
##          179          183          194          197          205          229
## 0.018981432 0.316878951 0.193481316 0.035400902 0.081053467 0.551346108
##          237          256          261          294
## 0.008377500 0.026817859 0.007546219 0.016887859

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[2]]))

decisiontree.model <- tree(DEATH_EVENT ~., data = trainset)
plot(decisiontree.model)

```

```
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[2]], type = 'vector')
decisiontree.valid
```

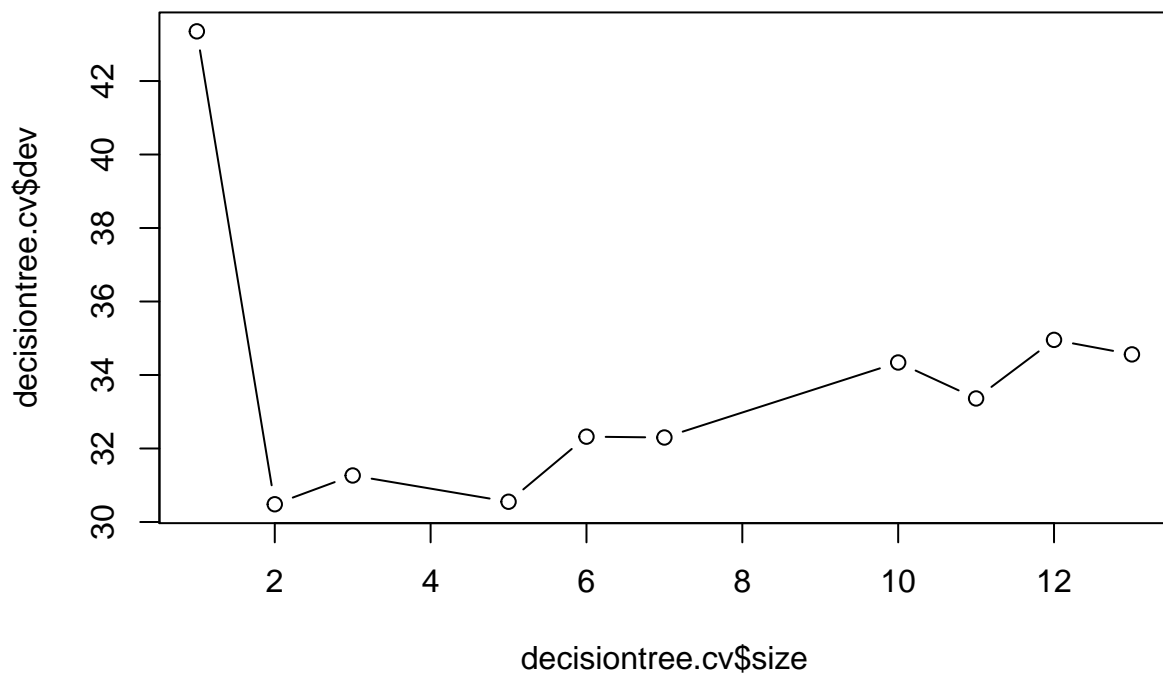
```
##      1      28      43      47      54      62      70
## 0.95454545 0.95454545 0.83333333 0.42857143 0.95454545 0.95454545 0.95454545
##      97     104     138     140     143     151     153
## 0.37500000 0.01923077 0.55555556 0.01923077 0.01923077 1.00000000 0.01923077
##     154     169     171     172     179     183     194
## 0.01923077 0.01923077 0.55555556 0.01923077 0.01923077 0.37500000 0.01923077
##     197     205     229     237     256     261     294
## 0.01923077 0.01923077 0.00000000 0.01923077 0.01923077 0.01923077 0.01923077
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - decisiontree.valid)^2))/n)
```

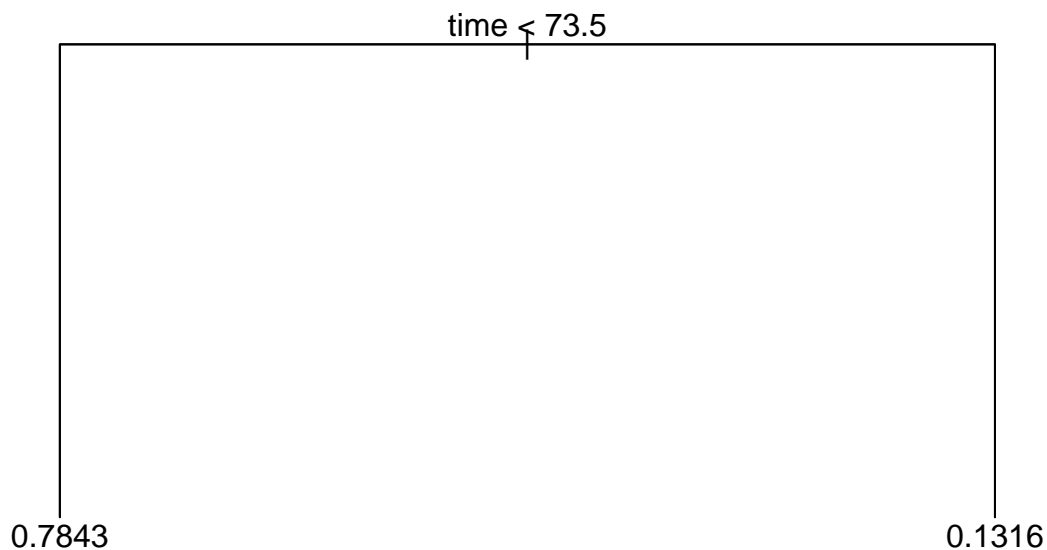
```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[2]], type = 'vector')
prunedectree.valid
```

```
##      1      28      43      47      54      62      70      97
## 0.7843137 0.7843137 0.7843137 0.7843137 0.7843137 0.7843137 0.7843137 0.1315789
##      104      138      140      143      151      153      154      169
## 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789
##      171      172      179      183      194      197      205      229
## 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789 0.1315789
##      237      256      261      294
## 0.1315789 0.1315789 0.1315789 0.1315789
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##3rd fold-out training
```

```
trainset <- data.frame()
```

```
for (i in c(1:2,4:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ ., data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ ., data = trainset)
```

```
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.293e+00                6.921e-03                1.379e-02
## creatinine_phosphokinase                diabetes                ejection_fraction
##             2.458e-05                2.515e-02                -9.698e-03
##             high_blood_pressure                platelets                serum_creatinine
##             5.323e-03                -1.737e-08                5.417e-02
##             serum_sodium                sex                smoking
##             -5.327e-03                -6.146e-02                1.416e-02
##             time
##             -2.865e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7495 -0.2776 -0.0333  0.2526  1.0086
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.293e+00  9.217e-01   1.402   0.1625
## age             6.921e-03  2.276e-03   3.041   0.0027 **
## anaemia        1.379e-02  5.499e-02   0.251   0.8023
## creatinine_phosphokinase 2.458e-05  2.665e-05   0.922   0.3577
## diabetes       2.515e-02  5.506e-02   0.457   0.6484
## ejection_fraction -9.698e-03  2.319e-03  -4.182  4.45e-05 ***
## high_blood_pressure  5.323e-03  5.503e-02   0.097   0.9230
## platelets      -1.737e-08  2.990e-07  -0.058   0.9537
## serum_creatinine  5.417e-02  2.696e-02   2.009   0.0460 *
## serum_sodium    -5.327e-03  6.649e-03  -0.801   0.4241
## sex            -6.146e-02  6.451e-02  -0.953   0.3420
## smoking         1.416e-02  6.502e-02   0.218   0.8279
## time          -2.865e-03  3.689e-04  -7.767  5.35e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3639 on 185 degrees of freedom
## Multiple R-squared:  0.4248, Adjusted R-squared:  0.3875
## F-statistic: 11.39 on 12 and 185 DF,  p-value: < 2.2e-16
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.0370   4.0370  30.4892 1.124e-07 ***
## anaemia       1  0.0084   0.0084   0.0635  0.801371
## creatinine_phosphokinase 1  0.2281   0.2281   1.7229  0.190952
## diabetes      1  0.0048   0.0048   0.0360  0.849646
```

```
## ejection_fraction      1  3.9297  3.9297 29.6794 1.610e-07 ***
## high_blood_pressure    1  0.2668  0.2668  2.0151  0.157423
## platelets              1  0.0204  0.0204  0.1541  0.695111
## serum_creatinine       1  1.3084  1.3084  9.8819  0.001945 **
## serum_sodium           1  0.1572  0.1572  1.1872  0.277321
## sex                    1  0.1233  0.1233  0.9313  0.335788
## smoking                 1  0.0193  0.0193  0.1461  0.702713
## time                    1  7.9873  7.9873 60.3241 5.345e-13 ***
## Residuals              185 24.4951  0.1324
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[3]])
simplelinear.valid
```

```
##           8           9           21           40           42           57
## 0.45064240 0.42893622 0.81046425 0.68474061 0.64164901 0.65859101
##           58           85           95          105          106          108
## 0.44033122 0.58404628 0.39758775 0.33826694 0.63542820 0.34584571
##          109          110          113          146          147          155
## 0.42957064 0.29680640 0.46900032 0.26633803 0.30634695 0.40143500
##          157          174          217          243          247          252
## 0.33109885 0.28122516 0.05699896 -0.09170579 0.12488447 0.03468970
##          257          258          272          274          277          287
## 0.09825620 -0.05180276 -0.07079074 -0.27752092 0.04845192 -0.06152420
##          290          293          295
## 0.11056634 -0.17086124 -0.14049562
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - simplelinear.valid)^2)/n))
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##      0.590483          0.006470          -0.009841          0.057604
##              time
##      -0.002889
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72927 -0.28009 -0.03475  0.23485  1.04042
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5904829  0.1674370   3.527 0.000526 ***
## age            0.0064701  0.0021993   2.942 0.003661 **
## ejection_fraction -0.0098411  0.0021947  -4.484 1.25e-05 ***
## serum_creatinine 0.0576036  0.0259480   2.220 0.027585 *
## time           -0.0028891  0.0003573  -8.087 6.62e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3589 on 193 degrees of freedom
## Multiple R-squared:  0.4164, Adjusted R-squared:  0.4043
## F-statistic: 34.42 on 4 and 193 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##               Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  4.0370   4.0370  31.3486 7.334e-08 ***
## ejection_fraction 1  4.0101   4.0101  31.1399 8.045e-08 ***
## serum_creatinine 1  1.2633   1.2633   9.8098 0.002006 **
## time           1  8.4217   8.4217  65.3982 6.622e-14 ***
## Residuals      193 24.8538   0.1288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[3]])
revisedlinear.valid
```

```
##           8           9           21           40           42           57
## 0.422696184 0.428882633 0.793671854 0.690866149 0.595429267 0.698471870
##           58           85           95          105          106          108
## 0.463597618 0.558448000 0.392317058 0.367773517 0.604785738 0.334804376
##          109          110          113          146          147          155
## 0.433263718 0.357845829 0.500110412 0.335502138 0.348442354 0.365996045
##          157          174          217          243          247          252
## 0.316446822 0.341445330 0.036921272 -0.044269671 0.145414901 0.026833658
##          257          258          272          274          277          287
## 0.099409349 -0.015094614 -0.126274879 -0.198915643 0.054494374 0.009910143
##          290          293          295
## 0.111072749 -0.134809120 -0.099016475
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##         serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##           5.77081           0.05207           -0.07820           0.46704
##      serum_sodium              time
```



```
##           -0.03736           -0.02254
##
## Degrees of Freedom: 197 Total (i.e. Null);  192 Residual
## Null Deviance:      246.1
## Residual Deviance: 142.7      AIC: 154.7
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9890  -0.5729  -0.2140   0.4179   2.8825
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    5.770812   7.670448   0.752 0.451845
## age             0.052070   0.018676   2.788 0.005302 **
## ejection_fraction -0.078200   0.020565  -3.802 0.000143 ***
## serum_creatinine  0.467039   0.217270   2.150 0.031588 *
## serum_sodium    -0.037356   0.054889  -0.681 0.496138
## time            -0.022543   0.003836  -5.877 4.18e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.15  on 197  degrees of freedom
## Residual deviance: 142.67  on 192  degrees of freedom
## AIC: 154.67
##
## Number of Fisher Scoring iterations: 6
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			197	246.15
## age	1	19.207	196	226.94
## ejection_fraction	1	22.395	195	204.54
## serum_creatinine	1	6.973	194	197.57
## serum_sodium	1	0.369	193	197.20
## time	1	54.533	192	142.67

```
generlinear.valid <- predict(generlinear.model, groups[[3]], type = 'response')
generlinear.valid
```

```
##           8           9           21           40           42           57
## 0.400753911 0.352594951 0.911253860 0.793092985 0.687176510 0.827390272
##           58           85           95           105           106           108
## 0.419399684 0.581614027 0.263190142 0.223743701 0.696116538 0.207239514
##           109           110           113           146           147           155
## 0.411249884 0.213449917 0.514310472 0.192039736 0.241240414 0.270196549
##           157           174           217           243           247           252
## 0.199358176 0.214187692 0.022761856 0.014354353 0.058085819 0.019504195
##           257           258           272           274           277           287
## 0.040144866 0.016445240 0.008263929 0.003697947 0.025700417 0.019575795
##           290           293           295
## 0.037524652 0.006230030 0.007485236
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - generlinear.valid)^2))/nrow
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 2.26 1.00 2.53 2.64 6.37 total = 15.81
##
## REML score: 77.88315
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.31313    0.02237      14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)         2.261   2.848  3.759 0.012023 *
## s(creatinine_phosphokinase) 1.000   1.000  1.509 0.220818
## s(ejection_fraction)      2.531   3.132  6.582 0.000247 ***
```

```
## s(serum_creatinine)      2.644  3.282  2.539 0.049352 *
## s(time)                  6.370  7.539 13.612 1.77e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.542   Deviance explained = 57.6%
## -REML = 77.883   Scale est. = 0.099044   n = 198
```

```
generaddit.valid <- predict(generaddit.model, groups[[3]], type = 'response')
generaddit.valid
```

```
##           8           9           21           40           42           57
## 0.59541886 0.65379970 0.89443104 0.84212538 0.74811334 0.74913871
##           58           85           95          105          106          108
## 0.43108267 0.34343797 0.05827006 0.21058541 0.34444546 0.10207958
##          109          110          113          146          147          155
## 0.12012745 0.12381386 0.32678949 0.10614568 0.10681547 0.14000463
##          157          174          217          243          247          252
## 0.20287565 0.45262133 0.04550524 -0.07971949 0.15473774 -0.09377867
##          257          258          272          274          277          287
## 0.04497176 -0.08389103 -0.07129694 -0.13600766 0.06062197 0.09683618
##          290          293          295
## 0.27619653 -0.04041961 -0.01805790
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6868687 0.3131313
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.81373 0.4117647          607.5515 0.4264706          40.25735
## 1 67.02689 0.4516129          701.6129 0.4032258          33.17742
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3455882 261789.7          1.217279      137.0588 0.6470588
## 1          0.4516129 256050.8          1.796452      135.5645 0.6612903
##      smoking      time
## 0 0.3161765 159.98529
## 1 0.3064516  69.91935
##
## Coefficients of linear discriminants:
##                               LD1
## age                3.003853e-02
## anaemia             5.982976e-02
## creatinine_phosphokinase 1.066581e-04
## diabetes            1.091380e-01
## ejection_fraction    -4.208915e-02
## high_blood_pressure    2.310405e-02
## platelets            -7.540239e-08
```

```
## serum_creatinine      2.350919e-01
## serum_sodium         -2.311929e-02
## sex                  -2.667262e-01
## smoking               6.143551e-02
## time                 -1.243420e-02
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[3]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##           8           9           21           40           42           57
## 0.420577508 0.378931221 0.928158850 0.825311414 0.769936798 0.793067784
##           58           85           95           105           106           108
## 0.400613625 0.678533537 0.321928155 0.228009322 0.761001488 0.238859271
##           109           110           113           146           147           155
## 0.380126638 0.174893337 0.456730891 0.142444687 0.186184320 0.328684525
##           157           174           217           243           247           252
## 0.218071201 0.157624384 0.030172763 0.009377192 0.050835559 0.025365119
##           257           258           272           274           277           287
## 0.041484947 0.012859043 0.011066553 0.002135598 0.028234546 0.011908151
##           290           293           295
## 0.045583524 0.004999411 0.006365670
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[3]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model
```

```
## Call:
## qda(DEATH_EVENT ~., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6868687 0.3131313
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.81373 0.4117647                607.5515 0.4264706                40.25735
## 1 67.02689 0.4516129                701.6129 0.4032258                33.17742
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3455882 261789.7                1.217279    137.0588 0.6470588
## 1      0.4516129 256050.8                1.796452    135.5645 0.6612903
##      smoking      time
## 0 0.3161765 159.98529
## 1 0.3064516  69.91935
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[3]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid
```

```
##           8           9           21           40           42           57
## 0.010834319 0.016021352 0.841645466 0.971684202 0.856931689 0.717652465
##           58           85           95           105           106           108
## 0.057236118 0.503460826 0.169303865 0.296751922 0.829919470 0.129819689
##           109           110           113           146           147           155
## 0.145717371 0.161055925 0.166772380 0.054461766 0.111920606 0.164112004
```

```

##          157          174          217          243          247          252
## 0.068188747 0.246052035 0.002742932 0.001996286 0.053403305 0.041391183
##          257          258          272          274          277          287
## 0.008962892 0.001118192 0.006588483 0.004097752 0.006983578 0.007830793
##          290          293          295
## 0.061724554 0.001284900 0.001722222

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[3]]))

mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 66.84  91.70  95.86  99.29 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.12626 ( N = 198 )
##
## Deviance: 124.547

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[3]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 4.456681e-01 2.861060e-01 8.302798e-01 9.183038e-01 8.507274e-01
## [6] 7.548087e-01 4.583438e-01 4.675553e-01 2.189315e-01 1.331484e-01
## [11] 1.031771e-02 1.860902e-01 3.968687e-01 5.217129e-07 3.174544e-01
## [16] 1.591060e-01 2.477934e-01 2.792785e-01 9.122198e-02 5.347929e-01
## [21] 1.773546e-02 1.150862e-02 2.072636e-02 2.426335e-02 2.469822e-02
## [26] 1.645291e-02 1.339267e-02 2.989823e-03 1.766950e-02 7.499729e-03
## [31] 2.697751e-02 4.859916e-03 1.877581e-03

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[3]]))

flexdiscr.model <- fda(DEATH_EVENT ~., data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.14141 ( N = 198 )

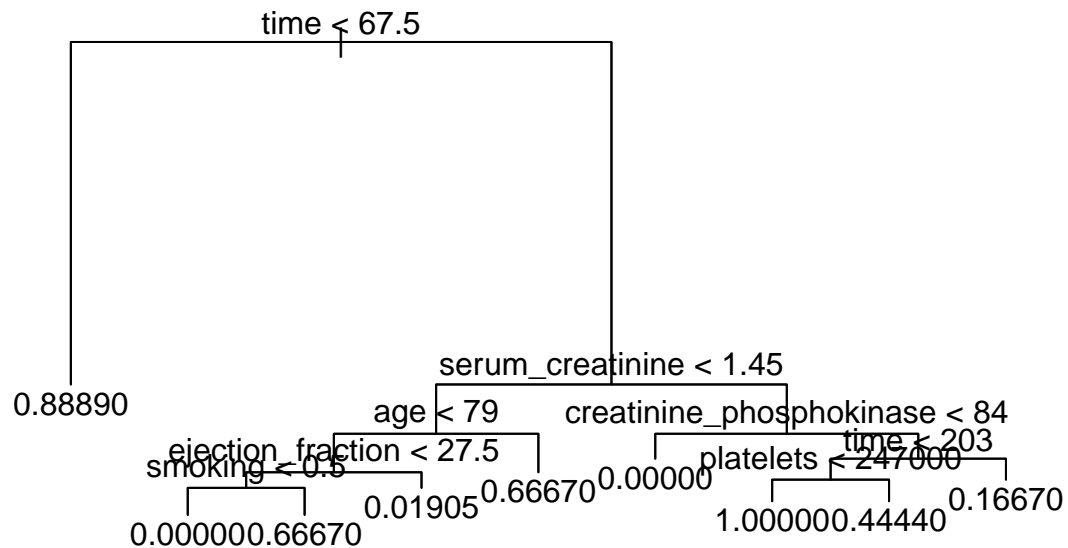
```

```
flexdiscr.valid <- predict(flexdiscr.model, groups[[3]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid
```

```
##           8           9           21           40           42           57
## 0.421734528 0.379631336 0.930401333 0.828724708 0.773520186 0.796610387
##           58           85           95           105           106           108
## 0.401551473 0.681935104 0.322018587 0.227230622 0.764590104 0.238167202
##           109           110           113           146           147           155
## 0.380839822 0.173768531 0.458280863 0.141190864 0.185120660 0.328845260
##           157           174           217           243           247           252
## 0.217217356 0.156421619 0.029381371 0.009016966 0.049791476 0.024652778
##           257           258           272           274           277           287
## 0.040540039 0.012406586 0.010660122 0.002022133 0.027473898 0.011479753
##           290           293           295
## 0.044592403 0.004775725 0.006096232
```

```
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[3]]))
```

```
decisiontree.model <- tree(DEATH_EVENT ~., data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[3]], type = 'vector')
decisiontree.valid
```

```
##           8           9           21           40           42           57           58
```

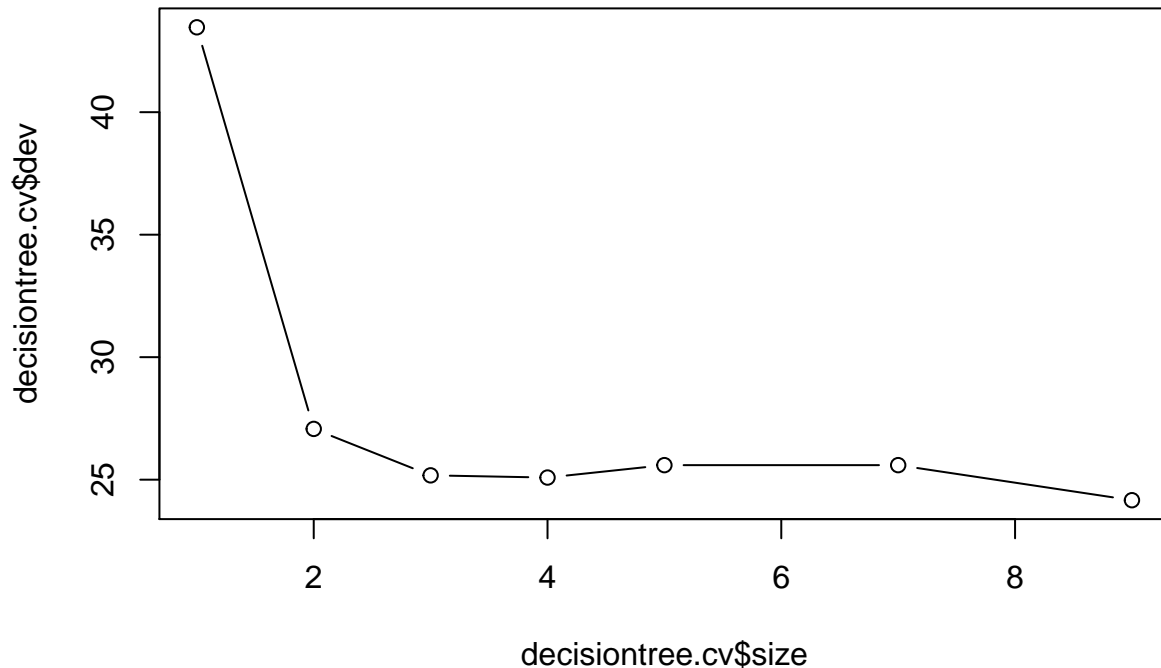
```
## 0.88888889 0.88888889 0.88888889 0.88888889 0.88888889 0.88888889 0.88888889
##      85      95      105      106      108      109      110
## 0.00000000 0.01904762 0.00000000 0.44444444 0.01904762 0.01904762 0.01904762
##      113      146      147      155      157      174      217
## 0.44444444 0.01904762 0.01904762 0.01904762 0.01904762 0.00000000 0.01904762
##      243      247      252      257      258      272      274
## 0.01904762 0.00000000 0.01904762 0.16666667 0.01904762 0.01904762 0.01904762
##      277      287      290      293      295
## 0.01904762 0.16666667 0.66666667 0.01904762 0.01904762
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - decisiontree.valid)^2))/n
```

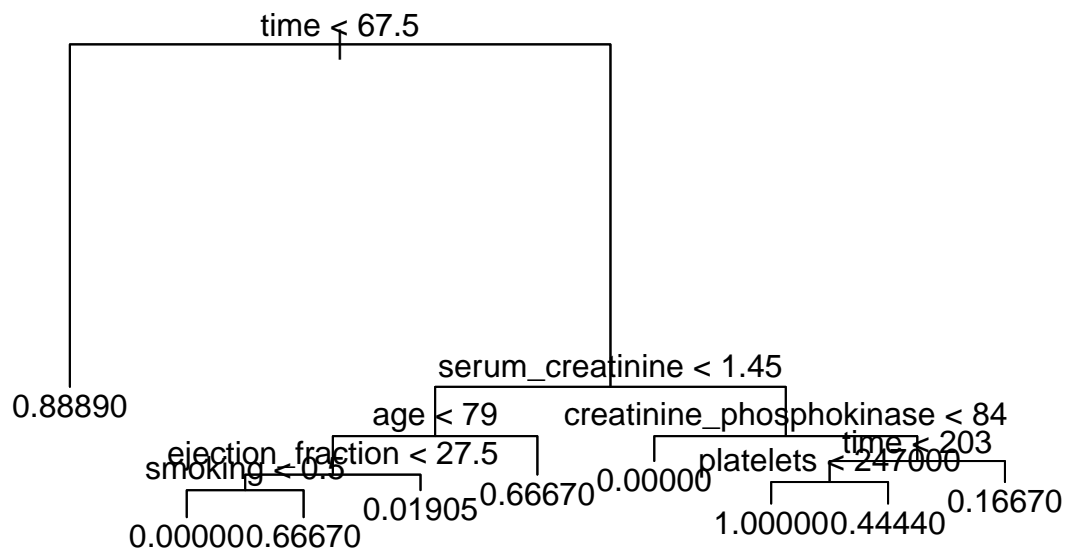
```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[3]], type = 'vector')
prunedectree.valid
```

```
##          8          9          21          40          42          57          58
## 0.88888889 0.88888889 0.88888889 0.88888889 0.88888889 0.88888889 0.88888889
##          85          95          105          106          108          109          110
## 0.00000000 0.01904762 0.00000000 0.44444444 0.01904762 0.01904762 0.01904762
##          113          146          147          155          157          174          217
## 0.44444444 0.01904762 0.01904762 0.01904762 0.01904762 0.00000000 0.01904762
##          243          247          252          257          258          272          274
## 0.01904762 0.00000000 0.01904762 0.16666667 0.01904762 0.01904762 0.01904762
##          277          287          290          293          295
## 0.01904762 0.16666667 0.66666667 0.01904762 0.01904762
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##4th fold-out training
```

```
trainset <- data.frame()
for (i in c(1:3,5:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ ., data = trainset)
simplelinear.model
```

```
##
```



```
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             2.181e+00                6.310e-03                1.697e-02
## creatinine_phosphokinase                diabetes                ejection_fraction
##             2.642e-05                5.470e-02                -8.610e-03
##             high_blood_pressure                platelets                serum_creatinine
##             2.484e-02                5.959e-08                3.983e-02
##             serum_sodium                sex                smoking
##             -1.216e-02                -8.932e-02                3.199e-02
##             time
##             -2.684e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77853 -0.29135 -0.04324  0.25178  0.97657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.181e+00  9.292e-01   2.347 0.019974 *
## age             6.310e-03  2.364e-03   2.669 0.008278 **
## anaemia         1.697e-02  5.610e-02   0.302 0.762646
## creatinine_phosphokinase 2.642e-05  2.745e-05   0.963 0.336994
## diabetes        5.470e-02  5.555e-02   0.985 0.326093
## ejection_fraction -8.610e-03  2.408e-03  -3.576 0.000444 ***
## high_blood_pressure  2.484e-02  5.646e-02   0.440 0.660403
## platelets        5.959e-08  2.709e-07   0.220 0.826117
## serum_creatinine  3.983e-02  3.034e-02   1.313 0.190935
## serum_sodium     -1.216e-02  6.712e-03  -1.811 0.071706 .
## sex             -8.932e-02  6.533e-02  -1.367 0.173191
## smoking          3.199e-02  6.421e-02   0.498 0.618909
## time            -2.684e-03  3.633e-04  -7.388 4.81e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3672 on 187 degrees of freedom
## Multiple R-squared:  0.3937, Adjusted R-squared:  0.3548
## F-statistic: 10.12 on 12 and 187 DF,  p-value: 3.429e-15
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age             1  2.6117   2.6117  19.3668 1.810e-05 ***
## anaemia         1  0.3029   0.3029   2.2460 0.135645
```

```
## creatinine_phosphokinase 1 0.1726 0.1726 1.2802 0.259315
## diabetes 1 0.1798 0.1798 1.3335 0.249648
## ejection_fraction 1 3.1277 3.1277 23.1931 3.017e-06 ***
## high_blood_pressure 1 0.3623 0.3623 2.6863 0.102895
## platelets 1 0.0517 0.0517 0.3837 0.536366
## serum_creatinine 1 0.8347 0.8347 6.1899 0.013723 *
## serum_sodium 1 0.9305 0.9305 6.9002 0.009334 **
## sex 1 0.1592 0.1592 1.1804 0.278672
## smoking 1 0.2830 0.2830 2.0989 0.149083
## time 1 7.3606 7.3606 54.5810 4.805e-12 ***
## Residuals 187 25.2181 0.1349
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[4]])
simplelinear.valid
```

```
##      32      35      36      44      51      53
## 0.70192987 0.42424995 0.73890747 0.53329087 0.61889693 0.58108478
##      56      59      78      81      93      99
## 0.82749422 0.56789585 0.23070751 0.51033045 0.12158364 0.61777273
##     115     117     118     120     144     166
## 0.53333395 0.28012242 0.44780264 0.61611900 0.30535383 0.53148039
##     196     199     201     202     206     209
## 0.05692468 0.36410949 0.03264339 -0.14384648 0.09099259 0.18750940
##     238     240     246     264     271     275
## 0.17068812 -0.06441301 0.05828738 -0.11683656 0.08179669 0.04785348
##     286
## -0.07414430
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Coefficients:
##      (Intercept)          age ejection_fraction  serum_creatinine
##      0.586026      0.005745      -0.008829      0.048506
##           time
##      -0.002811
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75683 -0.29979 -0.04048  0.26710  1.02247
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5860256  0.1703803   3.440 0.000713 ***
## age            0.0057454  0.0023151   2.482 0.013923 *
## ejection_fraction -0.0088292  0.0023599  -3.741 0.000241 ***
## serum_creatinine  0.0485060  0.0290831   1.668 0.096952 .
## time           -0.0028105  0.0003496  -8.039 8.52e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.367 on 195 degrees of freedom
## Multiple R-squared:  0.3684, Adjusted R-squared:  0.3555
## F-statistic: 28.44 on 4 and 195 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  2.6117   2.6117  19.3862 1.758e-05 ***
## ejection_fraction  1  3.2260   3.2260  23.9460 2.070e-06 ***
## serum_creatinine  1  0.7795   0.7795   5.7857  0.01709 *
## time            1  8.7072   8.7072  64.6313 8.521e-14 ***
## Residuals       195 26.2706   0.1347
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[4]])
revisedlinear.valid
```

```
##           32           35           36           44           51           53
## 0.743890805 0.485014684 0.758889138 0.513990264 0.683634421 0.592324942
##           56           59           78           81           93           99
## 0.823445800 0.589743038 0.324391879 0.506704790 0.124353157 0.529331427
##          115          117          118          120          144          166
## 0.380029977 0.170760892 0.435660237 0.566384867 0.317823940 0.407836758
##          196          199          201          202          206          209
## 0.212523791 0.222461605 0.061868932 -0.159434370 -0.001079598 0.100520258
##          238          240          246          264          271          275
## 0.191325834 -0.031073785 0.070253354 -0.170626596 -0.034208602 0.025803226
##          286
## -0.095567234
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - revisedlinear.valid)^2)))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##          serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##          (Intercept)              age ejection_fraction  serum_creatinine
```

```
##          15.58296          0.04492          -0.06626          0.26759
##      serum_sodium          time
##          -0.10788          -0.02196
##
## Degrees of Freedom: 199 Total (i.e. Null);  194 Residual
## Null Deviance:      242.6
## Residual Deviance: 149.5      AIC: 161.5
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0896  -0.6162  -0.2146   0.3804   2.6772
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    15.58296    8.444186   1.845  0.06498 .
## age              0.044923    0.018791   2.391  0.01682 *
## ejection_fraction -0.066258    0.019976  -3.317  0.00091 ***
## serum_creatinine  0.267594    0.253236   1.057  0.29065
## serum_sodium    -0.107879    0.059553  -1.811  0.07007 .
## time            -0.021957    0.003759  -5.841 5.18e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 242.63  on 199  degrees of freedom
## Residual deviance: 149.46  on 194  degrees of freedom
## AIC: 161.46
##
## Number of Fisher Scoring iterations: 6
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			199	242.63
## age	1	12.660	198	229.97
## ejection_fraction	1	17.532	197	212.44
## serum_creatinine	1	3.804	196	208.63
## serum_sodium	1	4.230	195	204.40

```

## time          1    54.940      194      149.46
generlinear.valid <- predict(generlinear.model, groups[[4]], type = 'response')
generlinear.valid

##          32          35          36          44          51          53
## 0.914335823 0.429924344 0.898958618 0.643764792 0.804933751 0.330814722
##          56          59          78          81          93          99
## 0.955221241 0.705620409 0.169754978 0.553254226 0.058972201 0.573764057
##          115         117         118         120         144         166
## 0.588621898 0.095751181 0.338708839 0.706097643 0.152845560 0.398614009
##          196         199         201         202         206         209
## 0.045101923 0.145754979 0.037518976 0.007698543 0.025026763 0.031635213
##          238         240         246         264         271         275
## 0.143058023 0.017557863 0.036395406 0.004599884 0.032866934 0.026577090
##          286
## 0.013015966

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - generlinear.valid)^2))/nrow)

generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 2.05 1.00 2.56 2.57 4.75 total = 13.94
##
## REML score: 83.21142

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29500    0.02321   12.71  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)          2.051  2.595  2.040 0.094906 .
## s(creatinine_phosphokinase) 1.000  1.000  1.940 0.165272

```

```
## s(ejection_fraction)      2.565  3.186  6.097 0.000471 ***
## s(serum_creatinine)      2.572  3.167  2.437 0.064196 .
## s(time)                  4.749  5.828 14.458 8.74e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.485   Deviance explained = 51.8%
## -REML = 83.211   Scale est. = 0.1077    n = 200
```

```
generaddit.valid <- predict(generaddit.model, groups[[4]], type = 'response')
generaddit.valid
```

```
##          32          35          36          44          51          53
## 0.96681800 0.59897430 0.84637730 0.61948675 0.74432965 0.66733486
##          56          59          78          81          93          99
## 1.00133100 0.73637993 0.22435746 0.35103669 0.16082969 0.36691527
##          115         117         118         120         144         166
## 0.14925198 0.01769895 0.44361034 0.43002855 0.03483946 0.29267127
##          196         199         201         202         206         209
## 0.29397791 0.37043385 0.05304624 0.02745698 0.03148572 0.11921858
##          238         240         246         264         271         275
## 0.14925662 -0.01258159 0.03580649 -0.04409592 0.13258296 0.04489273
##          286
## -0.03245799
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.705 0.295
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.62175 0.3546099          591.8298 0.3758865          39.06383
## 1 65.12995 0.4745763          657.2881 0.4237288          32.91525
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3262411 265023.6          1.204184      137.5816 0.6524823
## 1          0.4576271 265216.8          1.672034      135.5424 0.6101695
##      smoking      time
## 0 0.3475177 160.12057
## 1 0.3559322  69.72881
##
## Coefficients of linear discriminants:
##                                LD1
## age                2.817799e-02
## anaemia            7.576789e-02
## creatinine_phosphokinase 1.180028e-04
## diabetes           2.442658e-01
## ejection_fraction   -3.844957e-02
## high_blood_pressure  1.109418e-01
```

```
## platelets          2.660890e-07
## serum_creatinine   1.778524e-01
## serum_sodium       -5.429158e-02
## sex                -3.988549e-01
## smoking            1.428671e-01
## time               -1.198441e-02
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[4]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##          32          35          36          44          51          53
## 0.844233391 0.379861473 0.878723816 0.590491976 0.738490247 0.677269982
##          56          59          78          81          93          99
## 0.935591893 0.654233553 0.118184441 0.546298397 0.053833455 0.736782040
##          115         117         118         120         144         166
## 0.590573775 0.164962998 0.424283855 0.734256209 0.194089993 0.587050266
##          196         199         201         202         206         209
## 0.033111775 0.276412012 0.027522618 0.007029743 0.042831343 0.087152703
##          238         240         246         264         271         275
## 0.077202412 0.013036527 0.033456033 0.008675991 0.039967092 0.030905842
##          286
## 0.012089160
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[4]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model
```

```
## Call:
```

```
## qda(DEATH_EVENT ~., data = trainset)
```

```
##
```

```
## Prior probabilities of groups:
```

```
##      0      1
```

```
## 0.705 0.295
```

```
##
```

```
## Group means:
```

```
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
```

```
## 0 58.62175 0.3546099          591.8298 0.3758865          39.06383
```

```
## 1 65.12995 0.4745763          657.2881 0.4237288          32.91525
```

```
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
```

```
## 0          0.3262411 265023.6          1.204184      137.5816 0.6524823
```

```
## 1          0.4576271 265216.8          1.672034      135.5424 0.6101695
```

```
##      smoking      time
```

```
## 0 0.3475177 160.12057
```

```
## 1 0.3559322 69.72881
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[4]], type = 'response')
```

```
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
```

```
quaddiscr.valid
```

```
##          32          35          36          44          51          53
## 0.9770593303 0.1241542321 0.9069157091 0.2973903737 0.3407032642 1.0000000000
##          56          59          78          81          93          99
## 0.9932268406 0.4721503974 0.0563957880 0.2303220153 0.0264762002 0.6585661066
##          115         117         118         120         144         166
```

```
## 0.7578909671 0.1860518480 0.9485654521 0.2429117749 0.0564922015 0.0377310838
##          196          199          201          202          206          209
## 0.0705049553 0.2753309570 0.0145693915 0.0003619057 0.0132131450 0.1635650303
##          238          240          246          264          271          275
## 0.0800810477 0.0007239842 0.0026764249 0.0035329703 0.0287400902 0.0027098619
##          286
## 0.0006438637
```

```
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[4]]))

mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 64.49 89.36 95.69 98.92 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.135 ( N = 200 )
##
## Deviance: 139.442
```

```
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[4]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid
```

```
## [1] 0.749137382 0.374151351 0.736399660 0.696145524 0.690493050 0.999999983
## [7] 0.931771781 0.638920238 0.087742696 0.355767113 0.040606497 0.677040832
## [13] 0.742410381 0.191296044 0.979412846 0.627612153 0.176897258 0.606065006
## [19] 0.021056883 0.235147779 0.058130370 0.006585493 0.024169625 0.041854432
## [25] 0.103097332 0.008003140 0.028425862 0.007741464 0.038261929 0.015008615
## [31] 0.011069519
```

```
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[4]]))
```

```
flexdiscr.model <- fda(DEATH_EVENT ~., data = trainset)
flexdiscr.model
```

```
## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.13 ( N = 200 )
```



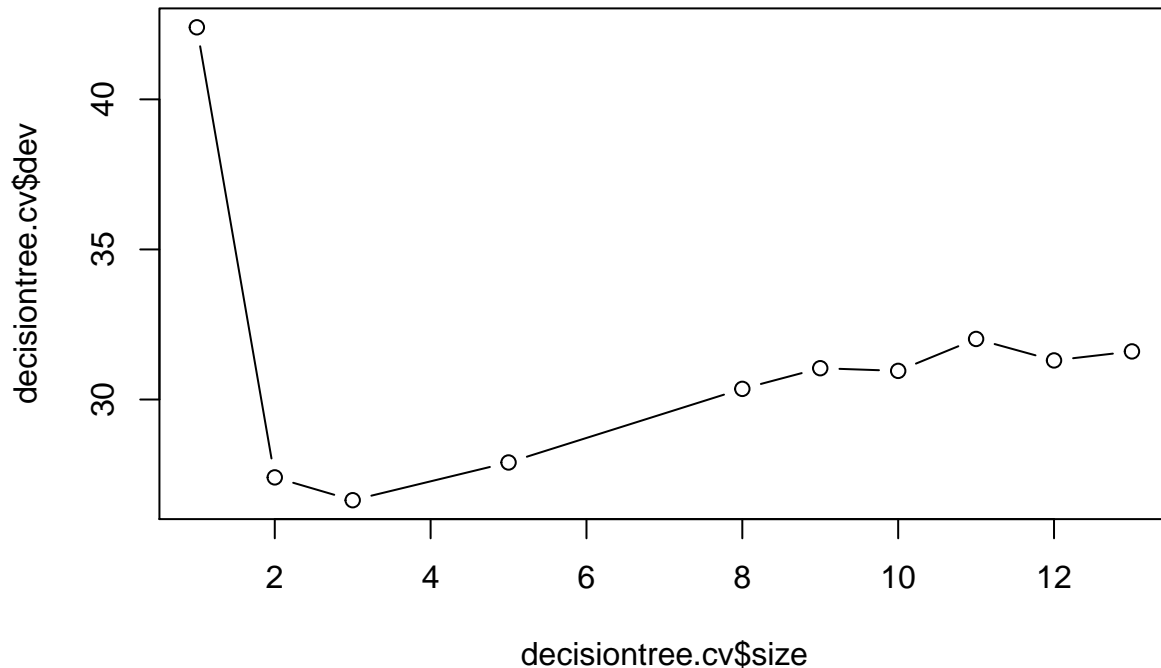
```
## 0.96000000 1.00000000 0.96000000 0.96000000 0.00000000 1.00000000 0.96000000
##      59      78      81      93      99     115     117
## 0.96000000 0.01941748 0.00000000 0.01941748 0.50000000 0.01941748 0.01941748
##     118     120     144     166     196     199     201
## 0.11111111 0.11111111 0.01941748 0.40000000 0.11111111 0.50000000 0.01941748
##     202     206     209     238     240     246     264
## 0.01941748 0.01941748 0.01941748 0.01941748 0.01941748 0.01941748 0.01941748
##     271     275     286
## 1.00000000 0.01941748 0.01941748
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - decisiontree.valid)^2))/n)
```

```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree.cv$dev)])
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)
```

```

graph TD
    Root["time < 73.5"]
    Leaf1["0.80000"]
    Node2["serum_creatinine < 1.45"]
    Leaf2["0.06557"]
    Leaf3["0.39290"]

    Root --> Leaf1
    Root --> Node2
    Node2 --> Leaf2
    Node2 --> Leaf3
  
```

```
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.233e+00                8.370e-03            -2.398e-02
## creatinine_phosphokinase            diabetes            ejection_fraction
##             2.694e-05                5.462e-02            -9.024e-03
##             high_blood_pressure        platelets            serum_creatinine
##             2.701e-02                3.944e-08                5.541e-02
##             serum_sodium                sex                smoking
##             -6.063e-03            -5.786e-02            1.294e-02
##             time
##             -2.687e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76507 -0.27032 -0.03721  0.24571  1.01953
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.233e+00  9.452e-01   1.304 0.193646
## age             8.370e-03  2.253e-03   3.716 0.000265 ***
## anaemia        -2.398e-02  5.418e-02  -0.443 0.658521
## creatinine_phosphokinase 2.694e-05  2.666e-05   1.011 0.313472
## diabetes        5.462e-02  5.334e-02   1.024 0.307119
## ejection_fraction -9.024e-03  2.303e-03  -3.919 0.000123 ***
## high_blood_pressure  2.701e-02  5.535e-02   0.488 0.626122
## platelets        3.944e-08  2.613e-07   0.151 0.880158
## serum_creatinine  5.541e-02  2.728e-02   2.031 0.043650 *
## serum_sodium     -6.063e-03  6.823e-03  -0.889 0.375312
## sex             -5.786e-02  6.395e-02  -0.905 0.366714
## smoking          1.294e-02  6.317e-02   0.205 0.837949
## time            -2.687e-03  3.613e-04  -7.438 3.28e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3656 on 193 degrees of freedom
## Multiple R-squared:  0.41, Adjusted R-squared:  0.3733
## F-statistic: 11.18 on 12 and 193 DF, p-value: < 2.2e-16
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  4.7993   4.7993 35.8966 1.002e-08 ***
## anaemia          1  0.0061   0.0061  0.0455  0.831344
```

```
## creatinine_phosphokinase 1 0.2676 0.2676 2.0016 0.158741
## diabetes 1 0.0961 0.0961 0.7190 0.397530
## ejection_fraction 1 2.6871 2.6871 20.0986 1.260e-05 ***
## high_blood_pressure 1 0.5908 0.5908 4.4191 0.036836 *
## platelets 1 0.0626 0.0626 0.4683 0.494602
## serum_creatinine 1 1.4840 1.4840 11.0994 0.001035 **
## serum_sodium 1 0.2071 0.2071 1.5488 0.214824
## sex 1 0.1860 0.1860 1.3911 0.239678
## smoking 1 0.1469 0.1469 1.0985 0.295905
## time 1 7.3957 7.3957 55.3165 3.277e-12 ***
## Residuals 193 25.8037 0.1337
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[5]])
simplelinear.valid
```

```
##          5          12          14          18          39          63
## 1.06911883 0.69870589 0.46981702 0.68428635 0.76091578 0.40238311
##          82          90          102          107          119          124
## 0.35413414 0.41536614 0.45073169 0.26473326 0.25262733 0.40352637
##          125          161          164          181          189          204
## 0.47508833 0.30987964 0.28234294 0.14612165 0.21796691 0.36835123
##          230          235          244          249          254          255
## 0.32419074 -0.10212863 0.14914807 -0.14503608 0.28464973 -0.26252017
##          296
## -0.09453922
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - simplelinear.valid)^2))/n)
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Coefficients:
## (Intercept)          age ejection_fraction  serum_creatinine
##    0.475054      0.007512      -0.008946      0.058902
##          time
##    -0.002760
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77213 -0.28897 -0.03961  0.26420  1.03592
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.4750539  0.1667885   2.848 0.004853 **
## age            0.0075123  0.0021701   3.462 0.000655 ***
## ejection_fraction -0.0089455  0.0022148  -4.039 7.63e-05 ***
## serum_creatinine  0.0589016  0.0265583   2.218 0.027686 *
## time           -0.0027596  0.0003464  -7.967 1.19e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3625 on 201 degrees of freedom
## Multiple R-squared:  0.3962, Adjusted R-squared:  0.3842
## F-statistic: 32.97 on 4 and 201 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##               Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  4.7993   4.7993  36.532 7.183e-09 ***
## ejection_fraction  1  2.7372   2.7372  20.836 8.705e-06 ***
## serum_creatinine  1  1.4518   1.4518  11.051 0.001053 **
## time            1  8.3391   8.3391  63.478 1.186e-13 ***
## Residuals       201 26.4055   0.1314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[5]])
revisedlinear.valid
```

```
##           5           12           14           18           39           63
## 0.92139737 0.74259076 0.54517234 0.69635500 0.71010935 0.47435116
##           82           90           102           107           119           124
## 0.39087395 0.52639925 0.46534442 0.31940795 0.22022918 0.44827477
##           125          161           164           181           189           204
## 0.52098432 0.34981440 0.24287770 0.15177228 0.15723297 0.39226602
##           230          235           244           249           254           255
## 0.29174783 -0.04130528 0.13084893 -0.06919715 0.16519125 -0.19793610
##           296
## -0.12886533
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##         serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##           5.16485           0.05605          -0.06917           0.45124
##      serum_sodium              time
##        -0.03768          -0.02141
##
```

```
## Degrees of Freedom: 205 Total (i.e. Null); 200 Residual
## Null Deviance: 253.7
## Residual Deviance: 152.7 AIC: 164.7
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1304  -0.6162  -0.2371   0.4296   2.8186
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    5.16485    7.50210   0.688 0.491167
## age             0.05605    0.01816   3.086 0.002031 **
## ejection_fraction -0.06917    0.01912  -3.618 0.000297 ***
## serum_creatinine  0.45124    0.22122   2.040 0.041371 *
## serum_sodium     -0.03768    0.05380  -0.700 0.483698
## time            -0.02142    0.00366  -5.851 4.88e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 253.68  on 205  degrees of freedom
## Residual deviance: 152.74  on 200  degrees of freedom
## AIC: 164.74
##
## Number of Fisher Scoring iterations: 6
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			205	253.68
## age	1	23.124	204	230.55
## ejection_fraction	1	15.309	203	215.24
## serum_creatinine	1	8.059	202	207.19
## serum_sodium	1	0.836	201	206.35
## time	1	53.611	200	152.74

```
generlinear.valid <- predict(generlinear.model, groups[[5]], type = 'response')
generlinear.valid
```

```
##           5           12           14           18           39           63
## 0.983702836 0.861392706 0.607550552 0.880172607 0.843624879 0.450184175
##           82           90           102           107           119           124
## 0.310864718 0.502140688 0.440921054 0.209884338 0.097551817 0.345311645
##           125           161           164           181           189           204
## 0.580580488 0.268542161 0.109404568 0.071991736 0.071844164 0.320456921
##           230           235           244           249           254           255
## 0.186958658 0.015997986 0.048382496 0.011022970 0.085517786 0.004470913
##           296
## 0.007533362
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - generlinear.valid)^2))/nrow
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 2.03 1.00 2.70 3.37 6.18 total = 16.27
##
## REML score: 79.01957
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.30583    0.02174   14.06  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df      F  p-value
## s(age)          2.030  2.551  3.758  0.0146 *
## s(creatinine_phosphokinase) 1.000  1.000  3.355  0.0685 .
## s(ejection_fraction)      2.696  3.340  7.132 9.79e-05 ***
## s(serum_creatinine)       3.365  4.141  3.091  0.0146 *
## s(time)              6.177  7.362 14.390 3.35e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```



```
## R-sq.(adj) = 0.543   Deviance explained = 57.7%
## -REML = 79.02   Scale est. = 0.097394   n = 206
```

```
generaddit.valid <- predict(generaddit.model, groups[[5]], type = 'response')
generaddit.valid
```

```
##           5           12           14           18           39           63
## 1.157648344 0.822610853 0.622720014 1.012570468 1.057877870 0.435242510
##           82           90           102           107           119           124
## 0.258848423 0.373785598 0.214367210 0.130385227 -0.002628561 0.123743837
##           125           161           164           181           189           204
## 0.146323766 0.118487285 0.147491834 0.197930405 0.138418661 0.419977808
##           230           235           244           249           254           255
## 0.294712615 -0.034982135 0.031475548 -0.071581401 0.059147135 -0.091995665
##           296
## 0.038640031
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model
```

```
## Call:
```

```
## lda(DEATH_EVENT ~., data = trainset)
```

```
##
```

```
## Prior probabilities of groups:
```

```
##           0           1
```

```
## 0.6941748 0.3058252
```

```
##
```

```
## Group means:
```

```
##           age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
```

```
## 0 58.47552 0.3916084                    566.9930 0.4335664           39.67133
```

```
## 1 67.37567 0.4444444                    680.9365 0.4285714           34.25397
```

```
##  high_blood_pressure platelets serum_creatinine serum_sodium           sex
```

```
## 0           0.3076923 268498.8           1.197133           137.3007 0.6363636
```

```
## 1           0.4603175 269801.7           1.775873           136.0794 0.6190476
```

```
##           smoking           time
```

```
## 0 0.3286713 161.37063
```

```
## 1 0.3333333 71.93651
```

```
##
```

```
## Coefficients of linear discriminants:
```

```
##                               LD1
```

```
## age                        3.675715e-02
```

```
## anaemia                    -1.053138e-01
```

```
## creatinine_phosphokinase  1.183206e-04
```

```
## diabetes                   2.398529e-01
```

```
## ejection_fraction         -3.962952e-02
```

```
## high_blood_pressure       1.186146e-01
```

```
## platelets                  1.732025e-07
```

```
## serum_creatinine          2.433131e-01
```

```
## serum_sodium              -2.662513e-02
```

```
## sex                        -2.540911e-01
```

```
## smoking                   5.681141e-02
```

```
## time                      -1.180140e-02
```

```

lindiscr.valid <- predict(lindiscr.model, groups[[5]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid

```

```

##          5          12          14          18          39          63
## 0.989903451 0.839833282 0.461931104 0.823898891 0.895557483 0.334991972
##          82          90          102          107          119          124
## 0.255947116 0.358230904 0.424712758 0.145054237 0.133583623 0.337008479
##          125          161          164          181          189          204
## 0.472304196 0.195130709 0.163186021 0.062289125 0.104925383 0.277931926
##          230          235          244          249          254          255
## 0.213514597 0.009245938 0.063701366 0.006603688 0.165691732 0.002619074
##          296
## 0.009812072

```

```

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[5]]))

```

```

quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model

```

```

## Call:
## qda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6941748 0.3058252
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.47552 0.3916084                566.9930 0.4335664                39.67133
## 1 67.37567 0.4444444                680.9365 0.4285714                34.25397
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3076923 268498.8                1.197133      137.3007 0.6363636
## 1      0.4603175 269801.7                1.775873      136.0794 0.6190476
##      smoking      time
## 0 0.3286713 161.37063
## 1 0.3333333  71.93651

```

```

quaddiscr.valid <- predict(quaddiscr.model, groups[[5]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid

```

```

##          5          12          14          18          39          63
## 5.081189e-02 1.925539e-01 3.020587e-02 8.410433e-01 9.759754e-01 4.465136e-02
##          82          90          102          107          119          124
## 7.389501e-02 1.026504e-01 1.006090e-01 4.203916e-02 2.081879e-01 2.542576e-01
##          125          161          164          181          189          204
## 9.382817e-01 7.528526e-02 4.018621e-01 3.081522e-02 3.259041e-02 8.983683e-01
##          230          235          244          249          254          255
## 5.726776e-02 7.573175e-04 9.197959e-03 6.950176e-04 5.877907e-03 1.799478e-05
##          296
## 3.639550e-03

```

```

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[5]]))

```

```

mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)

```

```
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 66.22  91.00  97.63  99.38 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.15534 ( N = 206 )
##
## Deviance: 157.95
```

```
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[5]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid
```

```
## [1] 0.969752973 0.826250129 0.478551083 0.894174636 0.907718603 0.289885328
## [7] 0.240448539 0.402026720 0.429542368 0.146186099 0.200148738 0.547929628
## [13] 0.394540798 0.215508101 0.263682363 0.070344466 0.152364993 0.196152904
## [19] 0.206983139 0.008152552 0.078271372 0.004138305 0.153950581 0.002118046
## [25] 0.013495088
```

```
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[5]]$DEATH_EVENT))
```

```
flexdiscr.model <- fda(DEATH_EVENT ~ ., data = trainset)
flexdiscr.model
```

```
## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.14078 ( N = 206 )
```

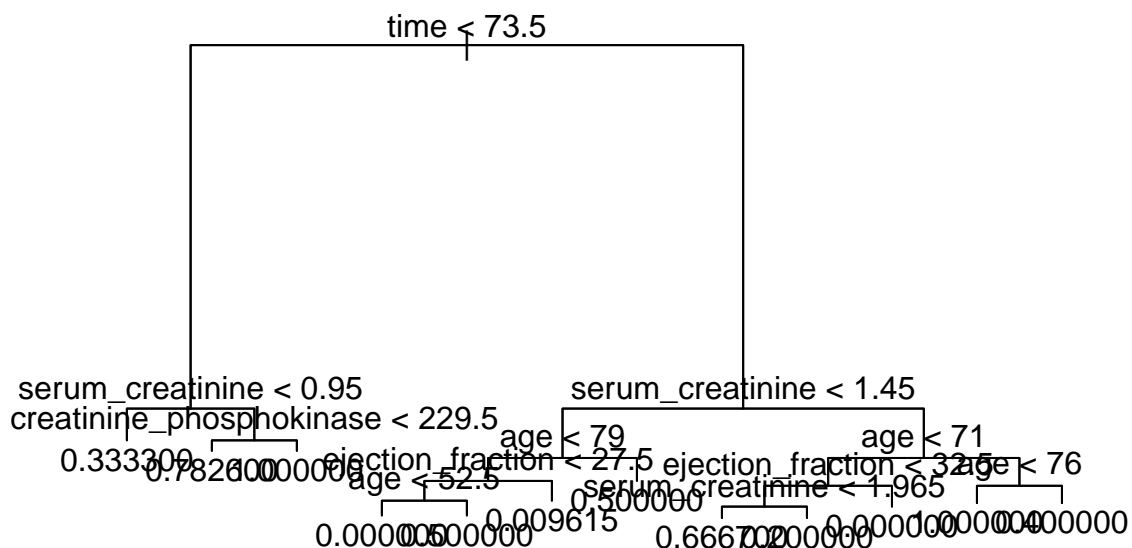
```
flexdiscr.valid <- predict(flexdiscr.model, groups[[5]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid
```

```
##           5           12           14           18           39           63
## 0.990419560 0.843072559 0.463557169 0.827234529 0.898248455 0.335284735
##           82           90          102          107          119          124
## 0.255485425 0.358764494 0.425949866 0.143897926 0.132396760 0.337321918
##          125          161          164          181          189          204
## 0.474036479 0.194212656 0.162097842 0.061214479 0.103712600 0.277666325
##          230          235          244          249          254          255
## 0.212718593 0.008906108 0.062617749 0.006339366 0.164614716 0.002491108
```

```
##          296
## 0.009457141

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups))

decisiontree.model <- tree(DEATH_EVENT ~., data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[5]], type = 'vector')
decisiontree.valid
```

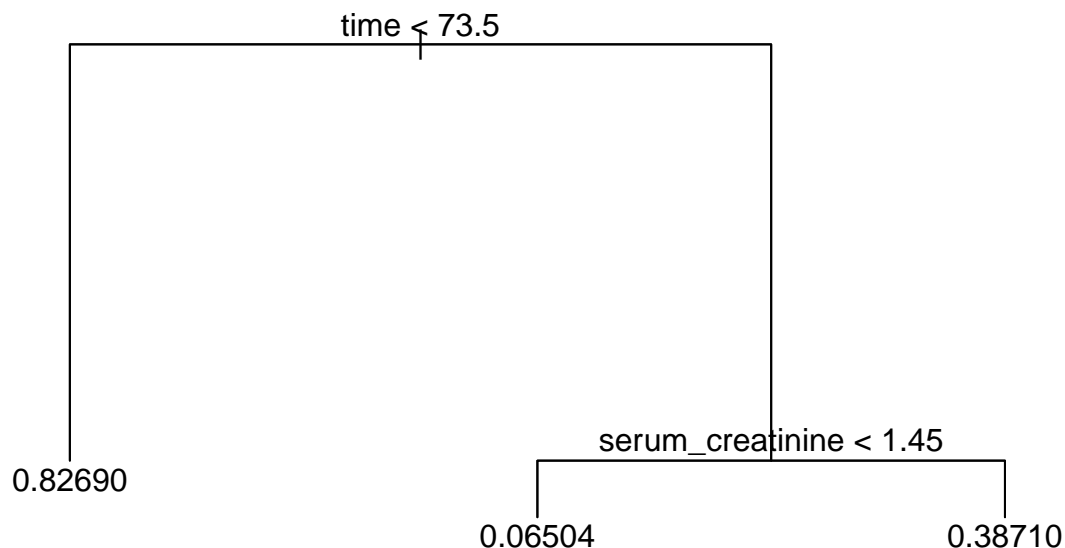
```
##          5          12          14          18          39          63
## 0.782608696 0.333333333 0.782608696 0.333333333 1.000000000 0.782608696
##          82          90          102          107          119          124
## 0.009615385 0.500000000 0.009615385 0.009615385 0.009615385 0.009615385
##          125          161          164          181          189          204
## 0.000000000 0.009615385 0.009615385 0.009615385 0.009615385 0.200000000
##          230          235          244          249          254          255
## 0.500000000 0.009615385 0.009615385 0.009615385 0.009615385 0.009615385
##          296
## 0.009615385
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - decisiontree.valid)^2))/nrow(groups))

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[5]], type = 'vector')
prunedectree.valid
```

```
##          5          12          14          18          39          63          82
## 0.82692308 0.82692308 0.82692308 0.82692308 0.82692308 0.82692308 0.06504065
##          90          102          107          119          124          125          161
## 0.06504065 0.06504065 0.06504065 0.06504065 0.06504065 0.38709677 0.06504065
##          164          181          189          204          230          235          244
## 0.06504065 0.06504065 0.06504065 0.38709677 0.06504065 0.06504065 0.06504065
##          249          254          255          296
## 0.06504065 0.06504065 0.06504065 0.06504065
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##6th fold-out training
```

```
trainset <- data.frame()
```

```
for (i in c(1:5,7:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ ., data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ ., data = trainset)
```

```
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.432e+00                7.336e-03            -4.244e-02
## creatinine_phosphokinase                diabetes            ejection_fraction
##             -3.592e-06                4.151e-02            -7.061e-03
##             high_blood_pressure                platelets            serum_creatinine
##             1.655e-02                8.266e-08                6.786e-02
##             serum_sodium                sex                smoking
##             -7.722e-03            -2.758e-02            -2.517e-02
##             time
##             -2.667e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71652 -0.26579 -0.02805  0.22184  0.99662
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.432e+00  9.158e-01   1.564  0.11960
## age             7.336e-03  2.361e-03   3.107  0.00219 **
## anaemia        -4.244e-02  5.498e-02  -0.772  0.44113
## creatinine_phosphokinase -3.592e-06  3.470e-05  -0.104  0.91766
## diabetes        4.151e-02  5.439e-02   0.763  0.44634
## ejection_fraction -7.061e-03  2.370e-03  -2.979  0.00327 **
## high_blood_pressure  1.655e-02  5.610e-02   0.295  0.76826
## platelets        8.266e-08  2.772e-07   0.298  0.76587
## serum_creatinine  6.786e-02  3.306e-02   2.053  0.04149 *
## serum_sodium    -7.722e-03  6.672e-03  -1.158  0.24854
## sex            -2.758e-02  6.413e-02  -0.430  0.66769
## smoking        -2.517e-02  6.498e-02  -0.387  0.69892
## time           -2.667e-03  3.728e-04  -7.155 1.85e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3677 on 187 degrees of freedom
## Multiple R-squared:  0.3861, Adjusted R-squared:  0.3467
## F-statistic:  9.8 on 12 and 187 DF,  p-value: 1.004e-14
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.6702   3.6702  27.1484 4.951e-07 ***
## anaemia       1  0.0527   0.0527   0.3896 0.5332517
## creatinine_phosphokinase 1  0.0060   0.0060   0.0447 0.8327318
## diabetes      1  0.1187   0.1187   0.8779 0.3499925
```

```
## ejection_fraction      1  2.4383  2.4383 18.0362 3.413e-05 ***
## high_blood_pressure    1  0.1977  0.1977  1.4621 0.2281163
## platelets              1  0.0157  0.0157  0.1162 0.7335863
## serum_creatinine       1  1.8851  1.8851 13.9441 0.0002499 ***
## serum_sodium           1  0.5429  0.5429  4.0161 0.0465116 *
## sex                    1  0.0330  0.0330  0.2444 0.6216549
## smoking                 1  0.0184  0.0184  0.1360 0.7126692
## time                    1  6.9204  6.9204 51.1894 1.846e-11 ***
## Residuals              187 25.2808  0.1352
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[6]])
simplelinear.valid
```

```
##          10          16          24          27          38          50
## 1.30874275 0.61618988 0.39424441 0.77443721 0.58756899 0.50266588
##          61          67          69          73          98         101
## 0.45496583 0.51447584 0.59730439 0.61100111 0.29787271 0.43948035
##          112         123         136         150         182         198
## 0.35299266 0.38013227 0.42848604 0.32873796 0.26970210 0.21024483
##          216         219         226         233         241         248
## 0.25729912 0.18826694 0.20238083 -0.12550792 0.24950229 0.26671312
##          267         278         279         280         282         292
## 0.10549898 0.02843136 -0.05749993 0.02742572 0.07856650 -0.05931440
##          298
## -0.28904563
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - simplelinear.valid)^2))/n)
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##          0.408373           0.006772           -0.007258           0.078171
##              time
##          -0.002711
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.72534 -0.28252 -0.03973  0.22045  1.03537
##
## Coefficients:
```



```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.408373   0.170849   2.390  0.01779 *
## age            0.006772   0.002278   2.972  0.00333 **
## ejection_fraction -0.007258  0.002254  -3.220  0.00150 **
## serum_creatinine  0.078171  0.031372   2.492  0.01355 *
## time           -0.002711  0.000359  -7.553 1.61e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3638 on 195 degrees of freedom
## Multiple R-squared:  0.3734, Adjusted R-squared:  0.3605
## F-statistic: 29.05 on 4 and 195 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##               Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  3.6702   3.6702  27.736 3.657e-07 ***
## ejection_fraction  1  2.3769   2.3769  17.962 3.474e-05 ***
## serum_creatinine  1  1.7806   1.7806  13.456  0.000315 ***
## time            1  7.5487   7.5487  57.046 1.606e-12 ***
## Residuals       195 25.8036   0.1323
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[6]])
revisedlinear.valid
```

```
##           10           16           24           27           38           50
## 1.40379175 0.66714623 0.33468611 0.77449103 0.59760353 0.54093131
##           61           67           69           73           98          101
## 0.44715473 0.50931597 0.61582362 0.61292065 0.30245314 0.51720950
##           112          123          136          150          182          198
## 0.37658914 0.33994171 0.42808476 0.31921678 0.29795011 0.18905165
##           216          219          226          233          241          248
## 0.22162098 0.16670385 0.03441288 -0.07108412 0.15521317 0.26772841
##           267          278          279          280          282          292
## 0.12530947 0.02562120 -0.08302280 -0.06032610 0.12533257 -0.02940895
##           298
## -0.21241303
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##         serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##           7.49433           0.05168           -0.05323            0.42217
##      serum_sodium              time
```

```
##           -0.05792           -0.02111
##
## Degrees of Freedom: 199 Total (i.e. Null);  194 Residual
## Null Deviance:      240.9
## Residual Deviance: 149.6      AIC: 161.6
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9759  -0.5948  -0.2385   0.4212   2.7373
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    7.494333   7.346679   1.020  0.30768
## age             0.051682   0.019035   2.715  0.00663 **
## ejection_fraction -0.053234   0.018876  -2.820  0.00480 **
## serum_creatinine  0.422173   0.211470   1.996  0.04589 *
## serum_sodium    -0.057918   0.052697  -1.099  0.27173
## time            -0.021114   0.003752  -5.628 1.82e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 240.86  on 199  degrees of freedom
## Residual deviance: 149.56  on 194  degrees of freedom
## AIC: 161.56
##
## Number of Fisher Scoring iterations: 6
anova(generlinear.model)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			199	240.86
## age	1	18.052	198	222.81
## ejection_fraction	1	13.112	197	209.70
## serum_creatinine	1	8.915	196	200.78
## serum_sodium	1	2.863	195	197.92
## time	1	48.362	194	149.56

```
generlinear.valid <- predict(generlinear.model, groups[[6]], type = 'response')
generlinear.valid
```

```
##          10          16          24          27          38          50
## 0.997040747 0.812755688 0.287762073 0.900022469 0.613291065 0.567100460
##          61          67          69          73          98         101
## 0.399845557 0.541410412 0.661486678 0.782598467 0.216000305 0.495957501
##          112         123         136         150         182         198
## 0.323236092 0.226992738 0.388245567 0.232732267 0.196382921 0.089032613
##          216         219         226         233         241         248
## 0.127250915 0.090252408 0.057290235 0.013799963 0.061280401 0.135524203
##          267         278         279         280         282         292
## 0.057069073 0.023000272 0.013467237 0.014656648 0.045967864 0.015144019
##          298
## 0.003567787
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - generlinear.valid)^2))/nrow
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 1.76 1.94 2.06 2.36 4.63 total = 13.76
##
## REML score: 83.27221
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2900     0.0233   12.45  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)         1.762   2.211   3.026   0.0423 *
## s(creatinine_phosphokinase) 1.945   2.411   1.261   0.3272
## s(ejection_fraction)      2.059   2.550   3.956   0.0162 *
```

```
## s(serum_creatinine)      2.363  2.953  3.288  0.0247 *
## s(time)                  4.629  5.690 14.811 9.45e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.475   Deviance explained = 50.9%
## -REML = 83.272   Scale est. = 0.10854   n = 200
```

```
generaddit.valid <- predict(generaddit.model, groups[[6]], type = 'response')
generaddit.valid
```

```
##          10          16          24          27          38          50
## 1.203711781 0.935393952 0.573027396 0.929527889 0.804952306 0.572133767
##          61          67          69          73          98         101
## 0.129948760 0.575939477 0.507915467 0.284144665 0.166483433 0.314706994
##          112         123         136         150         182         198
## 0.144550711 0.045070443 0.242854299 0.179326048 0.260716262 0.136368517
##          216         219         226         233         241         248
## 0.244002664 0.191817040 0.197467389 -0.060637077 0.110845573 0.320403416
##          267         278         279         280         282         292
## 0.351357020 0.057927217 0.003846427 -0.024461728 0.228476721 0.045944841
##          298
## 0.036890104
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.71 0.29
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.13850 0.4154930          570.6127 0.4154930          39.71127
## 1 65.94253 0.4137931          517.0345 0.4310345          34.13793
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3591549 268661.5          1.200845      137.5141 0.6338028
## 1          0.4655172 265347.4          1.740517      135.4828 0.6379310
##      smoking      time
## 0 0.3450704 157.12676
## 1 0.3103448  69.37931
##
## Coefficients of linear discriminants:
##                      LD1
## age                3.304087e-02
## anaemia            -1.911613e-01
## creatinine_phosphokinase -1.617822e-05
## diabetes           1.869398e-01
## ejection_fraction   -3.180299e-02
## high_blood_pressure   7.455656e-02
## platelets           3.723149e-07
```

```
## serum_creatinine      3.056285e-01
## serum_sodium          -3.478131e-02
## sex                   -1.242071e-01
## smoking               -1.133746e-01
## time                  -1.201196e-02
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[6]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##          10          16          24          27          38          50
## 0.998419495 0.735815635 0.328726224 0.905828347 0.690011393 0.533750761
##          61          67          69          73          98         101
## 0.440685688 0.556681467 0.706079927 0.727840516 0.187134483 0.411041273
##          112         123         136         150         182         198
## 0.261722138 0.304813808 0.390366901 0.226706843 0.155858537 0.103860956
##          216         219         226         233         241         248
## 0.143500914 0.088897275 0.098266691 0.008287630 0.136157655 0.152803342
##          267         278         279         280         282         292
## 0.048548959 0.027145807 0.014035530 0.026938579 0.039682579 0.013840223
##          298
## 0.002316215
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[6]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~ ., data = trainset)
quaddiscr.model
```

```
## Call:
```

```
## qda(DEATH_EVENT ~ ., data = trainset)
```

```
##
```

```
## Prior probabilities of groups:
```

```
##      0      1
```

```
## 0.71 0.29
```

```
##
```

```
## Group means:
```

```
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
```

```
## 0 58.13850 0.4154930          570.6127 0.4154930          39.71127
```

```
## 1 65.94253 0.4137931          517.0345 0.4310345          34.13793
```

```
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
```

```
## 0          0.3591549 268661.5          1.200845      137.5141 0.6338028
```

```
## 1          0.4655172 265347.4          1.740517      135.4828 0.6379310
```

```
##      smoking      time
```

```
## 0 0.3450704 157.12676
```

```
## 1 0.3103448  69.37931
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[6]], type = 'response')
```

```
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
```

```
quaddiscr.valid
```

```
##          10          16          24          27          38          50
## 1.000000e+00 7.192734e-01 3.338153e-01 4.967281e-01 7.423086e-01 1.801468e-01
##          61          67          69          73          98         101
## 4.520913e-27 4.153017e-01 1.165403e-01 3.011345e-20 1.291293e-01 7.860170e-01
##          112         123         136         150         182         198
## 1.339735e-01 2.298780e-01 7.223163e-01 3.647294e-03 3.511222e-02 2.793684e-02
```

```
##           216           219           226           233           241           248
## 9.937631e-02 1.558724e-02 4.952711e-01 3.728859e-03 6.230939e-03 1.030656e-01
##           267           278           279           280           282           292
## 8.576676e-03 1.991990e-02 8.430954e-03 5.732876e-04 3.560958e-03 1.204775e-02
##           298
## 3.826890e-04
```

```
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[6]]$DEATH_EVENT))
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 71.54 95.11 98.59 99.78 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.16 ( N = 200 )
##
## Deviance: 143.683
```

```
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[6]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid
```

```
## [1] 0.999616798 0.797481281 0.285107363 0.903935802 0.699135667 0.409965197
## [7] 0.643942164 0.533021941 0.678207783 0.609454594 0.137245936 0.293543855
## [13] 0.259991353 0.289474985 0.331611057 0.231239938 0.131207492 0.078888033
## [19] 0.168454491 0.062080342 0.096739016 0.007913320 0.805522571 0.134199206
## [25] 0.042130064 0.055862853 0.012653327 0.045280736 0.050553517 0.036265826
## [31] 0.002692321
```

```
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[6]]$DEATH_EVENT))
```

```
flexdiscr.model <- fda(DEATH_EVENT ~., data = trainset)
flexdiscr.model
```

```
## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.15 ( N = 200 )
```



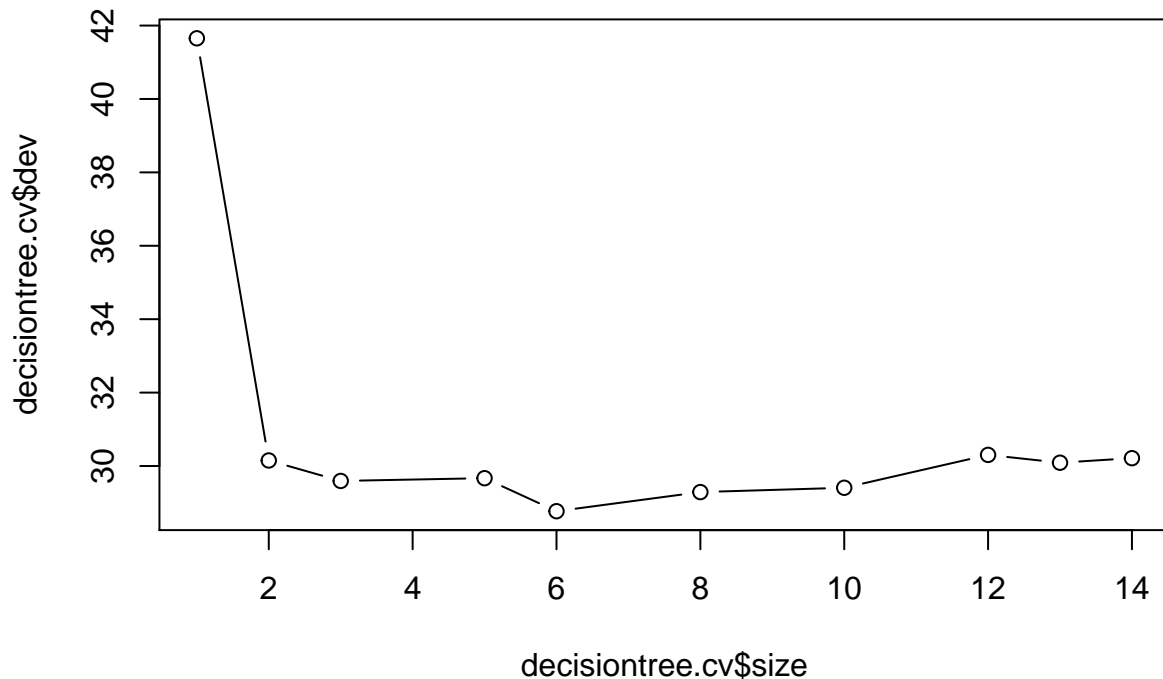
```
## 0.95833333 0.95833333 0.95833333 0.80000000 1.00000000 1.00000000 0.80000000
##      67      69      73      98     101     112     123
## 0.95833333 1.00000000 0.50000000 0.01886792 0.80000000 0.01886792 0.01886792
##     136     150     182     198     216     219     226
## 0.01886792 0.01886792 0.20000000 0.01886792 0.01886792 0.01886792 0.01886792
##     233     241     248     267     278     279     280
## 0.01886792 0.01886792 0.00000000 0.00000000 0.01886792 0.01886792 0.01886792
##     282     292     298
## 0.00000000 0.01886792 0.01886792
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - decisiontree.valid)^2))/n)
```

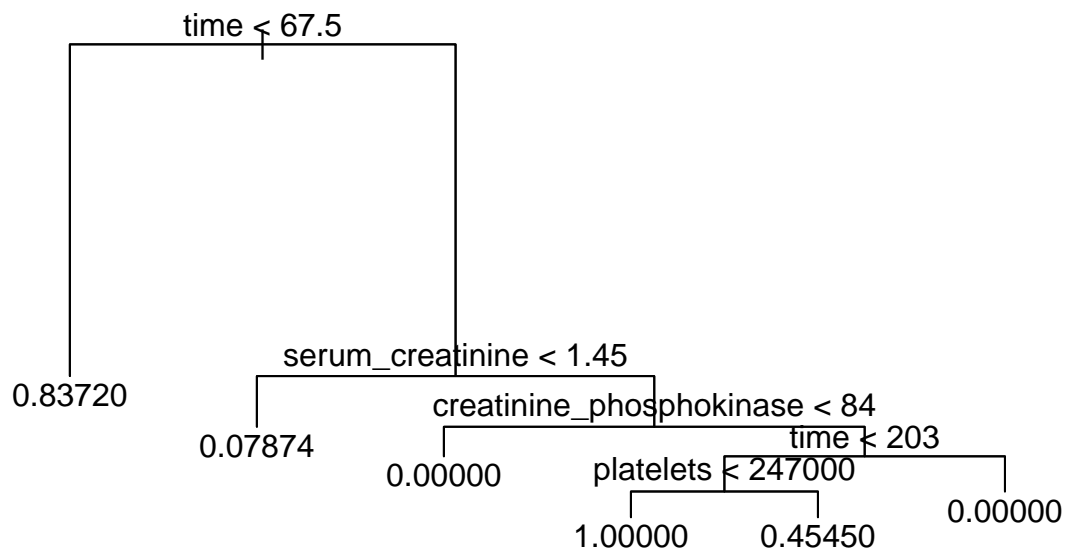
```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree.cv$dev)])
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)
```

```
prunedectree.valid <- predict(prunedectree.model, groups[[6]], type = 'vector')
prunedectree.valid
```

```
##      10      16      24      27      38      50      61
## 0.83720930 0.83720930 0.83720930 0.83720930 0.83720930 0.83720930 0.83720930
##      67      69      73      98     101     112     123
## 0.83720930 0.83720930 0.07874016 0.07874016 0.07874016 0.07874016 0.07874016
##      136     150     182     198     216     219     226
## 0.07874016 0.07874016 0.07874016 0.07874016 0.07874016 0.07874016 0.07874016
##      233     241     248     267     278     279     280
## 0.07874016 0.07874016 0.00000000 0.00000000 0.07874016 0.07874016 0.07874016
##      282     292     298
## 0.00000000 0.07874016 0.07874016
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##7th fold-out training
trainset <- data.frame()
for (i in c(1:6, 8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ ., data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.178e+00                6.719e-03                1.677e-04
## creatinine_phosphokinase                diabetes                ejection_fraction
##             3.325e-05                4.761e-02                -8.601e-03
##             high_blood_pressure                platelets                serum_creatinine
##             2.996e-02                5.315e-08                5.182e-02
##             serum_sodium                sex                smoking
##             -5.252e-03                -3.478e-02                1.150e-02
##             time
##             -2.862e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75453 -0.27646 -0.02411  0.26190  0.89843
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.178e+00  9.097e-01   1.295 0.196818
## age            6.719e-03  2.317e-03   2.900 0.004171 **
## anaemia        1.677e-04  5.376e-02   0.003 0.997514
## creatinine_phosphokinase 3.325e-05  2.665e-05   1.248 0.213730
## diabetes       4.761e-02  5.428e-02   0.877 0.381546
## ejection_fraction -8.601e-03  2.379e-03  -3.616 0.000384 ***
## high_blood_pressure  2.996e-02  5.592e-02   0.536 0.592777
## platelets       5.315e-08  2.638e-07   0.201 0.840572
## serum_creatinine  5.182e-02  2.657e-02   1.950 0.052615 .
## serum_sodium    -5.252e-03  6.599e-03  -0.796 0.427105
## sex            -3.478e-02  6.257e-02  -0.556 0.579025
## smoking         1.150e-02  6.261e-02   0.184 0.854510
## time          -2.862e-03  3.514e-04  -8.144 5.07e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3622 on 189 degrees of freedom
## Multiple R-squared:  0.4121, Adjusted R-squared:  0.3748
## F-statistic: 11.04 on 12 and 189 DF,  p-value: < 2.2e-16
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  2.7521   2.7521  20.9777 8.416e-06 ***
## anaemia       1  0.0179   0.0179   0.1366 0.7120680
```

```
## creatinine_phosphokinase 1 0.3461 0.3461 2.6381 0.1059973
## diabetes 1 0.0210 0.0210 0.1599 0.6896741
## ejection_fraction 1 2.7714 2.7714 21.1242 7.856e-06 ***
## high_blood_pressure 1 0.6551 0.6551 4.9937 0.0266104 *
## platelets 1 0.1034 0.1034 0.7879 0.3758495
## serum_creatinine 1 1.5455 1.5455 11.7803 0.0007353 ***
## serum_sodium 1 0.3712 0.3712 2.8296 0.0941931 .
## sex 1 0.0614 0.0614 0.4677 0.4948728
## smoking 1 0.0361 0.0361 0.2755 0.6002671
## time 1 8.7015 8.7015 66.3260 5.067e-14 ***
## Residuals 189 24.7955 0.1312
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[7]])
simplelinear.valid
```

```
##          6          7          26          71          80          84
## 0.85082771 0.84751288 0.70244195 0.22278868 0.31868993 0.45686460
##          89          111          128          167          175          184
## 0.26808820 0.34194090 0.14657676 -0.03247997 0.29905088 0.32789728
##          186          187          188          190          192          207
## 0.23907390 -0.05325395 0.35673178 -0.07392592 -0.01836777 -0.11570779
##          210          212          214          222          224          228
## 0.15413065 -0.22848722 0.18562972 -0.10099411 0.03298593 0.06953620
##          236          245          251          269          276
## 0.02548997 0.04041716 0.07493392 -0.15012326 -0.15368212
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - simplelinear.valid)^2)/n))
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Coefficients:
## (Intercept)          age ejection_fraction  serum_creatinine
##      0.542098      0.006288      -0.008651      0.054647
##          time
##     -0.002949
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7584 -0.3025 -0.0227  0.2764  0.8992
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5420980  0.1649269   3.287 0.001199 **
## age            0.0062877  0.0022352   2.813 0.005405 **
## ejection_fraction -0.0086509  0.0022505  -3.844 0.000163 ***
## serum_creatinine  0.0546468  0.0256455   2.131 0.034341 *
## time           -0.0029486  0.0003352  -8.796 7.13e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3583 on 197 degrees of freedom
## Multiple R-squared:  0.4005, Adjusted R-squared:  0.3883
## F-statistic: 32.9 on 4 and 197 DF, p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##               Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  2.7521   2.7521   21.442 6.614e-06 ***
## ejection_fraction  1  2.7375   2.7375   21.328 6.978e-06 ***
## serum_creatinine  1  1.4726   1.4726   11.473 0.0008526 ***
## time            1  9.9301   9.9301   77.365 7.132e-16 ***
## Residuals       197 25.2858   0.1284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[7]])
revisedlinear.valid
```

```
##           6           7           26           71           80
## 0.8531211798 0.9199997976 0.7523875764 0.2970668761 0.3296126586
##           84           89           111           128           167
## 0.4746498934 0.2780303775 0.3576951412 0.1247884591 -0.0005732683
##           175           184           186           187           188
## 0.2666978670 0.3438787937 0.2417805257 -0.0504393699 0.3106807375
##           190           192           207           210           212
## -0.0595647020 -0.0056362742 -0.0601068464 0.0397368616 -0.2022935547
##           214           222           224           228           236
## 0.1027295288 -0.0113622498 0.0723900508 0.0323684561 0.0375522592
##           245           251           269           276
## 0.0232050994 -0.0067276810 -0.1739734036 -0.1823867071
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call: glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##          serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##          (Intercept)              age ejection_fraction  serum_creatinine
##             5.00517             0.05389             -0.07217             0.43307
##          serum_sodium              time
```

```
##           -0.03296           -0.02637
##
## Degrees of Freedom: 201 Total (i.e. Null);  196 Residual
## Null Deviance:      245.8
## Residual Deviance: 140.6      AIC: 152.6
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2010  -0.5650  -0.1700   0.3984   2.6753
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    5.005168   7.684287   0.651 0.514820
## age             0.053889   0.019793   2.723 0.006477 **
## ejection_fraction -0.072166  0.020310  -3.553 0.000381 ***
## serum_creatinine  0.433073  0.216307   2.002 0.045272 *
## serum_sodium    -0.032960  0.055226  -0.597 0.550622
## time            -0.026369  0.004389  -6.008 1.88e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 245.76  on 201  degrees of freedom
## Residual deviance: 140.60  on 196  degrees of freedom
## AIC: 152.6
##
## Number of Fisher Scoring iterations: 6
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			201	245.76
## age	1	13.319	200	232.44
## ejection_fraction	1	14.998	199	217.45
## serum_creatinine	1	7.617	198	209.83
## serum_sodium	1	1.015	197	208.81
## time	1	68.212	196	140.60

```
generlinear.valid <- predict(generlinear.model, groups[[7]], type = 'response')
generlinear.valid
```

```
##           6           7           26           71           80           84
## 0.964981621 0.975984305 0.885339787 0.150184645 0.189317714 0.498265429
##           89           111           128           167           175           184
## 0.133322236 0.279170743 0.042664919 0.016581156 0.119873865 0.213938189
##           186           187           188           190           192           207
## 0.096849710 0.009941119 0.125861792 0.007393570 0.013867630 0.006978169
##           210           212           214           222           224           228
## 0.022986921 0.002225359 0.037169327 0.009219902 0.025662920 0.015213339
##           236           245           251           269           276
## 0.017913845 0.016609971 0.011279785 0.002545240 0.002620472
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - generlinear.valid)^2))/nrow
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 1.92 1.00 2.82 2.06 4.95 total = 13.75
##
## REML score: 76.88339
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29703    0.02229   13.32  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)         1.923   2.422  4.219 0.010797 *
## s(creatinine_phosphokinase) 1.000   1.000  3.331 0.069546 .
## s(ejection_fraction)      2.816   3.493  6.645 0.000148 ***
## s(serum_creatinine)       2.060   2.559  1.905 0.182389
## s(time)          4.953   6.073 18.484 < 2e-16 ***
```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.522   Deviance explained = 55.2%
## -REML = 76.883   Scale est. = 0.10037   n = 202

generaddit.valid <- predict(generaddit.model, groups[[7]], type = 'response')
generaddit.valid

##           6           7           26           71           80           84
## 1.11595645 1.32210844 0.92301999 0.21782672 0.20272280 0.39345965
##           89           111           128           167           175           184
## 0.10126527 0.28694199 -0.01807916 -0.02554598 0.10931179 0.27458523
##           186           187           188           190           192           207
## 0.15367516 -0.05012244 0.37858572 -0.04793121 0.02774131 -0.08785381
##           210           212           214           222           224           228
## -0.01151360 -0.11211069 0.07942159 -0.01840754 0.11383693 0.07602746
##           236           245           251           269           276
## 0.08098302 0.03703076 0.05334998 -0.07065227 -0.08136017

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr

lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7029703 0.2970297
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 59.31456 0.4084507                572.0704 0.4295775                38.94366
## 1 65.88333 0.4500000                726.2333 0.4166667                33.40000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0           0.3028169 265411.2           1.239085      137.3592 0.6478873
## 1           0.4666667 269175.1           1.788000      135.7667 0.6166667
##      smoking      time
## 0 0.3309859 158.64789
## 1 0.3166667  63.26667
##
## Coefficients of linear discriminants:
##                                LD1
## age                2.972356e-02
## anaemia             7.420638e-04
## creatinine_phosphokinase 1.471113e-04
## diabetes            2.106103e-01
## ejection_fraction    -3.804963e-02
## high_blood_pressure   1.325305e-01
## platelets            2.351206e-07
## serum_creatinine      2.292617e-01
## serum_sodium          -2.323578e-02
## sex                  -1.538453e-01
## smoking              5.085763e-02

```

```

## time                                -1.266018e-02

lindiscr.valid <- predict(lindiscr.model, groups[[7]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid

##           6           7           26           71           80           84
## 0.949344799 0.948043464 0.849904561 0.105739008 0.203998946 0.438569251
##           89          111          128          167          175          184
## 0.145587297 0.236142888 0.060100879 0.014861371 0.179477131 0.216323840
##          186          187          188          190          192          207
## 0.118818404 0.012597495 0.258334966 0.010683439 0.016623277 0.007650294
##          210          212          214          222          224          228
## 0.063636386 0.003094564 0.080561118 0.008606010 0.024941361 0.033209169
##          236          245          251          269          276
## 0.023512440 0.026441327 0.034635782 0.005806660 0.005643272

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[7]]))

quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7029703 0.2970297
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 59.31456 0.4084507                    572.0704 0.4295775          38.94366
## 1 65.88333 0.4500000                    726.2333 0.4166667          33.40000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3028169 265411.2          1.239085      137.3592 0.6478873
## 1      0.4666667 269175.1          1.788000      135.7667 0.6166667
##      smoking      time
## 0 0.3309859 158.64789
## 1 0.3166667  63.26667

quaddiscr.valid <- predict(quaddiscr.model, groups[[7]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid

##           6           7           26           71           80           84
## 9.609574e-01 4.832920e-01 7.170210e-01 8.539259e-03 3.585279e-02 1.284621e-01
##           89          111          128          167          175          184
## 6.576032e-02 3.525159e-01 2.630608e-02 4.632291e-04 2.955175e-02 3.704639e-02
##          186          187          188          190          192          207
## 4.037569e-03 2.748444e-03 1.903290e-01 1.445015e-03 1.672665e-04 2.796571e-03
##          210          212          214          222          224          228
## 3.286033e-03 1.481361e-05 1.227449e-03 2.779674e-03 3.626330e-02 4.753468e-02
##          236          245          251          269          276
## 2.727441e-04 2.782436e-04 3.114245e-02 1.929091e-04 1.489659e-05

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[7]]))

```



```

mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 63.65  89.94  97.97  99.45 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.15842 ( N = 202 )
##
## Deviance: 146.167

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[7]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

##      [1] 0.936115878 0.975357620 0.734394958 0.105753888 0.187413241 0.415440060
##      [7] 0.176026179 0.372453271 0.051381463 0.016522551 0.172016349 0.196454885
##     [13] 0.112890034 0.024067004 0.242412387 0.008116437 0.014269614 0.014571715
##     [19] 0.057231533 0.003377998 0.068204552 0.006091990 0.073646872 0.037906949
##     [25] 0.032006219 0.022154194 0.034441562 0.004863360 0.006300652

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(gr

flexdiscr.model <- fda(DEATH_EVENT ~., data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.12871 ( N = 202 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[7]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

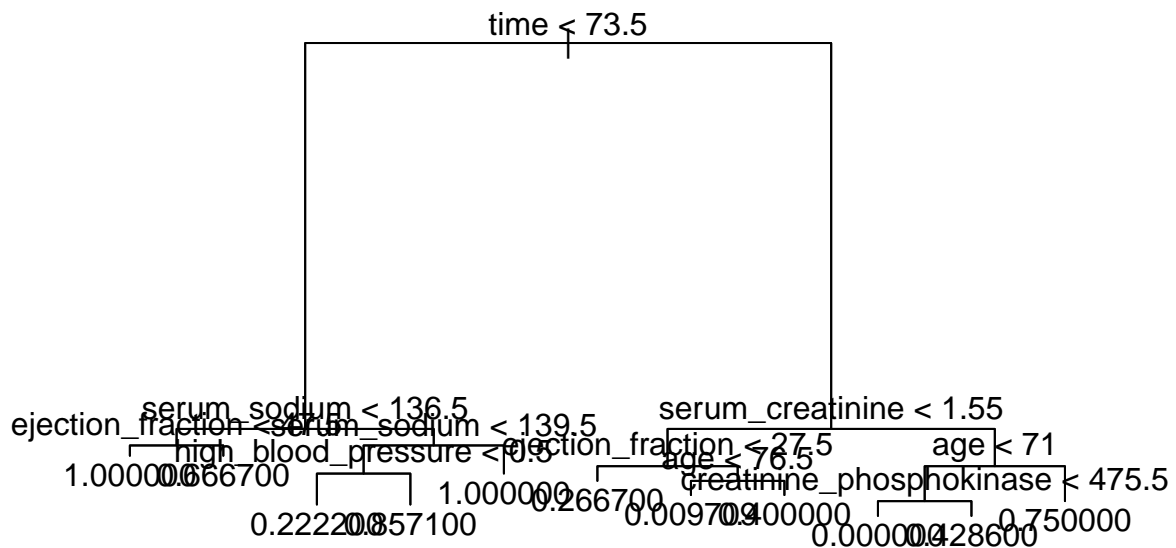
##           6           7           26           71           80           84
## 0.951137680 0.949867242 0.853185369 0.104540802 0.203188225 0.440082907
##           89          111          128          167          175          184
## 0.144461259 0.235579737 0.059043040 0.014381277 0.178509566 0.215602969
##          186          187          188          190          192          207
## 0.117627726 0.012169471 0.257965052 0.010302753 0.016105222 0.007352246
##          210          212          214          222          224          228

```

```
## 0.062556178 0.002946624 0.079403448 0.008280780 0.024268340 0.032412777
##      236      245      251      269      276
## 0.022863561 0.025743928 0.033820598 0.005564652 0.005406493

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[7]]))

decisiontree.model <- tree(DEATH_EVENT ~., data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



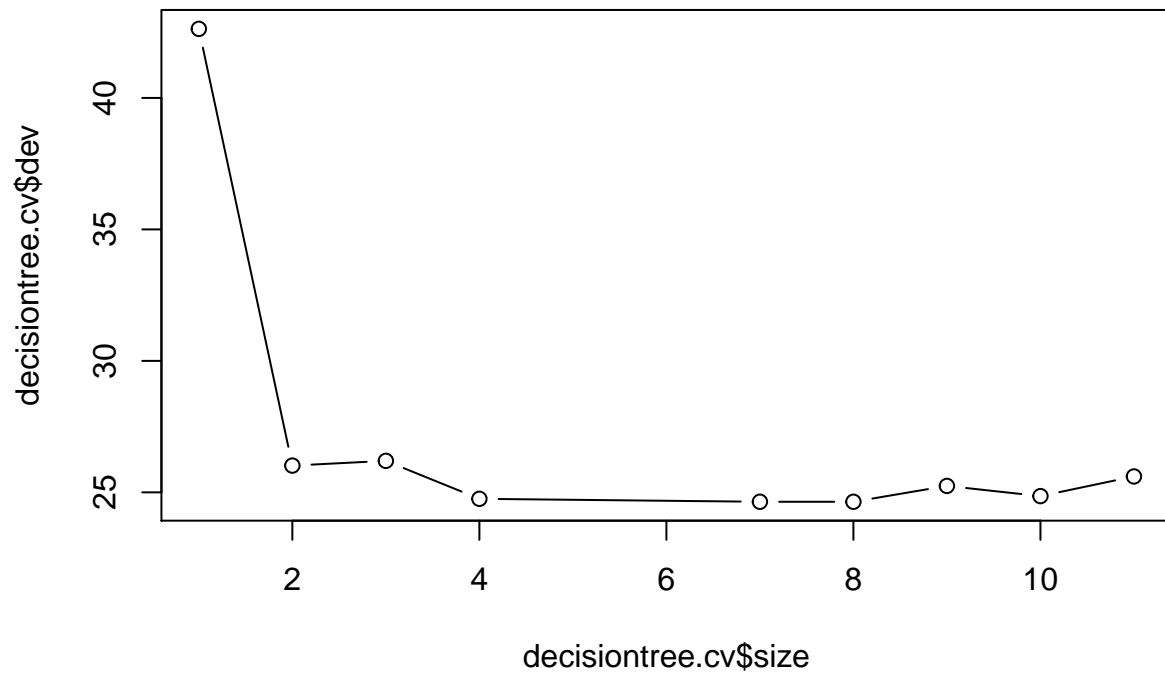
```
decisiontree.valid <- predict(decisiontree.model, groups[[7]], type = 'vector')
decisiontree.valid

##      6      7      26      71      80      84
## 1.000000000 0.222222222 1.000000000 1.000000000 0.009708738 0.750000000
##      89     111     128     167     175     184
## 0.009708738 0.400000000 0.009708738 0.009708738 0.009708738 0.750000000
##      186     187     188     190     192     207
## 0.009708738 0.009708738 0.428571429 0.009708738 0.009708738 0.009708738
##      210     212     214     222     224     228
## 0.009708738 0.009708738 0.000000000 0.009708738 0.266666667 0.009708738
##      236     245     251     269     276
## 0.400000000 0.428571429 0.009708738 0.009708738 0.009708738 0.009708738

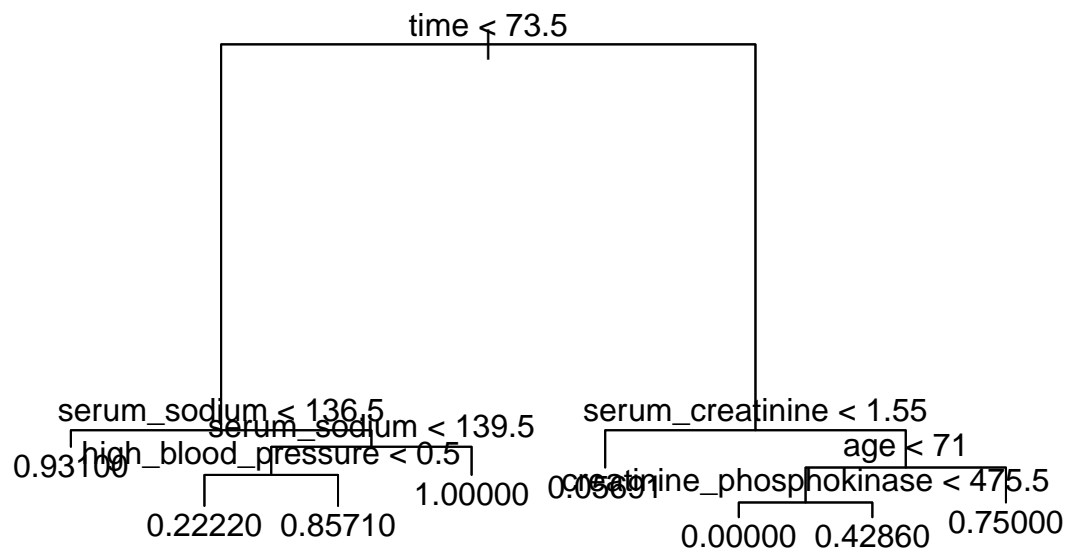
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - decisiontree.valid)^2))/nrow(groups[[7]]))

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[7]], type = 'vector')
prunedectree.valid
```

```
##          6          7          26          71          80          84          89
## 0.93103448 0.22222222 1.00000000 1.00000000 0.05691057 0.75000000 0.05691057
##          111         128         167         175         184         186         187
## 0.05691057 0.05691057 0.05691057 0.05691057 0.75000000 0.05691057 0.05691057
##          188         190         192         207         210         212         214
## 0.42857143 0.05691057 0.05691057 0.05691057 0.05691057 0.05691057 0.00000000
##          222         224         228         236         245         251         269
## 0.05691057 0.05691057 0.05691057 0.05691057 0.42857143 0.05691057 0.05691057
##          276
## 0.05691057
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##8th fold-out training
```

```
trainset <- data.frame()
```

```
for (i in c(1:7)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ ., data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             9.527e-01                7.709e-03            -1.056e-02
## creatinine_phosphokinase            diabetes            ejection_fraction
##             3.748e-05                6.027e-02            -9.153e-03
##             high_blood_pressure            platelets            serum_creatinine
##             -2.704e-02            -1.885e-07                5.409e-02
##             serum_sodium                sex                smoking
##             -3.078e-03            -8.543e-02                5.606e-02
##             time
##             -2.847e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77489 -0.28060 -0.03937  0.22957  0.99614
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.527e-01  8.956e-01   1.064 0.288808
## age           7.709e-03  2.350e-03   3.280 0.001236 **
## anaemia       -1.056e-02  5.356e-02  -0.197 0.843974
## creatinine_phosphokinase 3.748e-05  2.742e-05   1.367 0.173233
## diabetes      6.027e-02  5.356e-02   1.125 0.261930
## ejection_fraction -9.153e-03  2.335e-03  -3.920 0.000124 ***
## high_blood_pressure -2.704e-02  5.416e-02  -0.499 0.618177
## platelets     -1.885e-07  2.762e-07  -0.682 0.495867
## serum_creatinine  5.409e-02  2.711e-02   1.995 0.047439 *
## serum_sodium   -3.078e-03  6.473e-03  -0.475 0.635030
## sex           -8.543e-02  6.130e-02  -1.394 0.165048
## smoking        5.606e-02  6.303e-02   0.889 0.374980
## time          -2.847e-03  3.577e-04  -7.959 1.53e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3625 on 190 degrees of freedom
## Multiple R-squared:  0.4094, Adjusted R-squared:  0.3721
## F-statistic: 10.98 on 12 and 190 DF,  p-value: < 2.2e-16
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.4067   3.4067  25.9304 8.488e-07 ***
## anaemia       1  0.0003   0.0003   0.0022 0.9630234
```

```
## creatinine_phosphokinase 1 0.5838 0.5838 4.4435 0.0363463 *
## diabetes 1 0.1146 0.1146 0.8726 0.3514357
## ejection_fraction 1 2.6836 2.6836 20.4263 1.088e-05 ***
## high_blood_pressure 1 0.0642 0.0642 0.4887 0.4853756
## platelets 1 0.0193 0.0193 0.1473 0.7015860
## serum_creatinine 1 1.5734 1.5734 11.9761 0.0006654 ***
## serum_sodium 1 0.2132 0.2132 1.6229 0.2042384
## sex 1 0.0871 0.0871 0.6628 0.4166081
## smoking 1 0.2361 0.2361 1.7974 0.1816247
## time 1 8.3218 8.3218 63.3422 1.534e-13 ***
## Residuals 190 24.9619 0.1314
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[8]])
simplelinear.valid
```

```
##          25          33          34          37          41          45
## 0.882762968 0.570101926 0.619930814 0.669053077 0.823332941 0.429762498
##          72          74          75          100          103          133
## 0.527925543 0.451034394 0.688523196 0.466373356 0.685405730 0.229028689
##          135          141          142          145          148          162
## 0.691561766 0.453474391 0.229212673 0.531725087 0.136945491 0.182398831
##          193          195          203          232          234          281
## 0.031406119 0.160751596 -0.112708503 0.166146943 -0.003080904 0.085890801
##          283          288          289          297
## 0.034268126 -0.287945055 0.071149786 -0.424815893
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - simplelinear.valid)^2)/n))
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Coefficients:
##      (Intercept)          age ejection_fraction  serum_creatinine
##      0.525797      0.006780      -0.009180      0.057582
##      time
##      -0.002824
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76667 -0.29469 -0.02755  0.22270  1.04540
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5257970  0.1649396   3.188  0.00167 **
## age            0.0067802  0.0022565   3.005  0.00300 **
## ejection_fraction -0.0091803  0.0022541  -4.073  6.72e-05 ***
## serum_creatinine  0.0575818  0.0263489   2.185  0.03004 *
## time           -0.0028241  0.0003455  -8.174  3.50e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3603 on 198 degrees of freedom
## Multiple R-squared:  0.3918, Adjusted R-squared:  0.3795
## F-statistic: 31.88 on 4 and 198 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##               Df Sum Sq Mean Sq F value    Pr(>F)
## age             1  3.4067   3.4067  26.238 7.138e-07 ***
## ejection_fraction 1  2.8519   2.8519  21.965 5.158e-06 ***
## serum_creatinine 1  1.6237   1.6237  12.505 0.0005053 ***
## time            1  8.6755   8.6755  66.817 3.500e-14 ***
## Residuals       198 25.7082   0.1298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[8]])
revisedlinear.valid
```

```
##           25           33           34           37           41           45
## 0.799322636 0.522004191 0.576597770 0.649858616 0.834631931 0.351937274
##           72           74           75           100          103          133
## 0.449052777 0.379018711 0.672964792 0.406207777 0.656350270 0.236246381
##           135          141          142          145          148          162
## 0.519453245 0.474877658 0.260376643 0.568880666 0.147375061 0.213949604
##           193          195          203          232          234          281
## 0.043691790 0.231099429 -0.023858424 0.155035049 0.026682047 -0.006762688
##           283          288          289          297
## 0.047952238 -0.322442691 -0.014420552 -0.458933964
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##         serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##           2.76051           0.05536           -0.06998           0.42883
##      serum_sodium              time
##          -0.01966          -0.02173
##
```

```
## Degrees of Freedom: 202 Total (i.e. Null); 197 Residual
## Null Deviance: 246.5
## Residual Deviance: 149.9 AIC: 161.9
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1156  -0.6160  -0.2323   0.4224   2.8645
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.760513    7.468452   0.370  0.71166
## age            0.055357    0.019714   2.808  0.00498 **
## ejection_fraction -0.069983    0.019730  -3.547  0.00039 ***
## serum_creatinine  0.428832    0.210172   2.040  0.04131 *
## serum_sodium    -0.019663    0.053249  -0.369  0.71193
## time           -0.021732    0.003681  -5.904 3.54e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.47  on 202  degrees of freedom
## Residual deviance: 149.88  on 197  degrees of freedom
## AIC: 161.88
##
## Number of Fisher Scoring iterations: 6
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			202	246.47
## age	1	16.550	201	229.92
## ejection_fraction	1	15.918	200	214.00
## serum_creatinine	1	8.630	199	205.37
## serum_sodium	1	0.536	198	204.83
## time	1	54.952	197	149.88

```
generlinear.valid <- predict(generlinear.model, groups[[8]], type = 'response')
generlinear.valid
```



```

##          25          33          34          37          41          45
## 0.9215021860 0.5944687069 0.6456087159 0.7998856294 0.9377383447 0.2396515223
##          72          74          75          100          103          133
## 0.4089495599 0.3012339964 0.8136450316 0.3315216863 0.7665855778 0.1186309786
##          135          141          142          145          148          162
## 0.5700287045 0.4994699504 0.1326109741 0.6411417324 0.0680987110 0.0981964788
##          193          195          203          232          234          281
## 0.0345566985 0.1162805736 0.0192070042 0.0760556366 0.0265896450 0.0219560456
##          283          288          289          297
## 0.0344167449 0.0020312363 0.0184439969 0.0006362863

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - generlinear.valid)^2))/nrow

generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 2.17 1.00 2.11 2.36 5.34 total = 13.98
##
## REML score: 81.85365

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29557    0.02273      13    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          2.174   2.740   2.988  0.03016 *
## s(creatinine_phosphokinase) 1.000   1.000   3.405  0.06653 .
## s(ejection_fraction)      2.109   2.622   6.058  0.00149 **
## s(serum_creatinine)       2.358   2.926   2.680  0.06055 .
## s(time)           5.338   6.490  14.993 2.58e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```
## R-sq.(adj) = 0.499   Deviance explained = 53.1%
## -REML = 81.854   Scale est. = 0.10487   n = 203
```

```
generaddit.valid <- predict(generaddit.model, groups[[8]], type = 'response')
generaddit.valid
```

```
##          25          33          34          37          41          45
## 0.98760090 0.66618766 0.74468886 0.87355862 1.04853102 0.58844810
##          72          74          75          100          103          133
## 0.29783168 0.27480138 0.57243202 0.13600989 0.48566586 0.07148415
##          135          141          142          145          148          162
## 0.48165203 0.26437828 0.04894624 0.41714621 0.07127034 0.04858933
##          193          195          203          232          234          281
## 0.04580058 0.40571073 0.03361599 0.07102274 0.05487338 0.11255692
##          283          288          289          297
## 0.18144432 -0.11526498 0.04210247 -0.13336485
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model
```

```
## Call:
```

```
## lda(DEATH_EVENT ~ ., data = trainset)
```

```
##
```

```
## Prior probabilities of groups:
```

```
##          0          1
```

```
## 0.7044335 0.2955665
```

```
##
```

```
## Group means:
```

```
##          age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
```

```
## 0 58.67599 0.4055944          529.7762 0.4055944          39.35664
```

```
## 1 66.09445 0.4500000          699.7167 0.4166667          33.86667
```

```
##  high_blood_pressure platelets serum_creatinine serum_sodium          sex
```

```
## 0          0.3636364 264906.9          1.209021          137.4266 0.6363636
```

```
## 1          0.4333333 267540.5          1.827000          135.9167 0.6166667
```

```
##          smoking          time
```

```
## 0 0.3076923 160.1119
```

```
## 1 0.3166667 69.6500
```

```
##
```

```
## Coefficients of linear discriminants:
```

```
##          LD1
```

```
## age          3.418632e-02
```

```
## anaemia      -4.681336e-02
```

```
## creatinine_phosphokinase 1.662088e-04
```

```
## diabetes     2.672813e-01
```

```
## ejection_fraction -4.059063e-02
```

```
## high_blood_pressure -1.199213e-01
```

```
## platelets     -8.358088e-07
```

```
## serum_creatinine 2.398766e-01
```

```
## serum_sodium  -1.364901e-02
```

```
## sex          -3.788516e-01
```

```
## smoking      2.485975e-01
```

```
## time        -1.262541e-02
```

```

lindiscr.valid <- predict(lindiscr.model, groups[[8]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid

```

```

##          25          33          34          37          41          45
## 0.9602970276 0.6610961675 0.7444864081 0.8122909012 0.9374542390 0.3865424994
##          72          74          75          100          103          133
## 0.5814106579 0.4278614024 0.8350354976 0.4583297899 0.8315484262 0.1112304682
##          135          141          142          145          148          162
## 0.8383783439 0.4326777134 0.1113770103 0.5888375519 0.0562681650 0.0791676528
##          193          195          203          232          234          281
## 0.0248545573 0.0673567426 0.0079232695 0.0701377742 0.0189420985 0.0380222542
##          283          288          289          297
## 0.0254192695 0.0019440227 0.0339109369 0.0006465767

```

```

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[8]]))

```

```

quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model

```

```

## Call:
## qda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.67599 0.4055944                    529.7762 0.4055944          39.35664
## 1 66.09445 0.4500000                    699.7167 0.4166667          33.86667
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3636364 264906.9          1.209021      137.4266 0.6363636
## 1      0.4333333 267540.5          1.827000      135.9167 0.6166667
##      smoking      time
## 0 0.3076923 160.1119
## 1 0.3166667  69.6500

```

```

quaddiscr.valid <- predict(quaddiscr.model, groups[[8]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid

```

```

##          25          33          34          37          41          45
## 0.588945502 0.723610250 0.505606603 0.861145464 0.816950464 0.394379920
##          72          74          75          100          103          133
## 0.517575799 0.452741653 0.611975120 0.279828500 0.776882991 0.032928676
##          135          141          142          145          148          162
## 0.999951279 0.564350861 0.038425145 0.483127107 0.093977862 0.005117779
##          193          195          203          232          234          281
## 0.004886616 0.039753191 0.001807596 0.021327632 0.001317098 0.143375835
##          283          288          289          297
## 0.678585601 0.001901758 0.006464185 0.117353192

```

```

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[8]]))

```

```

mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)

```

```
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 58.53 87.34 95.86 98.92 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.15271 ( N = 203 )
##
## Deviance: 143.136
```

```
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[8]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid
```

```
## [1] 9.325167e-01 6.237149e-01 5.842070e-01 8.398703e-01 9.111071e-01
## [6] 3.920496e-01 9.094719e-01 6.863898e-01 8.065239e-01 4.818798e-01
## [11] 9.454793e-01 1.291964e-01 7.230618e-01 4.722684e-01 9.753750e-02
## [16] 3.870600e-01 8.769482e-02 1.878488e-01 2.763884e-02 6.070326e-02
## [21] 1.389776e-02 5.340000e-02 1.650661e-02 4.341011e-02 2.550158e-02
## [26] 3.469503e-05 3.132356e-02 1.052038e-06
```

```
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[8]]$DEATH_EVENT))
```

```
flexdiscr.model <- fda(DEATH_EVENT ~ ., data = trainset)
flexdiscr.model
```

```
## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.14286 ( N = 203 )
```

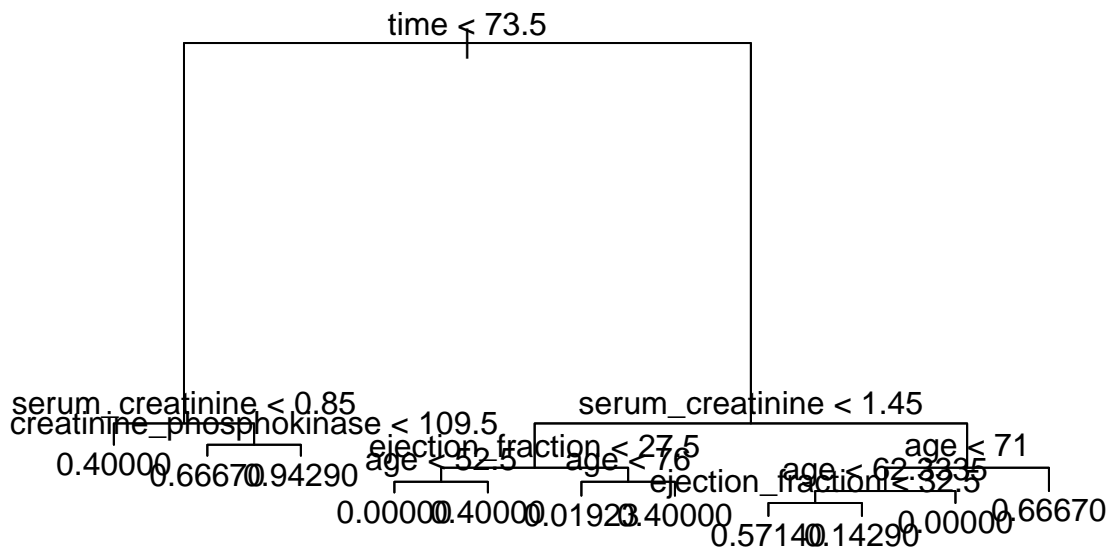
```
flexdiscr.valid <- predict(flexdiscr.model, groups[[8]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid
```

```
##           25           33           34           37           41           45
## 0.9618068710 0.6645134624 0.7481371770 0.8158054948 0.9395082312 0.3875023748
##           72           74           75          100          103          133
## 0.5843066509 0.4292696648 0.8384204982 0.4600630228 0.8349564229 0.1100460536
##           135          141          142          145          148          162
## 0.8417401880 0.4341378267 0.1101926741 0.5917917885 0.0552460235 0.0780253878
##           193          195          203          232          234          281
```

```
## 0.0241879221 0.0662652106 0.0076193599 0.0690322858 0.0183811237 0.0371717723
##          283          288          289          297
## 0.0247434051 0.0018430142 0.0331114148 0.0006062591

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[8]]))

decisiontree.model <- tree(DEATH_EVENT ~., data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```

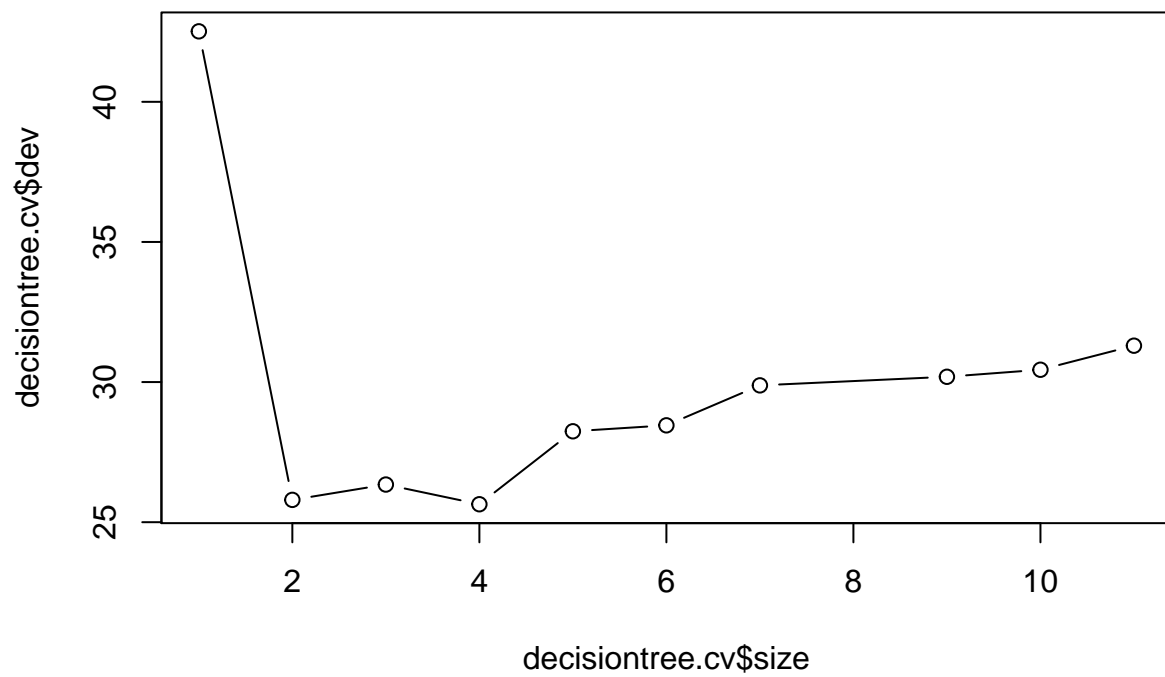


```
decisiontree.valid <- predict(decisiontree.model, groups[[8]], type = 'vector')
decisiontree.valid

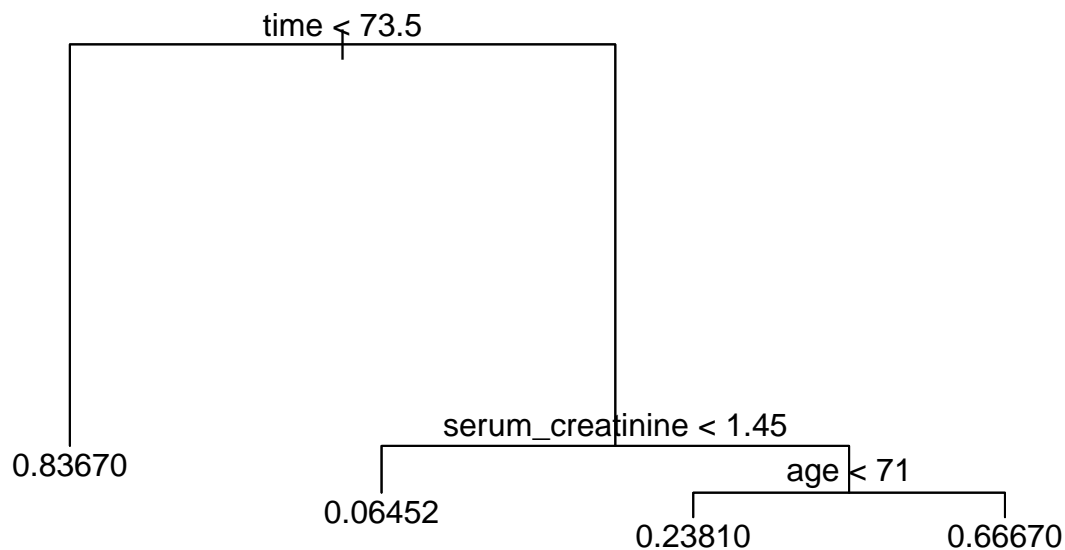
##          25          33          34          37          41          45          72
## 0.94285714 0.94285714 0.94285714 0.66666667 0.94285714 0.94285714 0.94285714
##          74          75          100          103          133          135          141
## 0.94285714 0.94285714 0.01923077 0.40000000 0.01923077 0.40000000 0.40000000
##          142          145          148          162          193          195          203
## 0.01923077 0.66666667 0.01923077 0.01923077 0.01923077 0.57142857 0.01923077
##          232          234          281          283          288          289          297
## 0.01923077 0.01923077 0.01923077 0.57142857 0.01923077 0.01923077 0.01923077

decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - decisiontree.valid)^2))/nrow(groups[[8]]))

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[8]], type = 'vector')
prunedectree.valid
```

```
##      25      33      34      37      41      45      72
## 0.83673469 0.83673469 0.83673469 0.83673469 0.83673469 0.83673469 0.83673469
##      74      75      100     103     133     135     141
## 0.83673469 0.83673469 0.06451613 0.06451613 0.06451613 0.06451613 0.06451613
##      142     145     148     162     193     195     203
## 0.06451613 0.66666667 0.06451613 0.06451613 0.06451613 0.23809524 0.06451613
##      232     234     281     283     288     289     297
## 0.06451613 0.06451613 0.06451613 0.23809524 0.06451613 0.06451613 0.06451613
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##Model comparison over training cross-validation
sum(simplelinear.RMSE)/8
```

```
## [1] 0.07002033
```

```
sum(revisedlinear.RMSE)/8
```

```
## [1] 0.06794522
```

```
sum(generlinear.RMSE)/8
```

```
## [1] 0.06561213
```

```

sum(generaddit.RMSE)/8

## [1] 0.06482254
sum(lindiscr.RMSE)/8

## [1] 0.06725358
sum(quaddiscr.RMSE)/8

## [1] 0.08396521
sum(mixeddiscr.RMSE)/8

## [1] 0.07155041
sum(flexdiscr.RMSE)/8

## [1] 0.06725752
sum(decisiontree.RMSE)/8

## [1] 0.06971369
sum(prunedectree.RMSE)/8

## [1] 0.06542507
##Conclusions from Model Comparison over Training Cross-Validation

##We can note that the Revised Linear Regression has an improvement
##over the Simple Linear Regression, when it comes to Error Rate.
##The Best Discriminant Analysis Types are Linear and Flexible,
##which is somewhat predictable for Linear over Quadratic when taking
##into account the small size of the dataset. Surprisingly, the Pruned
##Tree is as good as most models, even with with the relatively low
##amount of predictors it uses. However, the Generalized Additive Model
##seems to be the best model so far, being the only model with mean
##error under 0.065 for the cross-validation comparison

##Applications over Test set (groups 9 & 10)

trainset <- data.frame()
for (i in c(1:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
testset <- data.frame()
for (i in c(9,10)) {
  testset <- rbind(testset, groups[[i]])
}

simplelinear.model <- lm(DEATH_EVENT ~ ., data = trainset)
simplelinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Coefficients:

```



```
##           (Intercept)                age                anaemia
##           1.405e+00                7.183e-03                -6.544e-03
## creatinine_phosphokinase            diabetes            ejection_fraction
##           3.067e-05                4.792e-02                -8.613e-03
##           high_blood_pressure        platelets            serum_creatinine
##           1.149e-02                6.153e-09                5.644e-02
##           serum_sodium                sex                smoking
##           -6.901e-03                -5.912e-02                1.538e-02
##           time
##           -2.717e-03
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ ., data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75303 -0.28163 -0.03551  0.25099  1.00550
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.405e+00  8.537e-01   1.646 0.101122
## age             7.183e-03  2.148e-03   3.343 0.000974 ***
## anaemia        -6.544e-03  5.093e-02  -0.129 0.897868
## creatinine_phosphokinase 3.067e-05  2.589e-05   1.184 0.237562
## diabetes        4.792e-02  5.080e-02   0.943 0.346630
## ejection_fraction -8.613e-03  2.189e-03  -3.935 0.000112 ***
## high_blood_pressure  1.149e-02  5.157e-02   0.223 0.823843
## platelets        6.153e-09  2.529e-07   0.024 0.980616
## serum_creatinine  5.644e-02  2.638e-02   2.139 0.033527 *
## serum_sodium     -6.901e-03  6.160e-03  -1.120 0.263794
## sex             -5.912e-02  5.915e-02  -1.000 0.318650
## smoking          1.538e-02  5.974e-02   0.257 0.797062
## time            -2.717e-03  3.352e-04  -8.106 3.74e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3656 on 218 degrees of freedom
## Multiple R-squared:  0.398, Adjusted R-squared:  0.3648
## F-statistic: 12.01 on 12 and 218 DF,  p-value: < 2.2e-16
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.0962   4.0962  30.6534 8.824e-08 ***
## anaemia       1  0.0180   0.0180   0.1347 0.7139375
## creatinine_phosphokinase 1  0.2943   0.2943   2.2024 0.1392392
## diabetes      1  0.0936   0.0936   0.7004 0.4035571
## ejection_fraction 1  3.2874   3.2874  24.6010 1.418e-06 ***
## high_blood_pressure 1  0.3220   0.3220   2.4096 0.1220467
```

```
## platelets          1  0.0457  0.0457  0.3418  0.5594151
## serum_creatinine   1  1.6999  1.6999 12.7209  0.0004445 ***
## serum_sodium       1  0.4102  0.4102  3.0698  0.0811677 .
## sex                1  0.1097  0.1097  0.8212  0.3658205
## smoking            1  0.1005  0.1005  0.7520  0.3867994
## time               1  8.7806  8.7806 65.7077  3.735e-14 ***
## Residuals          218 29.1315  0.1336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.test <- predict(simplelinear.model, testset)
simplelinear.test
```

```
##          2          3          4          15          29          55
## 0.764048159 0.825946231 0.674144263 0.580545232 0.763861476 0.666395111
##          60          64          65          79          86          87
## 0.753286054 0.278360504 0.009091613 0.562742639 0.145605996 0.355921913
##          91          96          114          116          121          130
## 0.417652658 0.114410027 0.274869479 0.334202887 0.199648149 0.382558471
##          131          137          156          159          160          165
## 0.086703004 0.214423615 0.415369794 0.368908021 0.183113939 0.288349742
##          168          178          200          208          218          220
## 0.492129555 0.014007838 0.394292789 0.241378387 0.239343140 0.116664271
##          221          231          250          263          273          284
## 0.358433352 0.240559791 0.086369246 0.198636534 0.051438557 -0.014869021
##          285          291          11          13          17          19
## -0.249423166 -0.303118071 0.877409493 0.522304030 0.690219145 0.752716452
##          20          23          30          31          48          83
## 0.567739747 0.592488976 0.829780939 0.878321901 0.518270269 0.686953060
##          88          92          94          122          127          139
## 0.217624174 0.388210433 0.502952130 0.467949685 0.646747752 0.431675013
##          152          163          180          185          213          225
## 0.110372921 0.254918767 0.113508183 0.222602598 0.106137001 0.163545286
##          227          242          253          262          265          266
## 0.164765944 0.196291827 -0.107016688 0.010112364 -0.025436078 -0.143265968
##          270          299
## -0.118183331 -0.280937420
```

```
simplelinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - simplelinear.test)^2)/nrow(testset))
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + time, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      time, data = trainset)
##
## Coefficients:
##      (Intercept)          age ejection_fraction  serum_creatinine
##          0.510099          0.006603          -0.008752          0.061201
##              time
##          -0.002764
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##     time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7558 -0.2917 -0.0355  0.2549  1.0361
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.5100989  0.1562814   3.264  0.00127 **
## age            0.0066028  0.0020870   3.164  0.00177 **
## ejection_fraction -0.0087523  0.0021095  -4.149 4.73e-05 ***
## serum_creatinine  0.0612012  0.0256299   2.388  0.01777 *
## time           -0.0027641  0.0003247  -8.512 2.40e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3629 on 226 degrees of freedom
## Multiple R-squared:  0.3848, Adjusted R-squared:  0.3739
## F-statistic: 35.33 on 4 and 226 DF,  p-value: < 2.2e-16
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  4.0962   4.0962   31.096 6.979e-08 ***
## ejection_fraction  1  3.3338   3.3338   25.308 9.961e-07 ***
## serum_creatinine  1  1.6449   1.6449   12.487 0.0004968 ***
## time            1  9.5435   9.5435   72.447 2.401e-15 ***
## Residuals      226 29.7712   0.1317
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.test <- predict(revisedlinear.model, testset)
revisedlinear.test
```

```
##           2           3           4           15           29           55
## 0.591403004 0.824449706 0.762127801 0.599099993 0.843575246 0.583938810
##           60           64           65           79           86           87
## 0.726936640 0.393486249 0.005118918 0.504900754 0.233704105 0.415881069
##           91           96           114          116          121          130
## 0.406275825 0.199704561 0.277950351 0.352638623 0.210341698 0.471571659
##           131          137          156          159          160          165
## 0.084755802 0.173466044 0.459811498 0.378829432 0.238677018 0.255409593
##           168          178          200          208          218          220
## 0.498349770 0.050900298 0.195977913 0.276946756 0.263036582 0.065237341
##           221          231          250          263          273          284
## 0.381766864 0.175573498 -0.008119071 0.156594384 0.016138438 -0.017007842
##           285          291           11           13           17           19
```

```
## -0.151915506 -0.335563500 0.889886923 0.581572842 0.768340932 0.773229086
##          20          23          30          31          48          83
## 0.420477411 0.652559394 0.790536669 0.835544696 0.499837372 0.627628554
##          88          92          94          122          127          139
## 0.244740594 0.352487649 0.564847121 0.411909657 0.517152832 0.375822075
##          152          163          180          185          213          225
## 0.126016566 0.291081661 0.159680401 0.277800450 0.142478892 0.168816874
##          227          242          253          262          265          266
## 0.187177233 0.170286750 -0.086693637 -0.031811907 -0.001365849 -0.074392727
##          270          299
## -0.145357404 -0.243459592
```

```
revisedlinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - revisedlinear.test)^2))/nrow(testset)
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine + serum_sodium + time
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##        serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)          age ejection_fraction  serum_creatinine
##          6.87239          0.05086          -0.06674          0.44398
##    serum_sodium           time
##       -0.04887       -0.02157
##
## Degrees of Freedom: 230 Total (i.e. Null);  225 Residual
## Null Deviance:      281.7
## Residual Deviance: 171.7    AIC: 183.7
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine +
##      serum_sodium + time, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0825  -0.6045  -0.2296   0.4275   2.8048
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.872392    7.016676   0.979 0.327364
## age             0.050858    0.017412   2.921 0.003491 **
## ejection_fraction -0.066742    0.018138  -3.680 0.000234 ***
## serum_creatinine  0.443978    0.206769   2.147 0.031776 *
## serum_sodium    -0.048866    0.050103  -0.975 0.329405
## time           -0.021571    0.003467  -6.222 4.91e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
## Null deviance: 281.71 on 230 degrees of freedom
## Residual deviance: 171.70 on 225 degrees of freedom
## AIC: 183.7
##
## Number of Fisher Scoring iterations: 6
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			230	281.71
## age	1	19.825	229	261.88
## ejection_fraction	1	18.514	228	243.37
## serum_creatinine	1	8.832	227	234.54
## serum_sodium	1	1.472	226	233.06
## time	1	61.366	225	171.70

```
generlinear.test <- predict(generlinear.model, testset, type = 'response')
generlinear.test
```

##	2	3	4	15	29	55
##	0.699855356	0.951006367	0.888787190	0.690794478	0.940037033	0.719800568
##	60	64	65	79	86	87
##	0.865073864	0.243728467	0.023938910	0.603314089	0.108567141	0.359689220
##	91	96	114	116	121	130
##	0.357288796	0.081748396	0.163712147	0.238880383	0.112968084	0.352274927
##	131	137	156	159	160	165
##	0.040861555	0.080954125	0.396606409	0.326804950	0.114618526	0.126986455
##	168	178	200	208	218	220
##	0.535848342	0.028173056	0.241046782	0.165097459	0.121328275	0.030880057
##	221	231	250	263	273	284
##	0.319432651	0.119755030	0.028256681	0.096070963	0.029097955	0.017883052
##	285	291	11	13	17	19
##	0.005585341	0.001374317	0.964748524	0.671557285	0.882596752	0.885251411
##	20	23	30	31	48	83
##	0.562182246	0.754311632	0.928386451	0.942618266	0.511939630	0.777997797
##	88	92	94	122	127	139
##	0.117485376	0.297267754	0.655747758	0.363972744	0.688508310	0.301157061
##	152	163	180	185	213	225
##	0.052377926	0.176321330	0.062888603	0.157721167	0.059452383	0.071231248
##	227	242	253	262	265	266
##	0.105124492	0.078298076	0.013640065	0.020664226	0.017445601	0.010567724
##	270	299				
##	0.009023393	0.003431947				

```
generlinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - generlinear.test)^2))/nrow(testset)
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
```

```
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Estimated degrees of freedom:
## 2.25 1.00 2.56 2.69 5.68 total = 15.18
##
## REML score: 90.81294
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine) + s(time)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29870    0.02118   14.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)          2.254  2.836  3.654 0.012863 *
## s(creatinine_phosphokinase) 1.000  1.000  3.112 0.079070 .
## s(ejection_fraction)      2.560  3.174  6.779 0.000178 ***
## s(serum_creatinine)       2.692  3.342  2.768 0.033903 *
## s(time)             5.679  6.861 15.712 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.507 Deviance explained = 53.8%
## -REML = 90.813 Scale est. = 0.10367 n = 231
```

```
generaddit.test <- predict(generaddit.model, testset, type = 'response')
generaddit.test
```

```
##           2           3           4           15           29           55
## 1.015999824 1.047871271 1.099297448 0.748125306 0.754695154 0.682247605
##           60           64           65           79           86           87
## 0.795864025 0.367899753 0.319699539 0.237980343 0.059568598 0.181480869
##           91           96          114          116          121          130
## 0.136101269 0.078065058 0.138655403 0.066652603 0.125501011 0.244130107
##          131          137          156          159          160          165
## 0.016323228 0.007299461 0.383152013 0.375155107 0.083440990 0.288388129
```

```
##          168          178          200          208          218          220
## 0.585879173 0.112867290 0.257778163 0.279193951 0.115970131 -0.006452569
##          221          231          250          263          273          284
## 0.522976327 0.171234444 -0.029980336 0.213205948 0.010689590 0.042645856
##          285          291          11          13          17          19
## -0.114655487 -0.127726782 0.950732198 0.795444199 0.912536619 0.904023685
##          20          23          30          31          48          83
## 0.773320456 0.698001664 0.946440801 1.123683543 0.506715849 0.525118959
##          88          92          94          122          127          139
## 0.093815726 0.154733982 0.422146791 0.071828047 0.528008577 0.082712257
##          152          163          180          185          213          225
## 0.020051642 0.093345457 0.116549980 0.349791783 0.223735382 0.176284925
##          227          242          253          262          265          266
## 0.206966096 0.128635664 -0.080468131 -0.096494921 -0.003372152 -0.059776504
##          270          299
## -0.054363046 -0.018047935
```

```
generaddit.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - generaddit.test)^2))/nrow(testset)
```

```
lindiscr.model <- lda(DEATH_EVENT ~., data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7012987 0.2987013
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.63375 0.3950617          569.6481 0.4135802          39.54321
## 1 66.32851 0.4492754          680.5217 0.4202899          33.76812
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3333333 265940.7          1.205864      137.3827 0.6358025
## 1          0.4492754 265384.1          1.767826      135.7826 0.6231884
##      smoking      time
## 0 0.3209877 159.61111
## 1 0.3188406  69.52174
##
## Coefficients of linear discriminants:
##                                LD1
## age                3.192527e-02
## anaemia            -2.908546e-02
## creatinine_phosphokinase 1.362989e-04
## diabetes           2.129650e-01
## ejection_fraction   -3.827788e-02
## high_blood_pressure  5.108550e-02
## platelets           2.734444e-08
## serum_creatinine     2.508403e-01
## serum_sodium        -3.067231e-02
## sex                -2.627368e-01
## smoking             6.835990e-02
## time              -1.207645e-02
```

```

lindiscr.test <- predict(lindiscr.model, testset, type = 'response')
lindiscr.test <- lindiscr.test$posterior[,2]
lindiscr.test

##           2           3           4           15           29           55
## 0.897963728 0.934708753 0.812764489 0.675309749 0.897829191 0.803317409
##           60           64           65           79           86           87
## 0.889947142 0.162036948 0.022757271 0.643905330 0.063761425 0.262416624
##           91           96          114          116          121          130
## 0.366285515 0.050596528 0.158345229 0.230732947 0.094327840 0.304902955
##          131          137          156          159          160          165
## 0.041100990 0.104728825 0.362130102 0.282648274 0.083794364 0.172983985
##          168          178          200          208          218          220
## 0.509316775 0.023632774 0.324796475 0.126323619 0.124568414 0.051454566
##          221          231          250          263          273          284
## 0.266255694 0.125615128 0.040997712 0.093650669 0.031463303 0.018924298
##          285          291           11           13           17           19
## 0.003042545 0.001997010 0.955465118 0.568190073 0.831239511 0.889507835
##           20           23           30           31           48           83
## 0.652860543 0.695548429 0.936524480 0.955769318 0.560394047 0.827607200
##           88           92           94          122          127          139
## 0.107111349 0.314400000 0.530548600 0.461873015 0.777769040 0.392227931
##          152          163          180          185          213          225
## 0.049093632 0.138545529 0.050257064 0.110912115 0.047562289 0.072715715
##          227          242          253          262          265          266
## 0.073365380 0.092097869 0.009261650 0.022936406 0.017442147 0.006981255
##          270          299
## 0.008489931 0.002376505

lindiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - lindiscr.test)^2))/nrow(testset)

quaddiscr.model <- qda(DEATH_EVENT ~., data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ ., data = trainset)
##
## Prior probabilities of groups:
##           0           1
## 0.7012987 0.2987013
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.63375 0.3950617                569.6481 0.4135802                39.54321
## 1 66.32851 0.4492754                680.5217 0.4202899                33.76812
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0           0.3333333 265940.7           1.205864      137.3827 0.6358025
## 1           0.4492754 265384.1           1.767826      135.7826 0.6231884
##      smoking      time
## 0 0.3209877 159.61111
## 1 0.3188406  69.52174

quaddiscr.test <- predict(quaddiscr.model, testset, type = 'response')
quaddiscr.test <- quaddiscr.test$posterior[,2]
quaddiscr.test

```



```
##          2          3          4          15          29          55
## 1.000000e+00 9.487571e-01 4.345082e-01 3.370239e-01 9.999998e-01 8.296393e-01
##          60          64          65          79          86          87
## 6.930503e-01 4.776465e-02 7.151643e-03 8.453959e-01 9.888894e-03 1.035907e-01
##          91          96          114          116          121          130
## 3.390937e-01 3.415412e-03 3.577947e-02 6.275011e-02 5.293765e-03 9.877625e-01
##          131          137          156          159          160          165
## 2.240977e-03 1.610754e-01 9.631117e-02 3.623110e-01 1.351925e-02 2.558846e-01
##          168          178          200          208          218          220
## 1.601627e-01 9.901368e-03 9.872399e-01 2.474982e-01 1.000000e+00 5.757391e-03
##          221          231          250          263          273          284
## 1.782683e-01 1.627629e-01 1.077263e-02 2.865198e-03 4.790003e-03 2.906133e-03
##          285          291          11          13          17          19
## 8.686034e-04 3.092156e-03 9.998622e-01 9.552709e-02 9.319778e-01 6.238029e-01
##          20          23          30          31          48          83
## 9.750401e-01 4.002883e-01 9.321308e-01 8.138975e-01 3.132965e-01 6.039924e-01
##          88          92          94          122          127          139
## 1.882018e-02 1.370410e-01 3.193702e-01 2.230775e-01 9.609236e-01 1.852488e-01
##          152          163          180          185          213          225
## 9.599252e-03 6.993426e-02 8.865620e-03 2.745299e-02 2.430913e-02 8.045282e-03
##          227          242          253          262          265          266
## 2.195829e-02 7.332278e-03 1.092294e-03 4.689156e-03 4.252316e-03 9.057801e-04
##          270          299
## 1.400628e-03 6.313003e-05
```

```
quaddiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - quaddiscr.test)^2))/nrow(testset)
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~., data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 64.20  89.63  97.51  99.39 100.00
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.15152 ( N = 231 )
##
## Deviance: 174.08
```

```
mixeddiscr.test <- predict(mixeddiscr.model, testset, type = 'posterior')
mixeddiscr.test <- mixeddiscr.test[,2]
mixeddiscr.test
```

```
## [1] 0.988993383 0.955776769 0.884420589 0.559741534 0.816028489 0.765165709
## [7] 0.883037098 0.083975996 0.017628212 0.709108956 0.053479057 0.351468583
## [13] 0.400399638 0.051869697 0.140629879 0.415411171 0.055855398 0.209212092
## [19] 0.025330873 0.159151401 0.437272868 0.295536322 0.087659809 0.094287441
## [25] 0.693723075 0.074066950 0.600867876 0.149384671 0.999889750 0.041841499
## [31] 0.246986749 0.288414724 0.042938302 0.108561566 0.036226154 0.016997356
## [37] 0.003136405 0.001851249 0.965027052 0.817147141 0.833043823 0.900056952
```

```

## [43] 0.822631889 0.649771469 0.947226862 0.962290954 0.660319858 0.796374045
## [49] 0.098554627 0.301724965 0.609639743 0.589023262 0.750882032 0.365041369
## [55] 0.048053067 0.183837842 0.045632545 0.100664084 0.074014065 0.039777433
## [61] 0.066582909 0.084107990 0.010665318 0.024340982 0.021022699 0.006117497
## [67] 0.009690891 0.003460350

mixeddiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - mixeddiscr.test)^2))/nrow(testset)

flexdiscr.model <- fda(DEATH_EVENT ~., data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ ., data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
## v1
## 100
##
## Degrees of Freedom (per dimension): 13
##
## Training Misclassification Error: 0.14286 ( N = 231 )

flexdiscr.test <- predict(flexdiscr.model, testset, type = 'posterior')
flexdiscr.test <- flexdiscr.test[,2]
flexdiscr.test

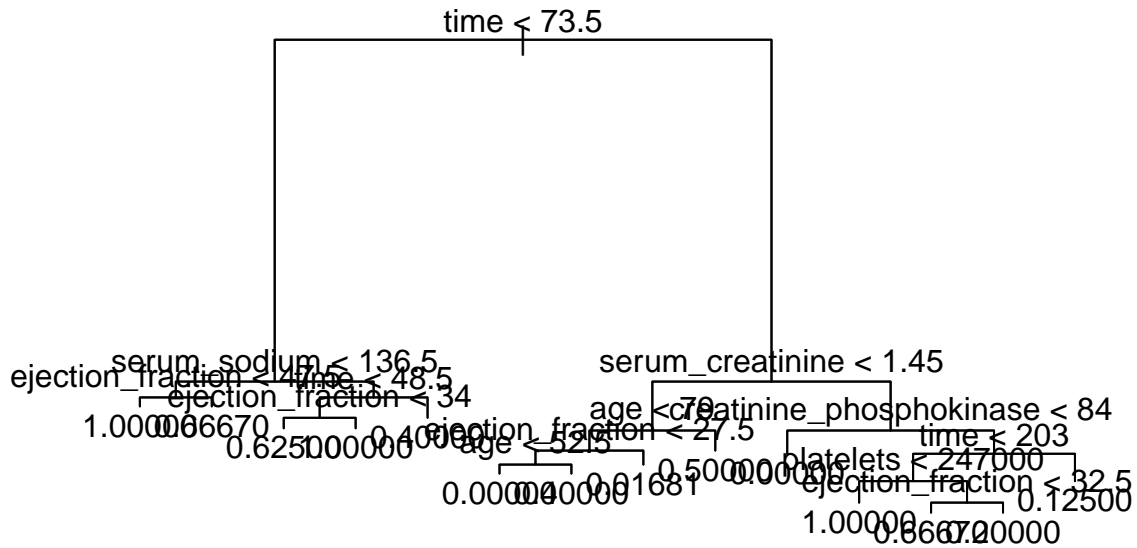
##          2          3          4          15          29          55
## 0.900361640 0.936557357 0.815830460 0.678339119 0.900228762 0.806418208
##          60          64          65          79          86          87
## 0.892439994 0.161102703 0.022199554 0.646795408 0.062812297 0.262112535
##          91          96          114          116          121          130
## 0.366904665 0.049731624 0.157396481 0.230189768 0.093282235 0.304957436
##          131          137          156          159          160          165
## 0.040317833 0.103675340 0.362710019 0.282510395 0.082768620 0.172097270
##          168          178          200          208          218          220
## 0.511260843 0.023061709 0.325029563 0.125285919 0.123527965 0.050583208
##          221          231          250          263          273          284
## 0.265982422 0.124576290 0.040215556 0.092605967 0.030785557 0.018428961
##          285          291          11          13          17          19
## 0.002914480 0.001905844 0.956899188 0.570605420 0.834217402 0.892005663
##          20          23          30          31          48          83
## 0.655794246 0.698645951 0.938340301 0.957196598 0.562751306 0.830604580
##          88          92          94          122          127          139
## 0.106057567 0.314539065 0.532670729 0.463394350 0.780932222 0.393093394
##          152          163          180          185          213          225
## 0.048240386 0.137533454 0.049394754 0.109858959 0.046721396 0.071725456
##          227          242          253          262          265          266
## 0.072372590 0.091055424 0.008960551 0.022375932 0.016972825 0.006737113
##          270          299
## 0.008207469 0.002271503

flexdiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - flexdiscr.test)^2))/nrow(testset)

decisiontree.model <- tree(DEATH_EVENT ~., data = trainset)

```

```
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



```
decisiontree.test <- predict(decisiontree.model, testset, type = 'vector')
decisiontree.test
```

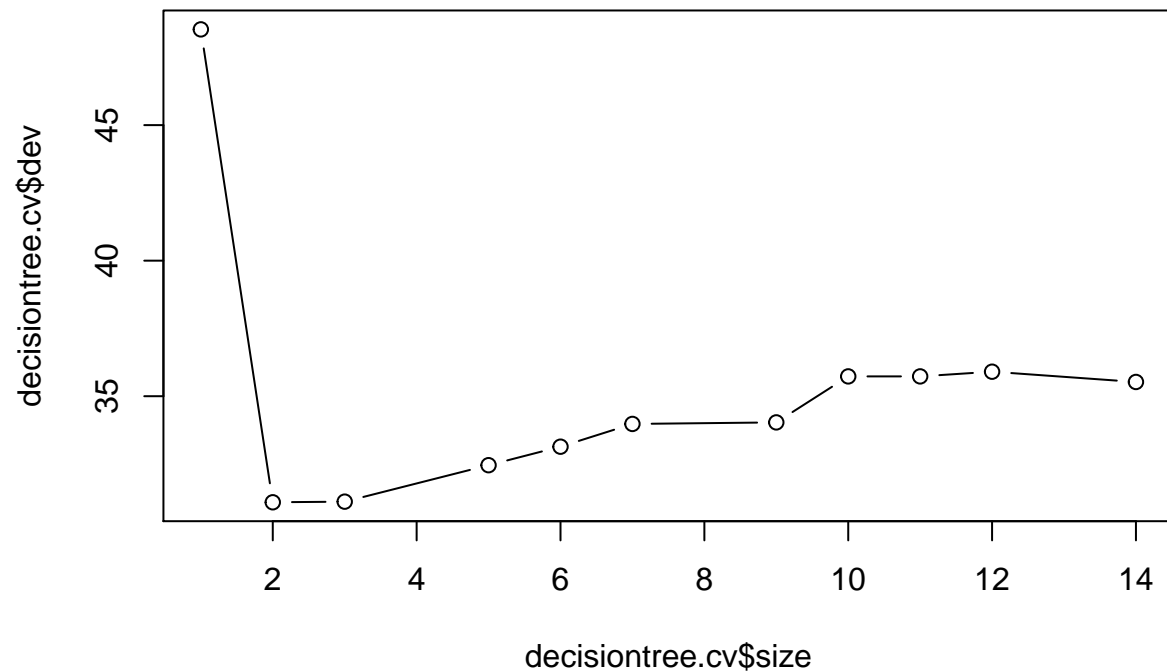
```
##      2      3      4      15      29      55      60
## 1.00000000 1.00000000 0.62500000 0.62500000 1.00000000 1.00000000 1.00000000
##      64      65      79      86      87      91      96
## 0.40000000 0.40000000 0.01680672 0.01680672 0.01680672 0.01680672 0.01680672
##      114     116     121     130     131     137     156
## 0.01680672 0.01680672 1.00000000 1.00000000 0.01680672 0.01680672 1.00000000
##      159     160     165     168     178     200     208
## 0.50000000 0.01680672 0.01680672 0.00000000 0.01680672 0.20000000 0.50000000
##      218     220     221     231     250     263     273
## 1.00000000 0.01680672 0.66666667 0.12500000 0.01680672 0.40000000 0.01680672
##      284     285     291      11      13      17      19
## 0.01680672 0.01680672 0.01680672 1.00000000 0.62500000 1.00000000 0.62500000
##      20      23      30      31      48      83      88
## 0.66666667 1.00000000 1.00000000 1.00000000 1.00000000 0.00000000 0.01680672
##      92      94     122     127     139     152     163
## 0.01680672 1.00000000 0.01680672 0.66666667 0.01680672 0.01680672 0.01680672
##      180     185     213     225     227     242     253
## 0.01680672 0.40000000 0.01680672 0.40000000 0.40000000 0.01680672 0.01680672
##      262     265     266     270     299
## 0.01680672 0.01680672 0.01680672 0.01680672 0.12500000
```

```

decisiontree.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - decisiontree.test)^2))/nrow(testset)

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

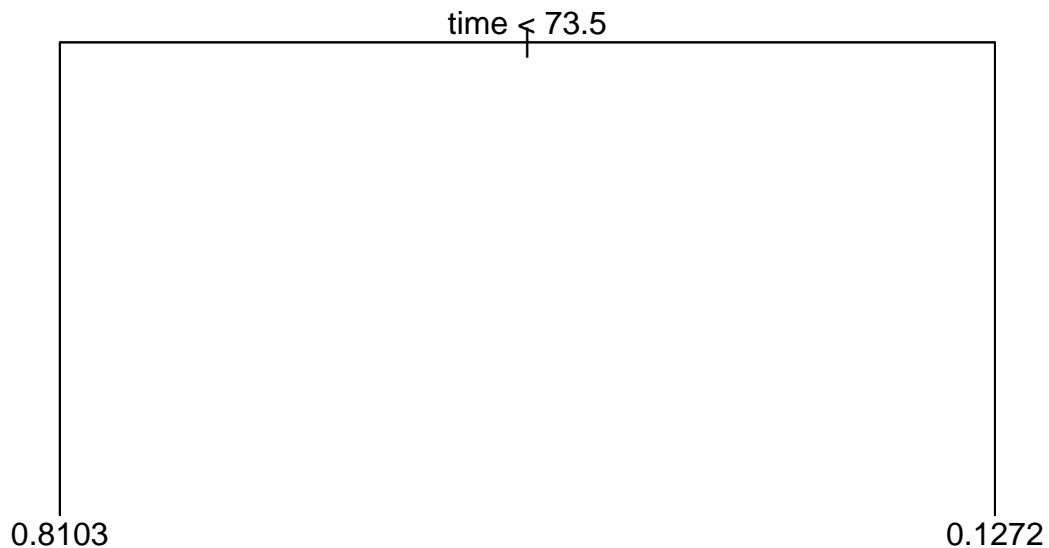
```



```

prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)

```



```
prunedectree.test <- predict(prunedectree.model, testset, type = 'vector')
prunedectree.test
```

```
##      2      3      4      15      29      55      60      64
## 0.8103448 0.8103448 0.8103448 0.8103448 0.8103448 0.8103448 0.8103448 0.8103448
##      65      79      86      87      91      96      114      116
## 0.8103448 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676
##      121      130      131      137      156      159      160      165
## 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676
##      168      178      200      208      218      220      221      231
## 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676
##      250      263      273      284      285      291      11      13
## 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.8103448 0.8103448
##      17      19      20      23      30      31      48      83
## 0.8103448 0.8103448 0.8103448 0.8103448 0.8103448 0.8103448 0.8103448 0.1271676
##      88      92      94      122      127      139      152      163
## 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676
##      180      185      213      225      227      242      253      262
## 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676 0.1271676
##      265      266      270      299
## 0.1271676 0.1271676 0.1271676 0.1271676
```

```
prunedectree.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - prunedectree.test)^2))/nrow(testset)
```

```
##Model Comparison over Test Set
```

```
lindiscr.test.RMSE
```

```
## [1] 0.04457358
revisedlinear.test.RMSE
```

```
## [1] 0.04571624
generlinear.test.RMSE
```

```
## [1] 0.04403429
generaddit.test.RMSE
```

```
## [1] 0.04052547
lindiscr.test.RMSE
```

```
## [1] 0.04457358
quaddiscr.test.RMSE
```

```
## [1] 0.05211896
mixeddiscr.test.RMSE
```

```
## [1] 0.0430242
flexdiscr.test.RMSE
```

```
## [1] 0.04457326
decisiontree.test.RMSE
```

```
## [1] 0.04870489
prunedectree.test.RMSE
```

```
## [1] 0.04855105
```

##Conclusions from Testing

*##Most Model Conclusions seem in line with our comments from
##earlier cross-validation. We still note that the decision trees
##seem like a less reliable option compared to other models now.
##We can also notice there is very little difference between linear
##models now (with the Simple Linear, Revised Linear and Generalized
##Linear Models all hovering around 0.044). We can note that only
##Quadratic Discriminant is over the 0.05 mark (again, likely due
##to the effect of a small dataset on Quadratic Discriminant Analysis)
##Finally, similarly to our Cross-Validation experiments, the
##Generalized Additive Model has best performance with a Root Mean
##Square Error which rounds up to 4.1% with the Mixed Discriminant
##Analysis Model with 4.3% and all others equal or above 4.4%.*

*##However we were building models which would give us the percentage
##of dying from heart failure
##One of the issues is that the models above are purely analytical
##Since we can not predict the follow-up period duration when
##patients are admitted, we cannot predict the actual risk of
##passing away from heart failure. We do however know that the longer
##the follow-up is, the better the chance of survival is.*

*##Then how about focusing on a purely predictive model ?
##Let us try all this again with the follow-up period to build our*

```
##analytical models

##Let us reset our error metrics
simplelinear.RMSE <- c()
##Once again, we only include the predictors with high significance
summary(lm(DEATH_EVENT~.-time, data = dataf))

##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = dataf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9112 -0.3146 -0.1264  0.3821  1.0122
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.498e+00  7.942e-01   1.886  0.0603 .
## age              9.012e-03  2.097e-03   4.298 2.36e-05 ***
## anaemia          5.488e-02  5.023e-02   1.093  0.2755
## creatinine_phosphokinase 4.880e-05  2.561e-05   1.905  0.0577 .
## diabetes         1.664e-02  5.038e-02   0.330  0.7415
## ejection_fraction -1.059e-02  2.105e-03  -5.032 8.57e-07 ***
## high_blood_pressure  6.795e-02  5.120e-02   1.327  0.1855
## platelets        -7.211e-08  2.523e-07  -0.286  0.7752
## serum_creatinine   1.063e-01  2.412e-02   4.409 1.47e-05 ***
## serum_sodium      -1.096e-02  5.726e-03  -1.914  0.0566 .
## sex              -6.256e-02  5.836e-02  -1.072  0.2847
## smoking           1.340e-02  5.844e-02   0.229  0.8187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4165 on 287 degrees of freedom
## Multiple R-squared:  0.2361, Adjusted R-squared:  0.2068
## F-statistic: 8.063 on 11 and 287 DF,  p-value: 2.917e-12

anova(lm(DEATH_EVENT~.-time, data = dataf))

## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.196   4.1960  24.1866 1.474e-06 ***
## anaemia       1  0.127   0.1268   0.7310  0.39326
## creatinine_phosphokinase 1  0.569   0.5695   3.2827  0.07106 .
## diabetes      1  0.043   0.0433   0.2494  0.61785
## ejection_fraction 1  5.203   5.2030  29.9914 9.485e-08 ***
## high_blood_pressure 1  0.275   0.2746   1.5830  0.20935
## platelets     1  0.025   0.0254   0.1462  0.70249
## serum_creatinine 1  4.107   4.1070  23.6738 1.885e-06 ***
## serum_sodium  1  0.631   0.6310   3.6372  0.05750 .
## sex           1  0.201   0.2015   1.1614  0.28209
## smoking       1  0.009   0.0091   0.0526  0.81874
## Residuals    287 49.790   0.1735
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##Note those predictors are age, ejection_fraction, & serum_creatinine
revisedlinear.RMSE <- c()
##Once again, for the Generalized Linear Model
generlinear.model <- glm(DEATH_EVENT~.-time, family = binomial(link = "logit"), data = dataf)
stepAIC(generlinear.model, direction = 'both')

## Start:  AIC=318.28
## DEATH_EVENT ~ (age + anaemia + creatinine_phosphokinase + diabetes +
##      ejection_fraction + high_blood_pressure + platelets + serum_creatinine +
##      serum_sodium + sex + smoking + time) - time
##
##              Df Deviance    AIC
## - smoking          1   294.43 316.43
## - platelets         1   294.47 316.47
## - diabetes          1   294.54 316.54
## - sex              1   295.58 317.58
## - high_blood_pressure 1   296.15 318.15
## - anaemia          1   296.22 318.22
## <none>              1   294.28 318.28
## - serum_sodium     1   297.21 319.21
## - creatinine_phosphokinase 1  298.46 320.46
## - serum_creatinine 1   314.04 336.04
## - age              1   314.06 336.06
## - ejection_fraction 1   320.96 342.96
##
## Step:  AIC=316.43
## DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes +
##      ejection_fraction + high_blood_pressure + platelets + serum_creatinine +
##      serum_sodium + sex
##
##              Df Deviance    AIC
## - platelets          1   294.61 314.61
## - diabetes           1   294.68 314.68
## - sex                1   295.59 315.59
## - anaemia            1   296.28 316.28
## - high_blood_pressure 1   296.29 316.29
## <none>               1   294.43 316.43
## - serum_sodium       1   297.35 317.35
## + smoking            1   294.28 318.28
## - creatinine_phosphokinase 1  298.51 318.51
## - serum_creatinine   1   314.07 334.07
## - age                1   314.17 334.17
## - ejection_fraction   1   321.18 341.18
##
## Step:  AIC=314.61
## DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + diabetes +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium + sex
##
##              Df Deviance    AIC
## - diabetes           1   294.83 312.83
## - sex                1   295.66 313.66
```



```

## - high_blood_pressure      1  296.38 314.38
## - anaemia                  1  296.49 314.49
## <none>                      294.61 314.61
## - serum_sodium             1  297.63 315.63
## + platelets                 1  294.43 316.43
## + smoking                   1  294.47 316.47
## - creatinine_phosphokinase 1  298.62 316.62
## - age                       1  314.29 332.29
## - serum_creatinine          1  314.39 332.39
## - ejection_fraction         1  321.53 339.53
##
## Step: AIC=312.83
## DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction +
##   high_blood_pressure + serum_creatinine + serum_sodium + sex
##
##               Df Deviance    AIC
## - sex          1   296.05 312.05
## - high_blood_pressure 1   296.58 312.58
## - anaemia       1   296.71 312.71
## <none>           294.83 312.83
## - serum_sodium  1   298.02 314.02
## + diabetes      1   294.61 314.61
## + platelets     1   294.68 314.68
## + smoking       1   294.71 314.71
## - creatinine_phosphokinase 1  298.84 314.84
## - age           1   314.29 330.29
## - serum_creatinine 1   314.46 330.46
## - ejection_fraction 1   321.68 337.68
##
## Step: AIC=312.05
## DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction +
##   high_blood_pressure + serum_creatinine + serum_sodium
##
##               Df Deviance    AIC
## <none>           296.05 312.05
## - anaemia       1   298.16 312.16
## - high_blood_pressure 1   298.42 312.42
## + sex           1   294.83 312.83
## - serum_sodium  1   299.06 313.06
## + diabetes      1   295.66 313.66
## - creatinine_phosphokinase 1  299.72 313.72
## + platelets     1   296.01 314.01
## + smoking       1   296.03 314.03
## - age           1   314.73 328.73
## - serum_creatinine 1   315.76 329.76
## - ejection_fraction 1   321.83 335.83
##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##   ejection_fraction + high_blood_pressure + serum_creatinine +
##   serum_sodium, family = binomial(link = "logit"), data = dataf)
##
## Coefficients:
##               (Intercept)                  age                anaemia

```

```

##              4.7298351              0.0530244              0.4309325
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##              0.0002674              -0.0679014              0.4606311
##              serum_creatinine      serum_sodium
##              0.6619016              -0.0568437
##
## Degrees of Freedom: 298 Total (i.e. Null); 291 Residual
## Null Deviance:      375.3
## Residual Deviance: 296.1      AIC: 312.1
##Note now the predictors are age, anaemia, creatinine_phosphokinase,
##ejection_fraction, high_blood_pressure, serum_creatinine, & serum_sodium
generlinear.RMSE <- c()
##Once again, for the Generalized Additive Model
generaddit.model <- gam(DEATH_EVENT ~ s(age) + anaemia + s(creatinine_phosphokinase) + diabetes + s(ejection_fraction) + s(serum_creatinine) + s(serum_sodium) + sex + smoking)
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + anaemia + s(creatinine_phosphokinase) +
##      diabetes + s(ejection_fraction) + high_blood_pressure + s(platelets) +
##      s(serum_creatinine) + s(serum_sodium) + sex + smoking
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.29144    0.05586   5.218 3.53e-07 ***
## anaemia           0.04888    0.04776   1.023   0.307
## diabetes          0.05062    0.04780   1.059   0.290
## high_blood_pressure 0.07030    0.04861   1.446   0.149
## sex              -0.08042    0.05486  -1.466   0.144
## smoking           0.04633    0.05517   0.840   0.402
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)         2.768   3.478  7.653 2.60e-05 ***
## s(creatinine_phosphokinase) 1.703   2.102   3.127  0.0392 *
## s(ejection_fraction)      3.522   4.420 11.171 6.97e-09 ***
## s(platelets)              1.000   1.000   0.499  0.4803
## s(serum_creatinine)       2.900   3.581   7.206 4.06e-05 ***
## s(serum_sodium)          1.000   1.000   0.588  0.4438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.312 Deviance explained = 35.3%
## -REML = 172.22 Scale est. = 0.15056 n = 299
##Note we use only the predictors that are significant; for our
##generalized additive model, they are s(age), s(creatinine_phosphokinase),
##s(ejection_fraction), & s(serum_creatinine)
generaddit.RMSE <- c()

```

```

lindiscr.RMSE <- c()
quaddiscr.RMSE <- c()
mixeddiscr.RMSE <- c()
flexdiscr.RMSE <- c()
decisiontree.RMSE <- c()
prunedectree.RMSE <- c()

##1st fold-out training
trainset <- data.frame()
for (i in 2:8){
  trainset <- rbind(trainset, groups[[i]])
}

simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model

```

```

##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.519e+00             1.104e-02             5.968e-03
## creatinine_phosphokinase            diabetes            ejection_fraction
##             4.329e-05             7.226e-02             -8.000e-03
##      high_blood_pressure            platelets            serum_creatinine
##             1.031e-01             2.574e-08             8.997e-02
##             serum_sodium                sex                smoking
##             -1.309e-02            -4.984e-02             2.943e-02

```

```
summary(simplelinear.model)
```

```

##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7923 -0.2872 -0.1336  0.3507  1.0009
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.519e+00  1.016e+00   1.494  0.13670
## age           1.104e-02  2.529e-03   4.364 2.08e-05 ***
## anaemia       5.968e-03  6.190e-02   0.096  0.92329
## creatinine_phosphokinase 4.329e-05  2.986e-05   1.450  0.14869
## diabetes      7.226e-02  6.123e-02   1.180  0.23941
## ejection_fraction -8.000e-03  2.712e-03  -2.950  0.00357 **
## high_blood_pressure  1.031e-01  6.116e-02   1.686  0.09334 .
## platelets      2.574e-08  3.001e-07   0.086  0.93172
## serum_creatinine  8.997e-02  3.296e-02   2.730  0.00692 **
## serum_sodium   -1.309e-02  7.262e-03  -1.803  0.07294 .
## sex           -4.984e-02  7.065e-02  -0.705  0.48142
## smoking       2.943e-02  7.322e-02   0.402  0.68813

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4163 on 193 degrees of freedom
## Multiple R-squared:  0.2194, Adjusted R-squared:  0.1749
## F-statistic: 4.932 on 11 and 193 DF,  p-value: 9.891e-07
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.985   3.9845  22.9917 3.244e-06 ***
## anaemia       1  0.008   0.0079   0.0454 0.8315287
## creatinine_phosphokinase 1  0.333   0.3334   1.9237 0.1670488
## diabetes      1  0.178   0.1782   1.0280 0.3118895
## ejection_fraction 1  2.138   2.1380  12.3366 0.0005531 ***
## high_blood_pressure 1  0.475   0.4755   2.7437 0.0992655 .
## platelets      1  0.064   0.0640   0.3694 0.5440584
## serum_creatinine 1  1.613   1.6134   9.3097 0.0026002 **
## serum_sodium   1  0.519   0.5186   2.9927 0.0852377 .
## sex            1  0.060   0.0599   0.3459 0.5571183
## smoking        1  0.028   0.0280   0.1616 0.6881301
## Residuals     193 33.447   0.1733
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[1]])
```

```
simplelinear.valid
```

```
##          22          46          49          52          66          68
## 0.55387271 0.25487161 0.98290497 0.37339129 0.60454092 0.36761618
##          76          77          126          129          132          134
## 0.25672884 0.19488966 -0.02983469 0.33454225 0.73614274 0.05110750
##          149          158          170          173          176          177
## 0.50371108 0.17693436 0.48896236 0.11233378 -0.01766896 0.32633060
##          191          211          215          223          239          259
## 0.61270153 0.63871017 0.34611651 -0.02013269 0.34136143 0.15692145
##          260          268
## -0.01105331 0.19026396
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - simplelinear.valid)^2)/n))
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
```

```
revisedlinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##      (Intercept)              age  ejection_fraction  serum_creatinine
##      -0.149534          0.010608          -0.008638           0.096302
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8056 -0.2965 -0.1555  0.3910  0.9932
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.149534   0.169405  -0.883  0.37845
## age             0.010608   0.002467   4.300 2.66e-05 ***
## ejection_fraction -0.008638  0.002647  -3.263  0.00130 **
## serum_creatinine  0.096302   0.032363   2.976  0.00328 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4183 on 201 degrees of freedom
## Multiple R-squared:  0.1791, Adjusted R-squared:  0.1668
## F-statistic: 14.62 on 3 and 201 DF,  p-value: 1.198e-08
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  3.985   3.9845  22.7683 3.51e-06 ***
## ejection_fraction  1  2.139   2.1391  12.2233 0.0005808 ***
## serum_creatinine  1  1.550   1.5496   8.8545 0.0032822 **
## Residuals       201 35.176   0.1750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[1]])
revisedlinear.valid
```

```
##           22           46           49           52           66
## 0.4349433863 0.2355995966 0.9500996243 0.3747664831 0.5934782104
##           68           76           77           126          129
## 0.4946137330 0.3816143015 0.1517900478 -0.0001021732 0.3058380375
##           132          134          149          158          170
## 0.6856821425 0.1256818741 0.5631771694 0.2612281094 0.3677518014
##           173          176          177          191          211
## 0.1806805808 0.0649660287 0.3332113686 0.5990284856 0.5425046193
##           215          223          239          259          260
## 0.3147096142 0.0996060861 0.2907776188 0.1889255668 0.0482011351
##           268
## 0.2799898659
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
```

```
generlinear.model
```

```
##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      4.851519          0.060717          0.165874
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##      0.000231          -0.048897          0.674270
##      serum_creatinine          serum_sodium
##      0.605507          -0.065952
##
## Degrees of Freedom: 204 Total (i.e. Null);  197 Residual
## Null Deviance:      249.6
## Residual Deviance: 202.1      AIC: 218.1
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0806  -0.7330  -0.4999   0.8162   2.4294
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.8515189   5.7477993   0.844  0.3986
## age             0.0607168   0.0155330   3.909 9.27e-05 ***
## anaemia        0.1658738   0.3696853   0.449  0.6537
## creatinine_phosphokinase 0.0002310   0.0001683   1.372  0.1700
## ejection_fraction -0.0488966   0.0174592  -2.801  0.0051 **
## high_blood_pressure  0.6742698   0.3574442   1.886  0.0592 .
## serum_creatinine   0.6055066   0.2279557   2.656  0.0079 **
## serum_sodium     -0.0659521   0.0420969  -1.567  0.1172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 249.60  on 204  degrees of freedom
## Residual deviance: 202.09  on 197  degrees of freedom
## AIC: 218.09
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
```

```
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                      204      249.60
## age                1  19.3005      203      230.30
## anaemia             1   0.0284      202      230.27
## creatinine_phosphokinase 1   1.7018      201      228.57
## ejection_fraction    1  11.4801      200      217.09
## high_blood_pressure   1   2.7000      199      214.39
## serum_creatinine      1   9.7780      198      204.61
## serum_sodium          1   2.5212      197      202.09

generlinear.valid <- predict(generlinear.model, groups[[1]], type = 'response')
generlinear.valid

##           22           46           49           52           66           68           76
## 0.54989365 0.16947277 0.97317925 0.40929075 0.71348072 0.39274277 0.25919305
##           77           126           129           132           134           149           158
## 0.13176121 0.05466525 0.32613003 0.85368663 0.08995587 0.62461095 0.16228838
##           170           173           176           177           191           211           215
## 0.49255250 0.13593577 0.06023697 0.29210457 0.63781198 0.65242235 0.40248385
##           223           239           259           260           268
## 0.06752722 0.23333569 0.13277354 0.05509268 0.16613172

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - generlinear.valid)^2))/nrow

generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##           s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.67 1.00 3.60 1.66 total = 9.93
##
## REML score: 112.7125

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
```

```
##      s(serum_creatinine)
##
## Parametric coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29756    0.02714   10.96  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df    F  p-value
## s(age)          2.667  3.350  7.623  3.90e-05 ***
## s(creatinine_phosphokinase) 1.000  1.000  1.652   0.2002
## s(ejection_fraction)      3.603  4.455  7.849  3.31e-06 ***
## s(serum_creatinine)       1.659  2.043  5.337   0.0055 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.281   Deviance explained = 31.3%
## -REML = 112.71   Scale est. = 0.15096    n = 205
```

```
generaddit.valid <- predict(generaddit.model, groups[[1]], type = 'response')
generaddit.valid
```

```
##          22          46          49          52          66          68          76
## 0.29797069 0.19197736 1.27010302 0.65824198 0.84440535 0.49133878 0.57680084
##          77          126          129          132          134          149          158
## 0.19703100 0.14470213 0.15554813 0.63755532 0.20080171 0.49081596 0.35310001
##          170          173          176          177          191          211          215
## 0.17272650 0.24200603 0.14295100 0.20929231 0.54413898 0.86640188 0.09145315
##          223          239          259          260          268
## 0.02694282 0.14221294 0.29024643 0.10345438 0.16938533
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~ . - time, data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.702439 0.297561
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.37963 0.3819444          594.0833 0.4236111          39.34722
## 1 66.58470 0.4098361          734.3443 0.4426230          34.86885
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3263889 265322.2          1.187847    137.4514 0.6180556
## 1          0.4590164 267614.9          1.741803    135.8361 0.6065574
##      smoking
## 0 0.2916667
## 1 0.2950820
##
## Coefficients of linear discriminants:
```



```
##                                LD1
## age                           5.805501e-02
## anaemia                       3.139012e-02
## creatinine_phosphokinase      2.277064e-04
## diabetes                      3.800273e-01
## ejection_fraction            -4.207352e-02
## high_blood_pressure           5.424288e-01
## platelets                    1.353866e-07
## serum_creatinine              4.731912e-01
## serum_sodium                 -6.886626e-02
## sex                          -2.621138e-01
## smoking                      1.548031e-01
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[1]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##          22          46          49          52          66          68          76
## 0.60518093 0.19979003 0.95395495 0.33888124 0.67581762 0.33107339 0.20159821
##          77          126          129          132          134          149          158
## 0.14783901 0.04247013 0.28821532 0.82249433 0.06759086 0.53062594 0.13463004
##          170          173          176          177          191          211          215
## 0.50828520 0.09511662 0.04557633 0.27809996 0.68657291 0.71950309 0.30283780
##          223          239          259          260          268
## 0.04493029 0.29677963 0.12109628 0.04735512 0.14433696
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[1]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~ . - time, data = trainset)
quaddiscr.model
```

```
## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.702439 0.297561
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.37963 0.3819444          594.0833 0.4236111          39.34722
## 1 66.58470 0.4098361          734.3443 0.4426230          34.86885
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3263889 265322.2      1.187847      137.4514 0.6180556
## 1      0.4590164 267614.9      1.741803      135.8361 0.6065574
##      smoking
## 0 0.2916667
## 1 0.2950820
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[1]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid
```

```
##          22          46          49          52          66          68
## 0.211808326 0.024042483 0.999927775 0.194989609 0.533433159 0.138049602
##          76          77          126          129          132          134
```

```
## 0.055606672 0.144404100 0.012455052 0.388266796 1.000000000 0.036247297
##      149      158      170      173      176      177
## 0.243355287 0.031770781 0.334606834 0.060086658 0.002778815 0.052286569
##      191      211      215      223      239      259
## 0.352666936 0.896428178 0.209535139 0.002023298 0.087616584 0.059153091
##      260      268
## 0.005790889 0.105997960

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[1]]))

mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 70.78 87.53 95.93 99.41 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23902 ( N = 205 )
##
## Deviance: 195.852

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[1]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 0.49058852 0.22470329 0.90238950 0.29344429 0.44175449 0.24197941
## [7] 0.17627527 0.19194567 0.05493033 0.24078956 0.99998902 0.04918388
## [13] 0.46157756 0.10560498 0.55911120 0.05671921 0.04478051 0.16293170
## [19] 0.82235031 0.88602671 0.28116587 0.03804501 0.23361792 0.20837214
## [25] 0.05360852 0.31033628

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[1]]))

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

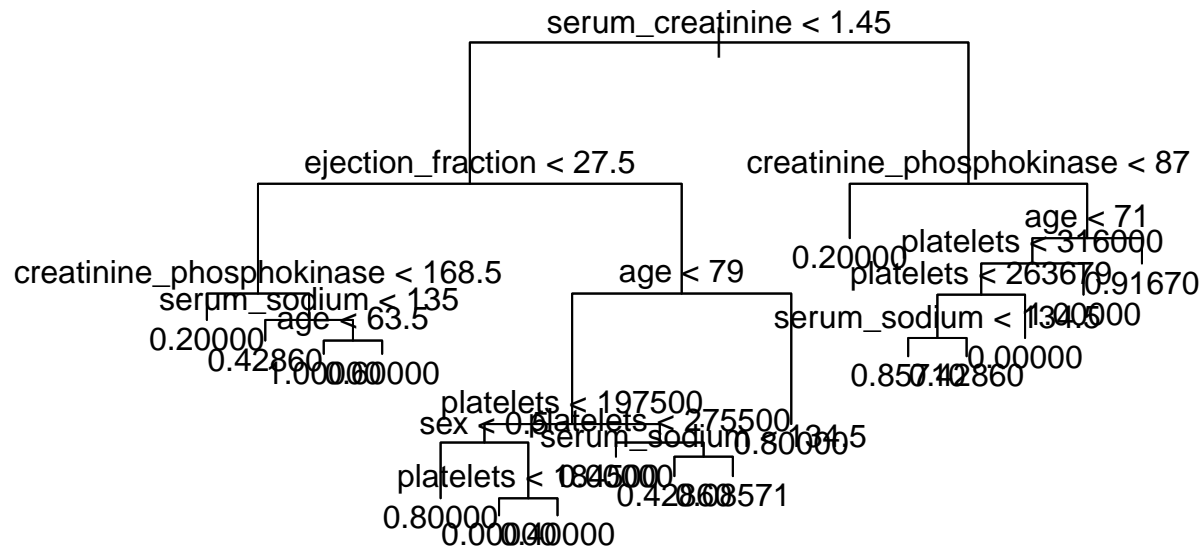
## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.24878 ( N = 205 )
```

```
flexdiscr.valid <- predict(flexdiscr.model, groups[[1]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid
```

```
##          22          46          49          52          66          68          76
## 0.60820428 0.19895862 0.95560937 0.33930220 0.67924778 0.33141299 0.20077899
##          77          126          129          132          134          149          158
## 0.14673439 0.04157515 0.28812412 0.82590926 0.06650294 0.53303386 0.13348442
##          170          173          176          177          191          211          215
## 0.51048168 0.09394129 0.04464996 0.27791216 0.69004585 0.72307001 0.30289005
##          223          239          259          260          268
## 0.04401025 0.29677194 0.11992348 0.04641185 0.14322017
```

```
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[1]]))
```

```
decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[1]], type = 'vector')
decisiontree.valid
```

```
##          22          46          49          52          66          68          76
## 0.00000000 0.00000000 0.91666667 0.20000000 0.20000000 0.20000000 0.20000000
##          77          126          129          132          134          149          158
## 0.08571429 0.00000000 0.00000000 0.85714286 0.08571429 0.91666667 1.00000000
##          170          173          176          177          191          211          215
```

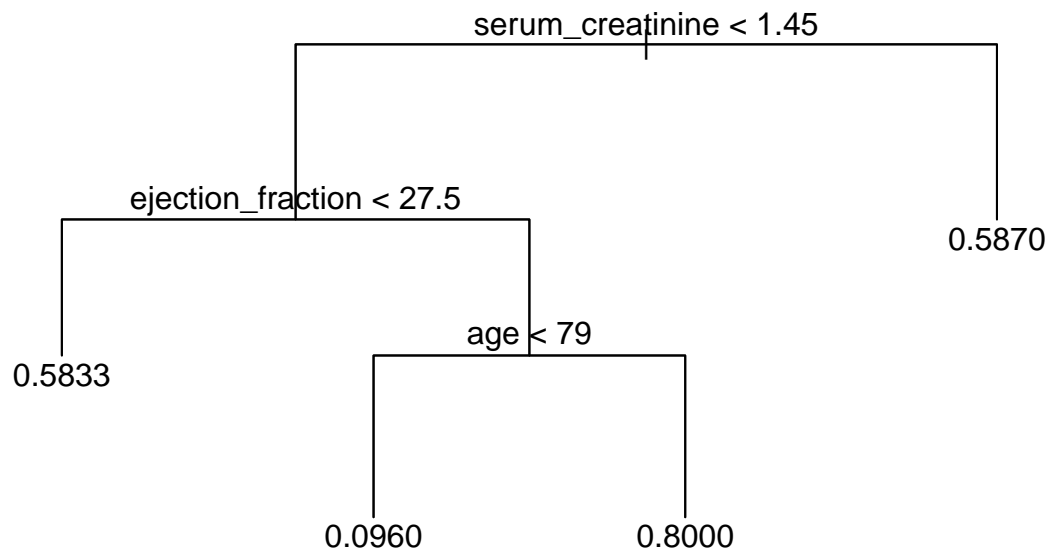
```
## 0.42857143 0.00000000 0.08571429 0.00000000 0.91666667 0.60000000 0.42857143
##      223      239      259      260      268
## 0.08571429 0.00000000 0.20000000 0.08571429 0.42857143

decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - decisiontree.valid)^2))/n)

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree.cv$dev)])
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[1]], type = 'vector')
prunedectree.valid
```

```
##      22      46      49      52      66      68      76      77
## 0.5869565 0.5869565 0.5869565 0.5833333 0.5869565 0.5833333 0.5833333 0.0960000
##      126      129      132      134      149      158      170      173
## 0.0960000 0.0960000 0.5869565 0.0960000 0.5869565 0.5833333 0.0960000 0.0960000
##      176      177      191      211      215      223      239      259
## 0.0960000 0.0960000 0.5869565 0.5833333 0.0960000 0.0960000 0.0960000 0.5833333
##      260      268
## 0.0960000 0.5869565
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##2nd fold-out training
```

```
trainset <- data.frame()
```

```
for (i in c(1,3:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
```

```
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.147e+00                9.836e-03                8.057e-02
## creatinine_phosphokinase                diabetes                ejection_fraction
##             4.982e-05                4.026e-02                -1.030e-02
##             high_blood_pressure                platelets                serum_creatinine
##             2.855e-02                -3.803e-08                9.135e-02
##             serum_sodium                sex                smoking
##             -9.020e-03                -7.793e-02                6.643e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7364 -0.3004 -0.1495  0.3724  1.0071
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.147e+00  1.042e+00   1.101 0.272250
## age            9.836e-03  2.497e-03   3.940 0.000114 ***
## anaemia        8.057e-02  6.088e-02   1.323 0.187284
## creatinine_phosphokinase 4.982e-05  3.223e-05   1.546 0.123834
## diabetes       4.026e-02  6.275e-02   0.642 0.521922
## ejection_fraction -1.030e-02  2.564e-03 -4.018 8.44e-05 ***
## high_blood_pressure 2.855e-02  6.165e-02   0.463 0.643884
## platelets      -3.803e-08  2.951e-07 -0.129 0.897583
## serum_creatinine 9.135e-02  3.082e-02   2.964 0.003421 **
## serum_sodium   -9.020e-03  7.411e-03 -1.217 0.225065
## sex           -7.793e-02  7.108e-02 -1.096 0.274312
## smoking        6.643e-02  7.339e-02   0.905 0.366545
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4121 on 191 degrees of freedom
## Multiple R-squared:  0.2325, Adjusted R-squared:  0.1883
## F-statistic: 5.261 on 11 and 191 DF,  p-value: 3.08e-07
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
age	1	3.530	3.5301	20.7858	9.158e-06 ***
anaemia	1	0.233	0.2334	1.3742	0.242554
creatinine_phosphokinase	1	0.521	0.5213	3.0693	0.081389 .
diabetes	1	0.057	0.0566	0.3333	0.564405
ejection_fraction	1	3.338	3.3384	19.6570	1.561e-05 ***
high_blood_pressure	1	0.019	0.0195	0.1146	0.735302
platelets	1	0.018	0.0179	0.1053	0.745943

```
## serum_creatinine      1  1.680  1.6795  9.8894  0.001928 **
## serum_sodium          1  0.188  0.1879  1.1061  0.294260
## sex                   1  0.105  0.1046  0.6159  0.433541
## smoking               1  0.139  0.1391  0.8192  0.366545
## Residuals            191 32.438  0.1698
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[2]])
simplelinear.valid

##           1           28           43           47           54           62
## 0.64907454 0.31647223 0.30641492 0.30969279 0.34248332 0.29130509
##           70           97          104          138          140          143
## 0.42723397 0.50650782 0.31896936 0.60067570 0.17770906 0.27300156
##          151          153          154          169          171          172
## 0.43341746 0.06445516 0.26092619 0.27976510 0.32660499 0.23947992
##          179          183          194          197          205          229
## -0.01097328 0.44996933 0.49759534 0.15791327 0.39198369 0.80661964
##          237          256          261          294
## 0.08217236 0.29453703 0.04005892 0.36917032

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - simplelinear.valid)^2))/n)

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##      -0.032934          0.009929          -0.010741          0.096445

summary(revisedlinear.model)

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6678 -0.2958 -0.1306  0.3846  1.0156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.032934   0.166519  -0.198  0.84342
## age           0.009929   0.002415   4.112 5.74e-05 ***
## ejection_fraction -0.010741  0.002489  -4.315 2.52e-05 ***
## serum_creatinine  0.096445   0.030094   3.205  0.00157 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.4107 on 199 degrees of freedom
## Multiple R-squared: 0.206, Adjusted R-squared: 0.194
## F-statistic: 17.21 on 3 and 199 DF, p-value: 5.622e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.530   3.5301   20.933 8.359e-06 ***
## ejection_fraction 1  3.445   3.4449   20.428 1.061e-05 ***
## serum_creatinine 1  1.732   1.7321   10.271 0.001574 **
## Residuals     199 33.559   0.1686
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[2]])
revisedlinear.valid
```

```
##          1          28          43          47          54          62          70
## 0.68018812 0.30415680 0.29451228 0.29173829 0.22151961 0.25572138 0.52044122
##          97          104          138          140          143          151          153
## 0.44946687 0.15831627 0.57626904 0.21846092 0.22810544 0.43974940 0.06699453
##          154          169          171          172          179          183          194
## 0.20140644 0.27928084 0.24219810 0.14055659 0.06389690 0.45968078 0.48348192
##          197          205          229          237          256          261          294
## 0.11953821 0.37942703 0.82617235 0.28081018 0.25760837 0.17998873 0.30348154
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - revisedlinear.valid)^2)))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##        ejection_fraction + high_blood_pressure + serum_creatinine +
##        serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##              (Intercept)              age              anaemia
##              2.3278620              0.0581904              0.5957786
## creatinine_phosphokinase ejection_fraction high_blood_pressure
##              0.0002591              -0.0694638              0.2205787
##          serum_creatinine          serum_sodium
##              0.6197315              -0.0415546
##
## Degrees of Freedom: 202 Total (i.e. Null); 195 Residual
## Null Deviance: 246.5
## Residual Deviance: 194.7 AIC: 210.7
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
```



```
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0223  -0.7358  -0.4582   0.8058   2.5762
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.3278620   6.0848539   0.383 0.702041
## age              0.0581904   0.0156050   3.729 0.000192 ***
## anaemia          0.5957786   0.3709056   1.606 0.108212
## creatinine_phosphokinase 0.0002591 0.0001778   1.458 0.144935
## ejection_fraction -0.0694638 0.0182822  -3.800 0.000145 ***
## high_blood_pressure 0.2205787 0.3715752   0.594 0.552759
## serum_creatinine  0.6197315 0.2190429   2.829 0.004665 **
## serum_sodium     -0.0415546 0.0443780  -0.936 0.349079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.47  on 202  degrees of freedom
## Residual deviance: 194.72  on 195  degrees of freedom
## AIC: 210.72
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			202	246.47
## age	1	17.1370	201	229.33
## anaemia	1	1.2452	200	228.08
## creatinine_phosphokinase	1	2.6307	199	225.45
## ejection_fraction	1	19.0596	198	206.39
## high_blood_pressure	1	0.0602	197	206.33
## serum_creatinine	1	10.7226	196	195.61
## serum_sodium	1	0.8869	195	194.72

```
generlinear.valid <- predict(generlinear.model, groups[[2]], type = 'response')
generlinear.valid
```

```
##      1      28      43      47      54      62      70
## 0.80994393 0.21110882 0.19963878 0.33021569 0.23391457 0.22158457 0.48180053
##      97     104     138     140     143     151     153
## 0.60924605 0.23861600 0.77042682 0.18791750 0.16123298 0.40422755 0.06607160
```

```

##           154           169           171           172           179           183           194
## 0.15746251 0.16351454 0.24770291 0.16015766 0.05581348 0.39370902 0.49450530
##           197           205           229           237           256           261           294
## 0.09249842 0.36386362 0.88940940 0.11941313 0.24134564 0.08723207 0.28729340

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - generlinear.valid)^2))/nrow

generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 3.08 1.00 3.06 1.96  total = 10.09
##
## REML score: 106.8318
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29557    0.02659   11.11  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          3.081   3.851  7.276 2.21e-05 ***
## s(creatinine_phosphokinase) 1.000   1.001  1.951  0.16410
## s(ejection_fraction)      3.055   3.786 11.201 8.61e-08 ***
## s(serum_creatinine)       1.957   2.426  5.181  0.00489 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.314  Deviance explained = 34.5%
## -REML = 106.83  Scale est. = 0.14356  n = 203

generaddit.valid <- predict(generaddit.model, groups[[2]], type = 'response')
generaddit.valid

##           1           28           43           47           54           62           70
## 0.89921624 0.15418324 0.15671721 0.43076777 0.12115898 0.23895718 0.53131394

```

```

##           97           104           138           140           143           151           153
## 0.46640046 0.36859528 0.60544061 0.24936128 0.22206952 0.34490302 0.03098249
##           154           169           171           172           179           183           194
## 0.20497952 0.08471568 0.20691201 0.19863545 0.15333221 0.45004439 0.37011826
##           197           205           229           237           256           261           294
## 0.07821592 0.23842276 0.76407467 0.23994242 0.23325784 0.05928992 0.09916006

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr

lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.66201 0.3916084                526.2867 0.3986014                40.02098
## 1 66.47778 0.5000000                719.5500 0.4000000                33.50000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3356643 267715.5                1.192378      137.3566 0.6153846
## 1      0.4000000 262375.1                1.797500      136.0500 0.6166667
##      smoking
## 0 0.3006993
## 1 0.3166667
##
## Coefficients of linear discriminants:
##                                LD1
## age                5.077408e-02
## anaemia            4.159176e-01
## creatinine_phosphokinase 2.572000e-04
## diabetes           2.078107e-01
## ejection_fraction   -5.317875e-02
## high_blood_pressure  1.473631e-01
## platelets          -1.963339e-07
## serum_creatinine     4.715391e-01
## serum_sodium        -4.656113e-02
## sex                -4.022895e-01
## smoking             3.429145e-01

lindiscr.valid <- predict(lindiscr.model, groups[[2]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid

##           1           28           43           47           54           62           70
## 0.73641889 0.26240211 0.25052007 0.25435299 0.29476682 0.23335409 0.41406414
##           97           104           138           140           143           151           153
## 0.53594493 0.26540793 0.67426437 0.13086163 0.21367984 0.42338969 0.06945213
##           154           169           171           172           179           183           194
## 0.20137707 0.22080607 0.27473435 0.18084976 0.04467978 0.44860453 0.52218601
##           197           205           229           237           256           261           294
## 0.11753056 0.36224889 0.88118089 0.07689148 0.23695597 0.06029560 0.33026675

```

```

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[2]]))

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.66201 0.3916084          526.2867 0.3986014          40.02098
## 1 66.47778 0.5000000          719.5500 0.4000000          33.50000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3356643 267715.5      1.192378      137.3566 0.6153846
## 1      0.4000000 262375.1      1.797500      136.0500 0.6166667
##      smoking
## 0 0.3006993
## 1 0.3166667

quaddiscr.valid <- predict(quaddiscr.model, groups[[2]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid

##      1      28      43      47      54      62
## 0.699817532 0.037902507 0.051039787 0.439949438 0.086813058 0.241735441
##      70      97     104     138     140     143
## 0.005188954 0.110984888 0.999998882 0.701665920 0.093806781 0.071614762
##      151     153     154     169     171     172
## 0.203401672 0.007380232 0.130107655 0.059972421 0.044385161 0.994836600
##      179     183     194     197     205     229
## 0.200464311 0.044263113 0.164932571 0.087492687 0.483700388 0.999995299
##      237     256     261     294
## 0.285914631 0.245272441 0.041370619 0.089097982

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[2]]))

mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 75.29 90.20 94.60 98.40 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.21182 ( N = 203 )

```

```
##
## Deviance: 187.627
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[2]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 0.76070578 0.28176722 0.21076945 0.27983225 0.14698836 0.18246390
## [7] 0.60121403 0.48901819 0.13976222 0.52280794 0.16961660 0.07747764
## [13] 0.35274451 0.08642931 0.17350471 0.14822808 0.21997754 0.14795885
## [19] 0.04019477 0.48794204 0.40510486 0.06823513 0.42869359 0.84278888
## [25] 0.10215485 0.18120008 0.11682378 0.32695819

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[2]]))

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

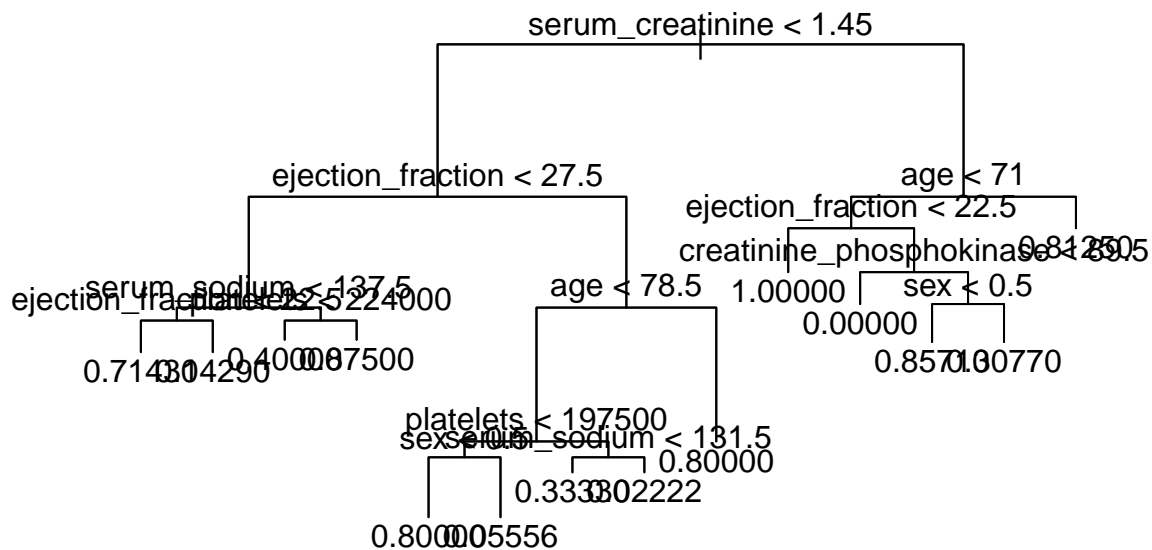
## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
## v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.24138 ( N = 203 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[2]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

##          1          28          43          47          54          62          70
## 0.74006434 0.26208443 0.25009558 0.25396250 0.29475887 0.23278323 0.41532318
##          97          104          138          140          143          151          153
## 0.53844971 0.26511792 0.67774264 0.12970614 0.21295452 0.42474969 0.06834997
##          154          169          171          172          179          183          194
## 0.20056344 0.22013490 0.27453170 0.17990520 0.04375716 0.45023491 0.52456215
##          197          205          229          237          256          261          294
## 0.11635146 0.36294543 0.88414067 0.07575731 0.23641498 0.05924559 0.33062234

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[2]]))

decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```

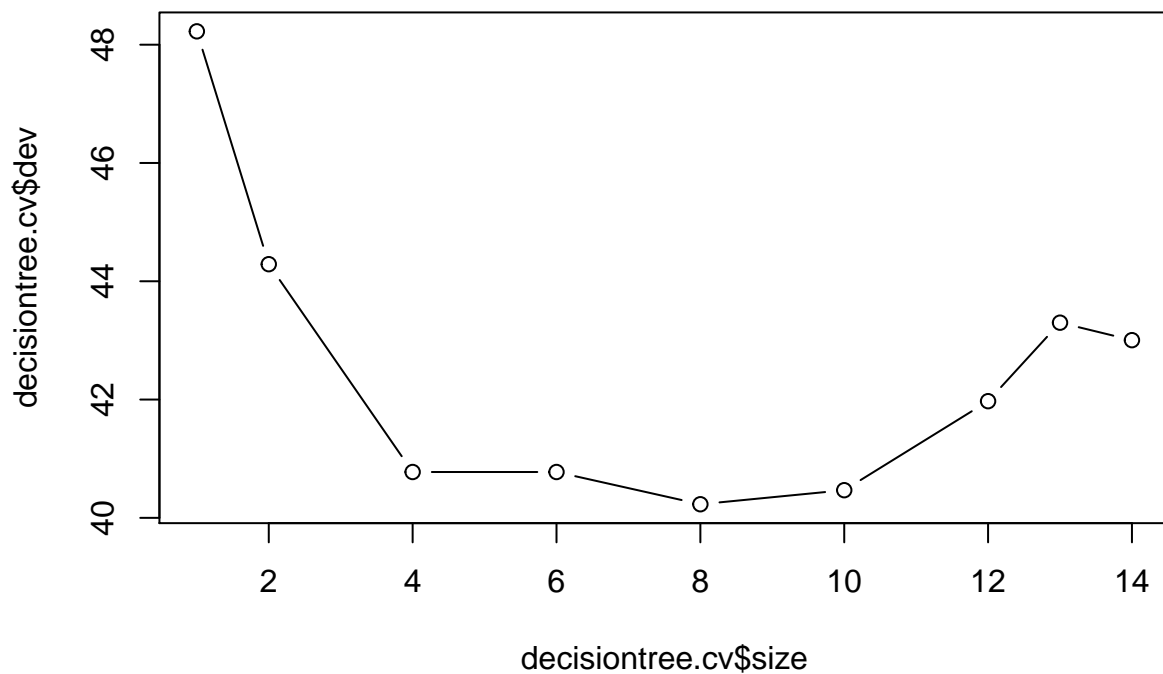


```
decisiontree.valid <- predict(decisiontree.model, groups[[2]], type = 'vector')
decisiontree.valid
```

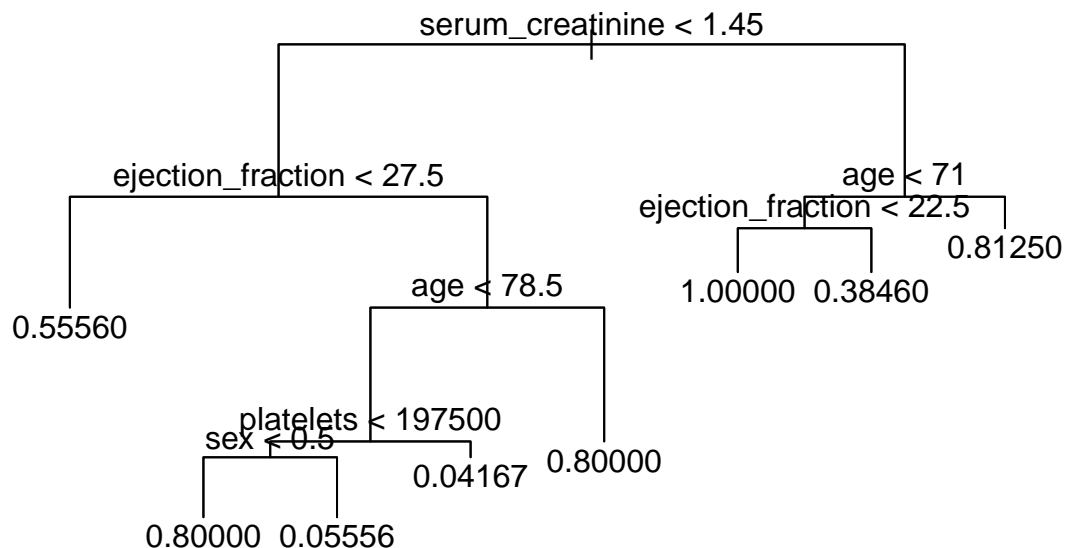
```
##          1          28          43          47          54          62          70
## 0.81250000 0.02222222 0.05555556 0.14285714 0.02222222 0.85714286 0.30769231
##          97          104          138          140          143          151          153
## 0.14285714 0.02222222 0.30769231 0.02222222 0.02222222 0.81250000 0.05555556
##          154          169          171          172          179          183          194
## 0.02222222 0.02222222 0.30769231 0.02222222 0.02222222 0.14285714 0.05555556
##          197          205          229          237          256          261          294
## 0.02222222 0.02222222 0.00000000 0.02222222 0.02222222 0.02222222 0.05555556
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - decisiontree.valid)^2))/n)
```

```
set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[2]], type = 'vector')
prunedectree.valid
```

```
##          1          28          43          47          54          62          70
## 0.81250000 0.04166667 0.05555556 0.55555556 0.04166667 0.38461538 0.38461538
##          97          104          138          140          143          151          153
## 0.55555556 0.04166667 0.38461538 0.04166667 0.04166667 0.81250000 0.05555556
##          154          169          171          172          179          183          194
## 0.04166667 0.04166667 0.38461538 0.04166667 0.04166667 0.55555556 0.05555556
##          197          205          229          237          256          261          294
## 0.04166667 0.04166667 0.38461538 0.04166667 0.04166667 0.04166667 0.05555556
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - prunedectree.valid)^2)/n))
```

```
##3rd fold-out training
```

```
trainset <- data.frame()
for (i in c(1:2,4:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
```



```
##
## Coefficients:
##             (Intercept)                age                anaemia
##             9.674e-01                1.112e-02                3.235e-02
## creatinine_phosphokinase                diabetes                ejection_fraction
##             3.085e-05                1.886e-02                -1.103e-02
##             high_blood_pressure                platelets                serum_creatinine
##             8.187e-02                -4.004e-08                7.746e-02
##             serum_sodium                sex                smoking
##             -7.701e-03                -6.679e-02                2.485e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7930 -0.3148 -0.1125  0.3732  1.0016
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.674e-01  1.057e+00   0.915   0.3615
## age           1.112e-02  2.539e-03   4.380 1.97e-05 ***
## anaemia       3.235e-02  6.309e-02   0.513   0.6087
## creatinine_phosphokinase 3.085e-05  3.060e-05   1.008   0.3146
## diabetes      1.886e-02  6.323e-02   0.298   0.7658
## ejection_fraction -1.103e-02  2.656e-03 -4.152 5.01e-05 ***
## high_blood_pressure  8.187e-02  6.218e-02   1.317   0.1896
## platelets      -4.004e-08  3.433e-07  -0.117   0.9073
## serum_creatinine  7.746e-02  3.077e-02   2.517   0.0127 *
## serum_sodium    -7.701e-03  7.628e-03  -1.010   0.3140
## sex            -6.679e-02  7.408e-02  -0.902   0.3684
## smoking        2.485e-02  7.465e-02   0.333   0.7396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4179 on 186 degrees of freedom
## Multiple R-squared:  0.2372, Adjusted R-squared:  0.1921
## F-statistic: 5.259 on 11 and 186 DF,  p-value: 3.278e-07
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
age	1	4.037	4.0370	23.1163	3.137e-06 ***
anaemia	1	0.008	0.0084	0.0481	0.826603
creatinine_phosphokinase	1	0.228	0.2281	1.3062	0.254546
diabetes	1	0.005	0.0048	0.0273	0.868890
ejection_fraction	1	3.930	3.9297	22.5024	4.169e-06 ***
high_blood_pressure	1	0.267	0.2668	1.5278	0.217999
platelets	1	0.020	0.0204	0.1168	0.732888

```
## serum_creatinine      1  1.308  1.3084  7.4923  0.006798 **
## serum_sodium          1  0.157  0.1572  0.9001  0.343991
## sex                   1  0.123  0.1233  0.7061  0.401825
## smoking               1  0.019  0.0193  0.1108  0.739629
## Residuals            186 32.482  0.1746
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[3]])
simplelinear.valid
```

```
##           8           9           21           40           42           57
## 0.05015620 0.02118427 0.56495730 0.36735070 0.36147278 0.49005753
##           58           85           95           105           106           108
## 0.17773546 0.46904768 0.24608117 0.23220635 0.63049594 0.12207016
##           109           110           113           146           147           155
## 0.28466083 0.04088815 0.27768373 0.11407000 0.17510397 0.43402676
##           157           174           217           243           247           252
## 0.24384561 0.25592215 0.28292866 0.01473032 0.30864729 0.18102938
##           257           258           272           274           277           287
## 0.33069825 0.22822539 0.15917746 -0.10296212 0.35748742 0.23203335
##           290           293           295
## 0.53708255 0.09340678 0.27623030
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##      (Intercept)           age  ejection_fraction  serum_creatinine
##      -0.03777      0.01101         -0.01150         0.08077
```

```
summary(revisedlinear.model)
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.7938 -0.3005 -0.1210  0.3919  1.0143
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.037771   0.171180  -0.221  0.82560
## age           0.011005   0.002454   4.484 1.25e-05 ***
## ejection_fraction -0.011505  0.002522  -4.562 8.96e-06 ***
## serum_creatinine  0.080774   0.029764   2.714  0.00725 **
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4142 on 194 degrees of freedom
## Multiple R-squared:  0.2186, Adjusted R-squared:  0.2065
## F-statistic: 18.09 on 3 and 194 DF,  p-value: 2.15e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.037   4.0370   23.536 2.512e-06 ***
## ejection_fraction 1  4.010   4.0101   23.379 2.701e-06 ***
## serum_creatinine 1  1.263   1.2633    7.365 0.007249 **
## Residuals     194 33.276   0.1715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[3]])
revisedlinear.valid
```

```
##           8           9           21           40           42           57           58
## 0.02112125 0.05093425 0.49496956 0.42769630 0.26428685 0.54803274 0.21082933
##           85           95           105           106           108           109           110
## 0.40470476 0.21990535 0.23309751 0.54679343 0.12750372 0.30724211 0.15981330
##           113          146           147           155           157           174           217
## 0.35412019 0.22389988 0.24591075 0.33953506 0.24492875 0.34702480 0.23006835
##           243          247           252           257           258           272           274
## 0.08863133 0.36876040 0.22948070 0.37771735 0.24413754 0.13601258 0.02080886
##           277          287           290           293           295
## 0.41879443 0.35720443 0.58823411 0.17810492 0.29623668
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##         ejection_fraction + high_blood_pressure + serum_creatinine +
##         serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##              (Intercept)              age              anaemia
##              0.949402              0.065301              0.311443
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##              0.000177              -0.071214              0.561488
##              serum_creatinine      serum_sodium
##              0.515844              -0.032717
##
## Degrees of Freedom: 197 Total (i.e. Null);  190 Residual
## Null Deviance:      246.1
## Residual Deviance: 193.4      AIC: 209.4
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9544  -0.7622  -0.4382   0.8507   2.5898
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.9494019   5.9395548   0.160   0.8730
## age            0.0653011   0.0159704   4.089 4.33e-05 ***
## anaemia        0.3114431   0.3726441   0.836   0.4033
## creatinine_phosphokinase 0.0001770   0.0001688   1.049   0.2944
## ejection_fraction -0.0712145   0.0180682  -3.941 8.10e-05 ***
## high_blood_pressure  0.5614878   0.3662246   1.533   0.1252
## serum_creatinine   0.5158444   0.2047519   2.519   0.0118 *
## serum_sodium     -0.0327170   0.0436223  -0.750   0.4533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.15  on 197  degrees of freedom
## Residual deviance: 193.40  on 190  degrees of freedom
## AIC: 209.4
##
## Number of Fisher Scoring iterations: 5
anova(generlinear.model)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                      197      246.15
## age                      1  19.2072      196      226.94
## anaemia                   1   0.0604      195      226.88
## creatinine_phosphokinase  1   1.1419      194      225.74
## ejection_fraction         1  22.1138      193      203.62
## high_blood_pressure        1   1.5039      192      202.12
## serum_creatinine           1   8.1487      191      193.97
## serum_sodium               1   0.5686      190      193.40

generlinear.valid <- predict(generlinear.model, groups[[3]], type = 'response')
generlinear.valid
```

```

##           8           9           21           40           42           57           58
## 0.05970918 0.04115590 0.61871844 0.29000272 0.23702791 0.55737648 0.14583235
##           85           95           105           106           108           109           110
## 0.46183457 0.15881992 0.16596759 0.69160242 0.11797544 0.22079510 0.07391223
##           113           146           147           155           157           174           217
## 0.24513030 0.10515250 0.13511909 0.40055674 0.17464304 0.26638822 0.19547190
##           243           247           252           257           258           272           274
## 0.05747470 0.30366542 0.10753628 0.24504719 0.19524945 0.08427531 0.03342367
##           277           287           290           293           295
## 0.28069275 0.21692022 0.56081083 0.08378478 0.22323708

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - generlinear.valid)^2))/nrow

generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 3.01 1.06 3.55 2.86 total = 11.49
##
## REML score: 107.5867

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.31313    0.02712   11.55  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df    F  p-value
## s(age)          3.014   3.778 6.980 4.38e-05 ***
## s(creatinine_phosphokinase) 1.056   1.109 1.275  0.2797
## s(ejection_fraction)      3.553   4.377 9.344 3.06e-07 ***
## s(serum_creatinine)       2.862   3.550 3.352  0.0139 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.327   Deviance explained = 36.2%

```

```
## -REML = 107.59  Scale est. = 0.14557  n = 198
```

```
generaddit.valid <- predict(generaddit.model, groups[[3]], type = 'response')
generaddit.valid
```

```
##           8           9           21           40           42           57
## 0.08113186 0.11661356 0.44916057 0.33162016 0.22701214 0.40099950
##           58           85           95           105           106           108
## 0.02335230 0.37392859 0.02162261 0.37550116 0.43742578 0.09150559
##           109           110           113           146           147           155
## 0.13754940 0.13067118 0.50238577 0.12191370 0.12439782 0.07457335
##           157           174           217           243           247           252
## 0.18128392 0.56275775 0.23432649 0.03064379 0.44980432 0.06695277
##           257           258           272           274           277           287
## 0.25303430 0.08481497 0.08645149 -0.03549458 0.19836457 0.24385225
##           290           293           295
## 0.69277640 0.08944110 0.10840521
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~ . - time, data = trainset)
lindiscr.model
```

```
## Call:
```

```
## lda(DEATH_EVENT ~ . - time, data = trainset)
```

```
##
```

```
## Prior probabilities of groups:
```

```
##           0           1
## 0.6868687 0.3131313
```

```
##
```

```
## Group means:
```

```
##           age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.81373 0.4117647                607.5515 0.4264706                40.25735
## 1 67.02689 0.4516129                701.6129 0.4032258                33.17742
## high_blood_pressure platelets serum_creatinine serum_sodium sex
## 0           0.3455882 261789.7                1.217279        137.0588 0.6470588
## 1           0.4516129 256050.8                1.796452        135.5645 0.6612903
```

```
## smoking
```

```
## 0 0.3161765
```

```
## 1 0.3064516
```

```
##
```

```
## Coefficients of linear discriminants:
```

```
##                               LD1
## age                5.608214e-02
## anaemia            1.631691e-01
## creatinine_phosphokinase 1.555866e-04
## diabetes           9.513530e-02
## ejection_fraction  -5.561200e-02
## high_blood_pressure  4.128612e-01
## platelets          -2.019486e-07
## serum_creatinine    3.906619e-01
## serum_sodium        -3.883829e-02
## sex                -3.368399e-01
## smoking            1.253098e-01
```

```

lindiscr.valid <- predict(lindiscr.model, groups[[3]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid

```

```

##           8           9           21           40           42           57           58
## 0.06661808 0.05653730 0.61455295 0.32610068 0.31835493 0.50363574 0.13353938
##           85           95          105          106          108          109          110
## 0.47197164 0.18883167 0.17634045 0.70306873 0.09922131 0.22708785 0.06322388
##          113          146          147          155          157          174          217
## 0.21978349 0.09498951 0.13171280 0.41981116 0.18677410 0.19809574 0.22525856
##          243          247          252          257          258          272          274
## 0.05449555 0.25348838 0.13585592 0.27948113 0.17287857 0.12110580 0.02755150
##          277          287          290          293          295
## 0.31315939 0.17618888 0.57402541 0.08479878 0.21828331

```

```

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[3]]))

```

```

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

```

```

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6868687 0.3131313
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.81373 0.4117647                607.5515 0.4264706          40.25735
## 1 67.02689 0.4516129                701.6129 0.4032258          33.17742
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0           0.3455882 261789.7          1.217279      137.0588 0.6470588
## 1           0.4516129 256050.8          1.796452      135.5645 0.6612903
##      smoking
## 0 0.3161765
## 1 0.3064516

```

```

quaddiscr.valid <- predict(quaddiscr.model, groups[[3]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid

```

```

##           8           9           21           40           42           57
## 0.0007273306 0.0056881965 0.3584796096 0.8602257620 0.4727883969 0.3603422489
##           58           85           95          105          106          108
## 0.0147642896 0.2632172291 0.0846651438 0.2056247613 0.7242354013 0.0464999770
##          109          110          113          146          147          155
## 0.0638072816 0.0487174462 0.0593421225 0.0318038251 0.0656524893 0.2071642852
##          157          174          217          243          247          252
## 0.0456260477 0.2617543482 0.0193213827 0.0070718957 0.2275982995 0.1572449406
##          257          258          272          274          277          287
## 0.0511621771 0.0449575681 0.0984239255 0.0225569761 0.0702828157 0.1062777288
##          290          293          295
## 0.8094222167 0.0354909579 0.1994234486

```

```

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[3]]))

mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 77.68 89.97 95.48 99.40 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.22727 ( N = 198 )
##
## Deviance: 174.662

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[3]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 6.115334e-02 4.568870e-02 4.596723e-01 7.174523e-01 3.096301e-01
## [6] 4.359974e-01 1.190677e-01 3.090155e-01 1.393961e-01 1.024025e-01
## [11] 7.763940e-03 7.062058e-02 2.802817e-01 1.348101e-08 1.257173e-01
## [16] 1.033331e-01 1.581057e-01 3.929687e-01 8.219584e-02 5.961269e-01
## [21] 1.814803e-01 6.000655e-02 1.022659e-01 1.110705e-01 1.584560e-01
## [26] 1.596174e-01 1.251567e-01 6.476694e-02 2.024547e-01 8.055239e-02
## [31] 8.427295e-01 7.600113e-02 1.423985e-01

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[3]]))

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.27273 ( N = 198 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[3]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

##           8           9          21          40          42          57          58
## 0.06545114 0.05544294 0.61757475 0.32623445 0.31840847 0.50567654 0.13226412
##          85          95         105         106         108         109         110

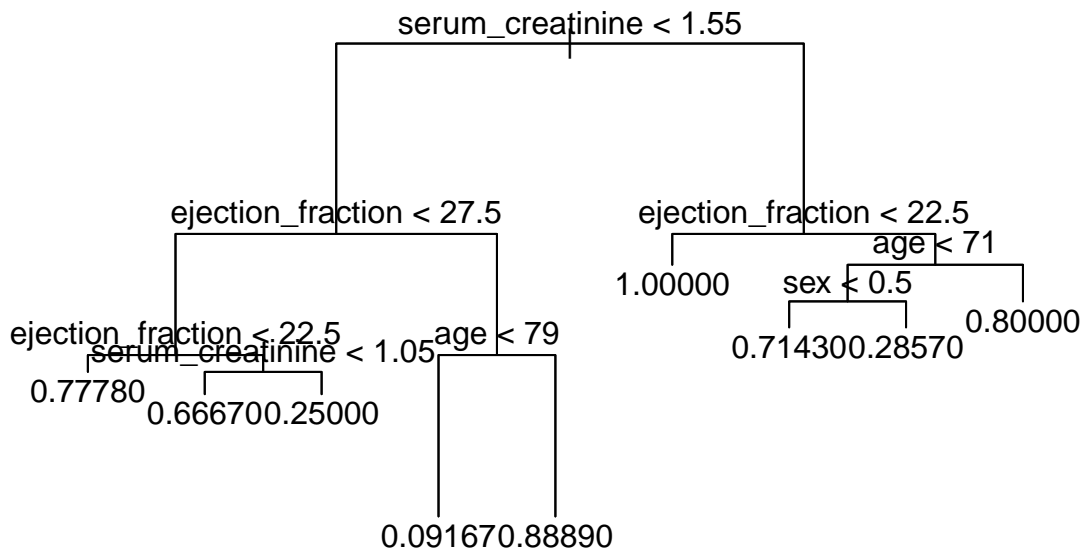
```



```
## 0.47368414 0.18778342 0.17522305 0.70656622 0.09793341 0.22630201 0.06207920
##      113      146      147      155      157      174      217
## 0.21894229 0.09370899 0.13043397 0.42095978 0.18571384 0.19710443 0.22445865
##      243      247      252      257      258      272      274
## 0.05341838 0.25292000 0.13458559 0.27914936 0.17174367 0.11981229 0.02680204
##      277      287      290      293      295
## 0.31315967 0.17507070 0.57672740 0.08354548 0.21743102

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[3]]))

decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```

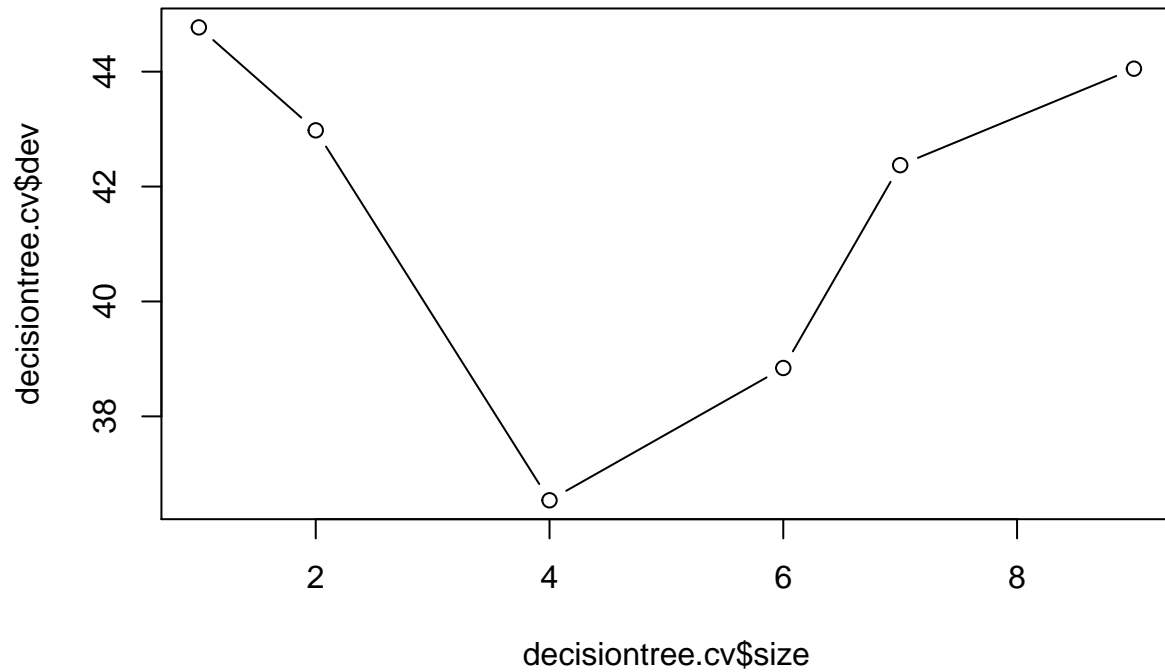


```
decisiontree.valid <- predict(decisiontree.model, groups[[3]], type = 'vector')
decisiontree.valid
```

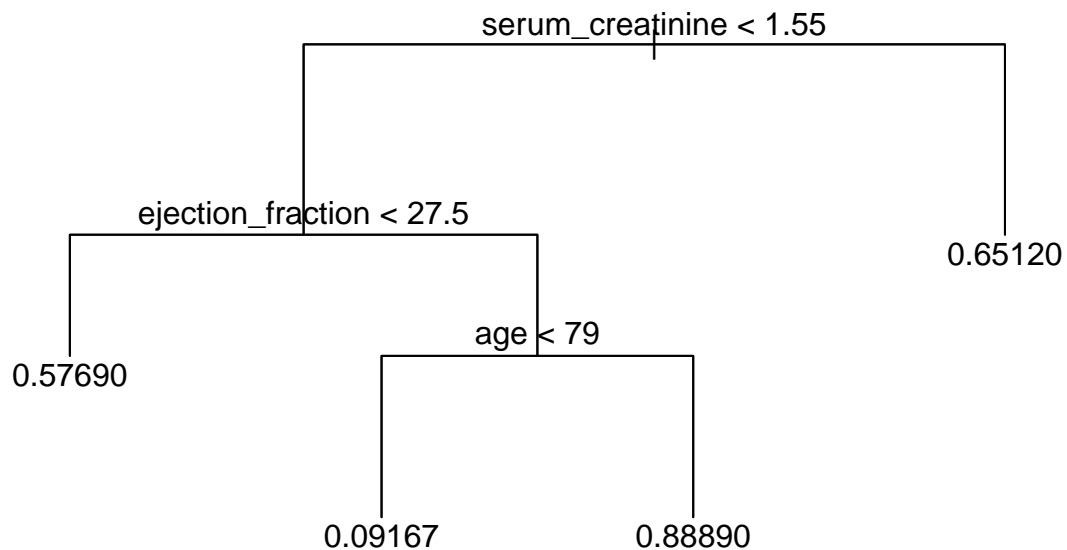
```
##      8      9      21      40      42      57      58
## 0.09166667 0.09166667 0.25000000 0.71428571 0.09166667 0.28571429 0.09166667
##      85      95      105      106      108      109      110
## 0.66666667 0.09166667 0.71428571 0.80000000 0.09166667 0.09166667 0.09166667
##      113      146      147      155      157      174      217
## 0.28571429 0.09166667 0.09166667 0.09166667 0.09166667 0.77777778 0.09166667
##      243      247      252      257      258      272      274
## 0.09166667 0.25000000 0.09166667 0.71428571 0.09166667 0.09166667 0.09166667
##      277      287      290      293      295
## 0.09166667 0.28571429 0.88888889 0.09166667 0.09166667
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - decisiontree.valid)^2))/n)

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree.cv$dev)])
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[3]], type = 'vector')
prunedectree.valid
```

```
##           8           9           21           40           42           57           58
## 0.09166667 0.09166667 0.57692308 0.65116279 0.09166667 0.65116279 0.09166667
##           85           95           105           106           108           109           110
## 0.57692308 0.09166667 0.65116279 0.65116279 0.09166667 0.09166667 0.09166667
##           113          146          147          155          157          174          217
## 0.65116279 0.09166667 0.09166667 0.09166667 0.09166667 0.57692308 0.09166667
##           243          247          252          257          258          272          274
## 0.09166667 0.57692308 0.09166667 0.65116279 0.09166667 0.09166667 0.09166667
##           277          287          290          293          295
## 0.09166667 0.65116279 0.88888889 0.09166667 0.09166667
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##4th fold-out training
```

```
trainset <- data.frame()
for (i in c(1:3,5:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             2.469e+00             9.343e-03             8.862e-02
## creatinine_phosphokinase             diabetes             ejection_fraction
##             3.376e-05             6.357e-02             -9.787e-03
##             high_blood_pressure             platelets             serum_creatinine
##             9.328e-02             5.290e-08             5.250e-02
##             serum_sodium                sex                smoking
##             -1.852e-02             -1.068e-01             9.227e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7003 -0.2881 -0.1276  0.3437  1.0183
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.469e+00  1.052e+00   2.346 0.020015 *
## age          9.343e-03  2.639e-03   3.540 0.000505 ***
## anaemia      8.862e-02  6.263e-02   1.415 0.158751
## creatinine_phosphokinase 3.376e-05  3.110e-05   1.086 0.278983
## diabetes     6.357e-02  6.296e-02   1.010 0.313927
## ejection_fraction -9.787e-03  2.723e-03  -3.594 0.000416 ***
## high_blood_pressure  9.328e-02  6.313e-02   1.478 0.141170
## platelets     5.290e-08  3.070e-07   0.172 0.863387
## serum_creatinine  5.250e-02  3.434e-02   1.529 0.127968
## serum_sodium  -1.852e-02  7.546e-03  -2.455 0.015002 *
## sex          -1.068e-01  7.400e-02  -1.444 0.150510
## smoking       9.227e-02  7.220e-02   1.278 0.202815
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4163 on 188 degrees of freedom
## Multiple R-squared:  0.2168, Adjusted R-squared:  0.1709
## F-statistic:  4.73 on 11 and 188 DF,  p-value: 2.147e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  2.612  2.61173  15.0714 0.0001432 ***
## anaemia       1  0.303  0.30289   1.7479 0.1877514
## creatinine_phosphokinase 1  0.173  0.17264   0.9962 0.3195040
## diabetes      1  0.180  0.17984   1.0378 0.3096481
## ejection_fraction 1  3.128  3.12772  18.0490 3.384e-05 ***
```

```
## high_blood_pressure      1  0.362 0.36227  2.0905 0.1498817
## platelets                1  0.052 0.05175  0.2986 0.5853943
## serum_creatinine         1  0.835 0.83475  4.8170 0.0294050 *
## serum_sodium             1  0.931 0.93053  5.3698 0.0215660 *
## sex                     1  0.159 0.15918  0.9186 0.3390739
## smoking                  1  0.283 0.28305  1.6334 0.2028153
## Residuals               188 32.579 0.17329
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[4]])
simplelinear.valid
```

```
##          32          35          36          44          51          53
## 0.44795850 0.10911872 0.46100814 0.28836720 0.45905772 0.28654618
##          56          59          78          81          93          99
## 0.74139861 0.39168204 -0.08781929 0.31923865 -0.16816816 0.57757369
##          115         117         118         120         144         166
## 0.54842985 0.22279858 0.40641067 0.54792496 0.27955354 0.60514854
##          196         199         201         202         206         209
## 0.16410292 0.59047266 0.16135887 -0.08428228 0.20585025 0.32278835
##          238         240         246         264         271         275
## 0.35727386 0.16057856 0.26560193 0.14463709 0.43814056 0.40481890
##          286
## 0.22241549
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##      (Intercept)          age ejection_fraction  serum_creatinine
##      0.009103      0.009590      -0.010422      0.069668
```

```
summary(revisedlinear.model)
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.7202 -0.2991 -0.1461  0.4309  0.9907
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.009103   0.177854   0.051 0.959235
## age           0.009590   0.002607   3.678 0.000303 ***
## ejection_fraction -0.010422 0.002707  -3.851 0.000160 ***
```

```
## serum_creatinine 0.069668 0.033335 2.090 0.037917 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4224 on 196 degrees of freedom
## Multiple R-squared: 0.1591, Adjusted R-squared: 0.1462
## F-statistic: 12.36 on 3 and 196 DF, p-value: 1.938e-07
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1  2.612   2.6117 14.6350 0.0001753 ***
## ejection_fraction 1  3.226   3.2260 18.0772 3.281e-05 ***
## serum_creatinine 1  0.779   0.7795  4.3677 0.0379168 *
## Residuals    196 34.978   0.1785
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[4]])
revisedlinear.valid
```

```
##           32           35           36           44           51           53
## 0.564283528 0.181038627 0.549893400 0.248169065 0.470351563 0.412099581
##           56           59           78           81           93           99
## 0.746842254 0.347215678 0.078617795 0.381973324 -0.131209755 0.407564572
##          115          117          118          120          144          166
## 0.251238923 0.007970949 0.421891389 0.565314528 0.239928721 0.470850351
##          196          199          201          202          206          209
## 0.403961968 0.349838985 0.193066786 -0.114979726 0.089296237 0.237305413
##          238          240          246          264          271          275
## 0.451356649 0.149786709 0.295605915 0.105591715 0.229882040 0.341522512
##          286
## 0.203288610
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model
```

```
##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
## ejection_fraction + high_blood_pressure + serum_creatinine +
## serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
## (Intercept)          age          anaemia
## 10.2881739       0.0512064       0.5257437
## creatinine_phosphokinase ejection_fraction high_blood_pressure
## 0.0001655       -0.0583792       0.5199828
## serum_creatinine      serum_sodium
## 0.2921661       -0.0962177
##
## Degrees of Freedom: 199 Total (i.e. Null); 192 Residual
```

```
## Null Deviance:      242.6
## Residual Deviance: 199.7      AIC: 215.7

summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6743  -0.7628  -0.4948   0.8324   2.4661
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    10.2881739   6.3645782   1.616  0.10599
## age              0.0512064   0.0156282   3.277  0.00105 **
## anaemia          0.5257437   0.3655187   1.438  0.15033
## creatinine_phosphokinase 0.0001655  0.0001712   0.967  0.33369
## ejection_fraction -0.0583792  0.0178972  -3.262  0.00111 **
## high_blood_pressure  0.5199828  0.3572359   1.456  0.14551
## serum_creatinine   0.2921661  0.2276989   1.283  0.19945
## serum_sodium     -0.0962177  0.0460996  -2.087  0.03687 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 242.63  on 199  degrees of freedom
## Residual deviance: 199.69  on 192  degrees of freedom
## AIC: 215.69
##
## Number of Fisher Scoring iterations: 4

anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			199	242.63
## age	1	12.6604	198	229.97
## anaemia	1	1.4938	197	228.47
## creatinine_phosphokinase	1	0.8243	196	227.65
## ejection_fraction	1	16.9990	195	210.65
## high_blood_pressure	1	1.8367	194	208.81
## serum_creatinine	1	4.5305	193	204.28

```
## serum_sodium          1    4.5938      192      199.69
generlinear.valid <- predict(generlinear.model, groups[[4]], type = 'response')
generlinear.valid

##          32          35          36          44          51          53          56
## 0.54815696 0.12508456 0.50097026 0.26817049 0.61445540 0.15923366 0.86670708
##          59          78          81          93          99         115         117
## 0.38131958 0.04745668 0.26114097 0.02178197 0.53529160 0.60422007 0.12328691
##         118         120         144         166         196         199         201
## 0.34017483 0.55263800 0.23730415 0.55110136 0.22675331 0.55526288 0.21891503
##         202         206         209         238         240         246         264
## 0.04177564 0.11726135 0.20654558 0.45252099 0.14135657 0.26106330 0.08634342
##         271         275         286
## 0.34645797 0.32911415 0.21155984

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - generlinear.valid)^2))/nrow(groups[[4]]))

generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(serum_creatinine))
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 3.13 1.27 3.37 3.33 total = 12.1
##
## REML score: 110.1503

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29500    0.02711   10.88  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)         3.128   3.917  4.743  0.00117 **
## s(creatinine_phosphokinase) 1.273   1.493  0.750  0.53988
## s(ejection_fraction)     3.371   4.173  9.317  4.63e-07 ***
## s(serum_creatinine)     3.332   4.077  3.915  0.00426 **
```



```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.297   Deviance explained = 33.6%
## -REML = 110.15   Scale est. = 0.14697   n = 200

generaddit.valid <- predict(generaddit.model, groups[[4]], type = 'response')
generaddit.valid

##          32          35          36          44          51          53          56
## 0.79181357 0.12762608 0.34299902 0.19338329 0.40615826 0.29135849 1.20426615
##          59          78          81          93          99         115         117
## 0.65207022 0.08111955 0.29486866 0.16260306 0.45316368 0.14870328 0.07316041
##         118         120         144         166         196         199         201
## 0.82425459 0.73941319 0.04773536 0.42271253 0.49294036 0.62413834 0.04428040
##         202         206         209         238         240         246         264
## 0.13695213 0.12314455 0.11470894 0.29927327 0.18190019 0.18859492 0.16751382
##         271         275         286
## 0.34768308 0.21410727 0.15316112

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr

lindiscr.model <- lda(DEATH_EVENT ~ . - time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.705 0.295
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.62175 0.3546099                591.8298 0.3758865          39.06383
## 1 65.12995 0.4745763                657.2881 0.4237288          32.91525
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3262411 265023.6          1.204184    137.5816 0.6524823
## 1          0.4576271 265216.8          1.672034    135.5424 0.6101695
##      smoking
## 0 0.3475177
## 1 0.3559322
##
## Coefficients of linear discriminants:
##                                LD1
## age                4.947082e-02
## anaemia            4.692593e-01
## creatinine_phosphokinase 1.787862e-04
## diabetes           3.366242e-01
## ejection_fraction   -5.182136e-02
## high_blood_pressure  4.939434e-01
## platelets           2.801214e-07
## serum_creatinine     2.780127e-01
## serum_sodium        -9.808889e-02
## sex                 -5.656914e-01
## smoking             4.885990e-01

```

```

lindiscr.valid <- predict(lindiscr.model, groups[[4]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid

```

```

##          32          35          36          44          51          53          56
## 0.44727886 0.09355310 0.46695801 0.23476408 0.46400867 0.23278166 0.82803147
##          59          78          81          93          99          115          117
## 0.36500710 0.03023911 0.27012691 0.01877563 0.64016438 0.59843611 0.17078093
##          118          120          144          166          196          199          201
## 0.38599414 0.59769850 0.22527802 0.67779711 0.12599637 0.65801761 0.12417128
##          202          206          209          238          240          246          264
## 0.03087590 0.15668437 0.27440136 0.31803517 0.12365645 0.21082475 0.11353436
##          271          275          286
## 0.43257797 0.38370388 0.17045147

```

```

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[4]]))

```

```

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

```

```

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.705 0.295
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.62175 0.3546099                591.8298 0.3758865                39.06383
## 1 65.12995 0.4745763                657.2881 0.4237288                32.91525
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3262411 265023.6                1.204184    137.5816 0.6524823
## 1      0.4576271 265216.8                1.672034    135.5424 0.6101695
##      smoking
## 0 0.3475177
## 1 0.3559322

```

```

quaddiscr.valid <- predict(quaddiscr.model, groups[[4]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid

```

```

##          32          35          36          44          51          53
## 0.810922684 0.015250726 0.462006437 0.050364955 0.135843920 1.000000000
##          56          59          78          81          93          99
## 0.975443243 0.275570001 0.015248647 0.067426352 0.008460296 0.525842157
##          115          117          118          120          144          166
## 0.770568364 0.202399766 0.931530254 0.130967963 0.060317945 0.114837361
##          196          199          201          202          206          209
## 0.259384620 0.774015682 0.064202319 0.002607573 0.072229409 0.494913552
##          238          240          246          264          271          275
## 0.470438870 0.005135785 0.039244575 0.320457549 0.571109225 0.105695163
##          286
## 0.027975503

```

```

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[4]]))

mixeddiscr.model <- mda(DEATH_EVENT ~ . - time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 74.89 88.05 95.20 98.79 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.195 ( N = 200 )
##
## Deviance: 185.311

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[4]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 0.36240303 0.09461318 0.27849212 0.34573618 0.36399551 0.99999991
## [7] 0.84913312 0.34393620 0.02157468 0.15720632 0.01336971 0.56940116
## [13] 0.73625904 0.19823440 0.97218111 0.49437043 0.19945213 0.68896180
## [19] 0.08732830 0.59837060 0.23590292 0.03059131 0.09368552 0.14883482
## [25] 0.38686091 0.08506849 0.18313336 0.10499503 0.39178218 0.29475788
## [31] 0.15228506

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[4]]))

flexdiscr.model <- fda(DEATH_EVENT ~ . - time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.26 ( N = 200 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[4]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

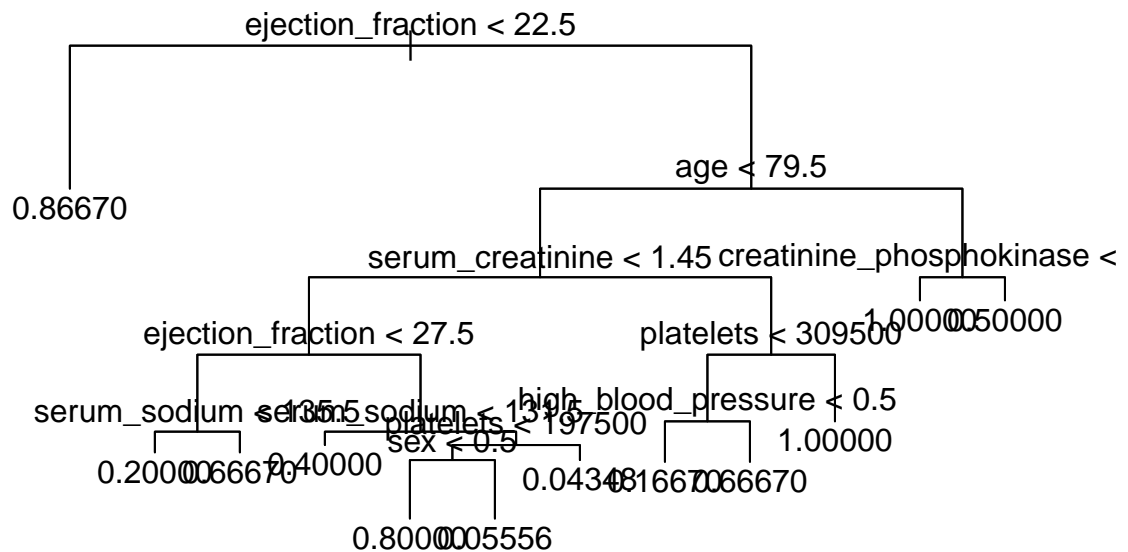
##      32      35      36      44      51      53      56
## 0.44892645 0.09236095 0.46881612 0.23420132 0.46583551 0.23220228 0.83151689
##      59      78      81      93      99     115     117
## 0.36575080 0.02947935 0.26988251 0.01821006 0.64352503 0.60151537 0.16976930

```

```
##          118          120          144          166          196          199          201
## 0.38696903 0.60077222 0.22463707 0.68134908 0.12481582 0.66147708 0.12298725
##          202          206          209          238          240          246          264
## 0.03010698 0.15560380 0.27419791 0.31827271 0.12247149 0.21007159 0.11233633
##          271          275          286
## 0.43406588 0.38465348 0.16943808
```

```
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[4]]))
```

```
decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[4]], type = 'vector')
decisiontree.valid
```

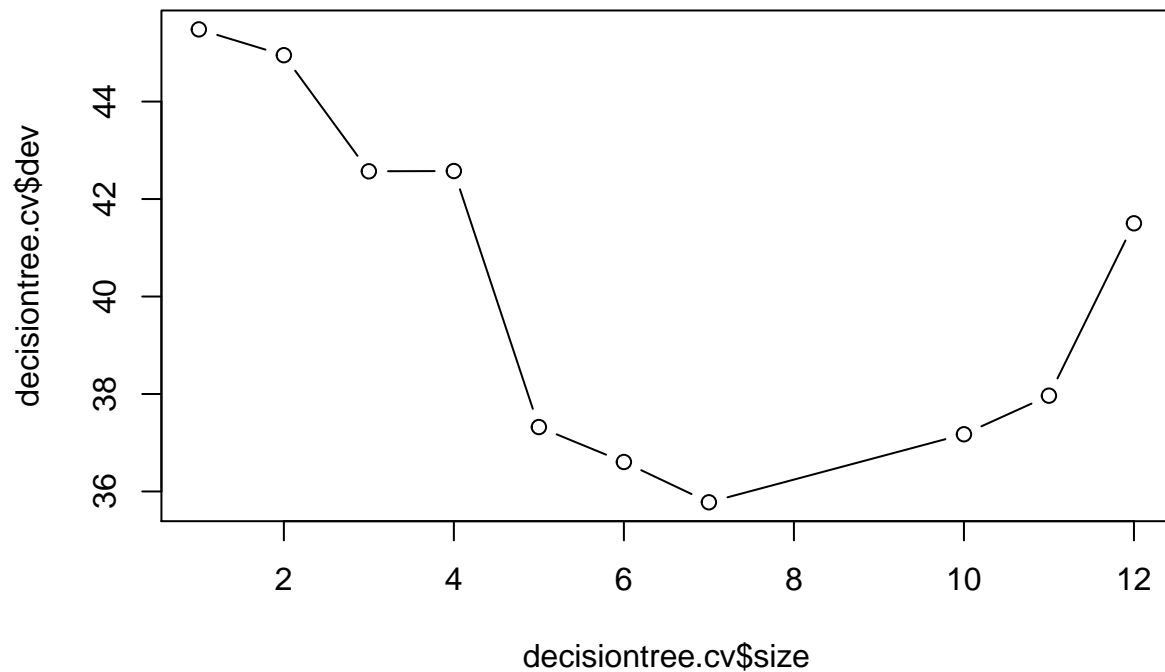
```
##          32          35          36          44          51          53          56
## 1.00000000 0.05555556 0.16666667 0.04347826 0.66666667 0.16666667 0.50000000
##          59          78          81          93          99          115          117
## 0.86666667 0.04347826 0.16666667 0.04347826 0.66666667 0.40000000 0.04347826
##          118          120          144          166          196          199          201
## 1.00000000 0.50000000 0.04347826 0.50000000 0.16666667 0.86666667 0.05555556
##          202          206          209          238          240          246          264
## 0.04347826 0.04347826 0.04347826 0.05555556 0.04347826 0.04347826 0.04347826
##          271          275          286
## 0.66666667 0.05555556 0.04347826
```

```

decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - decisiontree.valid)^2))/n)

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

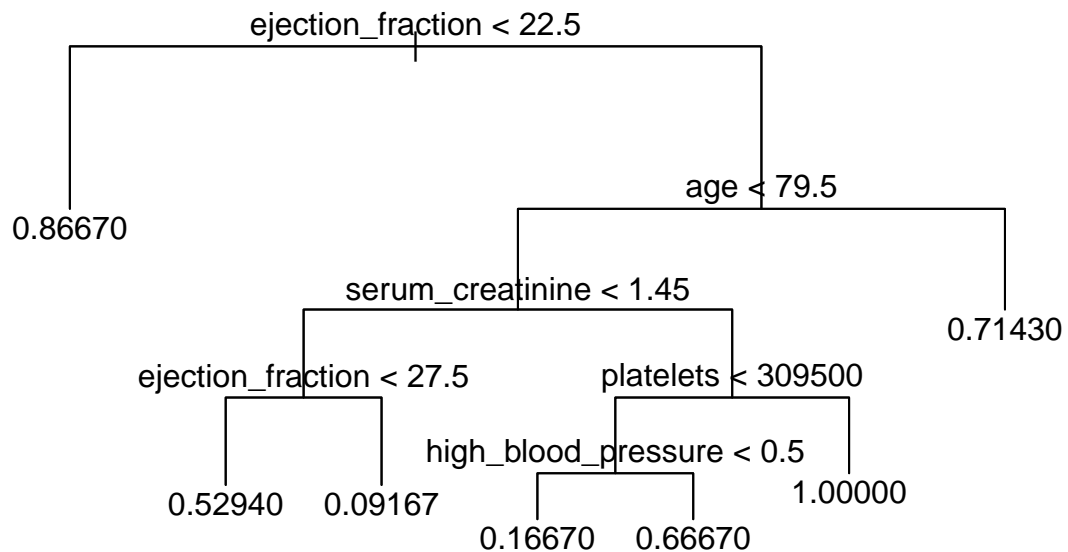
```



```

prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree.cv$dev)])
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)

```



```
prunedectree.valid <- predict(prunedectree.model, groups[[4]], type = 'vector')
prunedectree.valid
```

```
##      32      35      36      44      51      53      56
## 0.71428571 0.09166667 0.16666667 0.09166667 0.52941176 0.16666667 0.71428571
##      59      78      81      93      99      115      117
## 0.86666667 0.09166667 0.16666667 0.09166667 0.52941176 0.09166667 0.09166667
##      118      120      144      166      196      199      201
## 0.71428571 0.71428571 0.09166667 0.71428571 0.16666667 0.86666667 0.09166667
##      202      206      209      238      240      246      264
## 0.09166667 0.09166667 0.09166667 0.09166667 0.09166667 0.09166667 0.09166667
##      271      275      286
## 0.66666667 0.09166667 0.09166667
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##5th fold-out training
trainset <- data.frame()
for (i in c(1:4,6:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.094e+00                1.205e-02                2.001e-02
## creatinine_phosphokinase                diabetes                ejection_fraction
##             3.503e-05                5.412e-02                -9.290e-03
##             high_blood_pressure                platelets                serum_creatinine
##             1.084e-01                4.238e-08                8.266e-02
##             serum_sodium                sex                smoking
##             -9.853e-03                -9.747e-02                6.579e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7899 -0.3076 -0.1343  0.3461  1.0114
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.094e+00  1.069e+00   1.023 0.307376
## age            1.205e-02  2.486e-03   4.848 2.55e-06 ***
## anaemia        2.001e-02  6.093e-02   0.328 0.742957
## creatinine_phosphokinase 3.503e-05  3.014e-05   1.162 0.246518
## diabetes       5.412e-02  6.035e-02   0.897 0.370932
## ejection_fraction -9.290e-03  2.605e-03  -3.566 0.000456 ***
## high_blood_pressure  1.084e-01  6.139e-02   1.766 0.078962 .
## platelets       4.238e-08  2.956e-07   0.143 0.886147
## serum_creatinine  8.266e-02  3.059e-02   2.702 0.007493 **
## serum_sodium    -9.853e-03  7.698e-03  -1.280 0.202078
## sex            -9.747e-02  7.210e-02  -1.352 0.177990
## smoking         6.579e-02  7.102e-02   0.926 0.355388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4137 on 194 degrees of freedom
## Multiple R-squared:  0.2409, Adjusted R-squared:  0.1978
## F-statistic: 5.596 on 11 and 194 DF,  p-value: 8.922e-08
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
age	1	4.799	4.7993	28.0446	3.196e-07 ***
anaemia	1	0.006	0.0061	0.0355	0.8506784
creatinine_phosphokinase	1	0.268	0.2676	1.5638	0.2126160
diabetes	1	0.096	0.0961	0.5617	0.4544806
ejection_fraction	1	2.687	2.6871	15.7023	0.0001041 ***

```
## high_blood_pressure      1  0.591  0.5908  3.4525 0.0646723 .
## platelets                1  0.063  0.0626  0.3658 0.5459839
## serum_creatinine         1  1.484  1.4840  8.6716 0.0036272 **
## serum_sodium             1  0.207  0.2071  1.2100 0.2726937
## sex                     1  0.186  0.1860  1.0868 0.2984824
## smoking                 1  0.147  0.1469  0.8582 0.3553883
## Residuals               194 33.199  0.1711
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[5]])
simplelinear.valid
```

```
##          5          12          14          18          39          63
## 0.86576062 0.39983405 0.13349527 0.25132585 0.44163674 0.13629597
##          82          90          102          107          119          124
## 0.18488759 0.26378869 0.37039596 0.03661445 0.21041476 0.33866056
##          125          161          164          181          189          204
## 0.36352404 0.34080690 0.20610441 0.01828868 0.39290348 0.62230211
##          230          235          244          249          254          255
## 0.52775209 0.07773723 0.46694684 -0.06231158 0.58706652 -0.05348573
##          296
## 0.20899302
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##      (Intercept)           age  ejection_fraction  serum_creatinine
##      -0.157207       0.011563       -0.009596         0.087479
```

```
summary(revisedlinear.model)
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.7919 -0.2963 -0.1409  0.3985  1.0064
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.157207   0.167861  -0.937 0.350116
## age           0.011563   0.002414   4.790 3.22e-06 ***
## ejection_fraction -0.009596 0.002533  -3.789 0.000199 ***
## serum_creatinine  0.087479  0.030111   2.905 0.004078 **
```

```
## ---
```



```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4147 on 202 degrees of freedom
## Multiple R-squared:  0.2055, Adjusted R-squared:  0.1937
## F-statistic: 17.42 on 3 and 202 DF,  p-value: 4.252e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1  4.799   4.7993 27.9024 3.289e-07 ***
## ejection_fraction  1  2.737   2.7372 15.9138 9.267e-05 ***
## serum_creatinine  1  1.452   1.4518  8.4405 0.004078 **
## Residuals      202 34.745   0.1720
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[5]])
```

```
revisedlinear.valid
```

```
##           5           12           14           18           39           63
## 0.63865556 0.39852189 0.15250873 0.29876270 0.44988505 0.23911293
##           82           90           102           107           119           124
## 0.24092197 0.35820266 0.38140796 0.16064510 0.09733803 0.32741401
##           125           161           164           181           189           204
## 0.47639239 0.32707222 0.16380198 0.09615363 0.24791078 0.60284207
##           230           235           244           249           254           255
## 0.54039587 0.12002320 0.38176956 0.05691972 0.42130615 -0.03267027
##           296
## 0.21907176
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model
```

```
##
```

```
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##       ejection_fraction + high_blood_pressure + serum_creatinine +
##       serum_sodium, family = binomial(link = "logit"), data = trainset)
##
```

```
## Coefficients:
```

```
##           (Intercept)                age                anaemia
##           0.9965989                0.0666032                0.1809211
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##           0.0001724                -0.0579326                0.7697949
##           serum_creatinine                serum_sodium
##           0.5716267                -0.0376864
```

```
##
```

```
## Degrees of Freedom: 205 Total (i.e. Null);  198 Residual
```

```
## Null Deviance:      253.7
```

```
## Residual Deviance: 200.7      AIC: 216.7
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9963  -0.7467  -0.4581   0.8293   2.5367
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.9965989   6.0487558   0.165 0.869132
## age            0.0666032   0.0155966   4.270 1.95e-05 ***
## anaemia        0.1809211   0.3645460   0.496 0.619689
## creatinine_phosphokinase 0.0001724 0.0001686   1.023 0.306335
## ejection_fraction -0.0579326 0.0174474  -3.320 0.000899 ***
## high_blood_pressure  0.7697949 0.3624780   2.124 0.033695 *
## serum_creatinine    0.5716267 0.2119965   2.696 0.007009 **
## serum_sodium     -0.0376864 0.0441970  -0.853 0.393831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 253.68  on 205  degrees of freedom
## Residual deviance: 200.74  on 198  degrees of freedom
## AIC: 216.74
##
## Number of Fisher Scoring iterations: 5
anova(generlinear.model)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                205      253.68
## age              1  23.1241      204      230.55
## anaemia          1   0.0383      203      230.51
## creatinine_phosphokinase 1   1.3570      202      229.16
## ejection_fraction  1  14.7913      201      214.37
## high_blood_pressure  1   3.4173      200      210.95
## serum_creatinine    1   9.4869      199      201.46
## serum_sodium        1   0.7257      198      200.74

generlinear.valid <- predict(generlinear.model, groups[[5]], type = 'response')
generlinear.valid
```

```

##           5           12           14           18           39           63           82
## 0.82454292 0.43184607 0.19321666 0.26002690 0.46617797 0.12360871 0.13874213
##           90           102           107           119           124           125           161
## 0.38158150 0.44189039 0.09637801 0.12547437 0.34430255 0.46030496 0.41926566
##           164           181           189           204           230           235           244
## 0.12406322 0.06877156 0.29795410 0.76848683 0.50401787 0.08839688 0.47428805
##           249           254           255           296
## 0.04568789 0.53183134 0.06987906 0.14426734

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - generlinear.valid)^2))/nrow
generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.56 1.00 3.71 3.37 total = 11.65
##
## REML score: 110.8426

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.30583    0.02647   11.55  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          2.563   3.220  7.171 9.22e-05 ***
## s(creatinine_phosphokinase) 1.001   1.001  1.939  0.16543
## s(ejection_fraction)      3.712   4.571  8.186 1.62e-06 ***
## s(serum_creatinine)       3.369   4.146  4.255  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.324 Deviance explained = 35.9%
## -REML = 110.84 Scale est. = 0.1443 n = 206

```

```
generaddit.valid <- predict(generaddit.model, groups[[5]], type = 'response')
generaddit.valid
```

```
##           5           12           14           18           39           63
## 0.902886643 0.375875227 0.055530819 0.712933655 0.578444891 0.093019827
##           82           90           102           107           119           124
## 0.277125503 0.415028029 0.368726662 0.238060736 0.129566725 0.194321338
##           125           161           164           181           189           204
## 0.311038007 0.163223887 0.097090179 0.086121502 0.078051973 0.630980128
##           230           235           244           249           254           255
## 0.560935849 0.165063714 0.206115923 -0.005531087 0.211412592 0.125615525
##           296
## 0.171338778
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~ . - time, data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6941748 0.3058252
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.47552 0.3916084                566.9930 0.4335664                39.67133
## 1 67.37567 0.4444444                680.9365 0.4285714                34.25397
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3076923 268498.8                1.197133      137.3007 0.6363636
## 1      0.4603175 269801.7                1.775873      136.0794 0.6190476
##      smoking
## 0 0.3286713
## 1 0.3333333
##
## Coefficients of linear discriminants:
##                                LD1
## age                6.088490e-02
## anaemia            1.010686e-01
## creatinine_phosphokinase 1.769413e-04
## diabetes           2.733493e-01
## ejection_fraction   -4.692135e-02
## high_blood_pressure  5.475879e-01
## platelets           2.140396e-07
## serum_creatinine     4.175273e-01
## serum_sodium        -4.976644e-02
## sex                 -4.923186e-01
## smoking             3.322952e-01
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[5]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##           5           12           14           18           39           63           82
## 0.91161662 0.37064811 0.10284497 0.19123987 0.43227555 0.10444374 0.13584751
##           90           102           107           119           124           125           161
## 0.20336509 0.32952697 0.05945117 0.15533451 0.28795654 0.32026569 0.29066823
##          164          181          189          204          230          235          244
## 0.15189107 0.05345823 0.36077002 0.69795091 0.56379513 0.07525598 0.47077079
##          249          254          255          296
## 0.03327275 0.65045676 0.03506201 0.15419173
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[5]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model
```

```
## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6941748 0.3058252
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.47552 0.3916084                566.9930 0.4335664                39.67133
## 1 67.37567 0.4444444                680.9365 0.4285714                34.25397
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0           0.3076923 268498.8           1.197133      137.3007 0.6363636
## 1           0.4603175 269801.7           1.775873      136.0794 0.6190476
##      smoking
## 0 0.3286713
## 1 0.3333333
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[5]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid
```

```
##           5           12           14           18           39           63
## 0.0056091057 0.0536877843 0.0095223608 0.7067866379 0.7696200384 0.0152946521
##           82           90           102           107           119           124
## 0.0299856768 0.0493722337 0.0579236529 0.0164872331 0.2450464177 0.1795539013
##          125          161          164          181          189          204
## 0.8655458290 0.1042841273 0.3428884209 0.0230726561 0.2778401802 0.9829791906
##          230          235          244          249          254          255
## 0.2515281946 0.0045991626 0.1963722302 0.0016662802 0.2838788187 0.0002925118
##          296
## 0.0702345199
```

```
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[5]]))
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model
```

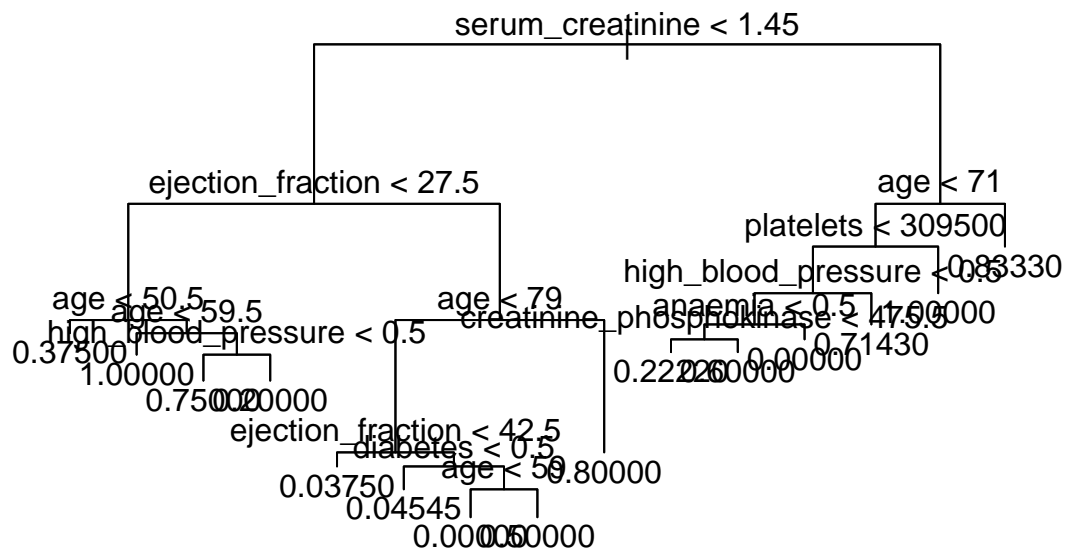
```
## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
```

```
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 76.28 89.91 97.34 99.35 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23301 ( N = 206 )
##
## Deviance: 200.702
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[5]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

##      [1] 0.75443596 0.30532268 0.10796526 0.27586674 0.53726177 0.08021419
##      [7] 0.12833765 0.20739349 0.34108527 0.06287683 0.23036083 0.45738455
##     [13] 0.26087853 0.31694776 0.22313849 0.06336795 0.45902630 0.61067668
##     [19] 0.51395811 0.06786151 0.53508646 0.02338832 0.59041539 0.03292827
##     [25] 0.18432639
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[5]]))
flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.26214 ( N = 206 )
flexdiscr.valid <- predict(flexdiscr.model, groups[[5]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

##           5           12           14           18           39           63           82
## 0.91407594 0.37131219 0.10163350 0.19029808 0.43359244 0.10323122 0.13466585
##           90          102          107          119          124          125          161
## 0.20249975 0.32976394 0.05839566 0.15421403 0.28778446 0.32040904 0.29052185
##          164          181          189          204          230          235          244
## 0.15075774 0.05244830 0.36133021 0.70136513 0.56638838 0.07411174 0.47248744
##          249          254          255          296
## 0.03247808 0.65366141 0.03424370 0.15306689
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[5]]))
decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[5]], type = 'vector')
decisiontree.valid
```

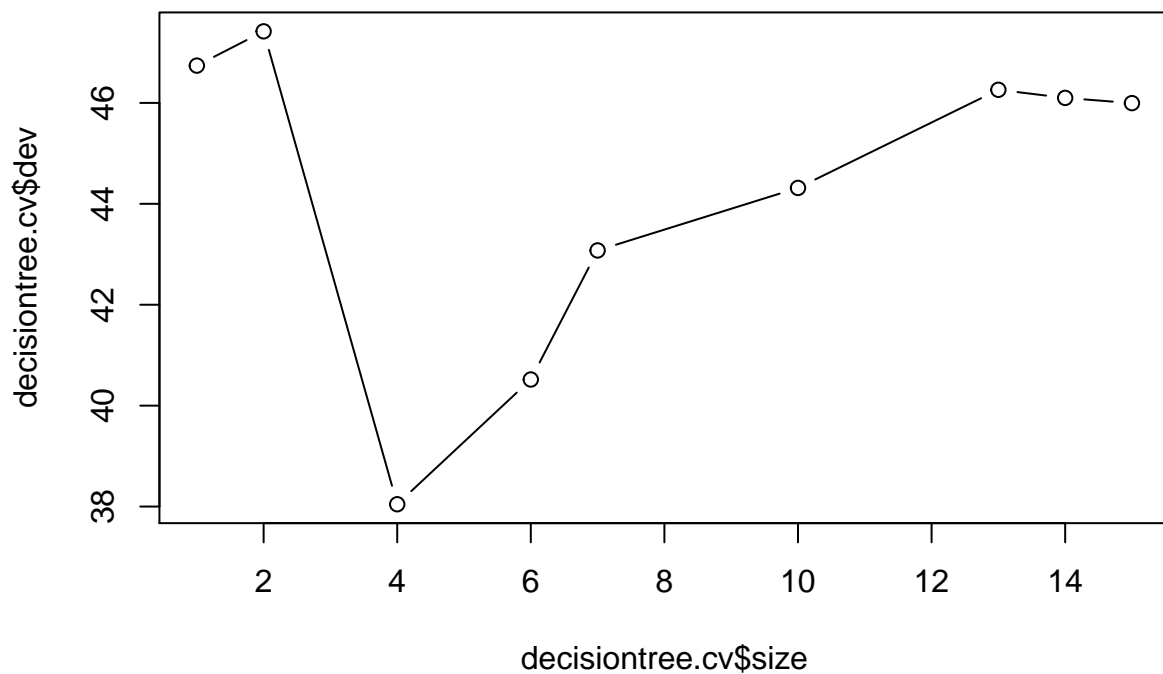
```
##          5          12          14          18          39          63          82
## 1.00000000 0.20000000 0.03750000 0.37500000 0.60000000 0.03750000 0.04545455
##          90          102          107          119          124          125          161
## 1.00000000 0.04545455 0.04545455 0.50000000 0.03750000 0.60000000 0.03750000
##          164          181          189          204          230          235          244
## 0.03750000 0.03750000 0.03750000 0.71428571 0.75000000 0.04545455 0.03750000
##          249          254          255          296
## 0.03750000 0.03750000 0.04545455 0.03750000
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - decisiontree.valid)^2))/n)
```

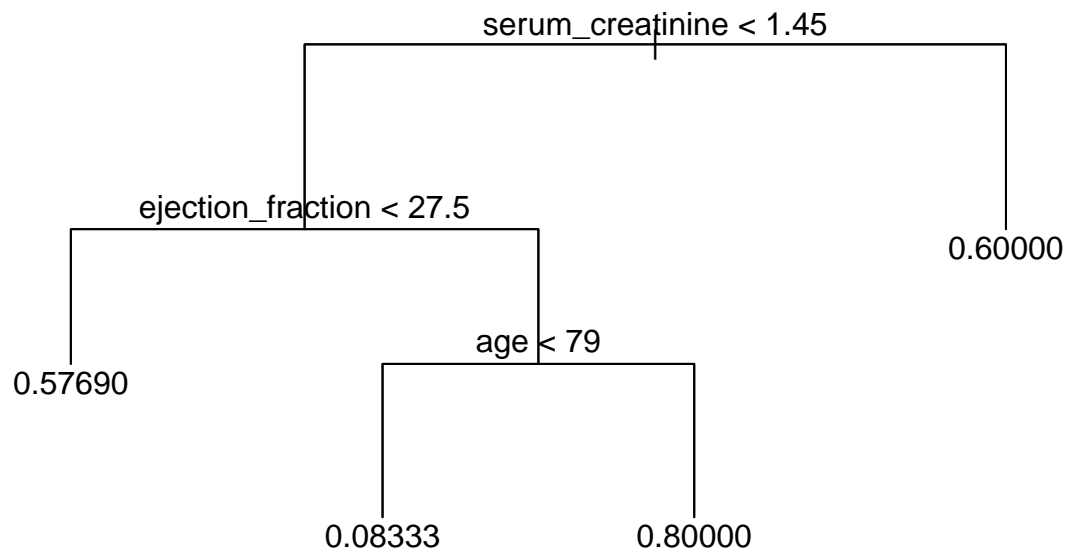
```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```

```
prunedectree.valid <- predict(prunedectree.model, groups[[5]], type = 'vector')
prunedectree.valid
```

```
##          5          12          14          18          39          63          82
## 0.60000000 0.57692308 0.08333333 0.57692308 0.60000000 0.08333333 0.08333333
##          90          102          107          119          124          125          161
## 0.57692308 0.08333333 0.08333333 0.08333333 0.08333333 0.60000000 0.08333333
##          164          181          189          204          230          235          244
## 0.08333333 0.08333333 0.08333333 0.60000000 0.57692308 0.08333333 0.08333333
##          249          254          255          296
## 0.08333333 0.08333333 0.08333333 0.08333333
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##6th fold-out training
```

```
trainset <- data.frame()
for (i in c(1:5,7:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
```

```
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.664e+00             1.027e-02             -2.391e-02
## creatinine_phosphokinase             diabetes             ejection_fraction
##             -2.991e-06             4.355e-02             -7.949e-03
##             high_blood_pressure             platelets             serum_creatinine
##             8.493e-02             2.466e-08             1.057e-01
##             serum_sodium                sex                smoking
##             -1.362e-02             -3.897e-02             2.383e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8968 -0.2851 -0.1216  0.3317  0.9916
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.664e+00  1.030e+00   1.615 0.107931
## age           1.027e-02  2.617e-03   3.925 0.000122 ***
## anaemia       -2.391e-02  6.182e-02  -0.387 0.699331
## creatinine_phosphokinase -2.991e-06  3.906e-05  -0.077 0.939044
## diabetes      4.355e-02  6.122e-02   0.711 0.477689
## ejection_fraction -7.949e-03  2.664e-03  -2.984 0.003225 **
## high_blood_pressure  8.493e-02  6.222e-02   1.365 0.173898
## platelets      2.466e-08  3.119e-07   0.079 0.937053
## serum_creatinine  1.057e-01  3.673e-02   2.878 0.004460 **
## serum_sodium   -1.362e-02  7.452e-03  -1.828 0.069196 .
## sex           -3.897e-02  7.217e-02  -0.540 0.589852
## smoking        2.383e-02  7.274e-02   0.328 0.743519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4139 on 188 degrees of freedom
## Multiple R-squared:  0.218, Adjusted R-squared:  0.1723
## F-statistic: 4.766 on 11 and 188 DF,  p-value: 1.889e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.670   3.6702  21.4279 6.835e-06 ***
## anaemia       1  0.053   0.0527   0.3075 0.579856
## creatinine_phosphokinase 1  0.006   0.0060   0.0353 0.851161
## diabetes      1  0.119   0.1187   0.6929 0.406237
## ejection_fraction 1  2.438   2.4383  14.2358 0.000216 ***
## high_blood_pressure 1  0.198   0.1977   1.1541 0.284080
## platelets     1  0.016   0.0157   0.0917 0.762352
```

```
## serum_creatinine      1  1.885  1.8851 11.0059  0.001090 **
## serum_sodium          1  0.543  0.5429  3.1698  0.076625 .
## sex                   1  0.033  0.0330  0.1929  0.661043
## smoking               1  0.018  0.0184  0.1074  0.743519
## Residuals            188 32.201  0.1713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[6]])
simplelinear.valid
```

```
##          10          16          24          27          38          50
## 1.445086827 0.331261985 -0.009074719 0.613725318 0.349757817 0.195513899
##          61          67          69          73          98         101
## 0.216252261 0.285545419 0.382814670 0.540204690 0.152398391 0.272514372
##          112         123         136         150         182         198
## 0.229327370 0.199757268 0.341175846 0.289871413 0.334271743 0.304241374
##          216         219         226         233         241         248
## 0.497358688 0.359923635 0.451164381 -0.031017616 0.490601479 0.504507724
##          267         278         279         280         282         292
## 0.426254517 0.294069050 0.183136001 0.266548599 0.482719430 0.220407685
##          298
## 0.046585183
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##      -0.138608           0.010143           -0.008886           0.113776
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8675 -0.2827 -0.1352  0.3573  1.0075
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.138608   0.175476  -0.790  0.430541
## age           0.010143   0.002534   4.003 8.85e-05 ***
## ejection_fraction -0.008886  0.002545  -3.492 0.000593 ***
## serum_creatinine  0.113776  0.035172   3.235 0.001429 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4125 on 196 degrees of freedom
## Multiple R-squared:  0.1901, Adjusted R-squared:  0.1777
## F-statistic: 15.33 on 3 and 196 DF,  p-value: 5.312e-09
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1  3.670   3.6702   21.569 6.248e-06 ***
## ejection_fraction  1  2.377   2.3769   13.968 0.0002439 ***
## serum_creatinine  1  1.781   1.7806   10.464 0.0014286 **
## Residuals      196 33.352   0.1702
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[6]])
revisedlinear.valid
```

```
##           10           16           24           27           38           50
## 1.43135469 0.39677063 -0.04313723 0.58336226 0.36263795 0.28676887
##           61           67           69           73           98          101
## 0.20947656 0.30203781 0.48581879 0.52635692 0.16343469 0.42372381
##          112          123          136          150          182          198
## 0.24480752 0.21766821 0.40097218 0.26139225 0.35148534 0.34516162
##          216          219          226          233          241          248
## 0.43876779 0.36529525 0.24828479 0.05852257 0.40833734 0.56148863
##          267          278          279          280          282          292
## 0.44977467 0.35892451 0.18163179 0.22952737 0.52319370 0.31828006
##          298
## 0.13947009
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model
```

```
##
```

```
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
## ejection_fraction + high_blood_pressure + serum_creatinine +
## serum_sodium, family = binomial(link = "logit"), data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##           (Intercept)                age                anaemia
##           5.536e+00                5.838e-02            -1.086e-01
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##           -1.632e-05                -4.928e-02                5.586e-01
##           serum_creatinine          serum_sodium
##           5.613e-01                -6.742e-02
```

```
##
```

```
## Degrees of Freedom: 199 Total (i.e. Null);  192 Residual
```

```
## Null Deviance:          240.9
```

```
## Residual Deviance: 195.5      AIC: 211.5
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1220  -0.7202  -0.4923   0.7769   2.4662
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    5.536e+00  5.912e+00   0.936 0.349107
## age             5.838e-02  1.588e-02   3.675 0.000238 ***
## anaemia        -1.086e-01  3.702e-01  -0.293 0.769173
## creatinine_phosphokinase -1.632e-05  2.443e-04  -0.067 0.946741
## ejection_fraction -4.928e-02  1.729e-02  -2.851 0.004360 **
## high_blood_pressure  5.586e-01  3.656e-01   1.528 0.126533
## serum_creatinine  5.613e-01  2.070e-01   2.711 0.006704 **
## serum_sodium    -6.742e-02  4.365e-02  -1.544 0.122489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 240.86  on 199  degrees of freedom
## Residual deviance: 195.47  on 192  degrees of freedom
## AIC: 211.47
##
## Number of Fisher Scoring iterations: 4
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			199	240.86
## age	1	18.0519	198	222.81
## anaemia	1	0.2154	197	222.59
## creatinine_phosphokinase	1	0.0324	196	222.56
## ejection_fraction	1	13.2278	195	209.33
## high_blood_pressure	1	1.2738	194	208.06
## serum_creatinine	1	10.1208	193	197.94
## serum_sodium	1	2.4695	192	195.47

```
generlinear.valid <- predict(generlinear.model, groups[[6]], type = 'response')
generlinear.valid
```

```
##          10          16          24          27          38          50          61
## 0.99472520 0.33296613 0.04833478 0.69351759 0.28512869 0.16761689 0.19066355
##          67          69          73          98          101          112          123
## 0.21366609 0.37478835 0.58411162 0.11970666 0.28836741 0.19676020 0.13553361
##          136          150          182          198          216          219          226
## 0.34522795 0.30407794 0.27477152 0.26795901 0.57877160 0.31847671 0.33297675
##          233          241          248          267          278          279          280
## 0.06324865 0.45355702 0.57048176 0.43414497 0.25339877 0.12730013 0.17276062
##          282          292          298
## 0.49717039 0.21816146 0.08305805
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - generlinear.valid)^2))/nrow
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 1.77 1.00 3.04 3.29 total = 10.11
##
## REML score: 110.2509
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29000    0.02745   10.57  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)          1.774   2.231  4.365 0.010287 *
## s(creatinine_phosphokinase) 1.000   1.000  0.027 0.870455
## s(ejection_fraction)      3.043   3.774  6.785 6.31e-05 ***
## s(serum_creatinine)       3.290   4.082  5.068 0.000594 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.272   Deviance explained = 30.5%
## -REML = 110.25   Scale est. = 0.15067   n = 200
```

```
generaddit.valid <- predict(generaddit.model, groups[[6]], type = 'response')
generaddit.valid
```

```
##          10          16          24          27          38          50
## 1.34525213 0.44569871 0.06196851 0.50298514 0.34996193 0.19756546
##          61          67          69          73          98         101
## 0.33201573 0.75395735 0.50188996 0.42011306 0.26366550 0.42643668
##          112         123         136         150         182         198
## 0.16763283 0.03581123 0.29894329 0.11071510 0.35572533 0.18629574
##          216         219         226         233         241         248
## 0.33685692 0.22274119 0.42186187 -0.00604270 0.30255667 0.73719276
##          267         278         279         280         282         292
## 0.79111115 0.22052649 0.08269669 0.18324646 0.54666729 0.26198152
##          298
## 0.18161822
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~ . - time, data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.71 0.29
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.13850 0.4154930          570.6127 0.4154930          39.71127
## 1 65.94253 0.4137931          517.0345 0.4310345          34.13793
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3591549 268661.5          1.200845    137.5141 0.6338028
## 1          0.4655172 265347.4          1.740517    135.4828 0.6379310
##      smoking
## 0 0.3450704
## 1 0.3103448
##
## Coefficients of linear discriminants:
##                                LD1
## age                5.455688e-02
## anaemia            -1.269881e-01
## creatinine_phosphokinase -1.588190e-05
## diabetes           2.312901e-01
## ejection_fraction  -4.221379e-02
## high_blood_pressure  4.510068e-01
## platelets           1.309748e-07
## serum_creatinine     5.614226e-01
## serum_sodium        -7.232371e-02
## sex                -2.069369e-01
```

```
## smoking 1.265695e-01
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[6]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##      10      16      24      27      38      50      61
## 0.99733670 0.28430332 0.04671302 0.69287802 0.30799988 0.14705242 0.16377668
##      67      69      73      98     101     112     123
## 0.23070779 0.35292076 0.58941363 0.11680712 0.21679456 0.17508698 0.15035530
##     136     150     182     198     216     219     226
## 0.29686823 0.23546256 0.28808392 0.25173874 0.52450245 0.32147999 0.45364517
##     233     241     248     267     278     279     280
## 0.04105912 0.51413107 0.53545209 0.41602319 0.24014067 0.13776022 0.21063077
##     282     292     298
## 0.50201845 0.16730612 0.06454630
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[6]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model
```

```
## Call:
```

```
## qda(DEATH_EVENT ~ . - time, data = trainset)
```

```
##
```

```
## Prior probabilities of groups:
```

```
##      0      1
```

```
## 0.71 0.29
```

```
##
```

```
## Group means:
```

```
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
```

```
## 0 58.13850 0.4154930 570.6127 0.4154930 39.71127
```

```
## 1 65.94253 0.4137931 517.0345 0.4310345 34.13793
```

```
## high_blood_pressure platelets serum_creatinine serum_sodium sex
```

```
## 0 0.3591549 268661.5 1.200845 137.5141 0.6338028
```

```
## 1 0.4655172 265347.4 1.740517 135.4828 0.6379310
```

```
## smoking
```

```
## 0 0.3450704
```

```
## 1 0.3103448
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[6]], type = 'response')
```

```
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
```

```
quaddiscr.valid
```

```
##      10      16      24      27      38      50
## 1.000000e+00 3.082548e-01 2.704354e-02 2.048173e-01 4.036418e-01 5.987646e-02
##      61      67      69      73      98     101
## 8.906949e-26 1.458700e-01 4.544346e-02 5.733998e-20 6.173980e-02 6.094068e-01
##     112     123     136     150     182     198
## 7.094744e-02 9.057824e-02 5.807622e-01 2.660839e-03 1.421530e-01 5.864944e-02
##     216     219     226     233     241     248
## 4.838818e-01 1.310295e-01 9.789675e-01 2.581658e-02 1.235118e-01 6.129669e-01
##     267     278     279     280     282     292
## 1.833841e-01 4.982150e-01 1.471822e-01 3.369083e-02 4.720457e-01 2.772555e-01
##     298
## 6.088059e-03
```



```

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[6]]))

mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 81.76  94.07  98.35  99.83 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.225 ( N = 200 )
##
## Deviance: 187.219

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[6]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 0.99935905 0.36439445 0.03367089 0.68621836 0.30409608 0.08406365
## [7] 0.29580231 0.21445477 0.33270734 0.38532966 0.08447031 0.14314598
## [13] 0.17569128 0.14356240 0.24942940 0.22432659 0.24911340 0.19315571
## [19] 0.55098762 0.24268510 0.47065325 0.03592715 0.97197501 0.47998911
## [25] 0.38385301 0.41119108 0.11678372 0.37857152 0.56214124 0.33430844
## [31] 0.06825715

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[6]]))

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.235 ( N = 200 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[6]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

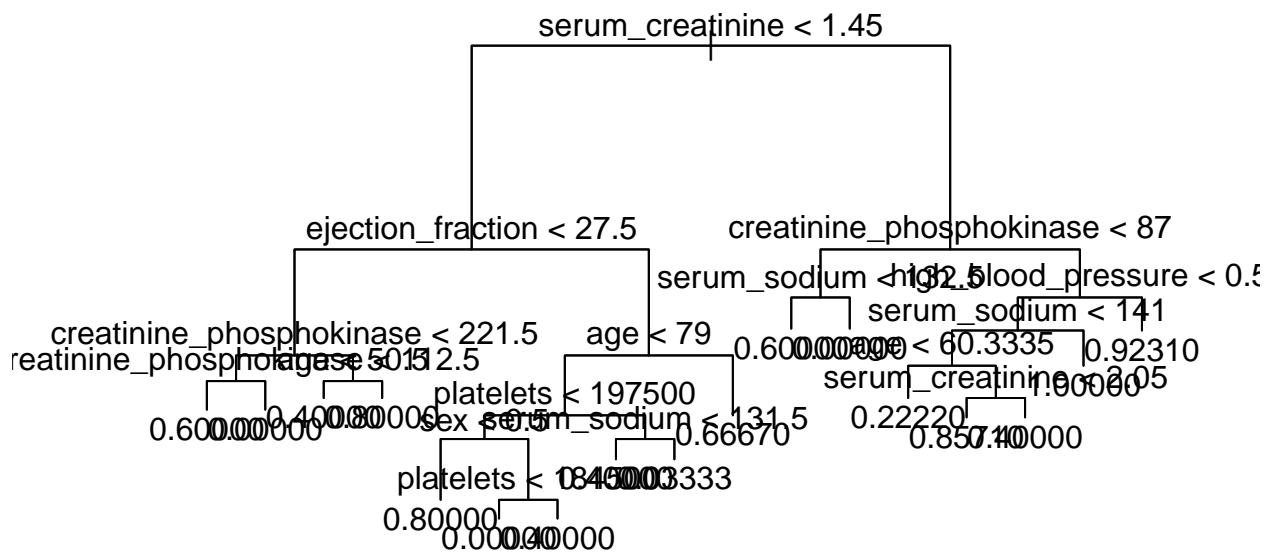
##      10      16      24      27      38      50      61
## 0.99751358 0.28424612 0.04576841 0.69653917 0.30818484 0.14596300 0.16276242
##      67      69      73      98     101     112     123
## 0.23015442 0.35358806 0.59248265 0.11563718 0.21612803 0.17413401 0.14927916

```

```
##          136          150          182          198          216          219          226
## 0.29693808 0.23494956 0.28806460 0.25136985 0.52700456 0.32180706 0.45542184
##          233          241          248          267          278          279          280
## 0.04017143 0.51653265 0.53805770 0.41738892 0.23966818 0.13663789 0.20991665
##          282          292          298
## 0.50429981 0.16631014 0.06347019
```

```
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[6]]))
```

```
decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[6]], type = 'vector')
decisiontree.valid
```

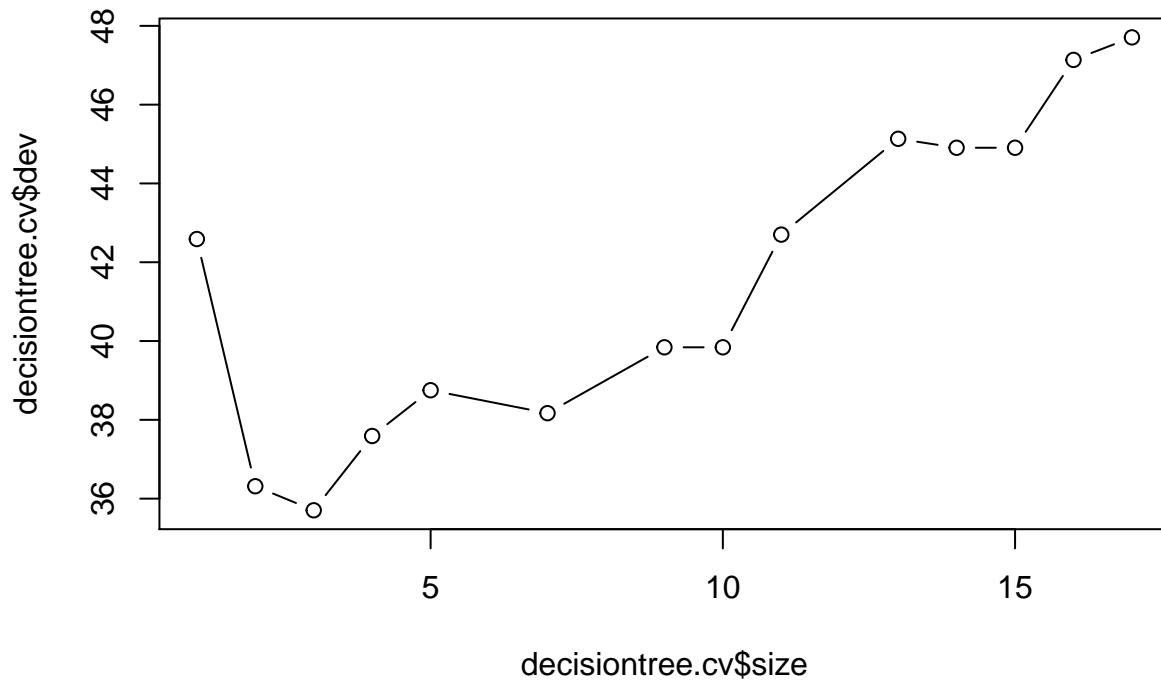
```
##          10          16          24          27          38          50          61
## 0.92307692 0.66666667 0.03333333 0.66666667 0.66666667 0.03333333 0.40000000
##          67          69          73          98          101          112          123
## 0.40000000 0.00000000 0.66666667 0.03333333 0.80000000 0.03333333 0.03333333
##          136          150          182          198          216          219          226
## 0.03333333 0.03333333 0.00000000 0.03333333 0.03333333 0.03333333 0.40000000
##          233          241          248          267          278          279          280
## 0.03333333 0.03333333 0.40000000 0.22222222 0.00000000 0.03333333 0.03333333
##          282          292          298
## 0.40000000 0.00000000 0.00000000
```

```

decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - decisiontree.valid)^2))/n)

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

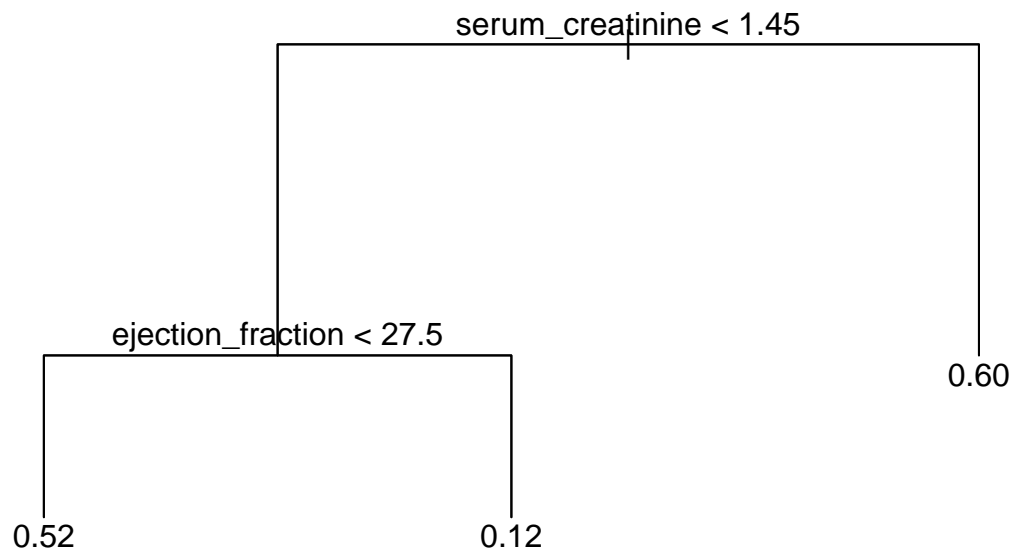
```



```

prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)

```



```
prunedectree.valid <- predict(prunedectree.model, groups[[6]], type = 'vector')
prunedectree.valid
```

```
## 10 16 24 27 38 50 61 67 69 73 98 101 112 123 136 150
## 0.60 0.12 0.12 0.12 0.12 0.12 0.52 0.52 0.52 0.12 0.12 0.52 0.12 0.12 0.12 0.12
## 182 198 216 219 226 233 241 248 267 278 279 280 282 292 298
## 0.52 0.12 0.12 0.12 0.12 0.12 0.12 0.60 0.60 0.12 0.12 0.12 0.60 0.12 0.12
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##7th fold-out training
trainset <- data.frame()
for (i in c(1:6, 8)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
## (Intercept)          age          anaemia
## 1.424e+00      9.350e-03      4.402e-02
```

```
## creatinine_phosphokinase      diabetes      ejection_fraction
##          4.819e-05              3.550e-02      -9.124e-03
##      high_blood_pressure      platelets      serum_creatinine
##          1.295e-01              1.170e-07      7.963e-02
##          serum_sodium              sex      smoking
##          -1.160e-02              -5.272e-02      3.283e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7238 -0.2842 -0.1399  0.3642  1.0047
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.424e+00  1.054e+00   1.351 0.178217
## age            9.350e-03  2.659e-03   3.516 0.000548 ***
## anaemia        4.402e-02  6.200e-02   0.710 0.478560
## creatinine_phosphokinase 4.819e-05  3.083e-05   1.563 0.119660
## diabetes       3.550e-02  6.290e-02   0.564 0.573176
## ejection_fraction -9.124e-03  2.756e-03  -3.310 0.001115 **
## high_blood_pressure  1.295e-01  6.326e-02   2.047 0.042023 *
## platelets       1.170e-07  3.057e-07   0.383 0.702464
## serum_creatinine  7.963e-02  3.055e-02   2.607 0.009861 **
## serum_sodium    -1.160e-02  7.597e-03  -1.528 0.128271
## sex            -5.272e-02  7.249e-02  -0.727 0.467928
## smoking         3.283e-02  7.251e-02   0.453 0.651209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4199 on 190 degrees of freedom
## Multiple R-squared:  0.2058, Adjusted R-squared:  0.1598
## F-statistic: 4.476 on 11 and 190 DF,  p-value: 5.269e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  2.752  2.75212  15.6105 0.0001096 ***
## anaemia       1  0.018  0.01792   0.1017 0.7501814
## creatinine_phosphokinase 1  0.346  0.34610   1.9631 0.1628128
## diabetes      1  0.021  0.02098   0.1190 0.7304921
## ejection_fraction 1  2.771  2.77135  15.7195 0.0001039 ***
## high_blood_pressure 1  0.655  0.65514   3.7161 0.0553820 .
## platelets     1  0.103  0.10337   0.5863 0.4447846
## serum_creatinine 1  1.545  1.54550   8.7663 0.0034597 **
## serum_sodium  1  0.371  0.37122   2.1056 0.1484066
## sex           1  0.061  0.06136   0.3481 0.5559162
## smoking       1  0.036  0.03615   0.2050 0.6512091
```

```
## Residuals          190 33.497 0.17630
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[7]])
simplelinear.valid

##           6           7           26           71           80           84
## 0.715899116 0.512318508 0.345881641 -0.087367878 0.158464684 0.450124701
##           89           111           128           167           175           184
## 0.057135879 0.257363650 0.017171041 -0.100081233 0.381945570 0.530530860
##           186           187           188           190           192           207
## 0.336237276 -0.025804837 0.422807857 0.008053369 0.111128939 -0.063983864
##           210           212           214           222           224           228
## 0.361282706 -0.080093161 0.461951317 -0.010844085 0.134761290 0.346010937
##           236           245           251           269           276
## 0.359233678 0.212398365 0.343022002 0.044120064 0.179001420

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - simplelinear.valid)^2))/n)

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Coefficients:
##      (Intercept)           age  ejection_fraction  serum_creatinine
##      -0.042274         0.009592         -0.009893          0.086021

summary(revisedlinear.model)

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7640 -0.3024 -0.1608  0.4333  0.9657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.042274   0.177694  -0.238  0.812202
## age           0.009592   0.002594   3.698  0.000281 ***
## ejection_fraction -0.009893  0.002644  -3.742  0.000239 ***
## serum_creatinine  0.086021  0.029895   2.877  0.004449 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4217 on 198 degrees of freedom
## Multiple R-squared:  0.1651, Adjusted R-squared:  0.1524
## F-statistic: 13.05 on 3 and 198 DF, p-value: 8.22e-08
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1  2.752  2.75212  15.4737 0.0001157 ***
## ejection_fraction 1  2.737  2.73749  15.3914 0.0001205 ***
## serum_creatinine 1  1.473  1.47260   8.2796 0.0044490 **
## Residuals    198 35.216  0.17786
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[7]])
```

```
revisedlinear.valid
```

```
##           6           7           26           71           80           84
## 0.605939166 0.631955812 0.512600973 0.024107897 0.117533846 0.375695969
##           89          111          128          167          175          184
## 0.044281484 0.282708798 -0.010681265 -0.067242941 0.312378725 0.516254254
##           186          187          188          190          192          207
## 0.371892708 -0.005695003 0.466570626 -0.026344773 0.107085141 0.014515989
##           210          212          214          222          224          228
## 0.150306060 -0.107201688 0.258994146 0.181194300 0.230048043 0.225061782
##           236          245          251          269          276
## 0.296297196 0.254609229 0.183554457 0.090862826 0.082260689
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model
```

```
##
```

```
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##       ejection_fraction + high_blood_pressure + serum_creatinine +
##       serum_sodium, family = binomial(link = "logit"), data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##           (Intercept)                age                anaemia
##           3.6484870                0.0536428                0.3298001
## creatinine_phosphokinase  ejection_fraction  high_blood_pressure
##           0.0002485                -0.0556756                0.8093114
##           serum_creatinine  serum_sodium
##           0.5128360                -0.0527211
```

```
##
```

```
## Degrees of Freedom: 201 Total (i.e. Null);  194 Residual
```

```
## Null Deviance:      245.8
```

```
## Residual Deviance: 201.5    AIC: 217.5
```

```
summary(generlinear.model)
```

```
##
```

```
## Call:
```

```
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##       ejection_fraction + high_blood_pressure + serum_creatinine +
##       serum_sodium, family = binomial(link = "logit"), data = trainset)
```

```
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8304  -0.7396  -0.5009   0.8429   2.4503
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.6484870   5.8708764   0.621 0.534300
## age            0.0536428   0.0157879   3.398 0.000679 ***
## anaemia        0.3298001   0.3658509   0.901 0.367344
## creatinine_phosphokinase 0.0002485 0.0001702   1.459 0.144447
## ejection_fraction -0.0556756 0.0178474  -3.120 0.001811 **
## high_blood_pressure  0.8093114 0.3603472   2.246 0.024709 *
## serum_creatinine    0.5128360 0.2007902   2.554 0.010647 *
## serum_sodium       -0.0527211 0.0431694  -1.221 0.221987
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 245.76  on 201  degrees of freedom
## Residual deviance: 201.50  on 194  degrees of freedom
## AIC: 217.5
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			201	245.76
## age	1	13.3194	200	232.44
## anaemia	1	0.0987	199	232.35
## creatinine_phosphokinase	1	1.6493	198	230.70
## ejection_fraction	1	14.8746	197	215.82
## high_blood_pressure	1	3.5555	196	212.27
## serum_creatinine	1	9.2516	195	203.01
## serum_sodium	1	1.5195	194	201.50

```
generlinear.valid <- predict(generlinear.model, groups[[7]], type = 'response')
generlinear.valid
```

	6	7	26	71	80	84	89
## 0.82022541	0.65024045	0.31941973	0.03512460	0.12640012	0.54152015	0.08648562	
## 111	128	167	175	184	186	187	
## 0.19090492	0.04680644	0.03067344	0.32825027	0.64734652	0.30733466	0.04455398	
## 188	190	192	207	210	212	214	


```
## 0.36155039 0.05948682 0.09433647 0.04305207 0.25648402 0.04156544 0.42348738
##          222          224          228          236          245          251          269
## 0.06299500 0.15033853 0.40501324 0.37818133 0.17258805 0.27377896 0.05582657
##          276
## 0.12877105
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - generlinear.valid)^2))/nrow
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.48 1.00 3.56 2.69 total = 10.74
##
## REML score: 113.0936
```

```
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29703    0.02747   10.81  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df    F  p-value
## s(age)          2.485   3.127  5.819 0.000647 ***
## s(creatinine_phosphokinase) 1.000   1.001  2.295 0.131482
## s(ejection_fraction)      3.563   4.400  8.517 1.19e-06 ***
## s(serum_creatinine)       2.691   3.338  3.444 0.013784 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.274 Deviance explained = 30.9%
## -REML = 113.09 Scale est. = 0.15243 n = 202
```

```
generaddit.valid <- predict(generaddit.model, groups[[7]], type = 'response')
generaddit.valid
```

```
##          6          7          26          71          80          84
```

```
## 0.752685319 1.050457099 0.487053860 -0.056235052 0.105420824 0.534501554
##      89      111      128      167      175      184
## -0.057638448 0.540768251 0.085163405 0.049073354 0.086220016 0.437080954
##      186      187      188      190      192      207
## 0.283845616 0.057098695 0.632984196 0.012790412 0.246514885 -0.066108605
##      210      212      214      222      224      228
## 0.036322901 0.084637135 0.256601730 0.201463401 0.296843379 0.182574672
##      236      245      251      269      276
## 0.366962896 0.223186780 0.143904906 0.004428891 -0.017270063
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7029703 0.2970297
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 59.31456 0.4084507      572.0704 0.4295775      38.94366
## 1 65.88333 0.4500000      726.2333 0.4166667      33.40000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3028169 265411.2      1.239085      137.3592 0.6478873
## 1      0.4666667 269175.1      1.788000      135.7667 0.6166667
##      smoking
## 0 0.3309859
## 1 0.3166667
##
## Coefficients of linear discriminants:
##
##      LD1
## age      5.035801e-02
## anaemia  2.371132e-01
## creatinine_phosphokinase 2.595360e-04
## diabetes  1.911815e-01
## ejection_fraction -4.914219e-02
## high_blood_pressure  6.974433e-01
## platelets  6.299304e-07
## serum_creatinine  4.289029e-01
## serum_sodium -6.250222e-02
## sex -2.839652e-01
## smoking  1.768441e-01
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[7]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##      6      7      26      71      80      84      89
## 0.80059616 0.54351681 0.30592640 0.03210754 0.12584121 0.45095053 0.07288170
##      111      128      167      175      184      186      187
## 0.20623903 0.05831227 0.02983055 0.35345033 0.57034258 0.29383783 0.04571846
##      188      190      192      207      210      212      214
```

```
## 0.41097622 0.05539383 0.09789212 0.03674162 0.32579222 0.03348513 0.46848735
##      222      224      228      236      245      251      269
## 0.04977762 0.11107900 0.30609035 0.32311073 0.16572682 0.30231305 0.06780178
##      276
## 0.13995993
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[7]]))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model
```

```
## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7029703 0.2970297
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 59.31456 0.4084507                    572.0704 0.4295775          38.94366
## 1 65.88333 0.4500000                    726.2333 0.4166667          33.40000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3028169 265411.2          1.239085      137.3592 0.6478873
## 1      0.4666667 269175.1          1.788000      135.7667 0.6166667
##      smoking
## 0 0.3309859
## 1 0.3166667
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[7]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid
```

```
##      6      7      26      71      80      84
## 0.799261464 0.315123448 0.277496144 0.002462024 0.016305193 0.148141544
##      89      111      128      167      175      184
## 0.026777755 0.294177243 0.038030471 0.001591881 0.154199315 0.422919952
##      186      187      188      190      192      207
## 0.040346119 0.027144205 0.298123364 0.024495764 0.009449111 0.019738808
##      210      212      214      222      224      228
## 0.428778477 0.001262947 0.410974206 0.019355355 0.220359049 0.495770082
##      236      245      251      269      276
## 0.204319039 0.034094604 0.497614480 0.044618080 0.021424387
```

```
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[7]]))
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
```

```

## 75.56 91.40 97.84 99.59 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.27228 ( N = 202 )
##
## Deviance: 205.604
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[7]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

## [1] 0.74365912 0.64568965 0.14795443 0.03757085 0.12420787 0.40678532
## [7] 0.08791225 0.35505196 0.05376619 0.03303539 0.35582982 0.53701565
## [13] 0.30907893 0.07944249 0.42972309 0.04314521 0.08845855 0.06904299
## [19] 0.29650551 0.03763892 0.41498432 0.03789826 0.26375596 0.32954326
## [25] 0.41871067 0.14778002 0.32081220 0.06124041 0.16467290

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[7]]))

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
## v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.26238 ( N = 202 )

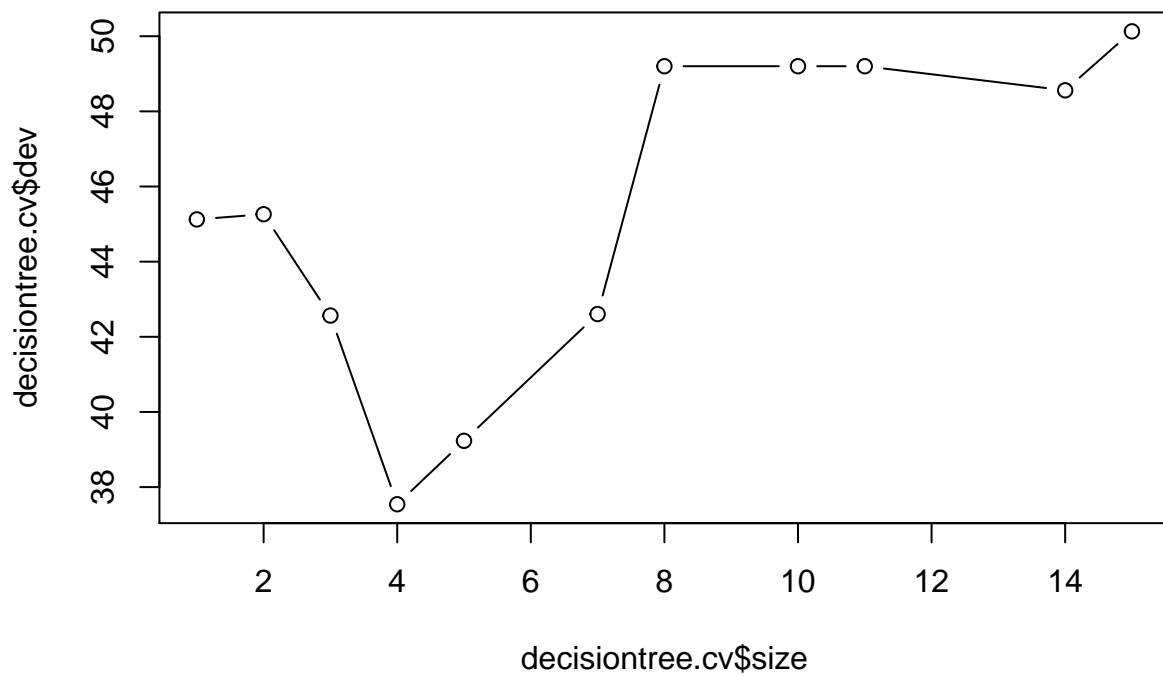
flexdiscr.valid <- predict(flexdiscr.model, groups[[7]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

##          6          7          26          71          80          84          89
## 0.80416620 0.54608599 0.30601612 0.03132612 0.12466148 0.45259670 0.07175346
##          111         128         167         175         184         186         187
## 0.20544413 0.05726666 0.02908145 0.35403917 0.57314547 0.29380601 0.04477802
##          188         190         192         207         210         212         214
## 0.41219089 0.05436953 0.09669820 0.03589989 0.32608707 0.03268505 0.47031862
##          222         224         228         236         245         251         269
## 0.04879939 0.10988175 0.30618173 0.32337754 0.16468594 0.30236615 0.06669814
##          276
## 0.13881536

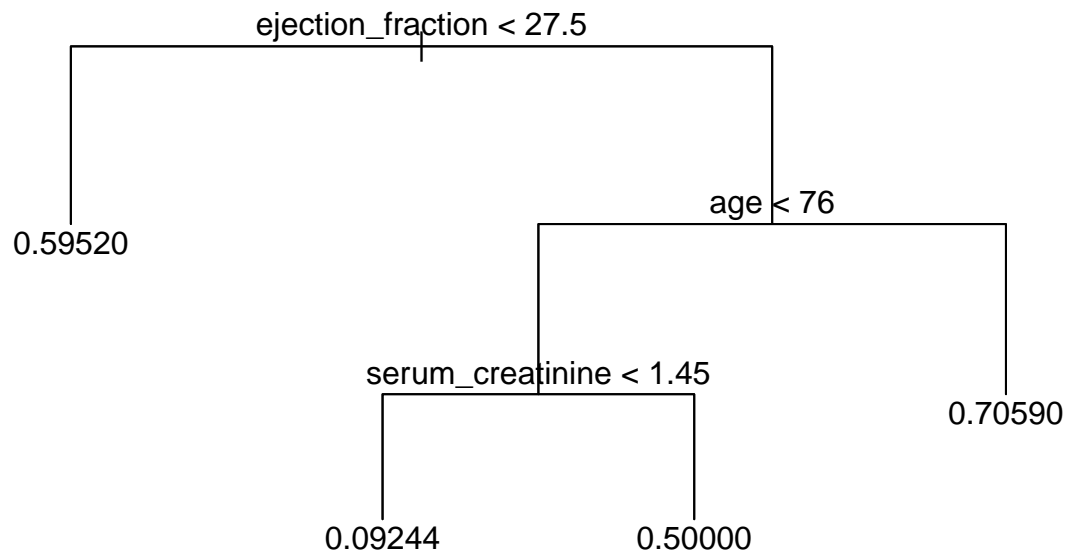
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[7]]))

decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```

```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[7]], type = 'vector')
prunedectree.valid
```

```
##          6          7          26          71          80          84          89
## 0.70588235 0.59523810 0.70588235 0.09243697 0.09243697 0.70588235 0.09243697
##          111         128         167         175         184         186         187
## 0.70588235 0.09243697 0.09243697 0.09243697 0.50000000 0.50000000 0.09243697
##          188         190         192         207         210         212         214
## 0.59523810 0.09243697 0.50000000 0.09243697 0.09243697 0.09243697 0.50000000
##          222         224         228         236         245         251         269
## 0.09243697 0.59523810 0.09243697 0.70588235 0.50000000 0.09243697 0.09243697
##          276
## 0.09243697
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##8th fold-out training
trainset <- data.frame()
for (i in c(1:7)) {
  trainset <- rbind(trainset, groups[[i]])
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
```

```
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.109e+00             1.047e-02             1.604e-02
## creatinine_phosphokinase            diabetes            ejection_fraction
##             4.944e-05             4.338e-02             -9.487e-03
##             high_blood_pressure            platelets            serum_creatinine
##             5.077e-02             -1.917e-08             8.660e-02
##             serum_sodium                sex                smoking
##             -9.164e-03             -8.123e-02             8.437e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7918 -0.2965 -0.1352  0.3752  0.9925
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.109e+00  1.031e+00   1.075 0.283681
## age            1.047e-02  2.677e-03   3.909 0.000129 ***
## anaemia        1.604e-02  6.156e-02   0.261 0.794712
## creatinine_phosphokinase 4.944e-05  3.153e-05   1.568 0.118527
## diabetes       4.338e-02  6.164e-02   0.704 0.482414
## ejection_fraction -9.487e-03  2.689e-03  -3.528 0.000524 ***
## high_blood_pressure  5.077e-02  6.135e-02   0.827 0.409015
## platelets       -1.917e-08  3.172e-07  -0.060 0.951866
## serum_creatinine  8.660e-02  3.086e-02   2.806 0.005537 **
## serum_sodium    -9.164e-03  7.403e-03  -1.238 0.217317
## sex            -8.123e-02  7.059e-02  -1.151 0.251300
## smoking         8.437e-02  7.248e-02   1.164 0.245838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4174 on 191 degrees of freedom
## Multiple R-squared:  0.2125, Adjusted R-squared:  0.1672
## F-statistic: 4.686 on 11 and 191 DF,  p-value: 2.45e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.407   3.4067  19.5495 1.643e-05 ***
## anaemia       1  0.000   0.0003   0.0016 0.9678906
## creatinine_phosphokinase 1  0.584   0.5838   3.3500 0.0687626 .
## diabetes     1  0.115   0.1146   0.6578 0.4183350
## ejection_fraction 1  2.684   2.6836  15.3998 0.0001213 ***
```



```
## high_blood_pressure      1  0.064  0.0642  0.3684 0.5445880
## platelets                1  0.019  0.0193  0.1110 0.7393391
## serum_creatinine         1  1.573  1.5734  9.0290 0.0030137 **
## serum_sodium             1  0.213  0.2132  1.2236 0.2700527
## sex                     1  0.087  0.0871  0.4997 0.4805089
## smoking                  1  0.236  0.2361  1.3551 0.2458375
## Residuals                191 33.284  0.1743
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.valid <- predict(simplelinear.model, groups[[8]])
simplelinear.valid
```

```
##          25          33          34          37          41          45
## 0.657473811 0.330001194 0.248230535 0.411800272 0.659778434 0.046242185
##          72          74          75          100          103          133
## 0.260846314 0.226555026 0.543935659 0.259139358 0.529195161 0.138694972
##          135          141          142          145          148          162
## 0.694336276 0.406334573 0.076884473 0.608655431 0.034215261 0.065444593
##          193          195          203          232          234          281
## 0.101065640 0.287302573 0.055580523 0.380814010 0.216534119 0.383792418
##          283          288          289          297
## 0.421997336 -0.003365902 0.333407266 -0.053852674
```

```
simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - simplelinear.valid)^2))/n)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##      (Intercept)          age ejection_fraction  serum_creatinine
##      -0.06696      0.01006      -0.01002      0.09198
```

```
summary(revisedlinear.model)
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.7966 -0.2915 -0.1471  0.3990  1.0095
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.06696    0.170895  -0.392  0.695609
## age           0.010063   0.002561   3.929  0.000118 ***
## ejection_fraction -0.010018  0.002598  -3.857  0.000155 ***
## serum_creatinine  0.091982   0.030005   3.066  0.002475 **
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4157 on 199 degrees of freedom
## Multiple R-squared:  0.1865, Adjusted R-squared:  0.1742
## F-statistic: 15.21 on 3 and 199 DF,  p-value: 6.004e-09
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age         1  3.407   3.4067  19.7166 1.488e-05 ***
## ejection_fraction  1  2.852   2.8519  16.5057 6.976e-05 ***
## serum_creatinine   1  1.624   1.6237   9.3974 0.002475 **
## Residuals       199 34.384   0.1728
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[8]])
```

```
revisedlinear.valid
```

```
##           25           33           34           37           41           45
## 0.55556661 0.17755579 0.24604075 0.42981379 0.60542839 0.03694176
##           72           74           75           100          103          133
## 0.24886241 0.20583084 0.53741687 0.26748559 0.58882415 0.10377146
##           135          141          142          145          148          162
## 0.50606851 0.45859370 0.12810524 0.56350834 0.06799599 0.10884398
##           193          195          203          232          234          281
## 0.09987267 0.33269517 0.11917641 0.38801590 0.21448455 0.32872911
##           283          288          289          297
## 0.40468819 -0.07311423 0.33770041 -0.14159919
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - revisedlinear.valid)^2))
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model
```

```
##
```

```
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##       ejection_fraction + high_blood_pressure + serum_creatinine +
##       serum_sodium, family = binomial(link = "logit"), data = trainset)
##
```

```
## Coefficients:
```

```
##           (Intercept)                age                anaemia
##           1.5546187                0.0598842                0.1213230
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##           0.0002703                -0.0604941                0.3822764
##           serum_creatinine          serum_sodium
##           0.5282822                -0.0371354
```

```
##
```

```
## Degrees of Freedom: 202 Total (i.e. Null);  195 Residual
```

```
## Null Deviance:          246.5
```

```
## Residual Deviance: 201.3      AIC: 217.3
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9606  -0.7671  -0.4752   0.8736   2.4606
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.5546187   5.8690077   0.265 0.791097
## age             0.0598842   0.0166652   3.593 0.000326 ***
## anaemia         0.1213230   0.3624572   0.335 0.737834
## creatinine_phosphokinase 0.0002703 0.0001781   1.518 0.129138
## ejection_fraction -0.0604941 0.0180989  -3.342 0.000831 ***
## high_blood_pressure  0.3822764 0.3599492   1.062 0.288223
## serum_creatinine    0.5282822 0.2082958   2.536 0.011206 *
## serum_sodium     -0.0371354 0.0427895  -0.868 0.385470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.47  on 202  degrees of freedom
## Residual deviance: 201.34  on 195  degrees of freedom
## AIC: 217.34
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
```

	Df	Deviance	Resid. Df	Resid. Dev
## NULL			202	246.47
## age	1	16.5498	201	229.92
## anaemia	1	0.0004	200	229.92
## creatinine_phosphokinase	1	3.0340	199	226.88
## ejection_fraction	1	15.4007	198	211.48
## high_blood_pressure	1	0.3926	197	211.09
## serum_creatinine	1	8.9877	196	202.10
## serum_sodium	1	0.7620	195	201.34

```
generlinear.valid <- predict(generlinear.model, groups[[8]], type = 'response')
generlinear.valid
```

```
##          25          33          34          37          41          45          72
## 0.68164288 0.22747942 0.16904211 0.40351922 0.74399776 0.05246030 0.16542227
##          74          75          100          103          133          135          141
## 0.12810282 0.57245473 0.17186480 0.57673132 0.11941262 0.74097463 0.46733663
##          142          145          148          162          193          195          203
## 0.08854665 0.70029565 0.08576477 0.07939396 0.10204535 0.35688902 0.10679921
##          232          234          281          283          288          289          297
## 0.32285778 0.16717224 0.37660280 0.38375960 0.03577678 0.24579158 0.02859741

generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - generlinear.valid)^2))/nrow

generaddit.model <- gam(DEATH_EVENT ~ s(age)+ s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.58 1.00 3.36 2.70 total = 10.64
##
## REML score: 111.8547

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2956      0.0272   10.87  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          2.585   3.250 6.006 0.000427 ***
## s(creatinine_phosphokinase) 1.000   1.000 3.205 0.074970 .
## s(ejection_fraction)      3.362   4.174 7.791 5.87e-06 ***
## s(serum_creatinine)       2.698   3.348 4.248 0.004799 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.282 Deviance explained = 31.7%
## -REML = 111.85 Scale est. = 0.15016 n = 203
```

```
generaddit.valid <- predict(generaddit.model, groups[[8]], type = 'response')
generaddit.valid
```

```
##          25          33          34          37          41          45
## 0.545354797 0.069458251 0.213890409 0.642655886 0.825420250 0.153718585
##          72          74          75          100          103          133
## 0.083137700 0.222114307 0.687768523 0.106468139 0.692489073 0.104450997
##          135          141          142          145          148          162
## 0.654743034 0.399952614 0.030499007 0.653862072 0.202333652 -0.006411645
##          193          195          203          232          234          281
## -0.017300359 0.661056885 0.157786432 0.183196550 0.174054889 0.285576770
##          283          288          289          297
## 0.404776264 0.090660544 0.174015093 0.114748156
```

```
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
```

```
lindiscr.model <- lda(DEATH_EVENT ~ . - time, data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.67599 0.4055944          529.7762 0.4055944          39.35664
## 1 66.09445 0.4500000          699.7167 0.4166667          33.86667
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3636364 264906.9          1.209021      137.4266 0.6363636
## 1          0.4333333 267540.5          1.827000      135.9167 0.6166667
##      smoking
## 0 0.3076923
## 1 0.3166667
##
## Coefficients of linear discriminants:
##
##          LD1
## age          5.579142e-02
## anaemia        8.551157e-02
## creatinine_phosphokinase 2.635256e-04
## diabetes        2.312565e-01
## ejection_fraction -5.057272e-02
## high_blood_pressure 2.706223e-01
## platelets       -1.021914e-07
## serum_creatinine 4.616477e-01
## serum_sodium    -4.884848e-02
## sex            -4.330050e-01
## smoking        4.497766e-01
```

```
lindiscr.valid <- predict(lindiscr.model, groups[[8]], type = 'response')
lindiscr.valid <- lindiscr.valid$posterior[,2]
lindiscr.valid
```

```
##          25          33          34          37          41          45          72
## 0.74167385 0.28436338 0.19517435 0.39438063 0.74433140 0.06682350 0.20742032
##          74          75          100          103          133          135          141
## 0.17542873 0.59122459 0.20573078 0.56954922 0.11123207 0.78198965 0.38652510
##          142          145          148          162          193          195          203
## 0.07932960 0.68132607 0.06243441 0.07442824 0.09067171 0.23491261 0.07042722
##          232          234          281          283          288          289          297
## 0.35067781 0.16684604 0.35478433 0.40918026 0.05039651 0.28856774 0.03765120
```

```
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[8]]$DEATH_EVENT))
```

```
quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model
```

```
## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##          0          1
## 0.7044335 0.2955665
##
## Group means:
##          age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.67599 0.4055944                    529.7762 0.4055944                39.35664
## 1 66.09445 0.4500000                    699.7167 0.4166667                33.86667
##  high_blood_pressure platelets serum_creatinine serum_sodium          sex
## 0          0.3636364 264906.9          1.209021      137.4266 0.6363636
## 1          0.4333333 267540.5          1.827000      135.9167 0.6166667
##          smoking
## 0 0.3076923
## 1 0.3166667
```

```
quaddiscr.valid <- predict(quaddiscr.model, groups[[8]], type = 'response')
quaddiscr.valid <- quaddiscr.valid$posterior[,2]
quaddiscr.valid
```

```
##          25          33          34          37          41          45
## 0.101890301 0.330503534 0.085879978 0.369714587 0.444381262 0.055301467
##          72          74          75          100          103          133
## 0.115793304 0.139599547 0.303407204 0.082882342 0.417648852 0.027265108
##          135          141          142          145          148          162
## 0.999928952 0.492844319 0.024102360 0.565834355 0.086593380 0.003929587
##          193          195          203          232          234          281
## 0.024153531 0.177824824 0.026467694 0.194770522 0.009312569 0.838986818
##          283          288          289          297
## 0.984754623 0.022460805 0.158064266 0.426730469
```

```
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[8]]$DEATH_EVENT))
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
```

```
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 71.50  89.46  96.05  99.50 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23645 ( N = 203 )
##
## Deviance: 202.363
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[8]], type = 'posterior')
mixeddiscr.valid <- mixeddiscr.valid[,2]
mixeddiscr.valid

##      [1] 6.807099e-01 3.006806e-01 1.297650e-01 4.325207e-01 7.043377e-01
##      [6] 5.995212e-02 2.498846e-01 1.884506e-01 5.833102e-01 2.282674e-01
##     [11] 6.097306e-01 1.173222e-01 6.504922e-01 3.931690e-01 6.753531e-02
##     [16] 5.593935e-01 7.992735e-02 1.110241e-01 8.161372e-02 2.585131e-01
##     [21] 7.632412e-02 4.559556e-01 1.146943e-01 2.803939e-01 3.369217e-01
##     [26] 2.194674e-03 2.212028e-01 2.492858e-05
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[8]]))

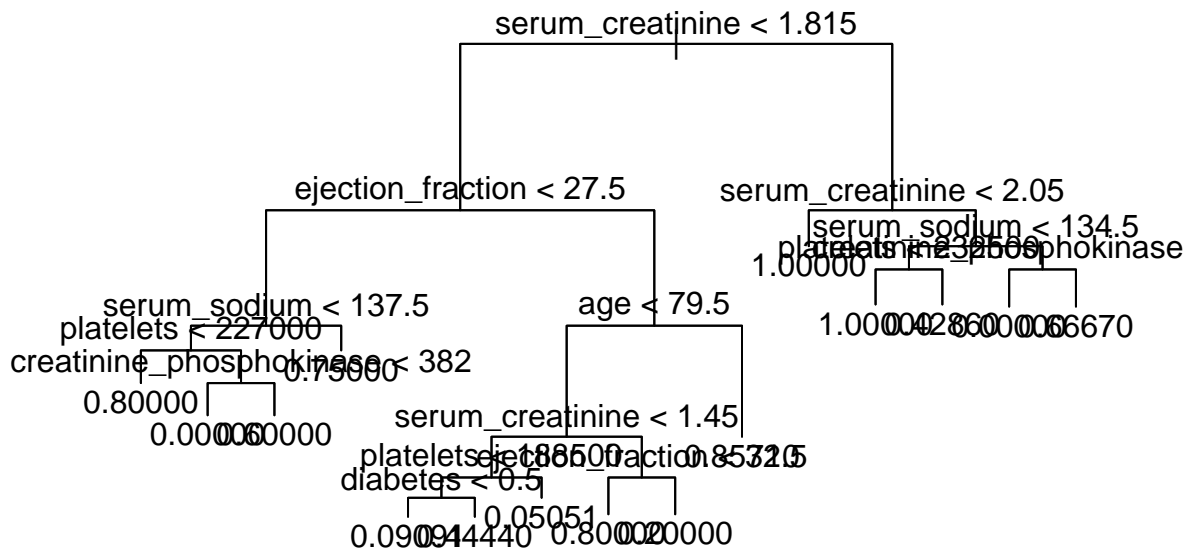
flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23153 ( N = 203 )
flexdiscr.valid <- predict(flexdiscr.model, groups[[8]], type = 'posterior')
flexdiscr.valid <- flexdiscr.valid[,2]
flexdiscr.valid

##           25           33           34           37           41           45           72
## 0.74532324 0.28425320 0.19431889 0.39542576 0.74798210 0.06573480 0.20664922
##           74           75          100          103          133          135          141
## 0.17445337 0.59419717 0.20494762 0.57234868 0.11004766 0.78560773 0.38748479
##          142          145          148          162          193          195          203
## 0.07818678 0.68483364 0.06137087 0.07330370 0.08949970 0.23435462 0.06932035
##          232          234          281          283          288          289          297
## 0.35124998 0.16582549 0.35540053 0.41038631 0.04942099 0.28849860 0.03680511
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[8]]))

decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
```

```
text(decisiontree.model, pretty = 0)
```



```
decisiontree.valid <- predict(decisiontree.model, groups[[8]], type = 'vector')
decisiontree.valid
```

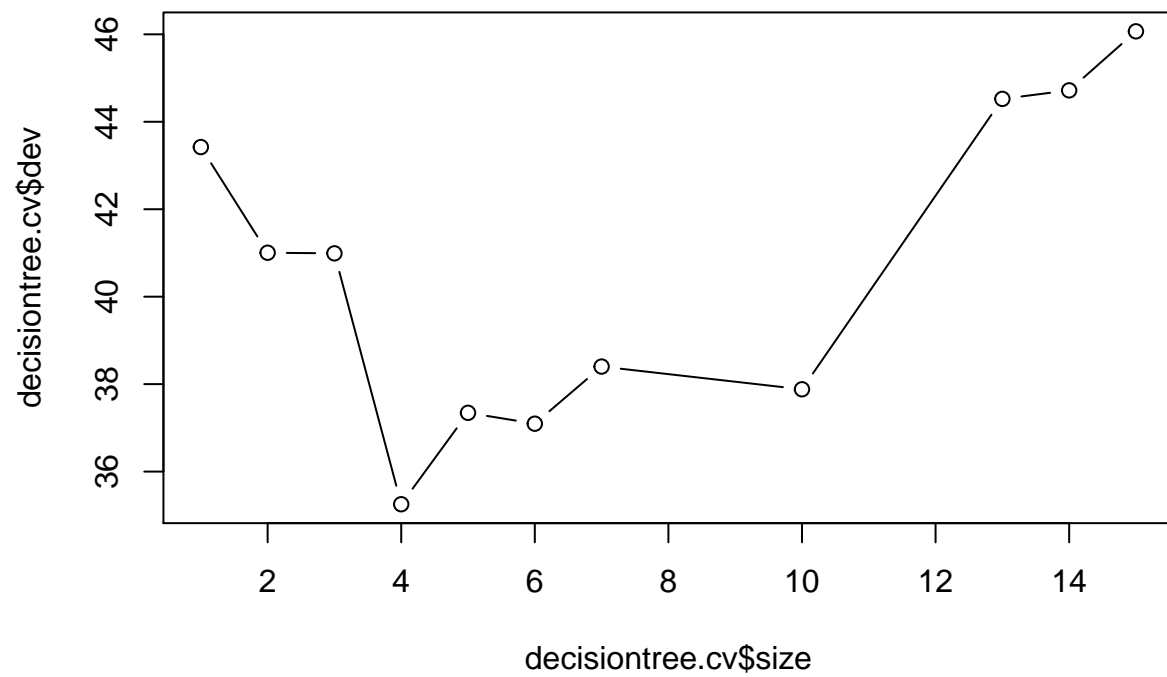
```
##      25      33      34      37      41      45      72
## 1.00000000 0.05050505 0.05050505 0.85714286 1.00000000 0.05050505 0.44444444
##      74      75      100     103     133     135     141
## 0.44444444 0.60000000 0.05050505 0.75000000 0.05050505 0.85714286 0.85714286
##      142     145     148     162     193     195     203
## 0.05050505 0.75000000 0.05050505 0.09090909 0.05050505 0.80000000 0.05050505
##      232     234     281     283     288     289     297
## 0.09090909 0.05050505 0.05050505 1.00000000 0.05050505 0.05050505 0.05050505
```

```
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - decisiontree.valid)^2))/n)
```

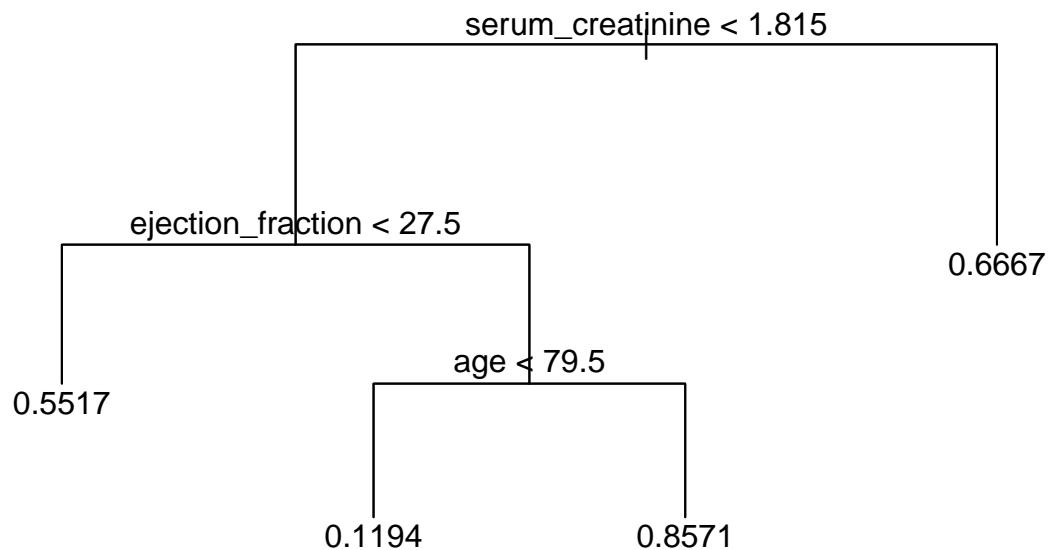
```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```

```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```
prunedectree.valid <- predict(prunedectree.model, groups[[8]], type = 'vector')
prunedectree.valid
```

```
##      25      33      34      37      41      45      72      74
## 0.6666667 0.1194030 0.1194030 0.8571429 0.6666667 0.1194030 0.1194030 0.1194030
##      75      100      103      133      135      141      142      145
## 0.5517241 0.1194030 0.5517241 0.1194030 0.8571429 0.8571429 0.1194030 0.5517241
##      148      162      193      195      203      232      234      281
## 0.1194030 0.1194030 0.1194030 0.5517241 0.1194030 0.1194030 0.1194030 0.1194030
##      283      288      289      297
## 0.6666667 0.1194030 0.1194030 0.1194030
```

```
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - prunedectree.valid)^2))/n)
```

```
##Model comparison over training cross-validation
sum(simplelinear.RMSE)/8
```

```
## [1] 0.08049216
```

```
sum(revisedlinear.RMSE)/8
```

```
## [1] 0.07796662
```

```
sum(generlinear.RMSE)/8
```

```
## [1] 0.08018486
```

```
sum(generaddit.RMSE)/8
```

```
## [1] 0.07540968
```

```
sum(lindiscr.RMSE)/8
```

```
## [1] 0.08101815
```

```
sum(quaddiscr.RMSE)/8
```

```
## [1] 0.0932875
```

```
sum(mixeddiscr.RMSE)/8
```

```
## [1] 0.0845517
```

```
sum(flexdiscr.RMSE)/8
```

```
## [1] 0.08106155
```

```
sum(decisiontree.RMSE)/8
```

```
## [1] 0.0848603
```

```
sum(prunedectree.RMSE)/8
```

```
## [1] 0.07600848
```

```
##Conclusions from Model Comparision over Training Cross-Validation  
##We note that our error rates are higher compared to previously; this  
##is because the longer a patient's follow-up period, the better chance  
##of survival is. However, some models seem less prone to error than  
## all of our models; our Revised Linear, General Additive, & Pruned  
##Decision Tree stand out from the rest (Average Cross-Validated RMSE of  
##7.8% or less). Let us do this again on the Test to verify these results.
```

```
##Applications over Test set (groups 9 & 10)
```

```
trainset <- data.frame()  
for (i in c(1:8)) {  
  trainset <- rbind(trainset, groups[[i]])  
}  
testset <- data.frame()  
for (i in c(9,10)) {  
  testset <- rbind(testset, groups[[i]])  
}
```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)  
simplelinear.model
```

```
##
```

```
## Call:
```

```
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
```

```
##
```

```
## Coefficients:
```

```
##          (Intercept)              age          anaemia  
##          1.423e+00          1.045e-02          3.230e-02  
## creatinine_phosphokinase          diabetes          ejection_fraction  
##          3.748e-05          4.680e-02          -9.334e-03
```

```
##      high_blood_pressure      platelets      serum_creatinine
##      8.462e-02      2.411e-08      8.338e-02
##      serum_sodium      sex      smoking
##      -1.159e-02      -7.086e-02      5.166e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7864 -0.2984 -0.1402  0.3740  0.9992
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.423e+00  9.716e-01   1.464 0.144615
## age            1.045e-02  2.402e-03   4.351 2.08e-05 ***
## anaemia        3.230e-02  5.771e-02   0.560 0.576254
## creatinine_phosphokinase 3.748e-05  2.946e-05   1.272 0.204550
## diabetes       4.680e-02  5.783e-02   0.809 0.419229
## ejection_fraction -9.334e-03  2.489e-03  -3.750 0.000226 ***
## high_blood_pressure  8.462e-02  5.780e-02   1.464 0.144597
## platelets       2.411e-08  2.879e-07   0.084 0.933333
## serum_creatinine  8.338e-02  2.979e-02   2.799 0.005587 **
## serum_sodium    -1.159e-02  6.980e-03  -1.660 0.098258 .
## sex            -7.086e-02  6.730e-02  -1.053 0.293563
## smoking         5.166e-02  6.781e-02   0.762 0.446944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4161 on 219 degrees of freedom
## Multiple R-squared:  0.2165, Adjusted R-squared:  0.1772
## F-statistic: 5.502 on 11 and 219 DF,  p-value: 9.525e-08
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.096   4.0962  23.6620 2.193e-06 ***
## anaemia       1  0.018   0.0180   0.1040 0.747392
## creatinine_phosphokinase 1  0.294   0.2943   1.7001 0.193645
## diabetes      1  0.094   0.0936   0.5407 0.462940
## ejection_fraction 1  3.287   3.2874  18.9900 2.020e-05 ***
## high_blood_pressure 1  0.322   0.3220   1.8600 0.174026
## platelets     1  0.046   0.0457   0.2638 0.608027
## serum_creatinine 1  1.700   1.6999   9.8195 0.001963 **
## serum_sodium  1  0.410   0.4102   2.3696 0.125162
## sex           1  0.110   0.1097   0.6339 0.426781
## smoking       1  0.100   0.1005   0.5805 0.446944
## Residuals    219 37.912   0.1731
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
simplelinear.test <- predict(simplelinear.model, testset)
simplelinear.test
```

```
##           2           3           4           15           29           55
## 0.38818804 0.51855369 0.29957520 0.26869937 0.57181508 0.49501608
##           60           64           65           79           86           87
## 0.65238539 -0.07088072 -0.31527138 0.54227289 -0.13363609 0.19414497
##           91           96          114          116          121          130
## 0.26052620 -0.02592737 0.16065180 0.17885790 0.18028853 0.27642073
##          131          137          156          159          160          165
## 0.04674543 0.11437955 0.35687671 0.40062774 0.21101543 0.16895796
##          168          178          200          208          218          220
## 0.47952874 -0.03331323 0.64257854 0.39646446 0.63252868 0.26850488
##          221          231          250          263          273          284
## 0.55552465 0.44702072 0.25627746 0.51272142 0.33972360 0.28979105
##          285          291           11           13           17           19
## -0.02565893 -0.11292029 0.77642899 0.11810321 0.40277512 0.50876680
##           20           23           30           31           48           83
## 0.26976044 0.37180307 0.60463315 0.73839516 0.29033317 0.58139472
##           88           92           94          122          127          139
## 0.08855490 0.18810799 0.36553080 0.43483045 0.62670626 0.31351625
##          152          163          180          185          213          225
## 0.04195964 0.24042966 0.07848513 0.33130076 0.28902578 0.28912420
##          227          242          253          262          265          266
## 0.39356826 0.40931559 0.09642755 0.27761876 0.17242763 0.10374860
##          270          299
## 0.07040634 0.07980271
```

```
simplelinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - simplelinear.test)^2))/nrow(testset)
```

```
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Coefficients:
##           (Intercept)              age  ejection_fraction  serum_creatinine
##           -0.07880           0.01035           -0.00995           0.08979
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7939 -0.2961 -0.1470  0.4008  1.0051
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.078798   0.160678  -0.490  0.62432
## age           0.010347   0.002339   4.423 1.51e-05 ***
## ejection_fraction -0.009950   0.002413  -4.123 5.25e-05 ***
## serum_creatinine 0.089788   0.029134   3.082 0.00231 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4162 on 227 degrees of freedom
## Multiple R-squared:  0.1875, Adjusted R-squared:  0.1768
## F-statistic: 17.47 on 3 and 227 DF,  p-value: 3.08e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##               Df Sum Sq Mean Sq F value    Pr(>F)
## age             1  4.096   4.0962 23.6514 2.157e-06 ***
## ejection_fraction 1  3.334   3.3338 19.2491 1.757e-05 ***
## serum_creatinine 1  1.645   1.6449  9.4977 0.002312 **
## Residuals      227 39.315   0.1732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.test <- predict(revisedlinear.model, testset)
revisedlinear.test
```

```
##           2           3           4           15           29           55
## 0.21093516 0.51147011 0.41013819 0.21947811 0.66397741 0.36143625
##           60           64           65           79           86           87
## 0.58389885 0.12833809 -0.30326971 0.37298064 0.01422709 0.24078647
##           91           96          114          116          121          130
## 0.26955030 0.01408774 0.16518754 0.21309643 0.07967542 0.42660385
##          131          137          156          159          160          165
## -0.06458333 0.07753772 0.44589814 0.41989632 0.18266997 0.18706902
##          168          178          200          208          218          220
## 0.54815462 0.02046942 0.35537254 0.50338652 0.59149573 0.20487146
##          221          231          250          263          273          284
## 0.64183320 0.39614596 0.17931912 0.47069669 0.34407746 0.31440478
##          285          291           11           13           17           19
## 0.11236323 -0.08862810 0.67825822 0.18706902 0.52408045 0.48651649
##           20           23           30           31           48           83
## 0.04117906 0.35733947 0.57888537 0.68001158 0.21777621 0.51772816
##           88           92           94          122          127          139
## 0.06855897 0.18403818 0.44589814 0.31577299 0.41655915 0.30423645
##          152          163          180          185          213          225
## 0.04649683 0.25585249 0.15511928 0.38031045 0.35644633 0.36235295
##          227          242          253          262          265          266
## 0.38928920 0.41196577 0.08056856 0.22754802 0.28199569 0.17109415
##          270          299
## 0.07660328 0.13444107
```

```
revisedlinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - revisedlinear.test)^2))/nrow(testset)
```

```
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
```

```
generlinear.model
```

```
##
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##        ejection_fraction + high_blood_pressure + serum_creatinine +
##        serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##              (Intercept)                age                anaemia
##              3.6165550                0.0586927                0.2615002
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##              0.0001998                -0.0580905                0.5620745
##              serum_creatinine      serum_sodium
##              0.5317051                -0.0528194
##
## Degrees of Freedom: 230 Total (i.e. Null);  223 Residual
## Null Deviance:      281.7
## Residual Deviance: 228.6    AIC: 244.6
```

```
summary(generlinear.model)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##        ejection_fraction + high_blood_pressure + serum_creatinine +
##        serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9688  -0.7403  -0.4847   0.8420   2.4676
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.6165550   5.5244721   0.655 0.512698
## age              0.0586927   0.0146583   4.004 6.23e-05 ***
## anaemia          0.2615002   0.3416900   0.765 0.444085
## creatinine_phosphokinase 0.0001998 0.0001631   1.225 0.220449
## ejection_fraction  -0.0580905 0.0164650  -3.528 0.000419 ***
## high_blood_pressure  0.5620745 0.3365968   1.670 0.094944 .
## serum_creatinine    0.5317051 0.1975575   2.691 0.007115 **
## serum_sodium       -0.0528194 0.0404205  -1.307 0.191299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 281.71  on 230  degrees of freedom
## Residual deviance: 228.57  on 223  degrees of freedom
## AIC: 244.57
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
```

```
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##           Df Deviance Resid. Df Resid. Dev
## NULL                      230      281.71
## age                1  19.8254      229      261.88
## anaemia             1   0.1074      228      261.78
## creatinine_phosphokinase 1   1.4760      227      260.30
## ejection_fraction    1  18.0546      226      242.25
## high_blood_pressure   1   1.8036      225      240.44
## serum_creatinine      1  10.1363      224      230.31
## serum_sodium          1   1.7392      223      228.57

generlinear.test <- predict(generlinear.model, testset, type = 'response')
generlinear.test

##           2           3           4           15           29           55
## 0.403611217 0.544003081 0.365180699 0.237213050 0.748923941 0.363948808
##           60           64           65           79           86           87
## 0.694919630 0.058072466 0.007524433 0.518716342 0.035517405 0.219527687
##           91           96          114          116          121          130
## 0.163296188 0.073756575 0.117601580 0.145338857 0.153226931 0.301229338
##          131          137          156          159          160          165
## 0.076550794 0.073218273 0.378177249 0.439716164 0.191188356 0.147542708
##          168          178          200          208          218          220
## 0.532231368 0.047437564 0.583080921 0.434249578 0.764520857 0.177606354
##          221          231          250          263          273          284
## 0.679362914 0.407192836 0.143877186 0.555537034 0.268384688 0.241819413
##          285          291           11           13           17           19
## 0.074309475 0.021113242 0.865100773 0.157251151 0.472648591 0.564226650
##           20           23           30           31           48           83
## 0.146256491 0.383636158 0.590618923 0.817550432 0.204079591 0.579239566
##           88           92           94          122          127          139
## 0.101012156 0.143076361 0.438172410 0.370114253 0.620462077 0.202097512
##          152          163          180          185          213          225
## 0.074109595 0.199513891 0.088296178 0.323166458 0.229726391 0.254910720
##          227          242          253          262          265          266
## 0.391842013 0.334691041 0.125086957 0.209406973 0.153968851 0.111342627
##          270          299
## 0.083704301 0.086557087

generlinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - generlinear.test)^2))/nrow(testset)

generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
```



```
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.83 1.00 3.50 3.07 total = 11.4
##
## REML score: 124.8923
summary(generaddit.model)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##       s(serum_creatinine)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29870    0.02532   11.79  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df      F p-value
## s(age)          2.831  3.552 6.561 0.000115 ***
## s(creatinine_phosphokinase) 1.000  1.001 1.844 0.175952
## s(ejection_fraction)      3.499  4.320 9.410 2.57e-07 ***
## s(serum_creatinine)       3.069  3.802 4.310 0.002711 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.296 Deviance explained = 32.8%
## -REML = 124.89 Scale est. = 0.14814 n = 231
```

```
generaddit.test <- predict(generaddit.model, testset, type = 'response')
generaddit.test
```

```
##           2           3           4           15           29           55
## 0.380217303 0.681139941 0.773014632 0.170836628 0.272358764 0.317459458
##           60           64           65           79           86           87
## 0.766036809 0.059548935 0.304093205 0.122358504 0.046104940 0.107998199
##           91           96           114          116          121          130
## 0.125794362 0.125137988 0.265968269 0.086884955 0.273568785 0.349020064
##          131          137          156          159          160          165
## 0.104700742 0.114233331 0.542323534 0.638593700 0.141722139 0.257750448
##          168          178          200          208          218          220
## 0.840486064 0.114155284 0.315090598 0.446781881 0.771087055 0.025405609
##          221          231          250          263          273          284
## 0.905312157 0.361752390 0.125270535 0.492870612 0.157067015 0.172169547
##          285          291           11           13           17           19
## 0.008239136 0.048498745 0.449640587 0.203767851 0.499944842 0.434104640
##          20          23          30          31          48          83
```

```
## 0.320623740 0.106396990 0.557663776 0.927741771 -0.011761165 0.630982193
##          88          92          94          122          127          139
## 0.088984168 0.145647860 0.539478551 0.092886575 0.915505882 0.099522568
##          152          163          180          185          213          225
## 0.101560815 0.076716848 0.023611659 0.421733011 0.456293428 0.386745672
##          227          242          253          262          265          266
## 0.443564927 0.302286731 0.104794165 0.025491566 0.143038017 0.053977691
##          270          299
## 0.017322748 0.248976803
```

```
generaddit.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - generaddit.test)^2)/nrow(testset))
```

```
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model
```

```
## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7012987 0.2987013
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.63375 0.3950617          569.6481 0.4135802          39.54321
## 1 66.32851 0.4492754          680.5217 0.4202899          33.76812
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3333333 265940.7          1.205864      137.3827 0.6358025
## 1          0.4492754 265384.1          1.767826      135.7826 0.6231884
##      smoking
## 0 0.3209877
## 1 0.3188406
##
## Coefficients of linear discriminants:
##                                LD1
## age                5.519907e-02
## anaemia            1.705953e-01
## creatinine_phosphokinase 1.979781e-04
## diabetes           2.471694e-01
## ejection_fraction   -4.930122e-02
## high_blood_pressure  4.469411e-01
## platelets           1.273423e-07
## serum_creatinine     4.403657e-01
## serum_sodium        -6.121693e-02
## sex                 -3.742431e-01
## smoking             2.728574e-01
```

```
lindiscr.test <- predict(lindiscr.model, testset, type = 'response')
lindiscr.test <- lindiscr.test$posterior[,2]
lindiscr.test
```

```
##          2          3          4          15          29          55
## 0.359798789 0.552607139 0.247593987 0.214504312 0.630165076 0.517252143
##          60          64          65          79          86          87
## 0.734893504 0.033922972 0.007959922 0.587703858 0.023471604 0.148259708
##          91          96          114          116          121          130
```

```
## 0.206303371 0.044039931 0.124485223 0.136973881 0.137998612 0.222467263
##      131      137      156      159      160      165
## 0.066691782 0.097077828 0.317482072 0.377282659 0.161594260 0.130057178
##      168      178      200      208      218      220
## 0.493872239 0.042199457 0.723193409 0.371392983 0.710878282 0.214306433
##      221      231      250      263      273      284
## 0.606953910 0.445005772 0.202132809 0.543882191 0.295467126 0.236749249
##      285      291      11      13      17      19
## 0.044108246 0.026517297 0.854315181 0.099067286 0.380334844 0.537950489
##      20      23      30      31      48      83
## 0.215586189 0.337331721 0.675058908 0.823336605 0.237341477 0.643546152
##      88      92      94      122      127      139
## 0.084237876 0.143713742 0.328915505 0.426901728 0.703596705 0.263612222
##      152      163      180      185      213      225
## 0.064914836 0.187133794 0.079662869 0.284987670 0.235914998 0.236022176
##      227      242      253      262      265      266
## 0.367318119 0.389688730 0.087979484 0.223721506 0.132446841 0.091593015
##      270      299
## 0.076157732 0.080248324
```

```
lindiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - lindiscr.test)^2))/nrow(testset)
```

```
quaddiscr.model <- qda(DEATH_EVENT ~ . - time, data = trainset)
quaddiscr.model
```

```
## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7012987 0.2987013
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.63375 0.3950617      569.6481 0.4135802      39.54321
## 1 66.32851 0.4492754      680.5217 0.4202899      33.76812
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3333333 265940.7      1.205864      137.3827 0.6358025
## 1      0.4492754 265384.1      1.767826      135.7826 0.6231884
##      smoking
## 0 0.3209877
## 1 0.3188406
```

```
quaddiscr.test <- predict(quaddiscr.model, testset, type = 'response')
quaddiscr.test <- quaddiscr.test$posterior[,2]
quaddiscr.test
```

```
##      2      3      4      15      29      55
## 0.999999997 0.704829125 0.153203504 0.109388938 0.999998675 0.469063639
##      60      64      65      79      86      87
## 0.379981098 0.020669891 0.002229068 0.811502264 0.003925512 0.048610546
##      91      96      114      116      121      130
## 0.152220532 0.002396428 0.022020184 0.030435286 0.006994977 0.978192236
##      131      137      156      159      160      165
## 0.002847065 0.144846346 0.073880684 0.466579169 0.022594425 0.181529748
```

```
##      168      178      200      208      218      220
## 0.152884134 0.013392535 0.999588276 0.658268182 1.000000000 0.051085569
##      221      231      250      263      273      284
## 0.511900591 0.656128069 0.165860283 0.180603160 0.131567029 0.059146767
##      285      291      11      13      17      19
## 0.012559617 0.110516367 0.999334736 0.026803288 0.687992853 0.241942477
##      20      23      30      31      48      83
## 0.749797211 0.143392883 0.548494373 0.366718884 0.086902203 0.309406554
##      88      92      94      122      127      139
## 0.013355649 0.045541619 0.163290420 0.201383173 0.940561426 0.097125161
##      152      163      180      185      213      225
## 0.015171590 0.074635723 0.010166118 0.064899851 0.160812352 0.040601681
##      227      242      253      262      265      266
## 0.165999835 0.100030870 0.019145142 0.094610831 0.088523678 0.010491661
##      270      299
## 0.035038271 0.004850719
```

```
quaddiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - quaddiscr.test)^2))/nrow(testset)
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~ . - time, data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 74.91 89.52 97.31 99.53 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.25108 ( N = 231 )
##
## Deviance: 229.971
```

```
mixeddiscr.test <- predict(mixeddiscr.model, testset, type = 'posterior')
mixeddiscr.test <- mixeddiscr.test[,2]
mixeddiscr.test
```

```
## [1] 0.860858289 0.583675563 0.327125661 0.163400585 0.451310894 0.477147750
## [7] 0.718552743 0.019483328 0.006686071 0.663278498 0.022445450 0.190589188
## [13] 0.235016547 0.044726745 0.114903125 0.254998630 0.086772688 0.149621250
## [19] 0.043117239 0.144702807 0.380392580 0.393647643 0.177808441 0.073504978
## [25] 0.659138964 0.126358050 0.879966549 0.428488768 0.999994409 0.170132610
## [31] 0.571188030 0.700891220 0.195074613 0.605368482 0.307123083 0.222270422
## [37] 0.052428033 0.024546974 0.897066335 0.203471717 0.387693607 0.556528411
## [43] 0.361362925 0.278010001 0.700231798 0.861106773 0.370387596 0.592069330
## [49] 0.082035270 0.147823166 0.388084583 0.536438621 0.664764576 0.246269190
## [55] 0.066112660 0.258423359 0.075315213 0.284397110 0.351423856 0.150923138
## [61] 0.390796059 0.376898982 0.099801533 0.213856524 0.181227554 0.083713641
## [67] 0.089097957 0.100681696
```

```
mixeddiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - mixeddiscr.test)^2))/nrow(testset)
```

```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.24242 ( N = 231 )

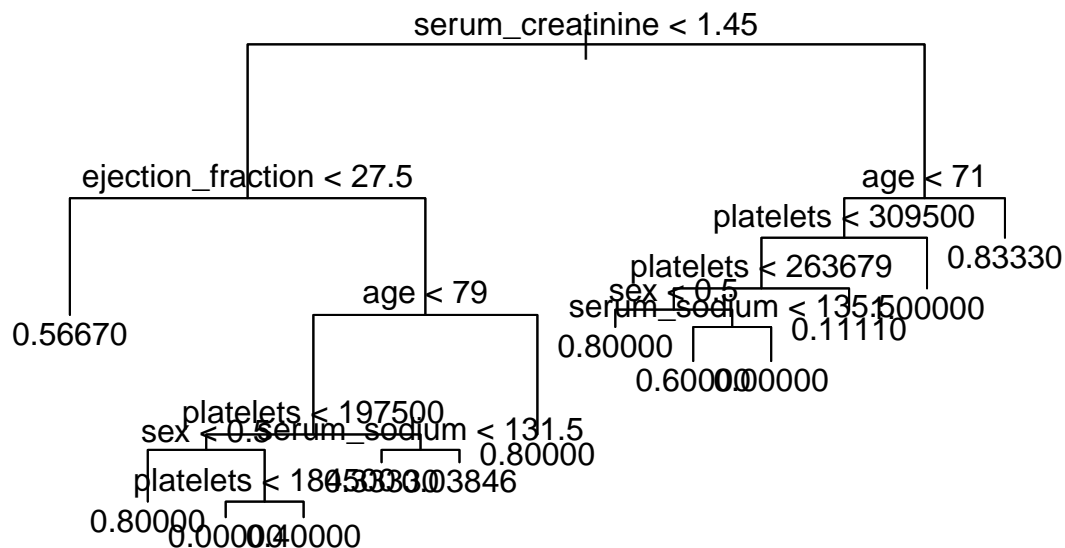
flexdiscr.test <- predict(flexdiscr.model, testset, type = 'posterior')
flexdiscr.test <- flexdiscr.test[,2]
flexdiscr.test

##           2           3           4           15           29           55
## 0.360356736 0.554904937 0.247174467 0.213850941 0.632982556 0.519263703
##           60           64           65           79           86           87
## 0.738068339 0.033215875 0.007690624 0.590257812 0.022902973 0.147275540
##           91           96          114          116          121          130
## 0.205598009 0.043229347 0.123444649 0.135957899 0.136985158 0.221866800
##          131          137          156          159          160          165
## 0.065727871 0.096029061 0.317648864 0.378005997 0.160658235 0.129026124
##          168          178          200          208          218          220
## 0.495682034 0.041405813 0.726353389 0.372060464 0.714015029 0.213651777
##          221          231          250          263          273          284
## 0.609634239 0.446370739 0.201402035 0.546111737 0.295438970 0.236249188
##          285          291           11           13           17           19
## 0.043297048 0.025904397 0.857142585 0.098016724 0.381087173 0.540132698
##           20           23           30           31           48           83
## 0.214939869 0.337679870 0.678087322 0.826355514 0.236845722 0.646434417
##           88           92           94          122          127          139
## 0.083211035 0.142715724 0.329186213 0.428096560 0.706716206 0.263317691
##          152          163          180          185          213          225
## 0.063959730 0.186317797 0.078648502 0.284869580 0.235408889 0.235516843
##          227          242          253          262          265          266
## 0.367947030 0.390530023 0.086944326 0.223129584 0.131420597 0.090551361
##          270          299
## 0.075154723 0.079232215

flexdiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - flexdiscr.test)^2))/nrow(testset)

decisiontree.model <- tree(DEATH_EVENT ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



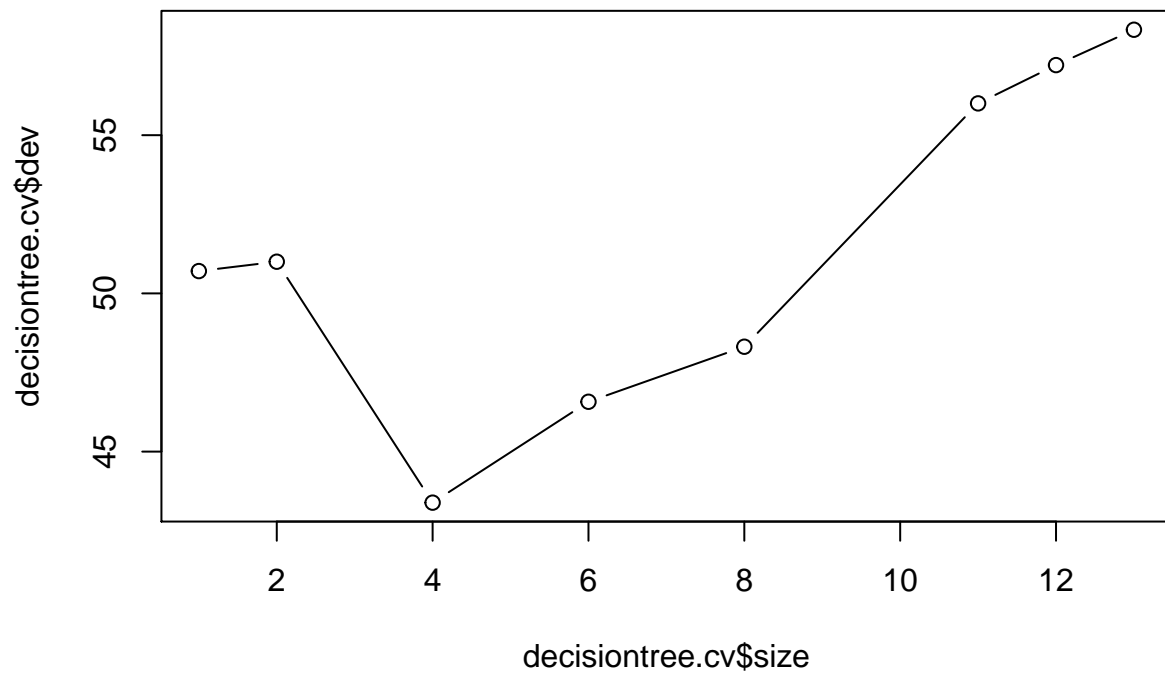
```
decisiontree.test <- predict(decisiontree.model, testset, type = 'vector')
decisiontree.test
```

```
##      2      3      4      15      29      55      60
## 0.03846154 0.56666667 0.00000000 0.03846154 0.60000000 0.80000000 0.56666667
##      64      65      79      86      87      91      96
## 0.03846154 0.03846154 0.33333333 0.03846154 0.00000000 0.03846154 0.03846154
##      114     116     121     130     131     137     156
## 0.03846154 0.80000000 0.60000000 0.00000000 0.03846154 0.80000000 0.00000000
##      159     160     165     168     178     200     208
## 0.80000000 0.03846154 0.03846154 0.60000000 0.80000000 0.60000000 0.80000000
##      218     220     221     231     250     263     273
## 0.80000000 0.03846154 0.83333333 0.80000000 0.33333333 0.56666667 0.03846154
##      284     285     291      11      13      17      19
## 0.03846154 0.03846154 0.03846154 0.83333333 0.00000000 0.80000000 0.56666667
##      20      23      30      31      48      83      88
## 0.80000000 0.03846154 0.80000000 0.83333333 0.03846154 0.80000000 0.03846154
##      92      94     122     127     139     152     163
## 0.03846154 0.60000000 0.80000000 0.11111111 0.03846154 0.03846154 0.03846154
##      180     185     213     225     227     242     253
## 0.03846154 0.56666667 0.03846154 0.56666667 0.56666667 0.03846154 0.03846154
##      262     265     266     270     299
## 0.03846154 0.00000000 0.03846154 0.03846154 1.00000000
```

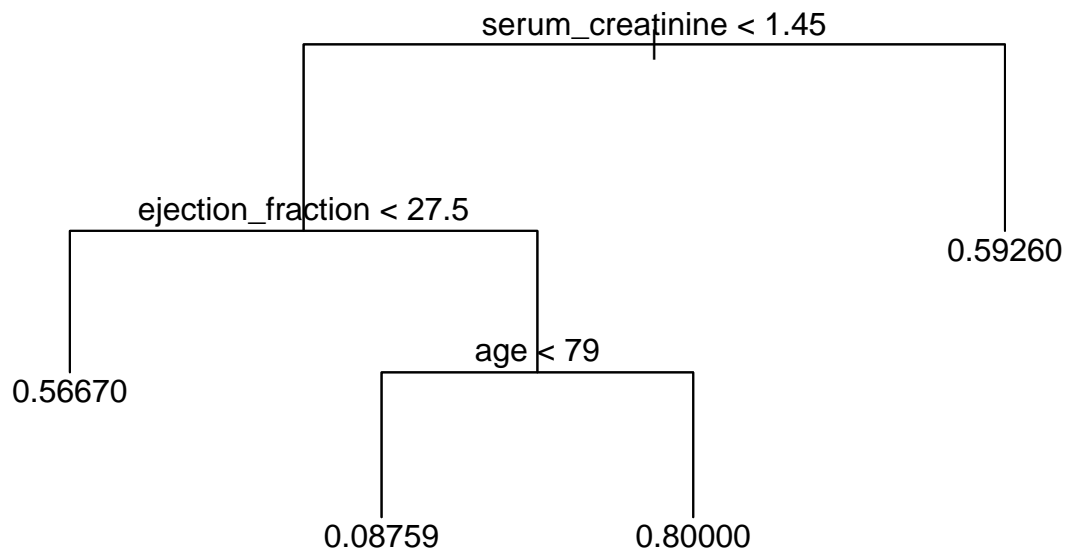
```
decisiontree.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - decisiontree.test)^2))/nrow(testset)
```

```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree.cv$dev)])
plot(prunedectree.model)
text(prunedectree.model, pretty = 0)
```



```
prunedectree.test <- predict(prunedectree.model, testset, type = 'vector')
prunedectree.test
```

```
##          2          3          4          15          29          55          60
## 0.08759124 0.56666667 0.59259259 0.08759124 0.59259259 0.59259259 0.56666667
##          64          65          79          86          87          91          96
## 0.08759124 0.08759124 0.08759124 0.08759124 0.08759124 0.08759124 0.08759124
##          114         116         121         130         131         137         156
## 0.08759124 0.08759124 0.59259259 0.59259259 0.08759124 0.08759124 0.59259259
##          159         160         165         168         178         200         208
## 0.80000000 0.08759124 0.08759124 0.59259259 0.08759124 0.59259259 0.80000000
##          218         220         221         231         250         263         273
## 0.59259259 0.08759124 0.59259259 0.59259259 0.08759124 0.56666667 0.08759124
##          284         285         291          11          13          17          19
## 0.08759124 0.08759124 0.08759124 0.59259259 0.08759124 0.80000000 0.56666667
##          20          23          30          31          48          83          88
## 0.59259259 0.08759124 0.80000000 0.59259259 0.08759124 0.59259259 0.08759124
##          92          94         122         127         139         152         163
## 0.08759124 0.59259259 0.08759124 0.59259259 0.08759124 0.08759124 0.08759124
##          180         185         213         225         227         242         253
## 0.08759124 0.56666667 0.08759124 0.56666667 0.56666667 0.08759124 0.08759124
##          262         265         266         270         299
## 0.08759124 0.08759124 0.08759124 0.08759124 0.59259259
```

```
prunedectree.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - prunedectree.test)^2))/nrow(testset)
```

```
##Model Comparison over Test Set
```



```
lindiscr.test.RMSE
```

```
## [1] 0.05095655
```

```
revisedlinear.test.RMSE
```

```
## [1] 0.05252098
```

```
generlinear.test.RMSE
```

```
## [1] 0.04982938
```

```
generaddit.test.RMSE
```

```
## [1] 0.04947659
```

```
lindiscr.test.RMSE
```

```
## [1] 0.05095655
```

```
quaddiscr.test.RMSE
```

```
## [1] 0.05742224
```

```
mixeddiscr.test.RMSE
```

```
## [1] 0.04980596
```

```
flexdiscr.test.RMSE
```

```
## [1] 0.05092428
```

```
decisiontree.test.RMSE
```

```
## [1] 0.05987955
```

```
prunedectree.test.RMSE
```

```
## [1] 0.05287502
```

```
##Conclusions from Testing
```

```
##We can note an inmpovement for all models here with the worst  
##being the Original Decision Tree with a 5.99% RMSE. While many  
##of our models are close to its RMSE, the best model we have  
##is our Generalized Additive model, since it had a low error score  
##in our cross-validation as well as in our test phase (others either  
##have a great test score, with unremarkable cross-validation results,  
##or great cross-validation results with a test score in the same range  
##as other models)
```

```
##However what we get our approximate percentages of dying from Heart  
##Failure according to biological indicators used as predictors (they are  
##approximate because some cases may give us negative probabilities or  
##probabilities above 100% [which could be categorised as impossible to  
##die from Heart Failure or death from Heart Failure inevitable respectively])  
##Thus, instead of relying on these approximate scores, what if we tried  
##classification instead.
```

```
##Classification section:
```

```
##Set up error metrics
```

```
simplelinear.RMSE <- c()
```

```
simplelinear accur <- c()
```

```

simplelinear.precis <- c()
simplelinear.recall <- c()
simplelinear.F1score <- c()
simplelinear.AUCsc <- c()
simplelinear.ROCcurv <- vector(mode = 'list', length = 8)
simplelinear.ConfMat <- vector(mode = 'list', length = 8)

revisedlinear.RMSE <- c()
revisedlinear accur <- c()
revisedlinear.precis <- c()
revisedlinear.recall <- c()
revisedlinear.F1score <- c()
revisedlinear.AUCsc <- c()
revisedlinear.ROCcurv <- vector(mode = 'list', length = 8)
revisedlinear.ConfMat <- vector(mode = 'list', length = 8)

generlinear.RMSE <- c()
generlinear accur <- c()
generlinear.precis <- c()
generlinear.recall <- c()
generlinear.F1score <- c()
generlinear.AUCsc <- c()
generlinear.ROCcurv <- vector(mode = 'list', length = 8)
generlinear.ConfMat <- vector(mode = 'list', length = 8)

generaddit.RMSE <- c()
generaddit accur <- c()
generaddit.precis <- c()
generaddit.recall <- c()
generaddit.F1score <- c()
generaddit.AUCsc <- c()
generaddit.ROCcurv <- vector(mode = 'list', length = 8)
generaddit.ConfMat <- vector(mode = 'list', length = 8)

lindiscr.RMSE <- c()
lindiscr accur <- c()
lindiscr.precis <- c()
lindiscr.recall <- c()
lindiscr.F1score <- c()
lindiscr.AUCsc <- c()
lindiscr.ROCcurv <- vector(mode = 'list', length = 8)
lindiscr.ConfMat <- vector(mode = 'list', length = 8)

quaddiscr.RMSE <- c()
quaddiscr accur <- c()
quaddiscr.precis <- c()
quaddiscr.recall <- c()
quaddiscr.F1score <- c()
quaddiscr.AUCsc <- c()
quaddiscr.ROCcurv <- vector(mode = 'list', length = 8)
quaddiscr.ConfMat <- vector(mode = 'list', length = 8)

mixeddiscr.RMSE <- c()

```

```

mixeddiscr accur <- c()
mixeddiscr precis <- c()
mixeddiscr recall <- c()
mixeddiscr F1score <- c()
mixeddiscr AUCsc <- c()
mixeddiscr ROCcurv <- vector(mode = 'list', length = 8)
mixeddiscr ConfMat <- vector(mode = 'list', length = 8)

flexdiscr RMSE <- c()
flexdiscr accur <- c()
flexdiscr precis <- c()
flexdiscr recall <- c()
flexdiscr F1score <- c()
flexdiscr AUCsc <- c()
flexdiscr ROCcurv <- vector(mode = 'list', length = 8)
flexdiscr ConfMat <- vector(mode = 'list', length = 8)

decisiontree RMSE <- c()
decisiontree accur <- c()
decisiontree precis <- c()
decisiontree recall <- c()
decisiontree F1score <- c()
decisiontree AUCsc <- c()
decisiontree ROCcurv <- vector(mode = 'list', length = 8)
decisiontree ConfMat <- vector(mode = 'list', length = 8)

prunedectree RMSE <- c()
prunedectree accur <- c()
prunedectree precis <- c()
prunedectree recall <- c()
prunedectree F1score <- c()
prunedectree AUCsc <- c()
prunedectree ROCcurv <- vector(mode = 'list', length = 8)
prunedectree ConfMat <- vector(mode = 'list', length = 8)

##1st fold-out training
trainset <- data.frame()
for (i in 2:8){
  trainset <- rbind(trainset, groups[[i]])
}

simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model

```

```

##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##              (Intercept)              age              anaemia
##              1.519e+00              1.104e-02              5.968e-03
## creatinine_phosphokinase              diabetes              ejection_fraction
##              4.329e-05              7.226e-02              -8.000e-03
##              high_blood_pressure              platelets              serum_creatinine

```

```
##          1.031e-01          2.574e-08          8.997e-02
##      serum_sodium          sex          smoking
##      -1.309e-02          -4.984e-02          2.943e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7923 -0.2872 -0.1336  0.3507  1.0009
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.519e+00  1.016e+00   1.494  0.13670
## age            1.104e-02  2.529e-03   4.364  2.08e-05 ***
## anaemia        5.968e-03  6.190e-02   0.096  0.92329
## creatinine_phosphokinase 4.329e-05  2.986e-05   1.450  0.14869
## diabetes       7.226e-02  6.123e-02   1.180  0.23941
## ejection_fraction -8.000e-03  2.712e-03  -2.950  0.00357 **
## high_blood_pressure  1.031e-01  6.116e-02   1.686  0.09334 .
## platelets       2.574e-08  3.001e-07   0.086  0.93172
## serum_creatinine  8.997e-02  3.296e-02   2.730  0.00692 **
## serum_sodium    -1.309e-02  7.262e-03  -1.803  0.07294 .
## sex            -4.984e-02  7.065e-02  -0.705  0.48142
## smoking        2.943e-02  7.322e-02   0.402  0.68813
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4163 on 193 degrees of freedom
## Multiple R-squared:  0.2194, Adjusted R-squared:  0.1749
## F-statistic: 4.932 on 11 and 193 DF,  p-value: 9.891e-07
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.985   3.9845 22.9917 3.244e-06 ***
## anaemia       1  0.008   0.0079  0.0454 0.8315287
## creatinine_phosphokinase 1  0.333   0.3334  1.9237 0.1670488
## diabetes      1  0.178   0.1782  1.0280 0.3118895
## ejection_fraction 1  2.138   2.1380 12.3366 0.0005531 ***
## high_blood_pressure 1  0.475   0.4755  2.7437 0.0992655 .
## platelets     1  0.064   0.0640  0.3694 0.5440584
## serum_creatinine 1  1.613   1.6134  9.3097 0.0026002 **
## serum_sodium  1  0.519   0.5186  2.9927 0.0852377 .
## sex           1  0.060   0.0599  0.3459 0.5571183
## smoking       1  0.028   0.0280  0.1616 0.6881301
## Residuals    193 33.447   0.1733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

simplelinear.valid <- predict(simplelinear.model, groups[[1]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 22 46 49 52 66 68 76 77 126 129 132 134 149 158 170 173 176 177 191 211
## 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1
## 215 223 239 259 260 268
## 0 0 0 0 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[1]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[1]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear.accur, simplelinear.ConfMat[[1]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[1]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[1]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[1]]$byClass[7])
simplelinear.ConfMat[[1]] <- simplelinear.ConfMat[[1]]$table
simplelinear.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[1]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Coefficients:
## (Intercept) age ejection_fraction serum_creatinine
## -0.149534 0.010608 -0.008638 0.096302
summary(revisedlinear.model)

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Residuals:
## Min 1Q Median 3Q Max
## -0.8056 -0.2965 -0.1555 0.3910 0.9932
##

```

```

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.149534   0.169405  -0.883   0.37845
## age           0.010608   0.002467   4.300 2.66e-05 ***
## ejection_fraction -0.008638  0.002647  -3.263  0.00130 **
## serum_creatinine  0.096302  0.032363   2.976  0.00328 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4183 on 201 degrees of freedom
## Multiple R-squared:  0.1791, Adjusted R-squared:  0.1668
## F-statistic: 14.62 on 3 and 201 DF,  p-value: 1.198e-08
anova(revisedlinear.model)

## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  3.985   3.9845 22.7683 3.51e-06 ***
## ejection_fraction  1  2.139   2.1391 12.2233 0.0005808 ***
## serum_creatinine  1  1.550   1.5496  8.8545 0.0032822 **
## Residuals       201 35.176   0.1750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

revisedlinear.valid <- predict(revisedlinear.model, groups[[1]])
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
revisedlinear.valid

## 22 46 49 52 66 68 76 77 126 129 132 134 149 158 170 173 176 177 191 211
##  0  0  1  0  1  0  0  0  0  0  1  0  1  0  0  0  0  0  1  1
## 215 223 239 259 260 268
##  0  0  0  0  0  0

revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - revisedlinear.valid)^2)))
revisedlinear.ConfMat[[1]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[1]]$DEATH_EVENT))
revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[1]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[1]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[1]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[1]]$byClass[7])
revisedlinear.ConfMat[[1]] <- revisedlinear.ConfMat[[1]]$table
revisedlinear.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[1]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1

```

```
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##       ejection_fraction + high_blood_pressure + serum_creatinine +
##       serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##                4.851519                0.060717                0.165874
## creatinine_phosphokinase    ejection_fraction    high_blood_pressure
##                0.000231                -0.048897                0.674270
##          serum_creatinine          serum_sodium
##                0.605507                -0.065952
##
## Degrees of Freedom: 204 Total (i.e. Null);  197 Residual
## Null Deviance:      249.6
## Residual Deviance: 202.1    AIC: 218.1
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##       ejection_fraction + high_blood_pressure + serum_creatinine +
##       serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0806  -0.7330  -0.4999   0.8162   2.4294
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.8515189   5.7477993   0.844   0.3986
## age           0.0607168   0.0155330   3.909 9.27e-05 ***
## anaemia       0.1658738   0.3696853   0.449   0.6537
## creatinine_phosphokinase 0.0002310   0.0001683   1.372   0.1700
## ejection_fraction -0.0488966   0.0174592  -2.801   0.0051 **
## high_blood_pressure  0.6742698   0.3574442   1.886   0.0592 .
## serum_creatinine  0.6055066   0.2279557   2.656   0.0079 **
## serum_sodium    -0.0659521   0.0420969  -1.567   0.1172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 249.60  on 204  degrees of freedom
## Residual deviance: 202.09  on 197  degrees of freedom
## AIC: 218.09
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                      204      249.60
## age                      1  19.3005      203      230.30
## anaemia                   1   0.0284      202      230.27
## creatinine_phosphokinase  1   1.7018      201      228.57
## ejection_fraction         1  11.4801      200      217.09
## high_blood_pressure        1   2.7000      199      214.39
## serum_creatinine           1   9.7780      198      204.61
## serum_sodium               1   2.5212      197      202.09
```

```
generlinear.valid <- predict(generlinear.model, groups[[1]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
## 22 46 49 52 66 68 76 77 126 129 132 134 149 158 170 173 176 177 191 211
## 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 1 1
## 215 223 239 259 260 268
## 0 0 0 0 0 0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - generlinear.valid)^2))/nrow)
generlinear.ConfMat[[1]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[1]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[1]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[1]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[1]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[1]]$byClass[7])
generlinear.ConfMat[[1]] <- generlinear.ConfMat[[1]]$table
generlinear.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, generlinear.valid)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[1]]$DEATH_EVENT, generlinear.valid))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model
```



```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.67 1.00 3.60 1.66 total = 9.93
##
## REML score: 112.7125

summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29756    0.02714   10.96  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)         2.667   3.350  7.623 3.90e-05 ***
## s(creatinine_phosphokinase) 1.000   1.000  1.652   0.2002
## s(ejection_fraction)      3.603   4.455  7.849 3.31e-06 ***
## s(serum_creatinine)       1.659   2.043  5.337   0.0055 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.281 Deviance explained = 31.3%
## -REML = 112.71 Scale est. = 0.15096 n = 205

generaddit.valid <- predict(generaddit.model, groups[[1]], type = 'response')
for (j in 1:length(generaddit.valid)) {
  if(generaddit.valid[j] < 0.5){
    generaddit.valid[j] <- 0
  }
  if(generaddit.valid[j] >= 0.5){
    generaddit.valid[j] <- 1
  }
}
generaddit.valid

## 22 46 49 52 66 68 76 77 126 129 132 134 149 158 170 173 176 177 191 211
## 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1
## 215 223 239 259 260 268
```

```

## 0 0 0 0 0 0
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[1]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[1]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[1]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[1]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[1]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[1]]$byClass[7])
generaddit.ConfMat[[1]] <- generaddit.ConfMat[[1]]$table
generaddit.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[1]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.702439 0.297561
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.37963 0.3819444          594.0833 0.4236111          39.34722
## 1 66.58470 0.4098361          734.3443 0.4426230          34.86885
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3263889 265322.2          1.187847    137.4514 0.6180556
## 1          0.4590164 267614.9          1.741803    135.8361 0.6065574
##      smoking
## 0 0.2916667
## 1 0.2950820
##
## Coefficients of linear discriminants:
##                                LD1
## age                5.805501e-02
## anaemia            3.139012e-02
## creatinine_phosphokinase 2.277064e-04
## diabetes           3.800273e-01
## ejection_fraction   -4.207352e-02
## high_blood_pressure  5.424288e-01
## platelets           1.353866e-07
## serum_creatinine     4.731912e-01
## serum_sodium        -6.886626e-02
## sex                 -2.621138e-01
## smoking             1.548031e-01

lindiscr.valid <- predict(lindiscr.model, groups[[1]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

```

```

## [1] 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0
lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[1]]))
lindiscr.ConfMat[[1]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[1]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[1]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[1]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[1]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[1]]$byClass[7])
lindiscr.ConfMat[[1]] <- lindiscr.ConfMat[[1]]$table
lindiscr.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[1]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.702439 0.297561
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.37963 0.3819444          594.0833 0.4236111          39.34722
## 1 66.58470 0.4098361          734.3443 0.4426230          34.86885
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3263889 265322.2          1.187847      137.4514 0.6180556
## 1          0.4590164 267614.9          1.741803      135.8361 0.6065574
##      smoking
## 0 0.2916667
## 1 0.2950820
quaddiscr.valid <- predict(quaddiscr.model, groups[[1]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

## [1] 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[1]]))
quaddiscr.ConfMat[[1]] <- confusionMatrix(factor(quaddiscr.valid), factor(groups[[1]]$DEATH_EVENT))
quaddiscr accur <- c(quaddiscr accur, quaddiscr.ConfMat[[1]]$overall[1])
quaddiscr.precis <- c(quaddiscr.precis, quaddiscr.ConfMat[[1]]$byClass[5])
quaddiscr.recall <- c(quaddiscr.recall, quaddiscr.ConfMat[[1]]$byClass[6])
quaddiscr.F1score <- c(quaddiscr.F1score, quaddiscr.ConfMat[[1]]$byClass[7])
quaddiscr.ConfMat[[1]] <- quaddiscr.ConfMat[[1]]$table
quaddiscr.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, quaddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

```

```
quaddiscr.AUCsc <- c(quaddiscr.AUCsc, auc(groups[[1]]$DEATH_EVENT, quaddiscr.valid))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model
```

```
## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 70.78  87.53  95.93  99.41 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23902 ( N = 205 )
##
## Deviance: 195.852
```

```
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[1]], type = 'class')
mixeddiscr.valid <- as.numeric(mixeddiscr.valid)-1
mixeddiscr.valid
```

```
## [1] 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0
```

```
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(gr
mixeddiscr.ConfMat[[1]] <- confusionMatrix(factor(mixeddiscr.valid), factor(groups[[1]]$DEATH_EVENT))
mixeddiscr accur <- c(mixeddiscr accur, mixeddiscr.ConfMat[[1]]$overall[1])
mixeddiscr.precis <- c(mixeddiscr.precis, mixeddiscr.ConfMat[[1]]$byClass[5])
mixeddiscr.recall <- c(mixeddiscr.recall, mixeddiscr.ConfMat[[1]]$byClass[6])
mixeddiscr.F1score <- c(mixeddiscr.F1score, mixeddiscr.ConfMat[[1]]$byClass[7])
mixeddiscr.ConfMat[[1]] <- mixeddiscr.ConfMat[[1]]$table
mixeddiscr.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, mixeddiscr.valid)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
mixeddiscr.AUCsc <- c(mixeddiscr.AUCsc, auc(groups[[1]]$DEATH_EVENT, mixeddiscr.valid))
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model
```

```
## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##      v1
## 100
##
```

```

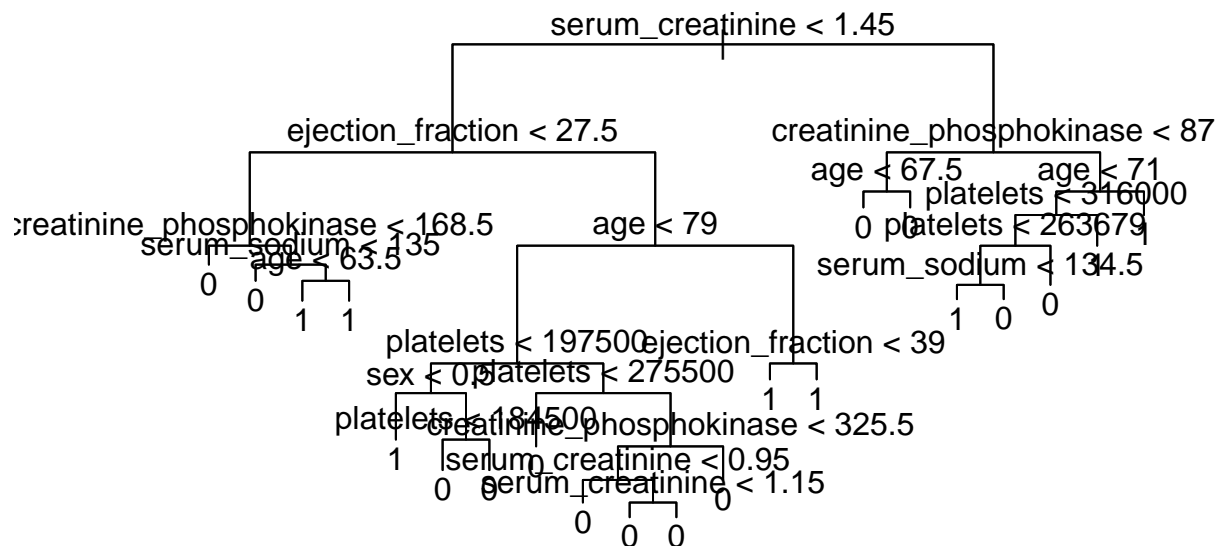
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.24878 ( N = 205 )
flexdiscr.valid <- predict(flexdiscr.model, groups[[1]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0
flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[1]]))
flexdiscr.ConfMat[[1]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[1]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[1]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[1]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[1]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[1]]$byClass[7])
flexdiscr.ConfMat[[1]] <- flexdiscr.ConfMat[[1]]$table
flexdiscr.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, flexdiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[1]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

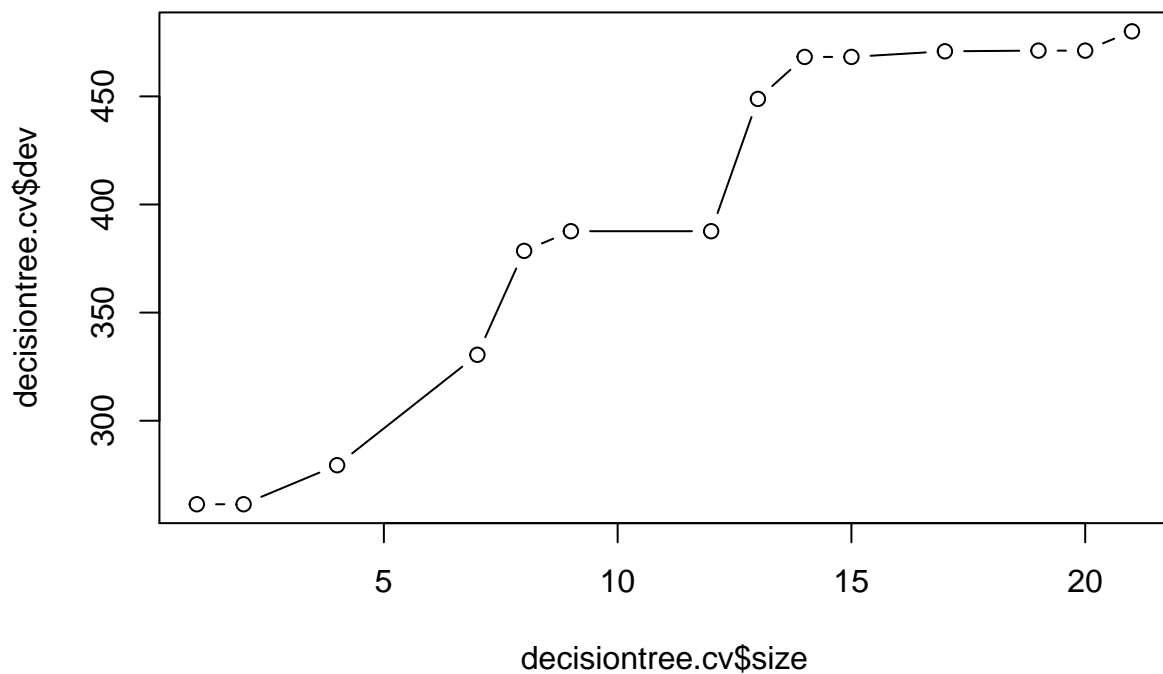
decisiontree.valid <- predict(decisiontree.model, groups[[1]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

## [1] 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 1 1 0 1 0 0 0 0
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[1]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[1]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[1]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[1]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[1]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[1]]$byClass[7])
decisiontree.ConfMat[[1]] <- decisiontree.ConfMat[[1]]$table
decisiontree.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, decisiontree.valid)

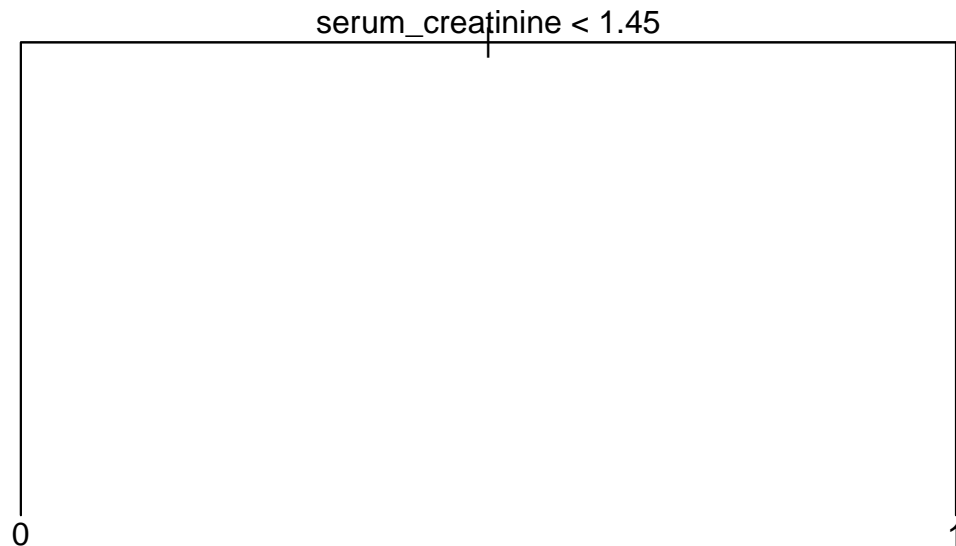
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[1]]$DEATH_EVENT, decisiontree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.valid <- predict(prunedectree.model, groups[[1]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 1 1 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[1]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[1]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[1]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[1]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[1]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[1]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[1]]$byClass[7])
prunedectree.ConfMat[[1]] <- prunedectree.ConfMat[[1]]$table
prunedectree.ROCcurv[[1]] <- roc(groups[[1]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[1]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##2nd fold-out training
trainset <- data.frame()
for (i in c(1,3:8)) {
  trainset <- rbind(trainset, groups[[i]])
}

```



```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##          (Intercept)              age          anaemia
##          1.147e+00          9.836e-03          8.057e-02
## creatinine_phosphokinase      diabetes      ejection_fraction
##          4.982e-05          4.026e-02          -1.030e-02
##      high_blood_pressure      platelets      serum_creatinine
##          2.855e-02          -3.803e-08          9.135e-02
##          serum_sodium              sex          smoking
##          -9.020e-03          -7.793e-02          6.643e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7364 -0.3004 -0.1495  0.3724  1.0071
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.147e+00  1.042e+00   1.101  0.272250
## age            9.836e-03  2.497e-03   3.940  0.000114 ***
## anaemia        8.057e-02  6.088e-02   1.323  0.187284
## creatinine_phosphokinase 4.982e-05  3.223e-05   1.546  0.123834
## diabetes       4.026e-02  6.275e-02   0.642  0.521922
## ejection_fraction -1.030e-02  2.564e-03 -4.018  8.44e-05 ***
## high_blood_pressure  2.855e-02  6.165e-02   0.463  0.643884
## platelets       -3.803e-08  2.951e-07 -0.129  0.897583
## serum_creatinine  9.135e-02  3.082e-02   2.964  0.003421 **
## serum_sodium    -9.020e-03  7.411e-03  -1.217  0.225065
## sex            -7.793e-02  7.108e-02  -1.096  0.274312
## smoking         6.643e-02  7.339e-02   0.905  0.366545
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4121 on 191 degrees of freedom
## Multiple R-squared:  0.2325, Adjusted R-squared:  0.1883
## F-statistic: 5.261 on 11 and 191 DF, p-value: 3.08e-07
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.530   3.5301 20.7858 9.158e-06 ***
```

```
## anaemia 1 0.233 0.2334 1.3742 0.242554
## creatinine_phosphokinase 1 0.521 0.5213 3.0693 0.081389 .
## diabetes 1 0.057 0.0566 0.3333 0.564405
## ejection_fraction 1 3.338 3.3384 19.6570 1.561e-05 ***
## high_blood_pressure 1 0.019 0.0195 0.1146 0.735302
## platelets 1 0.018 0.0179 0.1053 0.745943
## serum_creatinine 1 1.680 1.6795 9.8894 0.001928 **
## serum_sodium 1 0.188 0.1879 1.1061 0.294260
## sex 1 0.105 0.1046 0.6159 0.433541
## smoking 1 0.139 0.1391 0.8192 0.366545
## Residuals 191 32.438 0.1698
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[2]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 1 28 43 47 54 62 70 97 104 138 140 143 151 153 154 169 171 172 179 183
## 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0
## 194 197 205 229 237 256 261 294
## 0 0 0 1 0 0 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[2]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[2]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear accur, simplelinear.ConfMat[[2]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[2]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[2]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[2]]$byClass[7])
simplelinear.ConfMat[[2]] <- simplelinear.ConfMat[[2]]$table
simplelinear.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[2]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Coefficients:
## (Intercept) age ejection_fraction serum_creatinine
```

```
##           -0.032934           0.009929           -0.010741           0.096445
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6678 -0.2958 -0.1306  0.3846  1.0156
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.032934   0.166519  -0.198  0.84342
## age             0.009929   0.002415   4.112 5.74e-05 ***
## ejection_fraction -0.010741  0.002489  -4.315 2.52e-05 ***
## serum_creatinine  0.096445   0.030094   3.205  0.00157 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4107 on 199 degrees of freedom
## Multiple R-squared:  0.206, Adjusted R-squared:  0.194
## F-statistic: 17.21 on 3 and 199 DF, p-value: 5.622e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  3.530   3.5301  20.933 8.359e-06 ***
## ejection_fraction  1  3.445   3.4449  20.428 1.061e-05 ***
## serum_creatinine  1  1.732   1.7321  10.271 0.001574 **
## Residuals       199 33.559   0.1686
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[2]])
```

```
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
```

```
revisedlinear.valid
```

```
##      1  28  43  47  54  62  70  97 104 138 140 143 151 153 154 169 171 172 179 183
##      1   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0
## 194 197 205 229 237 256 261 294
##      0   0   0   1   0   0   0   0
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - revisedlinear.valid)^2))
revisedlinear.ConfMat[[2]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[2]]$DEATH_EVENT))
```

```

revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[2]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[2]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[2]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[2]]$byClass[7])
revisedlinear.ConfMat[[2]] <- revisedlinear.ConfMat[[2]]$table
revisedlinear.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[2]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      2.3278620             0.0581904             0.5957786
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##      0.0002591             -0.0694638             0.2205787
##      serum_creatinine      serum_sodium
##      0.6197315             -0.0415546
##
## Degrees of Freedom: 202 Total (i.e. Null); 195 Residual
## Null Deviance: 246.5
## Residual Deviance: 194.7 AIC: 210.7
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0223  -0.7358  -0.4582   0.8058   2.5762
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.3278620  6.0848539   0.383 0.702041
## age          0.0581904  0.0156050   3.729 0.000192 ***
## anaemia      0.5957786  0.3709056   1.606 0.108212
## creatinine_phosphokinase 0.0002591  0.0001778   1.458 0.144935
## ejection_fraction -0.0694638  0.0182822  -3.800 0.000145 ***
## high_blood_pressure  0.2205787  0.3715752   0.594 0.552759
## serum_creatinine  0.6197315  0.2190429   2.829 0.004665 **

```

```
## serum_sodium          -0.0415546  0.0443780  -0.936 0.349079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.47  on 202  degrees of freedom
## Residual deviance: 194.72  on 195  degrees of freedom
## AIC: 210.72
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                202      246.47
## age              1  17.1370      201      229.33
## anaemia          1   1.2452      200      228.08
## creatinine_phosphokinase 1   2.6307      199      225.45
## ejection_fraction  1  19.0596      198      206.39
## high_blood_pressure  1   0.0602      197      206.33
## serum_creatinine    1  10.7226      196      195.61
## serum_sodium        1   0.8869      195      194.72
```

```
generlinear.valid <- predict(generlinear.model, groups[[2]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
##      1  28  43  47  54  62  70  97 104 138 140 143 151 153 154 169 171 172 179 183
##      1   0   0   0   0   0   0   1   0   1   0   0   0   0   0   0   0   0   0
## 194 197 205 229 237 256 261 294
##      0   0   0   1   0   0   0   0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - generlinear.valid)^2))/nrow
generlinear.ConfMat[[2]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[2]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[2]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[2]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[2]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[2]]$byClass[7])
generlinear.ConfMat[[2]] <- generlinear.ConfMat[[2]]$table
```

```

generlinear.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, generlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[2]]$DEATH_EVENT, generlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 3.08 1.00 3.06 1.96 total = 10.09
##
## REML score: 106.8318
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29557    0.02659   11.11  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)          3.081  3.851  7.276 2.21e-05 ***
## s(creatinine_phosphokinase) 1.000  1.001  1.951  0.16410
## s(ejection_fraction)      3.055  3.786 11.201 8.61e-08 ***
## s(serum_creatinine)       1.957  2.426  5.181  0.00489 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.314 Deviance explained = 34.5%
## -REML = 106.83 Scale est. = 0.14356 n = 203
generaddit.valid <- predict(generaddit.model, groups[[2]], type = 'response')
for (j in 1:length(generaddit.valid)) {

```

```

if(generaddit.valid[j] < 0.5){
  generaddit.valid[j] <- 0
}
if(generaddit.valid[j] >= 0.5){
  generaddit.valid[j] <- 1
}
}
generaddit.valid

##      1  28  43  47  54  62  70  97 104 138 140 143 151 153 154 169 171 172 179 183
##      1   0   0   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0
## 194 197 205 229 237 256 261 294
##      0   0   0   1   0   0   0   0
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[2]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[2]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[2]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[2]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[2]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[2]]$byClass[7])
generaddit.ConfMat[[2]] <- generaddit.ConfMat[[2]]$table
generaddit.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[2]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.66201 0.3916084          526.2867 0.3986014          40.02098
## 1 66.47778 0.5000000          719.5500 0.4000000          33.50000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3356643 267715.5          1.192378      137.3566 0.6153846
## 1          0.4000000 262375.1          1.797500      136.0500 0.6166667
##      smoking
## 0 0.3006993
## 1 0.3166667
##
## Coefficients of linear discriminants:
##                                LD1
## age                5.077408e-02
## anaemia            4.159176e-01

```

```

## creatinine_phosphokinase 2.572000e-04
## diabetes                2.078107e-01
## ejection_fraction       -5.317875e-02
## high_blood_pressure     1.473631e-01
## platelets               -1.963339e-07
## serum_creatinine        4.715391e-01
## serum_sodium            -4.656113e-02
## sex                     -4.022895e-01
## smoking                 3.429145e-01

lindiscr.valid <- predict(lindiscr.model, groups[[2]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

## [1] 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[2]]))
lindiscr.ConfMat[[2]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[2]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[2]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[2]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[2]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[2]]$byClass[7])
lindiscr.ConfMat[[2]] <- lindiscr.ConfMat[[2]]$table
lindiscr.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[2]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.66201 0.3916084          526.2867 0.3986014          40.02098
## 1 66.47778 0.5000000          719.5500 0.4000000          33.50000
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3356643 267715.5          1.192378      137.3566 0.6153846
## 1          0.4000000 262375.1          1.797500      136.0500 0.6166667
##      smoking
## 0 0.3006993
## 1 0.3166667

quaddiscr.valid <- predict(quaddiscr.model, groups[[2]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

```



```

## [1] 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[2]]$DEATH_EVENT))
quaddiscr.ConfMat[[2]] <- confusionMatrix(factor(quaddiscr.valid), factor(groups[[2]]$DEATH_EVENT))
quaddiscr accur <- c(quaddiscr accur, quaddiscr.ConfMat[[2]]$overall[1])
quaddiscr.precis <- c(quaddiscr.precis, quaddiscr.ConfMat[[2]]$byClass[5])
quaddiscr.recall <- c(quaddiscr.recall, quaddiscr.ConfMat[[2]]$byClass[6])
quaddiscr.F1score <- c(quaddiscr.F1score, quaddiscr.ConfMat[[2]]$byClass[7])
quaddiscr.ConfMat[[2]] <- quaddiscr.ConfMat[[2]]$table
quaddiscr.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, quaddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
quaddiscr.AUCsc <- c(quaddiscr.AUCsc, auc(groups[[2]]$DEATH_EVENT, quaddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 75.29 90.20 94.60 98.40 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.21182 ( N = 203 )
##
## Deviance: 187.627
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[2]], type = 'class')
mixeddiscr.valid <- as.numeric(mixeddiscr.valid)-1
mixeddiscr.valid

## [1] 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[2]]$DEATH_EVENT))
mixeddiscr.ConfMat[[2]] <- confusionMatrix(factor(mixeddiscr.valid), factor(groups[[2]]$DEATH_EVENT))
mixeddiscr accur <- c(mixeddiscr accur, mixeddiscr.ConfMat[[2]]$overall[1])
mixeddiscr.precis <- c(mixeddiscr.precis, mixeddiscr.ConfMat[[2]]$byClass[5])
mixeddiscr.recall <- c(mixeddiscr.recall, mixeddiscr.ConfMat[[2]]$byClass[6])
mixeddiscr.F1score <- c(mixeddiscr.F1score, mixeddiscr.ConfMat[[2]]$byClass[7])
mixeddiscr.ConfMat[[2]] <- mixeddiscr.ConfMat[[2]]$table
mixeddiscr.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, mixeddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.AUCsc <- c(mixeddiscr.AUCsc, auc(groups[[2]]$DEATH_EVENT, mixeddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

```

```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.24138 ( N = 203 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[2]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[2]]$DEATH_EVENT))
flexdiscr.ConfMat[[2]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[2]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[2]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[2]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[2]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[2]]$byClass[7])
flexdiscr.ConfMat[[2]] <- flexdiscr.ConfMat[[2]]$table
flexdiscr.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, flexdiscr.valid)

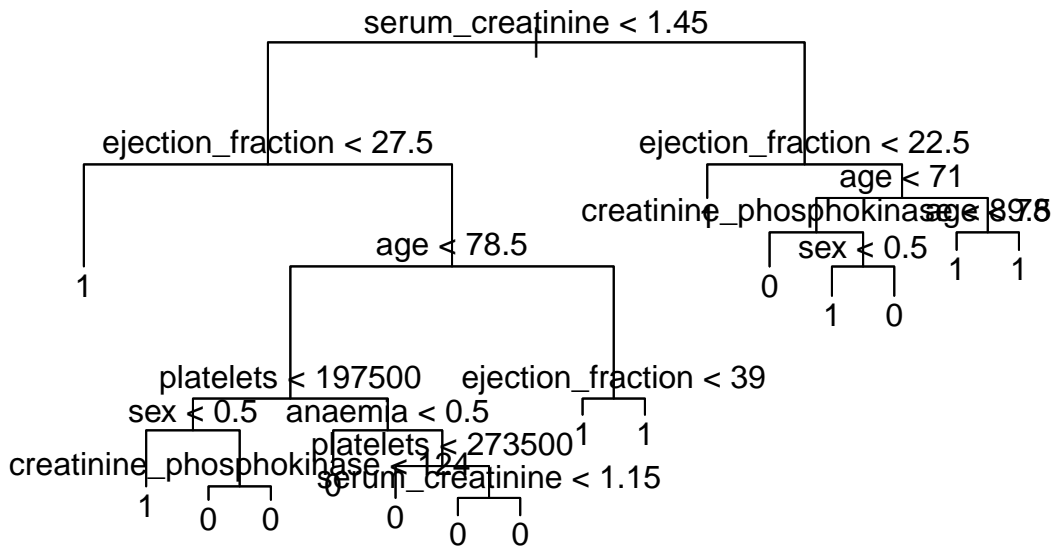
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[2]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

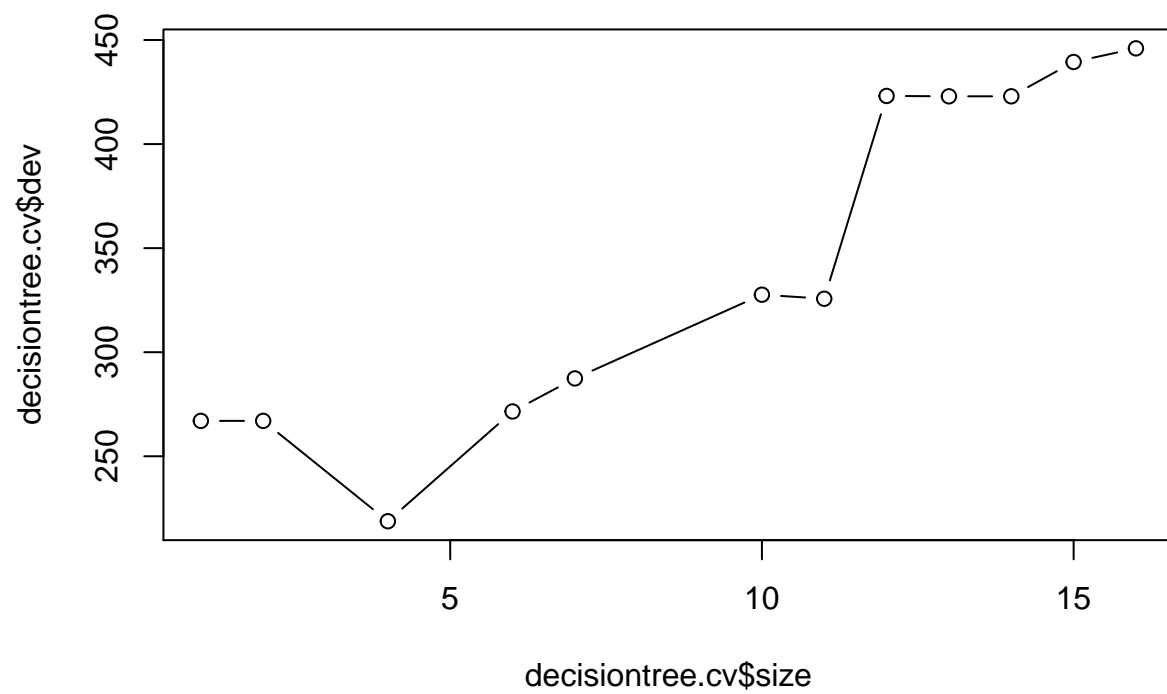
decisiontree.valid <- predict(decisiontree.model, groups[[2]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

## [1] 1 0 0 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[2]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[2]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[2]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[2]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[2]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[2]]$byClass[7])
decisiontree.ConfMat[[2]] <- decisiontree.ConfMat[[2]]$table
decisiontree.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, decisiontree.valid)

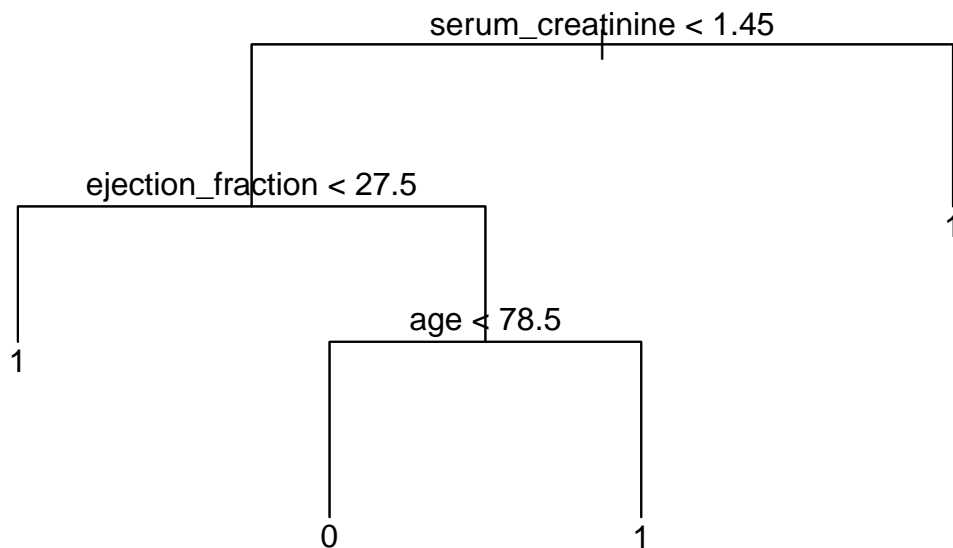
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[2]]$DEATH_EVENT, decisiontree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.valid <- predict(prunedectree.model, groups[[2]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 1 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[2]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[2]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[2]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[2]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[2]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[2]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[2]]$byClass[7])
prunedectree.ConfMat[[2]] <- prunedectree.ConfMat[[2]]$table
prunedectree.ROCcurv[[2]] <- roc(groups[[2]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[2]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##3rd fold-out training
trainset <- data.frame()
for (i in c(1:2,4:8)) {
  trainset <- rbind(trainset, groups[[i]])
}

```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##          (Intercept)                age                anaemia
##          9.674e-01            1.112e-02            3.235e-02
## creatinine_phosphokinase            diabetes            ejection_fraction
##          3.085e-05            1.886e-02            -1.103e-02
##          high_blood_pressure            platelets            serum_creatinine
##          8.187e-02            -4.004e-08            7.746e-02
##          serum_sodium                sex                smoking
##          -7.701e-03            -6.679e-02            2.485e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7930 -0.3148 -0.1125  0.3732  1.0016
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.674e-01  1.057e+00   0.915   0.3615
## age           1.112e-02  2.539e-03   4.380 1.97e-05 ***
## anaemia       3.235e-02  6.309e-02   0.513   0.6087
## creatinine_phosphokinase 3.085e-05  3.060e-05   1.008   0.3146
## diabetes      1.886e-02  6.323e-02   0.298   0.7658
## ejection_fraction -1.103e-02  2.656e-03  -4.152 5.01e-05 ***
## high_blood_pressure  8.187e-02  6.218e-02   1.317   0.1896
## platelets      -4.004e-08  3.433e-07  -0.117   0.9073
## serum_creatinine  7.746e-02  3.077e-02   2.517   0.0127 *
## serum_sodium    -7.701e-03  7.628e-03  -1.010   0.3140
## sex            -6.679e-02  7.408e-02  -0.902   0.3684
## smoking        2.485e-02  7.465e-02   0.333   0.7396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4179 on 186 degrees of freedom
## Multiple R-squared:  0.2372, Adjusted R-squared:  0.1921
## F-statistic: 5.259 on 11 and 186 DF, p-value: 3.278e-07
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.037   4.0370 23.1163 3.137e-06 ***
```

```

## anaemia 1 0.008 0.0084 0.0481 0.826603
## creatinine_phosphokinase 1 0.228 0.2281 1.3062 0.254546
## diabetes 1 0.005 0.0048 0.0273 0.868890
## ejection_fraction 1 3.930 3.9297 22.5024 4.169e-06 ***
## high_blood_pressure 1 0.267 0.2668 1.5278 0.217999
## platelets 1 0.020 0.0204 0.1168 0.732888
## serum_creatinine 1 1.308 1.3084 7.4923 0.006798 **
## serum_sodium 1 0.157 0.1572 0.9001 0.343991
## sex 1 0.123 0.1233 0.7061 0.401825
## smoking 1 0.019 0.0193 0.1108 0.739629
## Residuals 186 32.482 0.1746
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[3]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 8 9 21 40 42 57 58 85 95 105 106 108 109 110 113 146 147 155 157 174
## 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
## 217 243 247 252 257 258 272 274 277 287 290 293 295
## 0 0 0 0 0 0 0 0 0 0 0 1 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[3]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[3]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear accur, simplelinear.ConfMat[[3]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[3]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[3]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[3]]$byClass[7])
simplelinear.ConfMat[[3]] <- simplelinear.ConfMat[[3]]$table
simplelinear.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[3]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Coefficients:
## (Intercept) age ejection_fraction serum_creatinine

```

```
##           -0.03777           0.01101           -0.01150           0.08077
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7938 -0.3005 -0.1210  0.3919  1.0143
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.037771   0.171180  -0.221  0.82560
## age             0.011005   0.002454   4.484 1.25e-05 ***
## ejection_fraction -0.011505   0.002522  -4.562 8.96e-06 ***
## serum_creatinine  0.080774   0.029764   2.714  0.00725 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4142 on 194 degrees of freedom
## Multiple R-squared:  0.2186, Adjusted R-squared:  0.2065
## F-statistic: 18.09 on 3 and 194 DF,  p-value: 2.15e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  4.037   4.0370   23.536 2.512e-06 ***
## ejection_fraction  1  4.010   4.0101   23.379 2.701e-06 ***
## serum_creatinine  1  1.263   1.2633    7.365 0.007249 **
## Residuals       194 33.276   0.1715
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[3]])
```

```
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
```

```
revisedlinear.valid
```

```
##      8   9  21  40  42  57  58  85  95 105 106 108 109 110 113 146 147 155 157 174
##      0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0
## 217 243 247 252 257 258 272 274 277 287 290 293 295
##      0   0   0   0   0   0   0   0   0   0   1   0   0
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - revisedlinear.valid)^2)))
revisedlinear.ConfMat[[3]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[3]]$DEATH_EVENT))
```



```

revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[3]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[3]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[3]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[3]]$byClass[7])
revisedlinear.ConfMat[[3]] <- revisedlinear.ConfMat[[3]]$table
revisedlinear.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[3]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      0.949402              0.065301              0.311443
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##      0.000177              -0.071214              0.561488
##      serum_creatinine      serum_sodium
##      0.515844              -0.032717
##
## Degrees of Freedom: 197 Total (i.e. Null); 190 Residual
## Null Deviance: 246.1
## Residual Deviance: 193.4 AIC: 209.4
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9544  -0.7622  -0.4382   0.8507   2.5898
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.9494019  5.9395548   0.160   0.8730
## age          0.0653011  0.0159704   4.089 4.33e-05 ***
## anaemia      0.3114431  0.3726441   0.836   0.4033
## creatinine_phosphokinase 0.0001770  0.0001688   1.049   0.2944
## ejection_fraction -0.0712145  0.0180682  -3.941 8.10e-05 ***
## high_blood_pressure  0.5614878  0.3662246   1.533   0.1252
## serum_creatinine  0.5158444  0.2047519   2.519   0.0118 *

```

```
## serum_sodium          -0.0327170  0.0436223  -0.750   0.4533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.15  on 197  degrees of freedom
## Residual deviance: 193.40  on 190  degrees of freedom
## AIC: 209.4
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                197      246.15
## age              1  19.2072      196      226.94
## anaemia          1   0.0604      195      226.88
## creatinine_phosphokinase 1   1.1419      194      225.74
## ejection_fraction  1  22.1138      193      203.62
## high_blood_pressure  1   1.5039      192      202.12
## serum_creatinine    1   8.1487      191      193.97
## serum_sodium        1   0.5686      190      193.40
```

```
generlinear.valid <- predict(generlinear.model, groups[[3]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
##      8   9  21  40  42  57  58  85  95 105 106 108 109 110 113 146 147 155 157 174
##      0   0   1   0   0   1   0   0   0   0   1   0   0   0   0   0   0   0   0
## 217 243 247 252 257 258 272 274 277 287 290 293 295
##      0   0   0   0   0   0   0   0   0   0   1   0   0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - generlinear.valid)^2))/nrow
generlinear.ConfMat[[3]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[3]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[3]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[3]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[3]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[3]]$byClass[7])
generlinear.ConfMat[[3]] <- generlinear.ConfMat[[3]]$table
```

```

generlinear.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, generlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[3]]$DEATH_EVENT, generlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 3.01 1.06 3.55 2.86 total = 11.49
##
## REML score: 107.5867
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.31313    0.02712   11.55  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          3.014  3.778 6.980 4.38e-05 ***
## s(creatinine_phosphokinase) 1.056  1.109 1.275  0.2797
## s(ejection_fraction)      3.553  4.377 9.344 3.06e-07 ***
## s(serum_creatinine)       2.862  3.550 3.352  0.0139 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.327 Deviance explained = 36.2%
## -REML = 107.59 Scale est. = 0.14557 n = 198
generaddit.valid <- predict(generaddit.model, groups[[3]], type = 'response')
for (j in 1:length(generaddit.valid)) {

```

```

if(generaddit.valid[j] < 0.5){
  generaddit.valid[j] <- 0
}
if(generaddit.valid[j] >= 0.5){
  generaddit.valid[j] <- 1
}
}
generaddit.valid

##      8      9      21      40      42      57      58      85      95      105      106      108      109      110      113      146      147      155      157      174
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1      0      0      0      0      1
##    217    243    247    252    257    258    272    274    277    287    290    293    295
##      0      0      0      0      0      0      0      0      0      0      0      1      0      0

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[3]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[3]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[3]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[3]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[3]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[3]]$byClass[7])
generaddit.ConfMat[[3]] <- generaddit.ConfMat[[3]]$table
generaddit.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[3]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##          0          1
## 0.6868687 0.3131313
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.81373 0.4117647          607.5515 0.4264706          40.25735
## 1 67.02689 0.4516129          701.6129 0.4032258          33.17742
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3455882 261789.7          1.217279      137.0588 0.6470588
## 1          0.4516129 256050.8          1.796452      135.5645 0.6612903
##      smoking
## 0 0.3161765
## 1 0.3064516
##
## Coefficients of linear discriminants:
##                                LD1
## age                    5.608214e-02
## anaemia                1.631691e-01

```

```

## creatinine_phosphokinase 1.555866e-04
## diabetes                9.513530e-02
## ejection_fraction       -5.561200e-02
## high_blood_pressure     4.128612e-01
## platelets               -2.019486e-07
## serum_creatinine        3.906619e-01
## serum_sodium            -3.883829e-02
## sex                    -3.368399e-01
## smoking                 1.253098e-01

lindiscr.valid <- predict(lindiscr.model, groups[[3]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

## [1] 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[3]]))
lindiscr.ConfMat[[3]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[3]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[3]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[3]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[3]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[3]]$byClass[7])
lindiscr.ConfMat[[3]] <- lindiscr.ConfMat[[3]]$table
lindiscr.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[3]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6868687 0.3131313
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.81373 0.4117647          607.5515 0.4264706          40.25735
## 1 67.02689 0.4516129          701.6129 0.4032258          33.17742
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3455882 261789.7          1.217279      137.0588 0.6470588
## 1          0.4516129 256050.8          1.796452      135.5645 0.6612903
##      smoking
## 0 0.3161765
## 1 0.3064516

quaddiscr.valid <- predict(quaddiscr.model, groups[[3]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

```

```

## [1] 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[3]]))
quaddiscr.ConfMat[[3]] <- confusionMatrix(factor(quaddiscr.valid), factor(groups[[3]]$DEATH_EVENT))
quaddiscr accur <- c(quaddiscr accur, quaddiscr.ConfMat[[3]]$overall[1])
quaddiscr.precis <- c(quaddiscr.precis, quaddiscr.ConfMat[[3]]$byClass[5])
quaddiscr.recall <- c(quaddiscr.recall, quaddiscr.ConfMat[[3]]$byClass[6])
quaddiscr.F1score <- c(quaddiscr.F1score, quaddiscr.ConfMat[[3]]$byClass[7])
quaddiscr.ConfMat[[3]] <- quaddiscr.ConfMat[[3]]$table
quaddiscr.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, quaddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
quaddiscr.AUCsc <- c(quaddiscr.AUCsc, auc(groups[[3]]$DEATH_EVENT, quaddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 77.68 89.97 95.48 99.40 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.22727 ( N = 198 )
##
## Deviance: 174.662
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[3]], type = 'class')
mixeddiscr.valid <- as.numeric(mixeddiscr.valid)-1
mixeddiscr.valid

## [1] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[3]]))
mixeddiscr.ConfMat[[3]] <- confusionMatrix(factor(mixeddiscr.valid), factor(groups[[3]]$DEATH_EVENT))
mixeddiscr accur <- c(mixeddiscr accur, mixeddiscr.ConfMat[[3]]$overall[1])
mixeddiscr.precis <- c(mixeddiscr.precis, mixeddiscr.ConfMat[[3]]$byClass[5])
mixeddiscr.recall <- c(mixeddiscr.recall, mixeddiscr.ConfMat[[3]]$byClass[6])
mixeddiscr.F1score <- c(mixeddiscr.F1score, mixeddiscr.ConfMat[[3]]$byClass[7])
mixeddiscr.ConfMat[[3]] <- mixeddiscr.ConfMat[[3]]$table
mixeddiscr.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, mixeddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.AUCsc <- c(mixeddiscr.AUCsc, auc(groups[[3]]$DEATH_EVENT, mixeddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

```

```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.27273 ( N = 198 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[3]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[3]]$DEATH_EVENT))
flexdiscr.ConfMat[[3]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[3]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[3]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[3]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[3]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[3]]$byClass[7])
flexdiscr.ConfMat[[3]] <- flexdiscr.ConfMat[[3]]$table
flexdiscr.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, flexdiscr.valid)

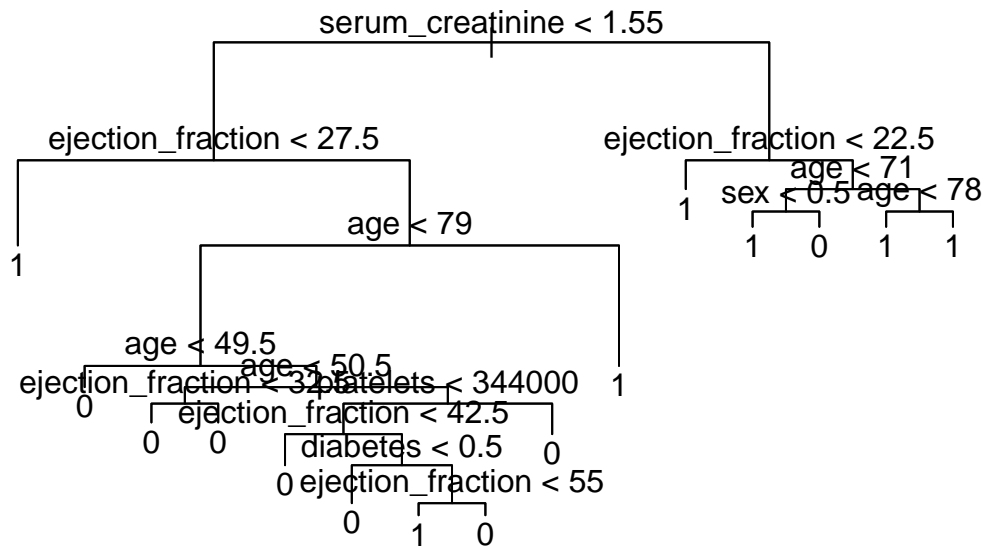
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[3]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

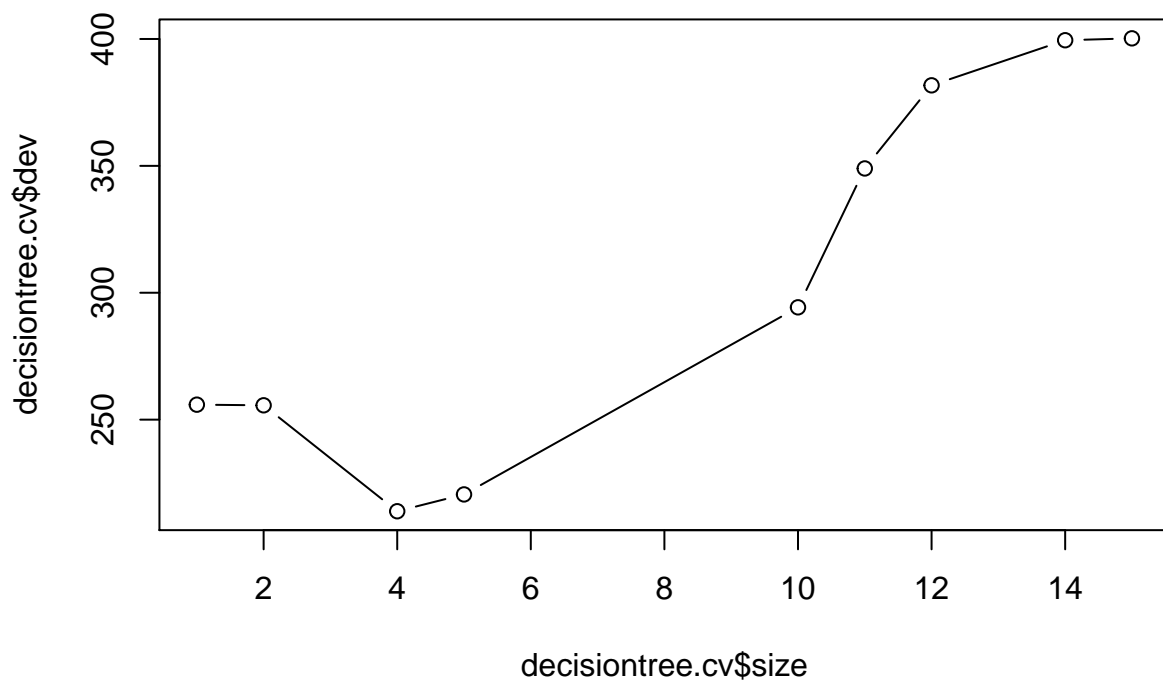
decisiontree.valid <- predict(decisiontree.model, groups[[3]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

## [1] 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 1 0 0
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[3]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[3]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[3]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[3]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[3]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[3]]$byClass[7])
decisiontree.ConfMat[[3]] <- decisiontree.ConfMat[[3]]$table
decisiontree.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, decisiontree.valid)

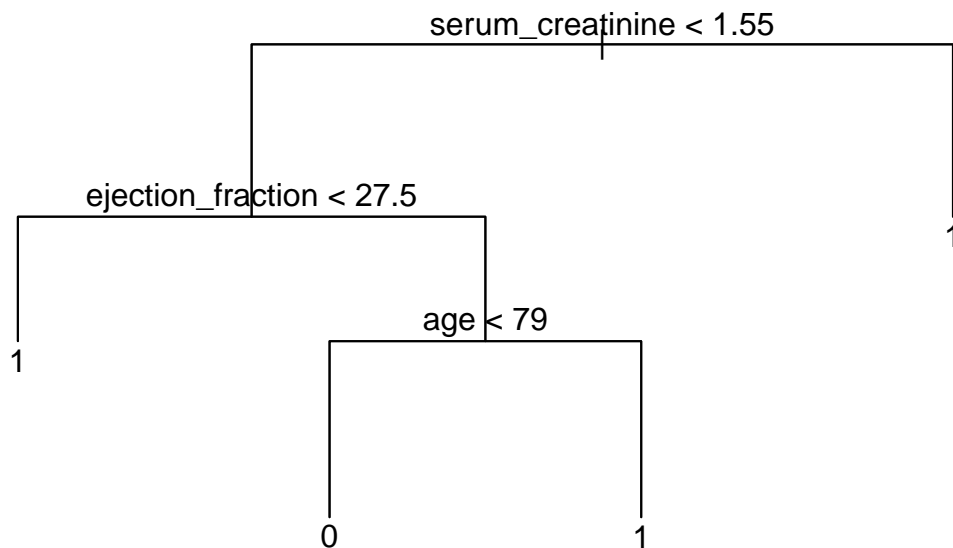
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[3]]$DEATH_EVENT, decisiontree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```

```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.valid <- predict(prunedectree.model, groups[[3]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 0 0 1 1 0 1 0 1 0 1 1 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1 1 0 0
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[3]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[3]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[3]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[3]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[3]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[3]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[3]]$byClass[7])
prunedectree.ConfMat[[3]] <- prunedectree.ConfMat[[3]]$table
prunedectree.ROCcurv[[3]] <- roc(groups[[3]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[3]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##4th fold-out training
trainset <- data.frame()
for (i in c(1:3,5:8)) {
  trainset <- rbind(trainset, groups[[i]])
}

```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             2.469e+00             9.343e-03             8.862e-02
## creatinine_phosphokinase            diabetes            ejection_fraction
##             3.376e-05             6.357e-02             -9.787e-03
##             high_blood_pressure            platelets            serum_creatinine
##             9.328e-02             5.290e-08             5.250e-02
##             serum_sodium                sex                smoking
##             -1.852e-02             -1.068e-01             9.227e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7003 -0.2881 -0.1276  0.3437  1.0183
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.469e+00  1.052e+00   2.346 0.020015 *
## age           9.343e-03  2.639e-03   3.540 0.000505 ***
## anaemia       8.862e-02  6.263e-02   1.415 0.158751
## creatinine_phosphokinase 3.376e-05  3.110e-05   1.086 0.278983
## diabetes      6.357e-02  6.296e-02   1.010 0.313927
## ejection_fraction -9.787e-03  2.723e-03  -3.594 0.000416 ***
## high_blood_pressure  9.328e-02  6.313e-02   1.478 0.141170
## platelets      5.290e-08  3.070e-07   0.172 0.863387
## serum_creatinine  5.250e-02  3.434e-02   1.529 0.127968
## serum_sodium   -1.852e-02  7.546e-03  -2.455 0.015002 *
## sex           -1.068e-01  7.400e-02  -1.444 0.150510
## smoking       9.227e-02  7.220e-02   1.278 0.202815
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4163 on 188 degrees of freedom
## Multiple R-squared:  0.2168, Adjusted R-squared:  0.1709
## F-statistic:  4.73 on 11 and 188 DF,  p-value: 2.147e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  2.612  2.61173  15.0714 0.0001432 ***
```

```

## anaemia                1  0.303 0.30289  1.7479 0.1877514
## creatinine_phosphokinase 1  0.173 0.17264  0.9962 0.3195040
## diabetes               1  0.180 0.17984  1.0378 0.3096481
## ejection_fraction      1  3.128 3.12772 18.0490 3.384e-05 ***
## high_blood_pressure    1  0.362 0.36227  2.0905 0.1498817
## platelets              1  0.052 0.05175  0.2986 0.5853943
## serum_creatinine       1  0.835 0.83475  4.8170 0.0294050 *
## serum_sodium           1  0.931 0.93053  5.3698 0.0215660 *
## sex                    1  0.159 0.15918  0.9186 0.3390739
## smoking                1  0.283 0.28305  1.6334 0.2028153
## Residuals              188 32.579 0.17329
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[4]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 32 35 36 44 51 53 56 59 78 81 93 99 115 117 118 120 144 166 196 199
## 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0 1
## 201 202 206 209 238 240 246 264 271 275 286
## 0 0 0 0 0 0 0 0 0 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[4]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[4]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear accur, simplelinear.ConfMat[[4]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[4]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[4]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[4]]$byClass[7])
simplelinear.ConfMat[[4]] <- simplelinear.ConfMat[[4]]$table
simplelinear.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[4]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Coefficients:
## (Intercept)                age ejection_fraction  serum_creatinine

```

```
##          0.009103          0.009590          -0.010422          0.069668
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7202 -0.2991 -0.1461  0.4309  0.9907
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.009103   0.177854   0.051 0.959235
## age            0.009590   0.002607   3.678 0.000303 ***
## ejection_fraction -0.010422  0.002707  -3.851 0.000160 ***
## serum_creatinine  0.069668   0.033335   2.090 0.037917 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4224 on 196 degrees of freedom
## Multiple R-squared:  0.1591, Adjusted R-squared:  0.1462
## F-statistic: 12.36 on 3 and 196 DF,  p-value: 1.938e-07
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age            1  2.612   2.6117  14.6350 0.0001753 ***
## ejection_fraction  1  3.226   3.2260  18.0772 3.281e-05 ***
## serum_creatinine  1  0.779   0.7795   4.3677 0.0379168 *
## Residuals      196 34.978   0.1785
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[4]])
```

```
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
```

```
revisedlinear.valid
```

```
## 32 35 36 44 51 53 56 59 78 81 93 99 115 117 118 120 144 166 196 199
## 1  0  1  0  0  0  1  0  0  0  0  0  0  0  0  1  0  0  0  0
## 201 202 206 209 238 240 246 264 271 275 286
## 0  0  0  0  0  0  0  0  0  0  0
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - revisedlinear.valid)^2)))
revisedlinear.ConfMat[[4]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[4]]$DEATH_EVENT))
```

```

revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[4]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[4]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[4]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[4]]$byClass[7])
revisedlinear.ConfMat[[4]] <- revisedlinear.ConfMat[[4]]$table
revisedlinear.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[4]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      10.2881739              0.0512064              0.5257437
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##      0.0001655              -0.0583792              0.5199828
##      serum_creatinine          serum_sodium
##      0.2921661              -0.0962177
##
## Degrees of Freedom: 199 Total (i.e. Null); 192 Residual
## Null Deviance: 242.6
## Residual Deviance: 199.7 AIC: 215.7
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6743  -0.7628  -0.4948   0.8324   2.4661
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  10.2881739  6.3645782   1.616  0.10599
## age          0.0512064  0.0156282   3.277  0.00105 **
## anaemia      0.5257437  0.3655187   1.438  0.15033
## creatinine_phosphokinase 0.0001655  0.0001712   0.967  0.33369
## ejection_fraction -0.0583792  0.0178972  -3.262  0.00111 **
## high_blood_pressure  0.5199828  0.3572359   1.456  0.14551
## serum_creatinine  0.2921661  0.2276989   1.283  0.19945

```

```
## serum_sodium          -0.0962177  0.0460996  -2.087  0.03687 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 242.63  on 199  degrees of freedom
## Residual deviance: 199.69  on 192  degrees of freedom
## AIC: 215.69
##
## Number of Fisher Scoring iterations: 4
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                199      242.63
## age              1  12.6604      198      229.97
## anaemia          1   1.4938      197      228.47
## creatinine_phosphokinase 1   0.8243      196      227.65
## ejection_fraction  1  16.9990      195      210.65
## high_blood_pressure  1   1.8367      194      208.81
## serum_creatinine    1   4.5305      193      204.28
## serum_sodium        1   4.5938      192      199.69
```

```
generlinear.valid <- predict(generlinear.model, groups[[4]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
## 32 35 36 44 51 53 56 59 78 81 93 99 115 117 118 120 144 166 196 199
## 1 0 1 0 1 0 1 0 0 0 0 1 1 0 0 1 0 1 0 1
## 201 202 206 209 238 240 246 264 271 275 286
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - generlinear.valid)^2))/nrow)
generlinear.ConfMat[[4]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[4]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[4]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[4]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[4]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[4]]$byClass[7])
generlinear.ConfMat[[4]] <- generlinear.ConfMat[[4]]$table
```

```

generlinear.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, generlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[4]]$DEATH_EVENT, generlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 3.13 1.27 3.37 3.33 total = 12.1
##
## REML score: 110.1503
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29500    0.02711   10.88  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          3.128  3.917  4.743  0.00117 **
## s(creatinine_phosphokinase) 1.273  1.493  0.750  0.53988
## s(ejection_fraction)      3.371  4.173  9.317  4.63e-07 ***
## s(serum_creatinine)       3.332  4.077  3.915  0.00426 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.297 Deviance explained = 33.6%
## -REML = 110.15 Scale est. = 0.14697 n = 200
generaddit.valid <- predict(generaddit.model, groups[[4]], type = 'response')
for (j in 1:length(generaddit.valid)) {

```



```

if(generaddit.valid[j] < 0.5){
  generaddit.valid[j] <- 0
}
if(generaddit.valid[j] >= 0.5){
  generaddit.valid[j] <- 1
}
}
generaddit.valid

## 32 35 36 44 51 53 56 59 78 81 93 99 115 117 118 120 144 166 196 199
## 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1
## 201 202 206 209 238 240 246 264 271 275 286
## 0 0 0 0 0 0 0 0 0 0 0 0
generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[4]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[4]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[4]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[4]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[4]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[4]]$byClass[7])
generaddit.ConfMat[[4]] <- generaddit.ConfMat[[4]]$table
generaddit.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[4]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
## 0 1
## 0.705 0.295
##
## Group means:
## age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 0 58.62175 0.3546099 591.8298 0.3758865 39.06383
## 1 65.12995 0.4745763 657.2881 0.4237288 32.91525
## high_blood_pressure platelets serum_creatinine serum_sodium sex
## 0 0.3262411 265023.6 1.204184 137.5816 0.6524823
## 1 0.4576271 265216.8 1.672034 135.5424 0.6101695
## smoking
## 0 0.3475177
## 1 0.3559322
##
## Coefficients of linear discriminants:
## LD1
## age 4.947082e-02
## anaemia 4.692593e-01

```

```

## creatinine_phosphokinase 1.787862e-04
## diabetes                 3.366242e-01
## ejection_fraction        -5.182136e-02
## high_blood_pressure      4.939434e-01
## platelets                 2.801214e-07
## serum_creatinine         2.780127e-01
## serum_sodium             -9.808889e-02
## sex                      -5.656914e-01
## smoking                   4.885990e-01

lindiscr.valid <- predict(lindiscr.model, groups[[4]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

## [1] 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[4]]))
lindiscr.ConfMat[[4]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[4]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[4]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[4]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[4]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[4]]$byClass[7])
lindiscr.ConfMat[[4]] <- lindiscr.ConfMat[[4]]$table
lindiscr.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[4]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.705 0.295
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.62175 0.3546099          591.8298 0.3758865          39.06383
## 1 65.12995 0.4745763          657.2881 0.4237288          32.91525
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3262411 265023.6          1.204184      137.5816 0.6524823
## 1          0.4576271 265216.8          1.672034      135.5424 0.6101695
##      smoking
## 0 0.3475177
## 1 0.3559322

quaddiscr.valid <- predict(quaddiscr.model, groups[[4]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

```

```

## [1] 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[4]]))
quaddiscr.ConfMat[[4]] <- confusionMatrix(factor(quaddiscr.valid), factor(groups[[4]]$DEATH_EVENT))
quaddiscr accur <- c(quaddiscr accur, quaddiscr.ConfMat[[4]]$overall[1])
quaddiscr.precis <- c(quaddiscr.precis, quaddiscr.ConfMat[[4]]$byClass[5])
quaddiscr.recall <- c(quaddiscr.recall, quaddiscr.ConfMat[[4]]$byClass[6])
quaddiscr.F1score <- c(quaddiscr.F1score, quaddiscr.ConfMat[[4]]$byClass[7])
quaddiscr.ConfMat[[4]] <- quaddiscr.ConfMat[[4]]$table
quaddiscr.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, quaddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.AUCsc <- c(quaddiscr.AUCsc, auc(groups[[4]]$DEATH_EVENT, quaddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 74.89 88.05 95.20 98.79 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.195 ( N = 200 )
##
## Deviance: 185.311

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[4]], type = 'class')
mixeddiscr.valid <- as.numeric(mixeddiscr.valid)-1
mixeddiscr.valid

## [1] 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[4]]))
mixeddiscr.ConfMat[[4]] <- confusionMatrix(factor(mixeddiscr.valid), factor(groups[[4]]$DEATH_EVENT))
mixeddiscr accur <- c(mixeddiscr accur, mixeddiscr.ConfMat[[4]]$overall[1])
mixeddiscr.precis <- c(mixeddiscr.precis, mixeddiscr.ConfMat[[4]]$byClass[5])
mixeddiscr.recall <- c(mixeddiscr.recall, mixeddiscr.ConfMat[[4]]$byClass[6])
mixeddiscr.F1score <- c(mixeddiscr.F1score, mixeddiscr.ConfMat[[4]]$byClass[7])
mixeddiscr.ConfMat[[4]] <- mixeddiscr.ConfMat[[4]]$table
mixeddiscr.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, mixeddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

mixeddiscr.AUCsc <- c(mixeddiscr.AUCsc, auc(groups[[4]]$DEATH_EVENT, mixeddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

```

```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.26 ( N = 200 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[4]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[4]]$DEATH_EVENT))
flexdiscr.ConfMat[[4]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[4]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[4]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[4]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[4]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[4]]$byClass[7])
flexdiscr.ConfMat[[4]] <- flexdiscr.ConfMat[[4]]$table
flexdiscr.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, flexdiscr.valid)

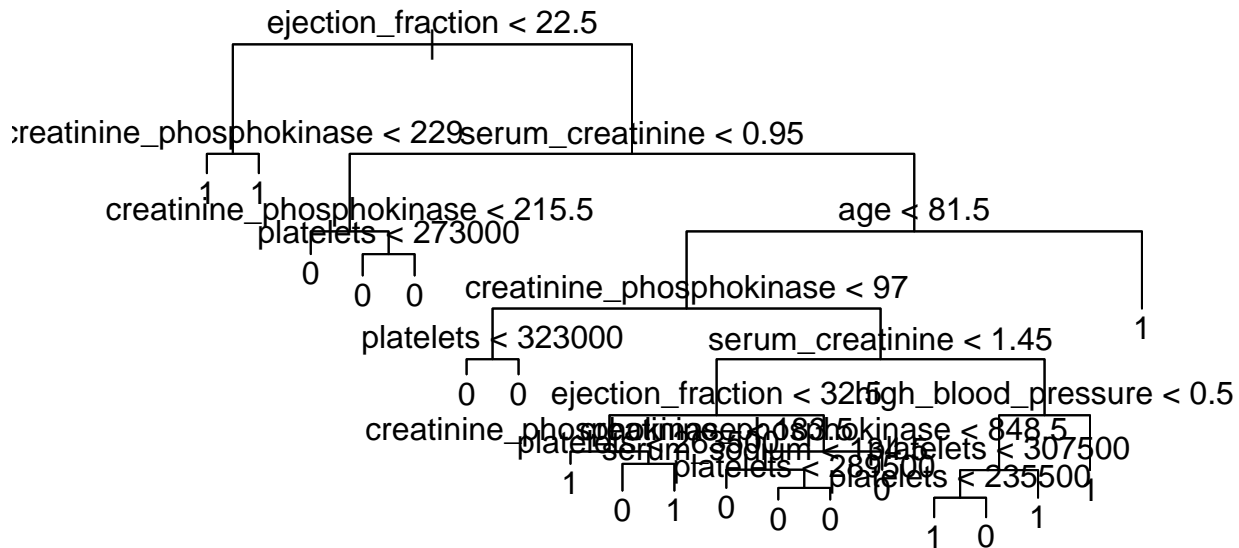
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[4]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

decisiontree.valid <- predict(decisiontree.model, groups[[4]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

```

```
## [1] 1 0 1 1 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0
```

```

decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[4]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[4]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[4]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[4]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[4]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[4]]$byClass[7])
decisiontree.ConfMat[[4]] <- decisiontree.ConfMat[[4]]$table
decisiontree.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, decisiontree.valid)

```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[4]]$DEATH_EVENT, decisiontree.valid))
```

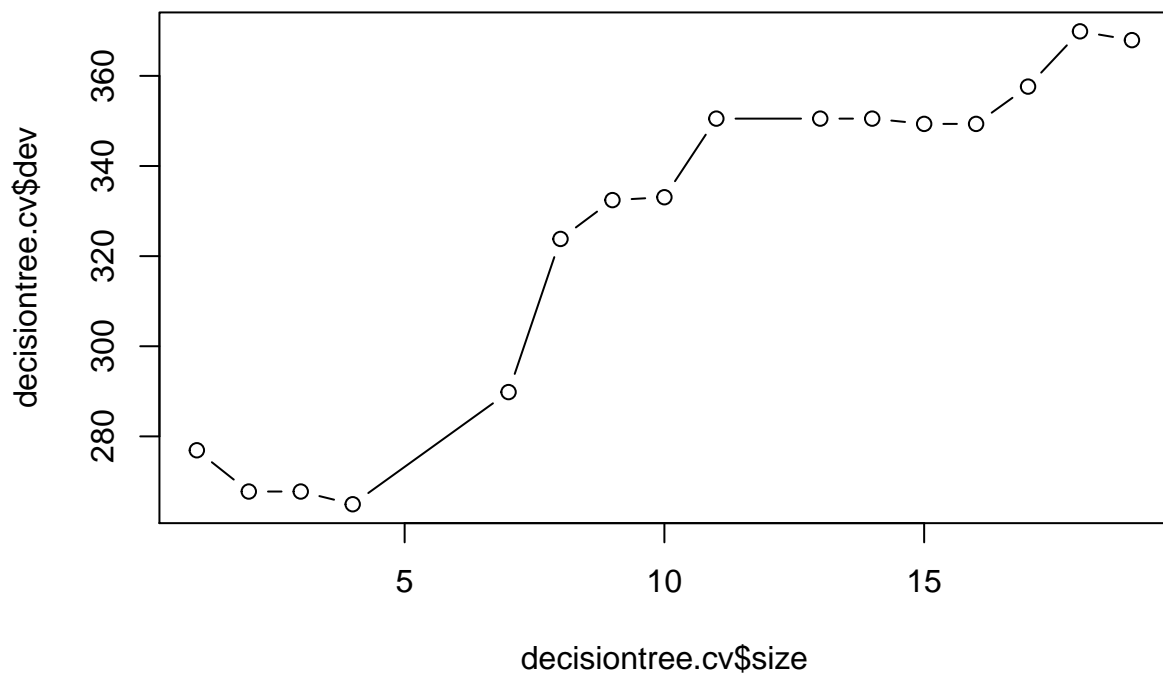
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

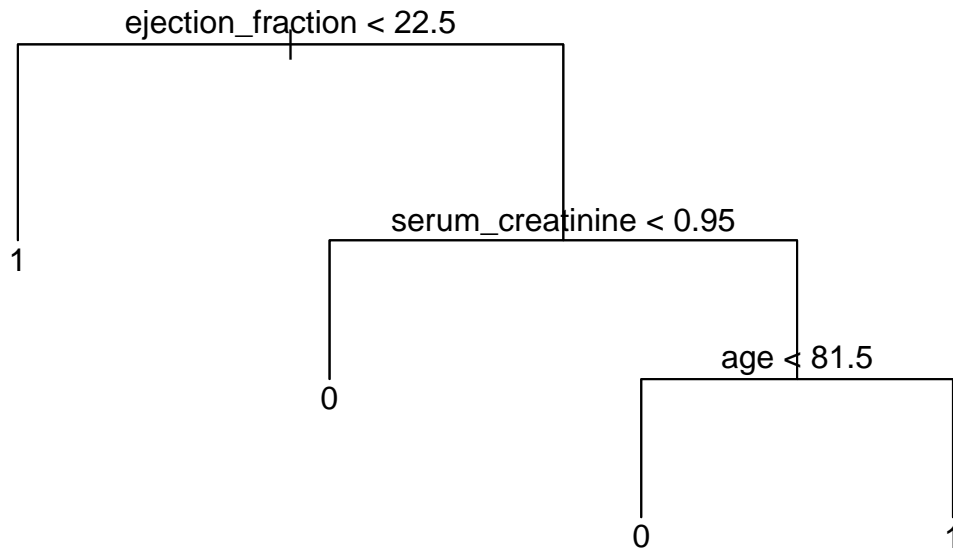
```
set.seed(5)
```

```
decisiontree.cv <- cv.tree(decisiontree.model)
```

```
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')
```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.valid <- predict(prunedectree.model, groups[[4]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[4]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[4]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[4]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[4]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[4]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[4]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[4]]$byClass[7])
prunedectree.ConfMat[[4]] <- prunedectree.ConfMat[[4]]$table
prunedectree.ROCcurv[[4]] <- roc(groups[[4]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[4]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

##5th fold-out training
trainset <- data.frame()
for (i in c(1:4,6:8)) {
  trainset <- rbind(trainset, groups[[i]])
}

```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##          (Intercept)              age              anaemia
##          1.094e+00          1.205e-02          2.001e-02
## creatinine_phosphokinase          diabetes          ejection_fraction
##          3.503e-05          5.412e-02          -9.290e-03
##          high_blood_pressure          platelets          serum_creatinine
##          1.084e-01          4.238e-08          8.266e-02
##          serum_sodium              sex              smoking
##          -9.853e-03          -9.747e-02          6.579e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7899 -0.3076 -0.1343  0.3461  1.0114
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.094e+00  1.069e+00   1.023 0.307376
## age            1.205e-02  2.486e-03   4.848 2.55e-06 ***
## anaemia        2.001e-02  6.093e-02   0.328 0.742957
## creatinine_phosphokinase 3.503e-05  3.014e-05   1.162 0.246518
## diabetes       5.412e-02  6.035e-02   0.897 0.370932
## ejection_fraction -9.290e-03  2.605e-03  -3.566 0.000456 ***
## high_blood_pressure 1.084e-01  6.139e-02   1.766 0.078962 .
## platelets       4.238e-08  2.956e-07   0.143 0.886147
## serum_creatinine 8.266e-02  3.059e-02   2.702 0.007493 **
## serum_sodium    -9.853e-03  7.698e-03  -1.280 0.202078
## sex            -9.747e-02  7.210e-02  -1.352 0.177990
## smoking        6.579e-02  7.102e-02   0.926 0.355388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4137 on 194 degrees of freedom
## Multiple R-squared:  0.2409, Adjusted R-squared:  0.1978
## F-statistic: 5.596 on 11 and 194 DF, p-value: 8.922e-08
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.799   4.7993 28.0446 3.196e-07 ***
```



```

## anaemia 1 0.006 0.0061 0.0355 0.8506784
## creatinine_phosphokinase 1 0.268 0.2676 1.5638 0.2126160
## diabetes 1 0.096 0.0961 0.5617 0.4544806
## ejection_fraction 1 2.687 2.6871 15.7023 0.0001041 ***
## high_blood_pressure 1 0.591 0.5908 3.4525 0.0646723 .
## platelets 1 0.063 0.0626 0.3658 0.5459839
## serum_creatinine 1 1.484 1.4840 8.6716 0.0036272 **
## serum_sodium 1 0.207 0.2071 1.2100 0.2726937
## sex 1 0.186 0.1860 1.0868 0.2984824
## smoking 1 0.147 0.1469 0.8582 0.3553883
## Residuals 194 33.199 0.1711
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[5]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 5 12 14 18 39 63 82 90 102 107 119 124 125 161 164 181 189 204 230 235
## 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
## 244 249 254 255 296
## 0 0 1 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[5]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[5]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear.accur, simplelinear.ConfMat[[5]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[5]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[5]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[5]]$byClass[7])
simplelinear.ConfMat[[5]] <- simplelinear.ConfMat[[5]]$table
simplelinear.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[5]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Coefficients:
## (Intercept) age ejection_fraction serum_creatinine

```

```
##           -0.157207           0.011563           -0.009596           0.087479
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7919 -0.2963 -0.1409  0.3985  1.0064
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.157207   0.167861  -0.937 0.350116
## age             0.011563   0.002414   4.790 3.22e-06 ***
## ejection_fraction -0.009596 0.002533  -3.789 0.000199 ***
## serum_creatinine  0.087479  0.030111   2.905 0.004078 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4147 on 202 degrees of freedom
## Multiple R-squared:  0.2055, Adjusted R-squared:  0.1937
## F-statistic: 17.42 on 3 and 202 DF,  p-value: 4.252e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  4.799   4.7993  27.9024 3.289e-07 ***
## ejection_fraction  1  2.737   2.7372  15.9138 9.267e-05 ***
## serum_creatinine  1  1.452   1.4518   8.4405 0.004078 **
## Residuals       202 34.745   0.1720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[5]])
```

```
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
```

```
revisedlinear.valid
```

```
##      5  12  14  18  39  63  82  90 102 107 119 124 125 161 164 181 189 204 230 235
##      1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0
## 244 249 254 255 296
##      0   0   0   0   0
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - revisedlinear.valid)^2))
revisedlinear.ConfMat[[5]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[5]]$DEATH_EVENT))
```

```

revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[5]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[5]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[5]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[5]]$byClass[7])
revisedlinear.ConfMat[[5]] <- revisedlinear.ConfMat[[5]]$table
revisedlinear.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[5]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      0.9965989          0.0666032          0.1809211
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##      0.0001724          -0.0579326          0.7697949
##      serum_creatinine      serum_sodium
##      0.5716267          -0.0376864
##
## Degrees of Freedom: 205 Total (i.e. Null); 198 Residual
## Null Deviance:      253.7
## Residual Deviance: 200.7      AIC: 216.7
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9963  -0.7467  -0.4581   0.8293   2.5367
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.9965989   6.0487558   0.165 0.869132
## age            0.0666032   0.0155966   4.270 1.95e-05 ***
## anaemia        0.1809211   0.3645460   0.496 0.619689
## creatinine_phosphokinase 0.0001724   0.0001686   1.023 0.306335
## ejection_fraction -0.0579326   0.0174474  -3.320 0.000899 ***
## high_blood_pressure  0.7697949   0.3624780   2.124 0.033695 *
## serum_creatinine  0.5716267   0.2119965   2.696 0.007009 **

```

```
## serum_sodium          -0.0376864  0.0441970  -0.853 0.393831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 253.68  on 205  degrees of freedom
## Residual deviance: 200.74  on 198  degrees of freedom
## AIC: 216.74
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                205      253.68
## age              1  23.1241      204      230.55
## anaemia          1   0.0383      203      230.51
## creatinine_phosphokinase 1   1.3570      202      229.16
## ejection_fraction  1  14.7913      201      214.37
## high_blood_pressure  1   3.4173      200      210.95
## serum_creatinine    1   9.4869      199      201.46
## serum_sodium        1   0.7257      198      200.74
```

```
generlinear.valid <- predict(generlinear.model, groups[[5]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
##      5  12  14  18  39  63  82  90 102 107 119 124 125 161 164 181 189 204 230 235
##      1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0
## 244 249 254 255 296
##      0  0  1  0  0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - generlinear.valid)^2))/nrow)
generlinear.ConfMat[[5]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[5]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[5]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[5]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[5]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[5]]$byClass[7])
generlinear.ConfMat[[5]] <- generlinear.ConfMat[[5]]$table
```

```

generlinear.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, generlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[5]]$DEATH_EVENT, generlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.56 1.00 3.71 3.37 total = 11.65
##
## REML score: 110.8426
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.30583    0.02647   11.55  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          2.563   3.220  7.171 9.22e-05 ***
## s(creatinine_phosphokinase) 1.001   1.001  1.939  0.16543
## s(ejection_fraction)      3.712   4.571  8.186 1.62e-06 ***
## s(serum_creatinine)       3.369   4.146  4.255  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.324 Deviance explained = 35.9%
## -REML = 110.84 Scale est. = 0.1443 n = 206
generaddit.valid <- predict(generaddit.model, groups[[5]], type = 'response')
for (j in 1:length(generaddit.valid)) {

```

```

if(generaddit.valid[j] < 0.5){
  generaddit.valid[j] <- 0
}
if(generaddit.valid[j] >= 0.5){
  generaddit.valid[j] <- 1
}
}
generaddit.valid

##      5  12  14  18  39  63  82  90 102 107 119 124 125 161 164 181 189 204 230 235
##      1   0   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0
## 244 249 254 255 296
##      0   0   0   0   0

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[5]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[5]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[5]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[5]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[5]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[5]]$byClass[7])
generaddit.ConfMat[[5]] <- generaddit.ConfMat[[5]]$table
generaddit.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[5]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6941748 0.3058252
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.47552 0.3916084          566.9930 0.4335664          39.67133
## 1 67.37567 0.4444444          680.9365 0.4285714          34.25397
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3076923 268498.8          1.197133      137.3007 0.6363636
## 1          0.4603175 269801.7          1.775873      136.0794 0.6190476
##      smoking
## 0 0.3286713
## 1 0.3333333
##
## Coefficients of linear discriminants:
##                                LD1
## age                6.088490e-02
## anaemia            1.010686e-01

```

```

## creatinine_phosphokinase 1.769413e-04
## diabetes                2.733493e-01
## ejection_fraction       -4.692135e-02
## high_blood_pressure     5.475879e-01
## platelets               2.140396e-07
## serum_creatinine        4.175273e-01
## serum_sodium            -4.976644e-02
## sex                     -4.923186e-01
## smoking                 3.322952e-01

lindiscr.valid <- predict(lindiscr.model, groups[[5]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

## [1] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[5]]))
lindiscr.ConfMat[[5]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[5]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[5]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[5]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[5]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[5]]$byClass[7])
lindiscr.ConfMat[[5]] <- lindiscr.ConfMat[[5]]$table
lindiscr.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[5]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.6941748 0.3058252
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.47552 0.3916084          566.9930 0.4335664          39.67133
## 1 67.37567 0.4444444          680.9365 0.4285714          34.25397
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3076923 268498.8          1.197133      137.3007 0.6363636
## 1          0.4603175 269801.7          1.775873      136.0794 0.6190476
##      smoking
## 0 0.3286713
## 1 0.3333333

quaddiscr.valid <- predict(quaddiscr.model, groups[[5]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

```

```

## [1] 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[5]]))
quaddiscr.ConfMat[[5]] <- confusionMatrix(factor(quaddiscr.valid), factor(groups[[5]]$DEATH_EVENT))
quaddiscr accur <- c(quaddiscr accur, quaddiscr.ConfMat[[5]]$overall[1])
quaddiscr.precis <- c(quaddiscr.precis, quaddiscr.ConfMat[[5]]$byClass[5])
quaddiscr.recall <- c(quaddiscr.recall, quaddiscr.ConfMat[[5]]$byClass[6])
quaddiscr.F1score <- c(quaddiscr.F1score, quaddiscr.ConfMat[[5]]$byClass[7])
quaddiscr.ConfMat[[5]] <- quaddiscr.ConfMat[[5]]$table
quaddiscr.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, quaddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
quaddiscr.AUCsc <- c(quaddiscr.AUCsc, auc(groups[[5]]$DEATH_EVENT, quaddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 76.28 89.91 97.34 99.35 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23301 ( N = 206 )
##
## Deviance: 200.702
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[5]], type = 'class')
mixeddiscr.valid <- as.numeric(mixeddiscr.valid)-1
mixeddiscr.valid

## [1] 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0 0
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[5]]))
mixeddiscr.ConfMat[[5]] <- confusionMatrix(factor(mixeddiscr.valid), factor(groups[[5]]$DEATH_EVENT))
mixeddiscr accur <- c(mixeddiscr accur, mixeddiscr.ConfMat[[5]]$overall[1])
mixeddiscr.precis <- c(mixeddiscr.precis, mixeddiscr.ConfMat[[5]]$byClass[5])
mixeddiscr.recall <- c(mixeddiscr.recall, mixeddiscr.ConfMat[[5]]$byClass[6])
mixeddiscr.F1score <- c(mixeddiscr.F1score, mixeddiscr.ConfMat[[5]]$byClass[7])
mixeddiscr.ConfMat[[5]] <- mixeddiscr.ConfMat[[5]]$table
mixeddiscr.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, mixeddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.AUCsc <- c(mixeddiscr.AUCsc, auc(groups[[5]]$DEATH_EVENT, mixeddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

```



```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.26214 ( N = 206 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[5]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[5]]))
flexdiscr.ConfMat[[5]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[5]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[5]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[5]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[5]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[5]]$byClass[7])
flexdiscr.ConfMat[[5]] <- flexdiscr.ConfMat[[5]]$table
flexdiscr.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, flexdiscr.valid)

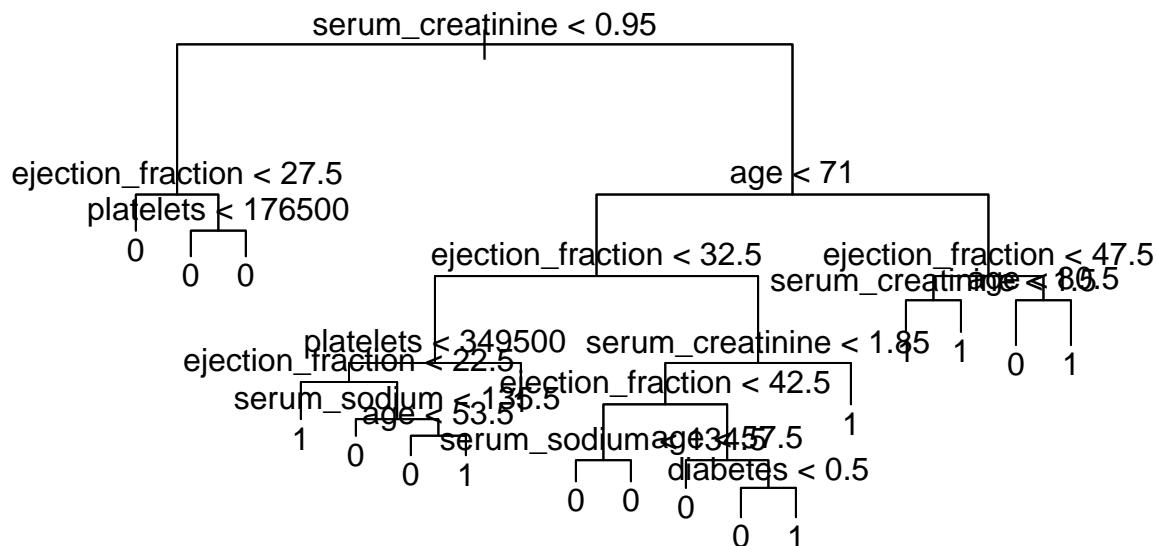
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[5]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

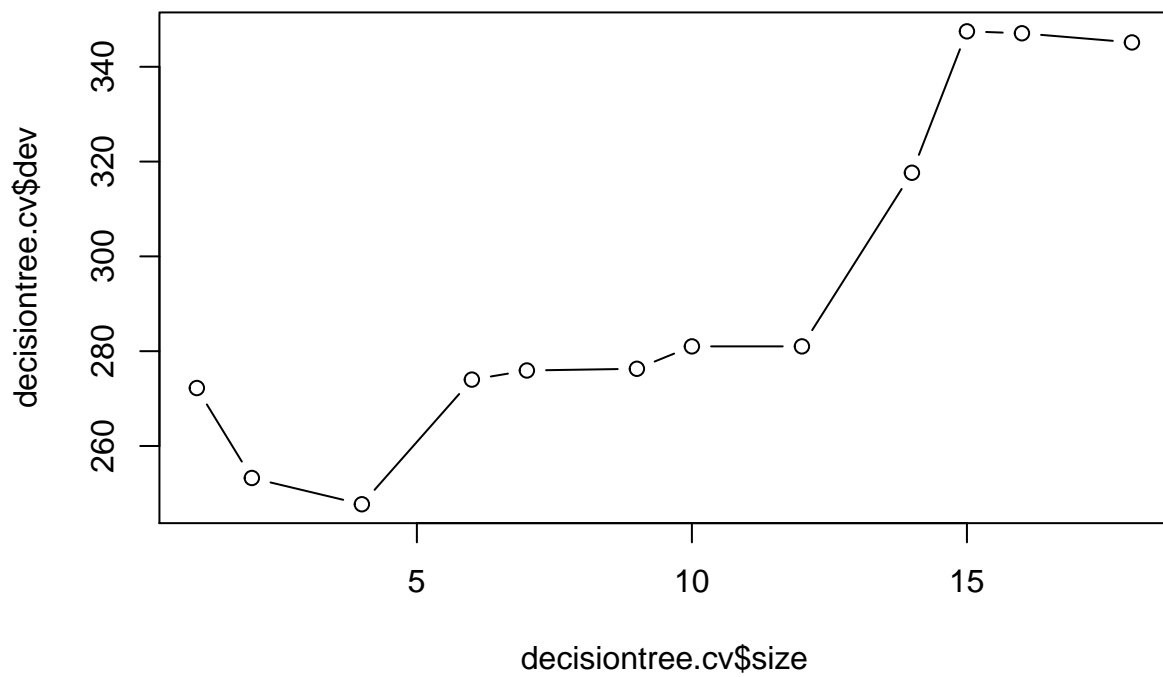
decisiontree.valid <- predict(decisiontree.model, groups[[5]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

## [1] 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[5]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[5]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[5]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[5]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[5]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[5]]$byClass[7])
decisiontree.ConfMat[[5]] <- decisiontree.ConfMat[[5]]$table
decisiontree.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, decisiontree.valid)

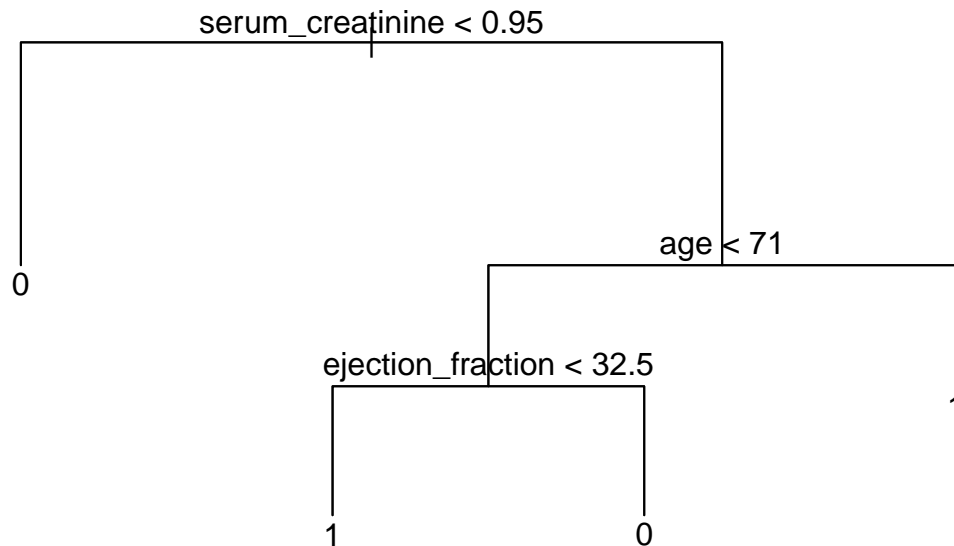
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[5]]$DEATH_EVENT, decisiontree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.valid <- predict(prunedectree.model, groups[[5]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0

prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[5]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[5]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[5]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[5]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[5]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[5]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[5]]$byClass[7])
prunedectree.ConfMat[[5]] <- prunedectree.ConfMat[[5]]$table
prunedectree.ROCcurv[[5]] <- roc(groups[[5]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[5]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

##6th fold-out training
trainset <- data.frame()
for (i in c(1:5,7:8)) {
  trainset <- rbind(trainset, groups[[i]])
}

```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.664e+00             1.027e-02             -2.391e-02
## creatinine_phosphokinase             diabetes             ejection_fraction
##             -2.991e-06             4.355e-02             -7.949e-03
##             high_blood_pressure             platelets             serum_creatinine
##             8.493e-02             2.466e-08             1.057e-01
##             serum_sodium                sex                smoking
##             -1.362e-02             -3.897e-02             2.383e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8968 -0.2851 -0.1216  0.3317  0.9916
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.664e+00  1.030e+00   1.615 0.107931
## age           1.027e-02  2.617e-03   3.925 0.000122 ***
## anaemia      -2.391e-02  6.182e-02  -0.387 0.699331
## creatinine_phosphokinase -2.991e-06  3.906e-05  -0.077 0.939044
## diabetes      4.355e-02  6.122e-02   0.711 0.477689
## ejection_fraction -7.949e-03  2.664e-03  -2.984 0.003225 **
## high_blood_pressure  8.493e-02  6.222e-02   1.365 0.173898
## platelets      2.466e-08  3.119e-07   0.079 0.937053
## serum_creatinine  1.057e-01  3.673e-02   2.878 0.004460 **
## serum_sodium   -1.362e-02  7.452e-03  -1.828 0.069196 .
## sex           -3.897e-02  7.217e-02  -0.540 0.589852
## smoking        2.383e-02  7.274e-02   0.328 0.743519
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4139 on 188 degrees of freedom
## Multiple R-squared:  0.218, Adjusted R-squared:  0.1723
## F-statistic: 4.766 on 11 and 188 DF, p-value: 1.889e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.670   3.6702  21.4279 6.835e-06 ***
```

```
## anaemia 1 0.053 0.0527 0.3075 0.579856
## creatinine_phosphokinase 1 0.006 0.0060 0.0353 0.851161
## diabetes 1 0.119 0.1187 0.6929 0.406237
## ejection_fraction 1 2.438 2.4383 14.2358 0.000216 ***
## high_blood_pressure 1 0.198 0.1977 1.1541 0.284080
## platelets 1 0.016 0.0157 0.0917 0.762352
## serum_creatinine 1 1.885 1.8851 11.0059 0.001090 **
## serum_sodium 1 0.543 0.5429 3.1698 0.076625 .
## sex 1 0.033 0.0330 0.1929 0.661043
## smoking 1 0.018 0.0184 0.1074 0.743519
## Residuals 188 32.201 0.1713
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[6]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 10 16 24 27 38 50 61 67 69 73 98 101 112 123 136 150 182 198 216 219
## 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
## 226 233 241 248 267 278 279 280 282 292 298
## 0 0 0 1 0 0 0 0 0 0 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[6]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[6]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear.accur, simplelinear.ConfMat[[6]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[6]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[6]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[6]]$byClass[7])
simplelinear.ConfMat[[6]] <- simplelinear.ConfMat[[6]]$table
simplelinear.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[6]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Coefficients:
## (Intercept) age ejection_fraction serum_creatinine
```

```
##           -0.138608           0.010143           -0.008886           0.113776
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8675 -0.2827 -0.1352  0.3573  1.0075
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.138608   0.175476  -0.790 0.430541
## age             0.010143   0.002534   4.003 8.85e-05 ***
## ejection_fraction -0.008886   0.002545  -3.492 0.000593 ***
## serum_creatinine  0.113776   0.035172   3.235 0.001429 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4125 on 196 degrees of freedom
## Multiple R-squared:  0.1901, Adjusted R-squared:  0.1777
## F-statistic: 15.33 on 3 and 196 DF,  p-value: 5.312e-09
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  3.670   3.6702   21.569 6.248e-06 ***
## ejection_fraction  1  2.377   2.3769   13.968 0.0002439 ***
## serum_creatinine  1  1.781   1.7806   10.464 0.0014286 **
## Residuals       196 33.352   0.1702
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[6]])
```

```
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
```

```
revisedlinear.valid
```

```
## 10 16 24 27 38 50 61 67 69 73 98 101 112 123 136 150 182 198 216 219
## 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
## 226 233 241 248 267 278 279 280 282 292 298
## 0 0 0 1 0 0 0 0 1 0 0
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - revisedlinear.valid)^2)))
revisedlinear.ConfMat[[6]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[6]]$DEATH_EVENT))
```

```

revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[6]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[6]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[6]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[6]]$byClass[7])
revisedlinear.ConfMat[[6]] <- revisedlinear.ConfMat[[6]]$table
revisedlinear.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[6]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      5.536e+00          5.838e-02         -1.086e-01
## creatinine_phosphokinase    ejection_fraction    high_blood_pressure
##      -1.632e-05          -4.928e-02          5.586e-01
##      serum_creatinine      serum_sodium
##      5.613e-01          -6.742e-02
##
## Degrees of Freedom: 199 Total (i.e. Null); 192 Residual
## Null Deviance: 240.9
## Residual Deviance: 195.5 AIC: 211.5
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1220  -0.7202  -0.4923   0.7769   2.4662
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.536e+00  5.912e+00   0.936 0.349107
## age          5.838e-02  1.588e-02   3.675 0.000238 ***
## anaemia     -1.086e-01  3.702e-01  -0.293 0.769173
## creatinine_phosphokinase -1.632e-05  2.443e-04  -0.067 0.946741
## ejection_fraction  -4.928e-02  1.729e-02  -2.851 0.004360 **
## high_blood_pressure  5.586e-01  3.656e-01   1.528 0.126533
## serum_creatinine  5.613e-01  2.070e-01   2.711 0.006704 **

```



```
## serum_sodium          -6.742e-02  4.365e-02  -1.544 0.122489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 240.86  on 199  degrees of freedom
## Residual deviance: 195.47  on 192  degrees of freedom
## AIC: 211.47
##
## Number of Fisher Scoring iterations: 4
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                199      240.86
## age              1  18.0519      198      222.81
## anaemia          1   0.2154      197      222.59
## creatinine_phosphokinase 1   0.0324      196      222.56
## ejection_fraction  1  13.2278      195      209.33
## high_blood_pressure  1   1.2738      194      208.06
## serum_creatinine    1  10.1208      193      197.94
## serum_sodium        1   2.4695      192      195.47
```

```
generlinear.valid <- predict(generlinear.model, groups[[6]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
## 10 16 24 27 38 50 61 67 69 73 98 101 112 123 136 150 182 198 216 219
## 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
## 226 233 241 248 267 278 279 280 282 292 298
## 0 0 0 1 0 0 0 0 0 0 0 0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - generlinear.valid)^2))/nrow
generlinear.ConfMat[[6]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[6]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[6]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[6]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[6]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[6]]$byClass[7])
generlinear.ConfMat[[6]] <- generlinear.ConfMat[[6]]$table
```

```

generlinear.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, generlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[6]]$DEATH_EVENT, generlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 1.77 1.00 3.04 3.29 total = 10.11
##
## REML score: 110.2509
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29000    0.02745   10.57  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df    F  p-value
## s(age)          1.774  2.231 4.365 0.010287 *
## s(creatinine_phosphokinase) 1.000  1.000 0.027 0.870455
## s(ejection_fraction)      3.043  3.774 6.785 6.31e-05 ***
## s(serum_creatinine)       3.290  4.082 5.068 0.000594 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.272 Deviance explained = 30.5%
## -REML = 110.25 Scale est. = 0.15067 n = 200
generaddit.valid <- predict(generaddit.model, groups[[6]], type = 'response')
for (j in 1:length(generaddit.valid)) {

```

```

if(generaddit.valid[j] < 0.5){
  generaddit.valid[j] <- 0
}
if(generaddit.valid[j] >= 0.5){
  generaddit.valid[j] <- 1
}
}
generaddit.valid

## 10 16 24 27 38 50 61 67 69 73 98 101 112 123 136 150 182 198 216 219
## 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
## 226 233 241 248 267 278 279 280 282 292 298
## 0 0 0 1 1 0 0 0 1 0 0

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[6]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[6]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[6]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[6]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[6]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[6]]$byClass[7])
generaddit.ConfMat[[6]] <- generaddit.ConfMat[[6]]$table
generaddit.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[6]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
## 0 1
## 0.71 0.29
##
## Group means:
## age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 0 58.13850 0.4154930 570.6127 0.4154930 39.71127
## 1 65.94253 0.4137931 517.0345 0.4310345 34.13793
## high_blood_pressure platelets serum_creatinine serum_sodium sex
## 0 0.3591549 268661.5 1.200845 137.5141 0.6338028
## 1 0.4655172 265347.4 1.740517 135.4828 0.6379310
## smoking
## 0 0.3450704
## 1 0.3103448
##
## Coefficients of linear discriminants:
## LD1
## age 5.455688e-02
## anaemia -1.269881e-01

```

```

## creatinine_phosphokinase -1.588190e-05
## diabetes                2.312901e-01
## ejection_fraction       -4.221379e-02
## high_blood_pressure     4.510068e-01
## platelets               1.309748e-07
## serum_creatinine        5.614226e-01
## serum_sodium            -7.232371e-02
## sex                     -2.069369e-01
## smoking                 1.265695e-01

lindiscr.valid <- predict(lindiscr.model, groups[[6]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

## [1] 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[6]]))
lindiscr.ConfMat[[6]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[6]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[6]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[6]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[6]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[6]]$byClass[7])
lindiscr.ConfMat[[6]] <- lindiscr.ConfMat[[6]]$table
lindiscr.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[6]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
## 0 1
## 0.71 0.29
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.13850 0.4154930          570.6127 0.4154930          39.71127
## 1 65.94253 0.4137931          517.0345 0.4310345          34.13793
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3591549 268661.5          1.200845      137.5141 0.6338028
## 1          0.4655172 265347.4          1.740517      135.4828 0.6379310
##      smoking
## 0 0.3450704
## 1 0.3103448

quaddiscr.valid <- predict(quaddiscr.model, groups[[6]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

```

```

## [1] 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0

quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[6]]))
quaddiscr.ConfMat[[6]] <- confusionMatrix(factor(quaddiscr.valid), factor(groups[[6]]$DEATH_EVENT))
quaddiscr accur <- c(quaddiscr accur, quaddiscr.ConfMat[[6]]$overall[1])
quaddiscr.precis <- c(quaddiscr.precis, quaddiscr.ConfMat[[6]]$byClass[5])
quaddiscr.recall <- c(quaddiscr.recall, quaddiscr.ConfMat[[6]]$byClass[6])
quaddiscr.F1score <- c(quaddiscr.F1score, quaddiscr.ConfMat[[6]]$byClass[7])
quaddiscr.ConfMat[[6]] <- quaddiscr.ConfMat[[6]]$table
quaddiscr.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, quaddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.AUCsc <- c(quaddiscr.AUCsc, auc(groups[[6]]$DEATH_EVENT, quaddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 81.76 94.07 98.35 99.83 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.225 ( N = 200 )
##
## Deviance: 187.219

mixeddiscr.valid <- predict(mixeddiscr.model, groups[[6]], type = 'class')
mixeddiscr.valid <- as.numeric(mixeddiscr.valid)-1
mixeddiscr.valid

## [1] 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0

mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[6]]))
mixeddiscr.ConfMat[[6]] <- confusionMatrix(factor(mixeddiscr.valid), factor(groups[[6]]$DEATH_EVENT))
mixeddiscr accur <- c(mixeddiscr accur, mixeddiscr.ConfMat[[6]]$overall[1])
mixeddiscr.precis <- c(mixeddiscr.precis, mixeddiscr.ConfMat[[6]]$byClass[5])
mixeddiscr.recall <- c(mixeddiscr.recall, mixeddiscr.ConfMat[[6]]$byClass[6])
mixeddiscr.F1score <- c(mixeddiscr.F1score, mixeddiscr.ConfMat[[6]]$byClass[7])
mixeddiscr.ConfMat[[6]] <- mixeddiscr.ConfMat[[6]]$table
mixeddiscr.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, mixeddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

mixeddiscr.AUCsc <- c(mixeddiscr.AUCsc, auc(groups[[6]]$DEATH_EVENT, mixeddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

```

```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.235 ( N = 200 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[6]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[6]]$DEATH_EVENT))
flexdiscr.ConfMat[[6]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[6]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[6]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[6]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[6]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[6]]$byClass[7])
flexdiscr.ConfMat[[6]] <- flexdiscr.ConfMat[[6]]$table
flexdiscr.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, flexdiscr.valid)

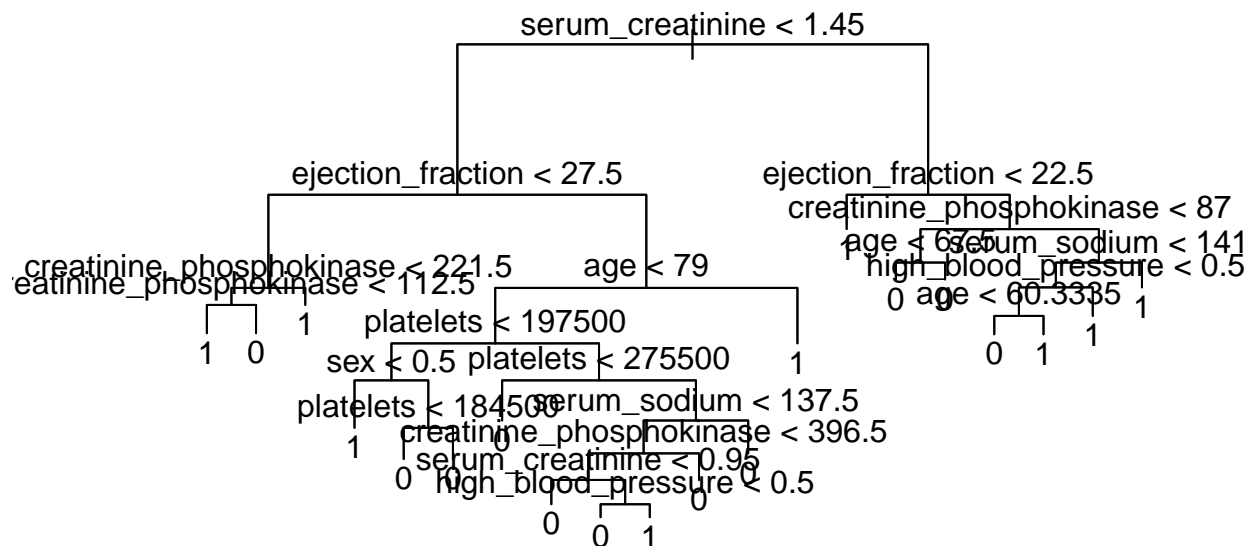
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[6]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

decisiontree.valid <- predict(decisiontree.model, groups[[6]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

## [1] 1 1 0 1 1 0 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0

decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[6]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[6]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[6]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[6]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[6]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[6]]$byClass[7])
decisiontree.ConfMat[[6]] <- decisiontree.ConfMat[[6]]$table
decisiontree.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, decisiontree.valid)

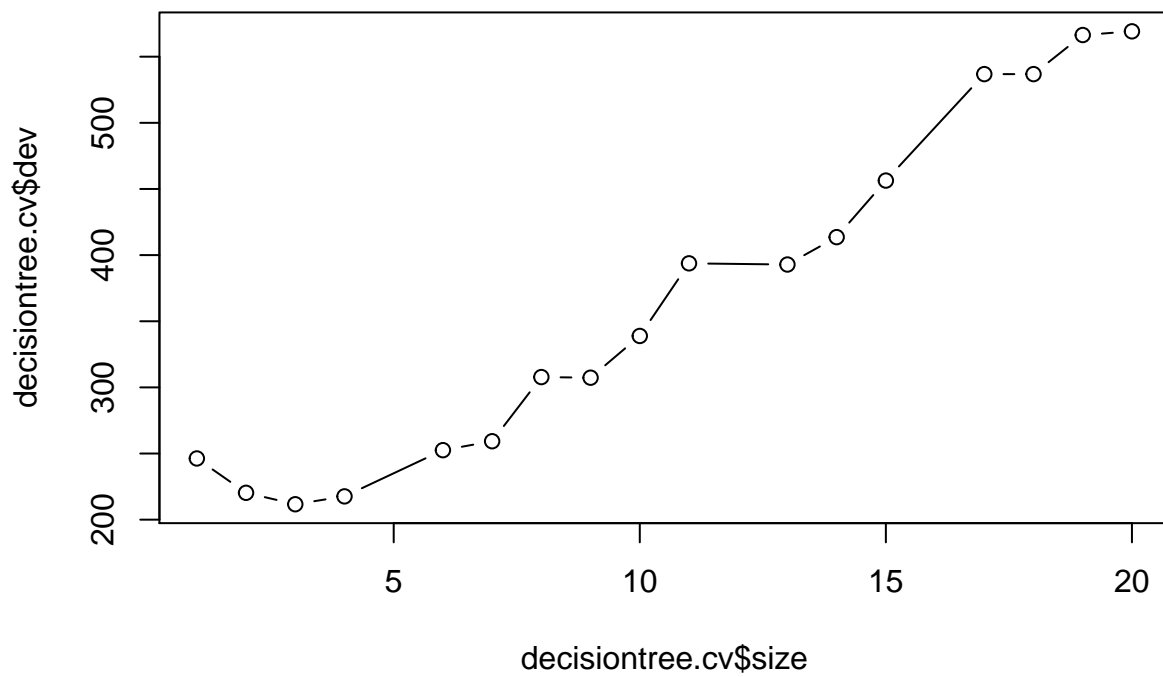
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[6]]$DEATH_EVENT, decisiontree.valid))

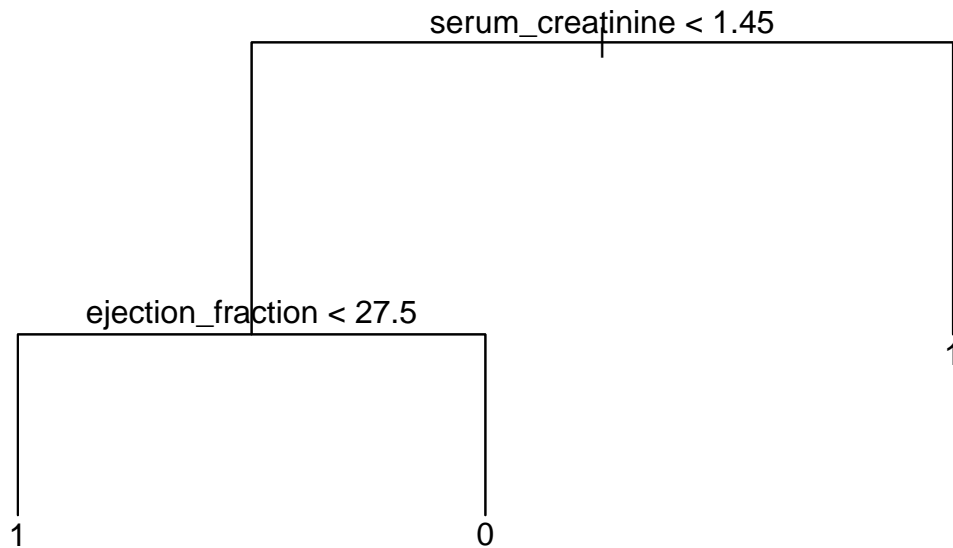
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```

```

prunedectree.valid <- predict(prunedectree.model, groups[[6]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 1 0 0 0 0 0 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[6]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[6]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[6]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[6]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[6]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[6]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[6]]$byClass[7])
prunedectree.ConfMat[[6]] <- prunedectree.ConfMat[[6]]$table
prunedectree.ROCcurv[[6]] <- roc(groups[[6]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[6]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##7th fold-out training
trainset <- data.frame()
for (i in c(1:6, 8)) {
  trainset <- rbind(trainset, groups[[i]])
}

```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##             (Intercept)                age                anaemia
##             1.424e+00             9.350e-03             4.402e-02
## creatinine_phosphokinase            diabetes            ejection_fraction
##             4.819e-05             3.550e-02             -9.124e-03
##             high_blood_pressure            platelets            serum_creatinine
##             1.295e-01             1.170e-07             7.963e-02
##             serum_sodium                sex                smoking
##             -1.160e-02             -5.272e-02             3.283e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7238 -0.2842 -0.1399  0.3642  1.0047
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.424e+00  1.054e+00   1.351 0.178217
## age           9.350e-03  2.659e-03   3.516 0.000548 ***
## anaemia       4.402e-02  6.200e-02   0.710 0.478560
## creatinine_phosphokinase 4.819e-05  3.083e-05   1.563 0.119660
## diabetes      3.550e-02  6.290e-02   0.564 0.573176
## ejection_fraction -9.124e-03  2.756e-03  -3.310 0.001115 **
## high_blood_pressure 1.295e-01  6.326e-02   2.047 0.042023 *
## platelets      1.170e-07  3.057e-07   0.383 0.702464
## serum_creatinine 7.963e-02  3.055e-02   2.607 0.009861 **
## serum_sodium   -1.160e-02  7.597e-03  -1.528 0.128271
## sex           -5.272e-02  7.249e-02  -0.727 0.467928
## smoking        3.283e-02  7.251e-02   0.453 0.651209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4199 on 190 degrees of freedom
## Multiple R-squared:  0.2058, Adjusted R-squared:  0.1598
## F-statistic: 4.476 on 11 and 190 DF, p-value: 5.269e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  2.752  2.75212 15.6105 0.0001096 ***
```

```
## anaemia 1 0.018 0.01792 0.1017 0.7501814
## creatinine_phosphokinase 1 0.346 0.34610 1.9631 0.1628128
## diabetes 1 0.021 0.02098 0.1190 0.7304921
## ejection_fraction 1 2.771 2.77135 15.7195 0.0001039 ***
## high_blood_pressure 1 0.655 0.65514 3.7161 0.0553820 .
## platelets 1 0.103 0.10337 0.5863 0.4447846
## serum_creatinine 1 1.545 1.54550 8.7663 0.0034597 **
## serum_sodium 1 0.371 0.37122 2.1056 0.1484066
## sex 1 0.061 0.06136 0.3481 0.5559162
## smoking 1 0.036 0.03615 0.2050 0.6512091
## Residuals 190 33.497 0.17630
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[7]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 6 7 26 71 80 84 89 111 128 167 175 184 186 187 188 190 192 207 210 212
## 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
## 214 222 224 228 236 245 251 269 276
## 0 0 0 0 0 0 0 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[7]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[7]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear accur, simplelinear.ConfMat[[7]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[7]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[7]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[7]]$byClass[7])
simplelinear.ConfMat[[7]] <- simplelinear.ConfMat[[7]]$table
simplelinear.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[7]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Coefficients:
## (Intercept) age ejection_fraction serum_creatinine
```

```
##           -0.042274           0.009592           -0.009893           0.086021
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7640 -0.3024 -0.1608  0.4333  0.9657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.042274   0.177694  -0.238 0.812202
## age             0.009592   0.002594   3.698 0.000281 ***
## ejection_fraction -0.009893  0.002644  -3.742 0.000239 ***
## serum_creatinine  0.086021  0.029895   2.877 0.004449 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4217 on 198 degrees of freedom
## Multiple R-squared:  0.1651, Adjusted R-squared:  0.1524
## F-statistic: 13.05 on 3 and 198 DF,  p-value: 8.22e-08
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  2.752  2.75212  15.4737 0.0001157 ***
## ejection_fraction  1  2.737  2.73749  15.3914 0.0001205 ***
## serum_creatinine  1  1.473  1.47260   8.2796 0.0044490 **
## Residuals       198 35.216  0.17786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[7]])
```

```
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
```

```
revisedlinear.valid
```

```
##      6      7      26      71      80      84      89      111      128      167      175      184      186      187      188      190      192      207      210      212
##      1      1      1      0      0      0      0      0      0      0      0      1      0      0      0      0      0      0      0      0
## 214 222 224 228 236 245 251 269 276
##      0      0      0      0      0      0      0      0      0
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - revisedlinear.valid)^2))
revisedlinear.ConfMat[[7]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[7]]$DEATH_EVENT))
```

```

revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[7]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[7]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[7]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[7]]$byClass[7])
revisedlinear.ConfMat[[7]] <- revisedlinear.ConfMat[[7]]$table
revisedlinear.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[7]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      3.6484870              0.0536428              0.3298001
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##      0.0002485              -0.0556756              0.8093114
##      serum_creatinine      serum_sodium
##      0.5128360              -0.0527211
##
## Degrees of Freedom: 201 Total (i.e. Null); 194 Residual
## Null Deviance: 245.8
## Residual Deviance: 201.5 AIC: 217.5
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8304  -0.7396  -0.5009   0.8429   2.4503
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.6484870   5.8708764   0.621 0.534300
## age             0.0536428   0.0157879   3.398 0.000679 ***
## anaemia         0.3298001   0.3658509   0.901 0.367344
## creatinine_phosphokinase 0.0002485   0.0001702   1.459 0.144447
## ejection_fraction -0.0556756   0.0178474  -3.120 0.001811 **
## high_blood_pressure  0.8093114   0.3603472   2.246 0.024709 *
## serum_creatinine  0.5128360   0.2007902   2.554 0.010647 *

```

```
## serum_sodium          -0.0527211  0.0431694  -1.221 0.221987
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 245.76  on 201  degrees of freedom
## Residual deviance: 201.50  on 194  degrees of freedom
## AIC: 217.5
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                201      245.76
## age              1  13.3194      200      232.44
## anaemia          1   0.0987      199      232.35
## creatinine_phosphokinase 1   1.6493      198      230.70
## ejection_fraction  1  14.8746      197      215.82
## high_blood_pressure  1   3.5555      196      212.27
## serum_creatinine    1   9.2516      195      203.01
## serum_sodium        1   1.5195      194      201.50
```

```
generlinear.valid <- predict(generlinear.model, groups[[7]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
##      6      7      26      71      80      84      89      111      128      167      175      184      186      187      188      190      192      207      210      212
##      1      1      0      0      0      1      0      0      0      0      0      1      0      0      0      0      0      0      0      0
## 214 222 224 228 236 245 251 269 276
##      0      0      0      0      0      0      0      0      0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - generlinear.valid)^2))/nrow
generlinear.ConfMat[[7]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[7]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[7]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[7]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[7]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[7]]$byClass[7])
generlinear.ConfMat[[7]] <- generlinear.ConfMat[[7]]$table
```

```

generlinear.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, generlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[7]]$DEATH_EVENT, generlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.48 1.00 3.56 2.69 total = 10.74
##
## REML score: 113.0936
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29703    0.02747   10.81  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)          2.485   3.127  5.819 0.000647 ***
## s(creatinine_phosphokinase) 1.000   1.001  2.295 0.131482
## s(ejection_fraction)      3.563   4.400  8.517 1.19e-06 ***
## s(serum_creatinine)       2.691   3.338  3.444 0.013784 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.274 Deviance explained = 30.9%
## -REML = 113.09 Scale est. = 0.15243 n = 202
generaddit.valid <- predict(generaddit.model, groups[[7]], type = 'response')
for (j in 1:length(generaddit.valid)) {

```

```

if(generaddit.valid[j] < 0.5){
  generaddit.valid[j] <- 0
}
if(generaddit.valid[j] >= 0.5){
  generaddit.valid[j] <- 1
}
}
generaddit.valid

##      6      7      26      71      80      84      89      111      128      167      175      184      186      187      188      190      192      207      210      212
##      1      1      0      0      0      1      0      1      0      0      0      0      0      0      1      0      0      0      0      0
##    214    222    224    228    236    245    251    269    276
##      0      0      0      0      0      0      0      0      0

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[7]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[7]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[7]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[7]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[7]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[7]]$byClass[7])
generaddit.ConfMat[[7]] <- generaddit.ConfMat[[7]]$table
generaddit.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[7]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7029703 0.2970297
##
## Group means:
##      age      anaemia creatinine_phosphokinase      diabetes      ejection_fraction
## 0 59.31456 0.4084507          572.0704 0.4295775          38.94366
## 1 65.88333 0.4500000          726.2333 0.4166667          33.40000
##      high_blood_pressure      platelets      serum_creatinine      serum_sodium      sex
## 0          0.3028169 265411.2          1.239085          137.3592 0.6478873
## 1          0.4666667 269175.1          1.788000          135.7667 0.6166667
##      smoking
## 0 0.3309859
## 1 0.3166667
##
## Coefficients of linear discriminants:
##
##              LD1
## age          5.035801e-02
## anaemia      2.371132e-01

```



```

## creatinine_phosphokinase 2.595360e-04
## diabetes 1.911815e-01
## ejection_fraction -4.914219e-02
## high_blood_pressure 6.974433e-01
## platelets 6.299304e-07
## serum_creatinine 4.289029e-01
## serum_sodium -6.250222e-02
## sex -2.839652e-01
## smoking 1.768441e-01

lindiscr.valid <- predict(lindiscr.model, groups[[7]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

## [1] 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[7]]))
lindiscr.ConfMat[[7]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[7]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[7]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[7]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[7]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[7]]$byClass[7])
lindiscr.ConfMat[[7]] <- lindiscr.ConfMat[[7]]$table
lindiscr.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[7]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
## 0 1
## 0.7029703 0.2970297
##
## Group means:
## age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 0 59.31456 0.4084507 572.0704 0.4295775 38.94366
## 1 65.88333 0.4500000 726.2333 0.4166667 33.40000
## high_blood_pressure platelets serum_creatinine serum_sodium sex
## 0 0.3028169 265411.2 1.239085 137.3592 0.6478873
## 1 0.4666667 269175.1 1.788000 135.7667 0.6166667
## smoking
## 0 0.3309859
## 1 0.3166667

quaddiscr.valid <- predict(quaddiscr.model, groups[[7]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

```

[illegible]

```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.26238 ( N = 202 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[7]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[7]]$DEATH_EVENT))
flexdiscr.ConfMat[[7]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[7]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[7]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[7]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[7]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[7]]$byClass[7])
flexdiscr.ConfMat[[7]] <- flexdiscr.ConfMat[[7]]$table
flexdiscr.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, flexdiscr.valid)

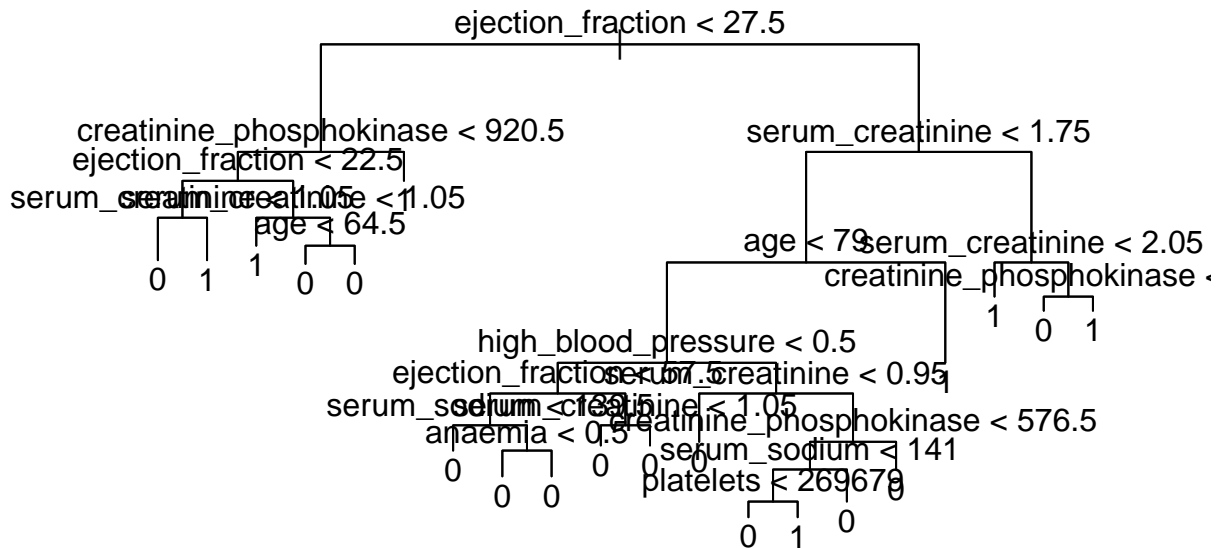
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[7]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

decisiontree.valid <- predict(decisiontree.model, groups[[7]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

## [1] 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0

decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[7]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[7]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[7]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[7]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[7]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[7]]$byClass[7])
decisiontree.ConfMat[[7]] <- decisiontree.ConfMat[[7]]$table
decisiontree.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, decisiontree.valid)

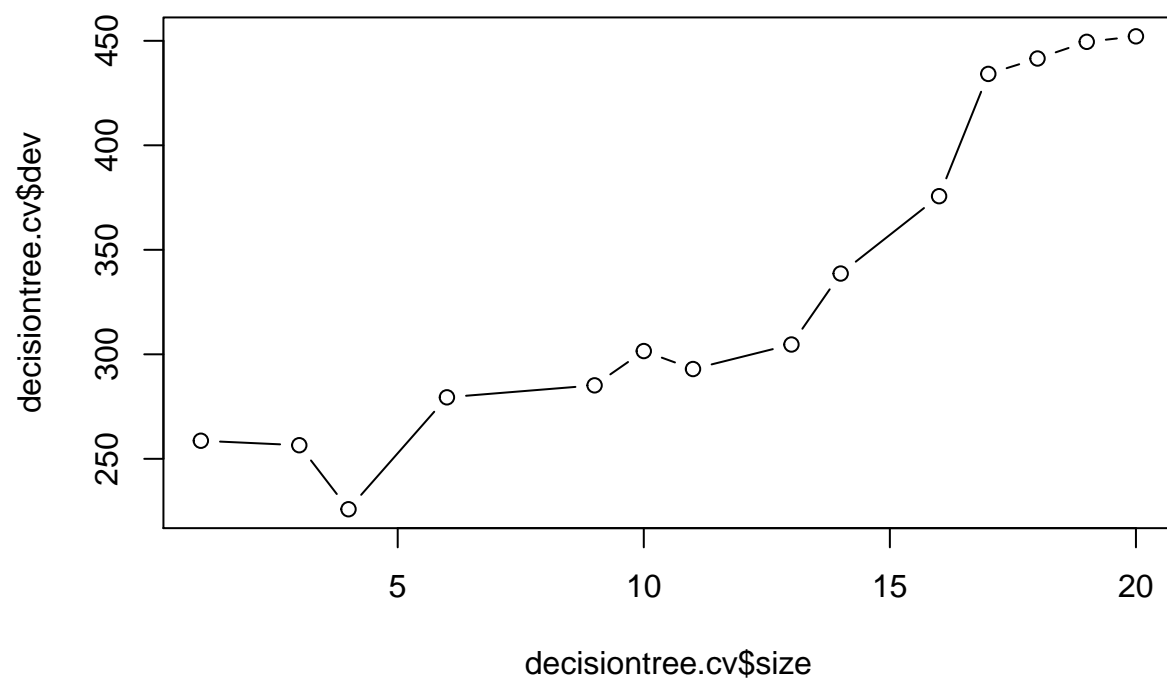
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[7]]$DEATH_EVENT, decisiontree.valid))

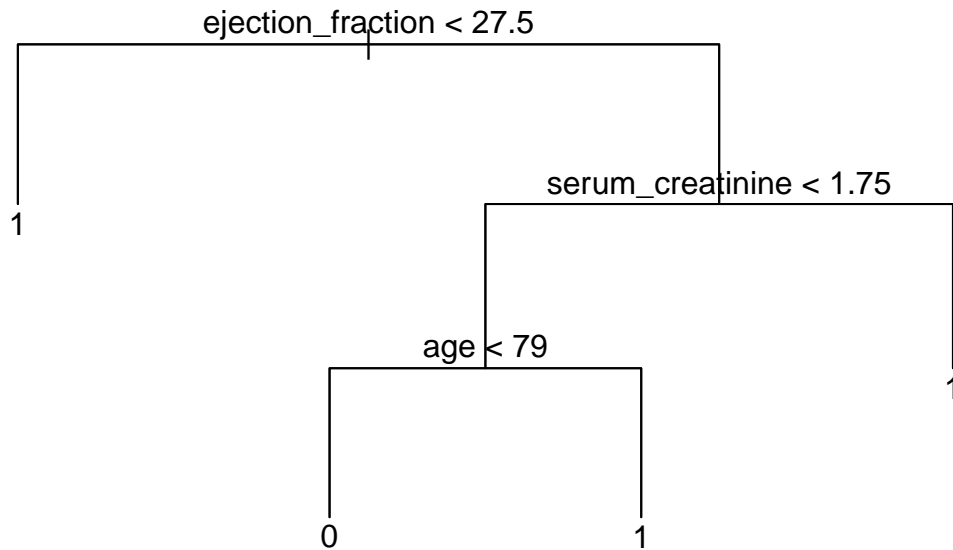
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.valid <- predict(prunedectree.model, groups[[7]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 1 1 1 0 0 1 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[7]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[7]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[7]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[7]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[7]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[7]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[7]]$byClass[7])
prunedectree.ConfMat[[7]] <- prunedectree.ConfMat[[7]]$table
prunedectree.ROCcurv[[7]] <- roc(groups[[7]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[7]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##8th fold-out training
trainset <- data.frame()
for (i in c(1:7)) {
  trainset <- rbind(trainset, groups[[i]])
}

```

```
simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##              (Intercept)              age              anaemia
##              1.109e+00              1.047e-02              1.604e-02
## creatinine_phosphokinase              diabetes              ejection_fraction
##              4.944e-05              4.338e-02              -9.487e-03
##              high_blood_pressure              platelets              serum_creatinine
##              5.077e-02              -1.917e-08              8.660e-02
##              serum_sodium              sex              smoking
##              -9.164e-03              -8.123e-02              8.437e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7918 -0.2965 -0.1352  0.3752  0.9925
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.109e+00  1.031e+00   1.075 0.283681
## age            1.047e-02  2.677e-03   3.909 0.000129 ***
## anaemia        1.604e-02  6.156e-02   0.261 0.794712
## creatinine_phosphokinase 4.944e-05  3.153e-05   1.568 0.118527
## diabetes       4.338e-02  6.164e-02   0.704 0.482414
## ejection_fraction -9.487e-03  2.689e-03  -3.528 0.000524 ***
## high_blood_pressure  5.077e-02  6.135e-02   0.827 0.409015
## platelets       -1.917e-08  3.172e-07  -0.060 0.951866
## serum_creatinine  8.660e-02  3.086e-02   2.806 0.005537 **
## serum_sodium    -9.164e-03  7.403e-03  -1.238 0.217317
## sex            -8.123e-02  7.059e-02  -1.151 0.251300
## smoking         8.437e-02  7.248e-02   1.164 0.245838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4174 on 191 degrees of freedom
## Multiple R-squared:  0.2125, Adjusted R-squared:  0.1672
## F-statistic: 4.686 on 11 and 191 DF, p-value: 2.45e-06
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  3.407   3.4067 19.5495 1.643e-05 ***
```

```

## anaemia                1 0.000 0.0003 0.0016 0.9678906
## creatinine_phosphokinase 1 0.584 0.5838 3.3500 0.0687626 .
## diabetes               1 0.115 0.1146 0.6578 0.4183350
## ejection_fraction      1 2.684 2.6836 15.3998 0.0001213 ***
## high_blood_pressure    1 0.064 0.0642 0.3684 0.5445880
## platelets              1 0.019 0.0193 0.1110 0.7393391
## serum_creatinine       1 1.573 1.5734 9.0290 0.0030137 **
## serum_sodium           1 0.213 0.2132 1.2236 0.2700527
## sex                    1 0.087 0.0871 0.4997 0.4805089
## smoking                1 0.236 0.2361 1.3551 0.2458375
## Residuals              191 33.284 0.1743
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.valid <- predict(simplelinear.model, groups[[8]])
for (j in 1:length(simplelinear.valid)) {
  if(simplelinear.valid[j] < 0.5){
    simplelinear.valid[j] <- 0
  }
  if(simplelinear.valid[j] >= 0.5){
    simplelinear.valid[j] <- 1
  }
}
simplelinear.valid

## 25 33 34 37 41 45 72 74 75 100 103 133 135 141 142 145 148 162 193 195
## 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0
## 203 232 234 281 283 288 289 297
## 0 0 0 0 0 0 0 0

simplelinear.RMSE <- c(simplelinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - simplelinear.valid)^2))/n)
simplelinear.ConfMat[[8]] <- confusionMatrix(factor(simplelinear.valid), factor(groups[[8]]$DEATH_EVENT))
simplelinear accur <- c(simplelinear accur, simplelinear.ConfMat[[8]]$overall[1])
simplelinear.precis <- c(simplelinear.precis, simplelinear.ConfMat[[8]]$byClass[5])
simplelinear.recall <- c(simplelinear.recall, simplelinear.ConfMat[[8]]$byClass[6])
simplelinear.F1score <- c(simplelinear.F1score, simplelinear.ConfMat[[8]]$byClass[7])
simplelinear.ConfMat[[8]] <- simplelinear.ConfMat[[8]]$table
simplelinear.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, simplelinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.AUCsc <- c(simplelinear.AUCsc, auc(groups[[8]]$DEATH_EVENT, simplelinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
## data = trainset)
##
## Coefficients:
## (Intercept) age ejection_fraction serum_creatinine

```



```
##           -0.06696           0.01006           -0.01002           0.09198
```

```
summary(revisedlinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7966 -0.2915 -0.1471  0.3990  1.0095
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.06696    0.170895  -0.392 0.695609
## age             0.010063   0.002561   3.929 0.000118 ***
## ejection_fraction -0.010018  0.002598  -3.857 0.000155 ***
## serum_creatinine  0.091982   0.030005   3.066 0.002475 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4157 on 199 degrees of freedom
## Multiple R-squared:  0.1865, Adjusted R-squared:  0.1742
## F-statistic: 15.21 on 3 and 199 DF,  p-value: 6.004e-09
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  3.407   3.4067  19.7166 1.488e-05 ***
## ejection_fraction  1  2.852   2.8519  16.5057 6.976e-05 ***
## serum_creatinine  1  1.624   1.6237   9.3974 0.002475 **
## Residuals       199 34.384   0.1728
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.valid <- predict(revisedlinear.model, groups[[8]])
```

```
for (j in 1:length(revisedlinear.valid)) {
  if(revisedlinear.valid[j] < 0.5){
    revisedlinear.valid[j] <- 0
  }
  if(revisedlinear.valid[j] >= 0.5){
    revisedlinear.valid[j] <- 1
  }
}
```

```
revisedlinear.valid
```

```
## 25 33 34 37 41 45 72 74 75 100 103 133 135 141 142 145 148 162 193 195
## 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0
## 203 232 234 281 283 288 289 297
## 0 0 0 0 0 0 0 0
```

```
revisedlinear.RMSE <- c(revisedlinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - revisedlinear.valid)^2)))
revisedlinear.ConfMat[[8]] <- confusionMatrix(factor(revisedlinear.valid), factor(groups[[8]]$DEATH_EVENT))
```

```

revisedlinear accur <- c(revisedlinear accur, revisedlinear.ConfMat[[8]]$overall[1])
revisedlinear.precis <- c(revisedlinear.precis, revisedlinear.ConfMat[[8]]$byClass[5])
revisedlinear.recall <- c(revisedlinear.recall, revisedlinear.ConfMat[[8]]$byClass[6])
revisedlinear.F1score <- c(revisedlinear.F1score, revisedlinear.ConfMat[[8]]$byClass[7])
revisedlinear.ConfMat[[8]] <- revisedlinear.ConfMat[[8]]$table
revisedlinear.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, revisedlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.AUCsc <- c(revisedlinear.AUCsc, auc(groups[[8]]$DEATH_EVENT, revisedlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call: glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##      (Intercept)              age              anaemia
##      1.5546187          0.0598842          0.1213230
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##      0.0002703          -0.0604941          0.3822764
##      serum_creatinine          serum_sodium
##      0.5282822          -0.0371354
##
## Degrees of Freedom: 202 Total (i.e. Null); 195 Residual
## Null Deviance: 246.5
## Residual Deviance: 201.3 AIC: 217.3
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9606  -0.7671  -0.4752   0.8736   2.4606
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.5546187  5.8690077   0.265 0.791097
## age          0.0598842  0.0166652   3.593 0.000326 ***
## anaemia      0.1213230  0.3624572   0.335 0.737834
## creatinine_phosphokinase 0.0002703  0.0001781   1.518 0.129138
## ejection_fraction -0.0604941  0.0180989  -3.342 0.000831 ***
## high_blood_pressure  0.3822764  0.3599492   1.062 0.288223
## serum_creatinine  0.5282822  0.2082958   2.536 0.011206 *

```

```
## serum_sodium          -0.0371354  0.0427895  -0.868 0.385470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 246.47  on 202  degrees of freedom
## Residual deviance: 201.34  on 195  degrees of freedom
## AIC: 217.34
##
## Number of Fisher Scoring iterations: 5
```

```
anova(generlinear.model)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                                202      246.47
## age              1  16.5498      201      229.92
## anaemia          1   0.0004      200      229.92
## creatinine_phosphokinase 1   3.0340      199      226.88
## ejection_fraction  1  15.4007      198      211.48
## high_blood_pressure  1   0.3926      197      211.09
## serum_creatinine    1   8.9877      196      202.10
## serum_sodium        1   0.7620      195      201.34
```

```
generlinear.valid <- predict(generlinear.model, groups[[8]], type = 'response')
for (j in 1:length(generlinear.valid)) {
  if(generlinear.valid[j] < 0.5){
    generlinear.valid[j] <- 0
  }
  if(generlinear.valid[j] >= 0.5){
    generlinear.valid[j] <- 1
  }
}
generlinear.valid
```

```
## 25 33 34 37 41 45 72 74 75 100 103 133 135 141 142 145 148 162 193 195
## 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0
## 203 232 234 281 283 288 289 297
## 0 0 0 0 0 0 0 0 0
```

```
generlinear.RMSE <- c(generlinear.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - generlinear.valid)^2))/nrow
generlinear.ConfMat[[8]] <- confusionMatrix(factor(generlinear.valid), factor(groups[[8]]$DEATH_EVENT))
generlinear accur <- c(generlinear accur, generlinear.ConfMat[[8]]$overall[1])
generlinear.precis <- c(generlinear.precis, generlinear.ConfMat[[8]]$byClass[5])
generlinear.recall <- c(generlinear.recall, generlinear.ConfMat[[8]]$byClass[6])
generlinear.F1score <- c(generlinear.F1score, generlinear.ConfMat[[8]]$byClass[7])
generlinear.ConfMat[[8]] <- generlinear.ConfMat[[8]]$table
```

```

generlinear.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, generlinear.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.AUCsc <- c(generlinear.AUCsc, auc(groups[[8]]$DEATH_EVENT, generlinear.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.58 1.00 3.36 2.70 total = 10.64
##
## REML score: 111.8547
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2956     0.0272   10.87  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F  p-value
## s(age)         2.585   3.250 6.006 0.000427 ***
## s(creatinine_phosphokinase) 1.000   1.000 3.205 0.074970 .
## s(ejection_fraction)      3.362   4.174 7.791 5.87e-06 ***
## s(serum_creatinine)       2.698   3.348 4.248 0.004799 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.282 Deviance explained = 31.7%
## -REML = 111.85 Scale est. = 0.15016 n = 203
generaddit.valid <- predict(generaddit.model, groups[[8]], type = 'response')
for (j in 1:length(generaddit.valid)) {

```

```

if(generaddit.valid[j] < 0.5){
  generaddit.valid[j] <- 0
}
if(generaddit.valid[j] >= 0.5){
  generaddit.valid[j] <- 1
}
}
generaddit.valid

## 25 33 34 37 41 45 72 74 75 100 103 133 135 141 142 145 148 162 193 195
## 1 0 0 1 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 1
## 203 232 234 281 283 288 289 297
## 0 0 0 0 0 0 0 0

generaddit.RMSE <- c(generaddit.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - generaddit.valid)^2))/nrow(gr
generaddit.ConfMat[[8]] <- confusionMatrix(factor(generaddit.valid), factor(groups[[8]]$DEATH_EVENT))
generaddit accur <- c(generaddit accur, generaddit.ConfMat[[8]]$overall[1])
generaddit.precis <- c(generaddit.precis, generaddit.ConfMat[[8]]$byClass[5])
generaddit.recall <- c(generaddit.recall, generaddit.ConfMat[[8]]$byClass[6])
generaddit.F1score <- c(generaddit.F1score, generaddit.ConfMat[[8]]$byClass[7])
generaddit.ConfMat[[8]] <- generaddit.ConfMat[[8]]$table
generaddit.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, generaddit.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generaddit.AUCsc <- c(generaddit.AUCsc, auc(groups[[8]]$DEATH_EVENT, generaddit.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
## 0 1
## 0.7044335 0.2955665
##
## Group means:
## age anaemia creatinine_phosphokinase diabetes ejection_fraction
## 0 58.67599 0.4055944 529.7762 0.4055944 39.35664
## 1 66.09445 0.4500000 699.7167 0.4166667 33.86667
## high_blood_pressure platelets serum_creatinine serum_sodium sex
## 0 0.3636364 264906.9 1.209021 137.4266 0.6363636
## 1 0.4333333 267540.5 1.827000 135.9167 0.6166667
## smoking
## 0 0.3076923
## 1 0.3166667
##
## Coefficients of linear discriminants:
## LD1
## age 5.579142e-02
## anaemia 8.551157e-02

```

```

## creatinine_phosphokinase 2.635256e-04
## diabetes                2.312565e-01
## ejection_fraction       -5.057272e-02
## high_blood_pressure     2.706223e-01
## platelets               -1.021914e-07
## serum_creatinine        4.616477e-01
## serum_sodium            -4.884848e-02
## sex                    -4.330050e-01
## smoking                 4.497766e-01

lindiscr.valid <- predict(lindiscr.model, groups[[8]], type = 'response')
lindiscr.valid <- as.numeric(lindiscr.valid$class)-1
lindiscr.valid

## [1] 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

lindiscr.RMSE <- c(lindiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - lindiscr.valid)^2))/nrow(groups[[8]]))
lindiscr.ConfMat[[8]] <- confusionMatrix(factor(lindiscr.valid), factor(groups[[8]]$DEATH_EVENT))
lindiscr accur <- c(lindiscr accur, lindiscr.ConfMat[[8]]$overall[1])
lindiscr.precis <- c(lindiscr.precis, lindiscr.ConfMat[[8]]$byClass[5])
lindiscr.recall <- c(lindiscr.recall, lindiscr.ConfMat[[8]]$byClass[6])
lindiscr.F1score <- c(lindiscr.F1score, lindiscr.ConfMat[[8]]$byClass[7])
lindiscr.ConfMat[[8]] <- lindiscr.ConfMat[[8]]$table
lindiscr.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, lindiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.AUCsc <- c(lindiscr.AUCsc, auc(groups[[8]]$DEATH_EVENT, lindiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7044335 0.2955665
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.67599 0.4055944          529.7762 0.4055944          39.35664
## 1 66.09445 0.4500000          699.7167 0.4166667          33.86667
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3636364 264906.9          1.209021      137.4266 0.6363636
## 1          0.4333333 267540.5          1.827000      135.9167 0.6166667
##      smoking
## 0 0.3076923
## 1 0.3166667

quaddiscr.valid <- predict(quaddiscr.model, groups[[8]], type = 'response')
quaddiscr.valid <- as.numeric(quaddiscr.valid$class)-1
quaddiscr.valid

```

```

## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0
quaddiscr.RMSE <- c(quaddiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - quaddiscr.valid)^2))/nrow(groups[[8]]$DEATH_EVENT))
quaddiscr.ConfMat[[8]] <- confusionMatrix(factor(quaddiscr.valid), factor(groups[[8]]$DEATH_EVENT))
quaddiscr accur <- c(quaddiscr accur, quaddiscr.ConfMat[[8]]$overall[1])
quaddiscr.precis <- c(quaddiscr.precis, quaddiscr.ConfMat[[8]]$byClass[5])
quaddiscr.recall <- c(quaddiscr.recall, quaddiscr.ConfMat[[8]]$byClass[6])
quaddiscr.F1score <- c(quaddiscr.F1score, quaddiscr.ConfMat[[8]]$byClass[7])
quaddiscr.ConfMat[[8]] <- quaddiscr.ConfMat[[8]]$table
quaddiscr.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, quaddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
quaddiscr.AUCsc <- c(quaddiscr.AUCsc, auc(groups[[8]]$DEATH_EVENT, quaddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 71.50 89.46 96.05 99.50 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23645 ( N = 203 )
##
## Deviance: 202.363
mixeddiscr.valid <- predict(mixeddiscr.model, groups[[8]], type = 'class')
mixeddiscr.valid <- as.numeric(mixeddiscr.valid)-1
mixeddiscr.valid

## [1] 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
mixeddiscr.RMSE <- c(mixeddiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - mixeddiscr.valid)^2))/nrow(groups[[8]]$DEATH_EVENT))
mixeddiscr.ConfMat[[8]] <- confusionMatrix(factor(mixeddiscr.valid), factor(groups[[8]]$DEATH_EVENT))
mixeddiscr accur <- c(mixeddiscr accur, mixeddiscr.ConfMat[[8]]$overall[1])
mixeddiscr.precis <- c(mixeddiscr.precis, mixeddiscr.ConfMat[[8]]$byClass[5])
mixeddiscr.recall <- c(mixeddiscr.recall, mixeddiscr.ConfMat[[8]]$byClass[6])
mixeddiscr.F1score <- c(mixeddiscr.F1score, mixeddiscr.ConfMat[[8]]$byClass[7])
mixeddiscr.ConfMat[[8]] <- mixeddiscr.ConfMat[[8]]$table
mixeddiscr.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, mixeddiscr.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.AUCsc <- c(mixeddiscr.AUCsc, auc(groups[[8]]$DEATH_EVENT, mixeddiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

```

```

flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.23153 ( N = 203 )

flexdiscr.valid <- predict(flexdiscr.model, groups[[8]], type = 'class')
flexdiscr.valid <- as.numeric(flexdiscr.valid)-1
flexdiscr.valid

## [1] 1 0 0 0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0

flexdiscr.RMSE <- c(flexdiscr.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - flexdiscr.valid)^2))/nrow(groups[[8]]$DEATH_EVENT))
flexdiscr.ConfMat[[8]] <- confusionMatrix(factor(flexdiscr.valid), factor(groups[[8]]$DEATH_EVENT))
flexdiscr accur <- c(flexdiscr accur, flexdiscr.ConfMat[[8]]$overall[1])
flexdiscr.precis <- c(flexdiscr.precis, flexdiscr.ConfMat[[8]]$byClass[5])
flexdiscr.recall <- c(flexdiscr.recall, flexdiscr.ConfMat[[8]]$byClass[6])
flexdiscr.F1score <- c(flexdiscr.F1score, flexdiscr.ConfMat[[8]]$byClass[7])
flexdiscr.ConfMat[[8]] <- flexdiscr.ConfMat[[8]]$table
flexdiscr.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, flexdiscr.valid)

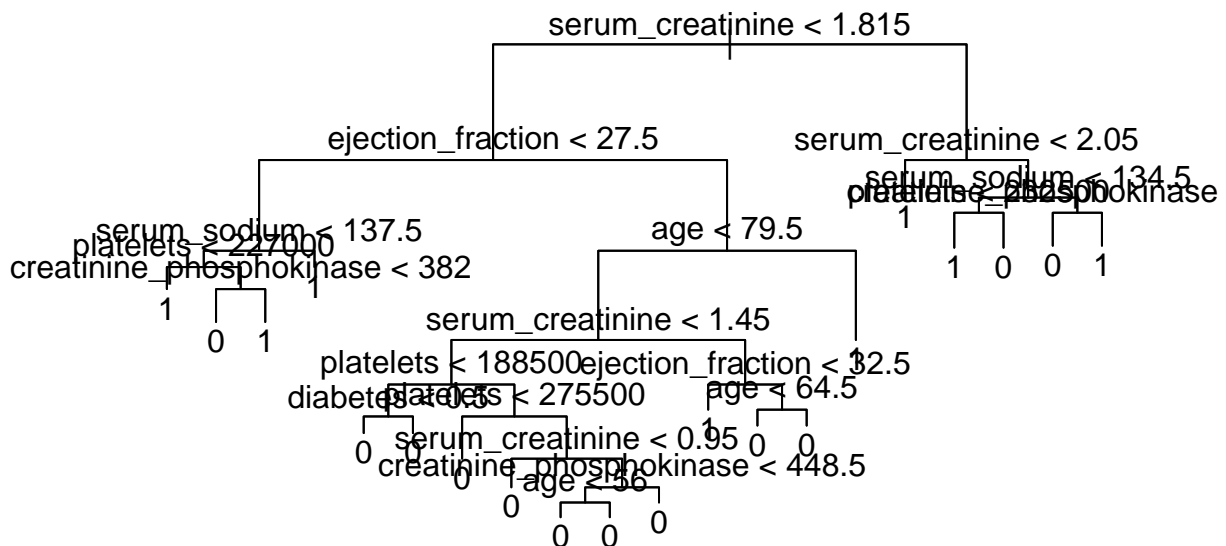
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

flexdiscr.AUCsc <- c(flexdiscr.AUCsc, auc(groups[[8]]$DEATH_EVENT, flexdiscr.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```

```

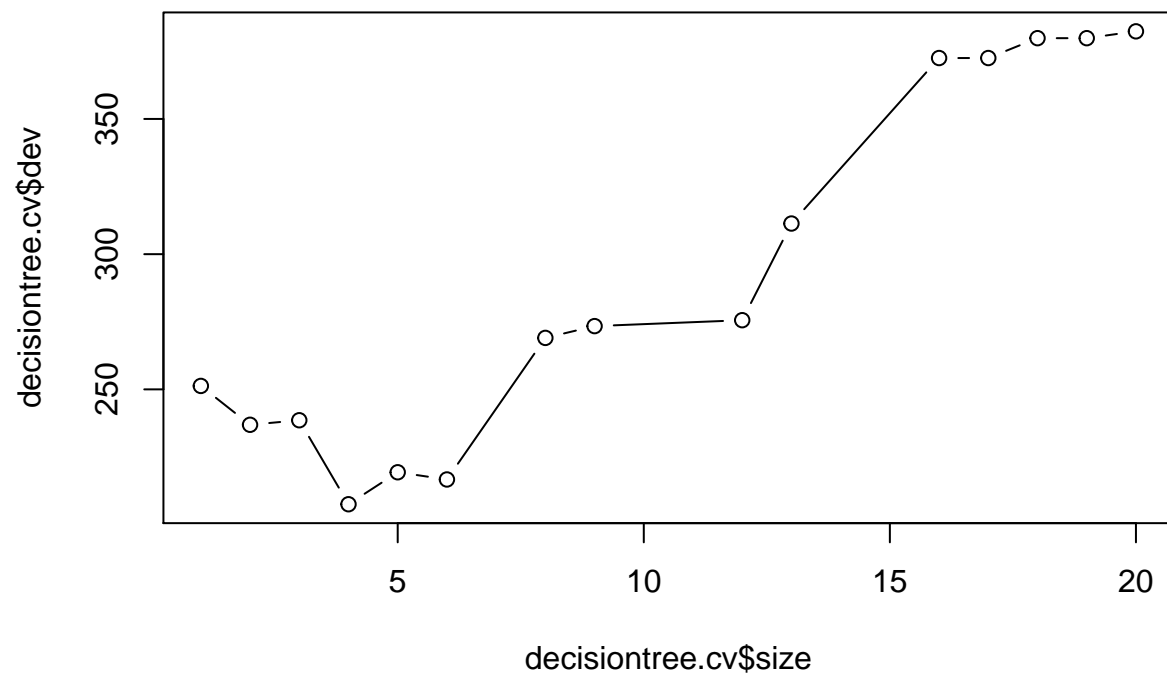
decisiontree.valid <- predict(decisiontree.model, groups[[8]], type = 'class')
decisiontree.valid <- as.numeric(decisiontree.valid)-1
decisiontree.valid

## [1] 1 0 0 1 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0
decisiontree.RMSE <- c(decisiontree.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - decisiontree.valid)^2))/n)
decisiontree.ConfMat[[8]] <- confusionMatrix(factor(decisiontree.valid), factor(groups[[8]]$DEATH_EVENT))
decisiontree accur <- c(decisiontree accur, decisiontree.ConfMat[[8]]$overall[1])
decisiontree.precis <- c(decisiontree.precis, decisiontree.ConfMat[[8]]$byClass[5])
decisiontree.recall <- c(decisiontree.recall, decisiontree.ConfMat[[8]]$byClass[6])
decisiontree.F1score <- c(decisiontree.F1score, decisiontree.ConfMat[[8]]$byClass[7])
decisiontree.ConfMat[[8]] <- decisiontree.ConfMat[[8]]$table
decisiontree.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, decisiontree.valid)

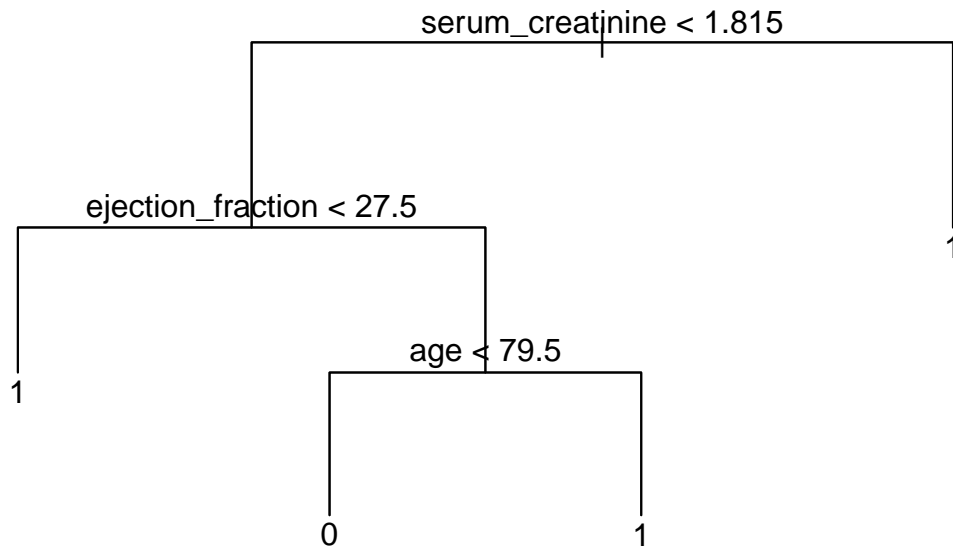
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
decisiontree.AUCsc <- c(decisiontree.AUCsc, auc(groups[[8]]$DEATH_EVENT, decisiontree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.valid <- predict(prunedectree.model, groups[[8]], type = 'class')
prunedectree.valid <- as.numeric(prunedectree.valid)-1
prunedectree.valid

## [1] 1 0 0 1 1 0 0 0 1 0 1 0 1 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0
prunedectree.RMSE <- c(prunedectree.RMSE, sqrt(sum((groups[[8]]$DEATH_EVENT - prunedectree.valid)^2))/n)
prunedectree.ConfMat[[8]] <- confusionMatrix(factor(prunedectree.valid), factor(groups[[8]]$DEATH_EVENT))
prunedectree accur <- c(prunedectree accur, prunedectree.ConfMat[[8]]$overall[1])
prunedectree.precis <- c(prunedectree.precis, prunedectree.ConfMat[[8]]$byClass[5])
prunedectree.recall <- c(prunedectree.recall, prunedectree.ConfMat[[8]]$byClass[6])
prunedectree.F1score <- c(prunedectree.F1score, prunedectree.ConfMat[[8]]$byClass[7])
prunedectree.ConfMat[[8]] <- prunedectree.ConfMat[[8]]$table
prunedectree.ROCcurv[[8]] <- roc(groups[[8]]$DEATH_EVENT, prunedectree.valid)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
prunedectree.AUCsc <- c(prunedectree.AUCsc, auc(groups[[8]]$DEATH_EVENT, prunedectree.valid))

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##Model comparison over training cross-validation
sum(simplelinear.RMSE)/8

## [1] 0.09959962

```

```
sum(simplelinear accur)/8
```

```
## [1] 0.713211
```

```
sum(simplelinear precis)/8
```

```
## [1] 0.7479634
```

```
sum(simplelinear recall)/8
```

```
## [1] 0.8928126
```

```
sum(simplelinear F1score)/8
```

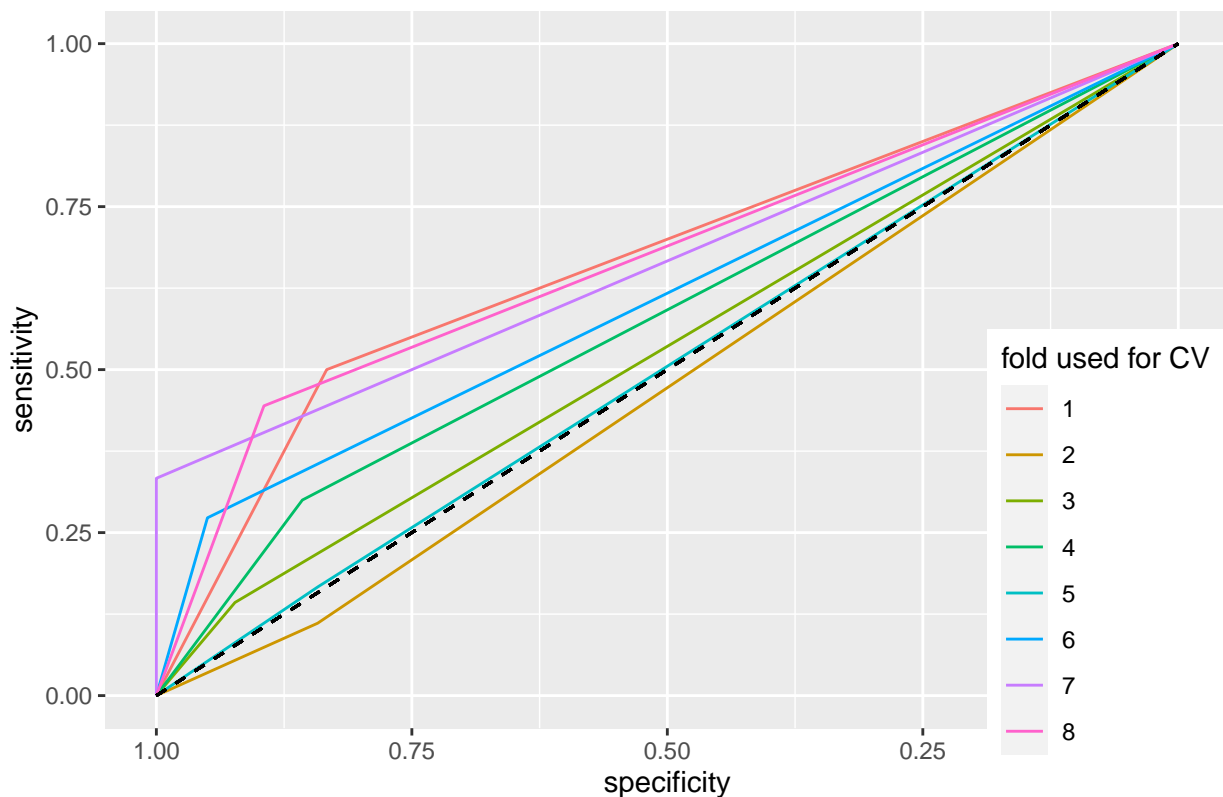
```
## [1] 0.8127616
```

```
sum(simplelinear AUCsc)/8
```

```
## [1] 0.5883525
```

```
ggroc(simplelinear.ROCcurv, aes = c('colour', linetype = 1, size = 2))+  
  ggtitle('ROC curves from 8-fold cross-validation for simplelinear.model')+  
  labs(colour = 'fold used for CV')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+  
  theme(legend.position = c(0.89, 0.25))
```

ROC curves from 8-fold cross-validation for simplelinear.model



```
simplelinear.ConfMat
```

```
## [[1]]
```

```
##      Reference
```

```

## Prediction 0 1
##           0 15 4
##           1 3 4
##
## [[2]]
##           Reference
## Prediction 0 1
##           0 16 8
##           1 3 1
##
## [[3]]
##           Reference
## Prediction 0 1
##           0 24 6
##           1 2 1
##
## [[4]]
##           Reference
## Prediction 0 1
##           0 18 7
##           1 3 3
##
## [[5]]
##           Reference
## Prediction 0 1
##           0 16 5
##           1 3 1
##
## [[6]]
##           Reference
## Prediction 0 1
##           0 19 8
##           1 1 3
##
## [[7]]
##           Reference
## Prediction 0 1
##           0 20 6
##           1 0 3
##
## [[8]]
##           Reference
## Prediction 0 1
##           0 17 5
##           1 2 4

sum(revisedlinear.RMSE)/8

## [1] 0.09501127

sum(revisedlinear accur)/8

## [1] 0.738739

```

```
sum(revisedlinear.precis)/8
```

```
## [1] 0.7592342
```

```
sum(revisedlinear.recall)/8
```

```
## [1] 0.9175776
```

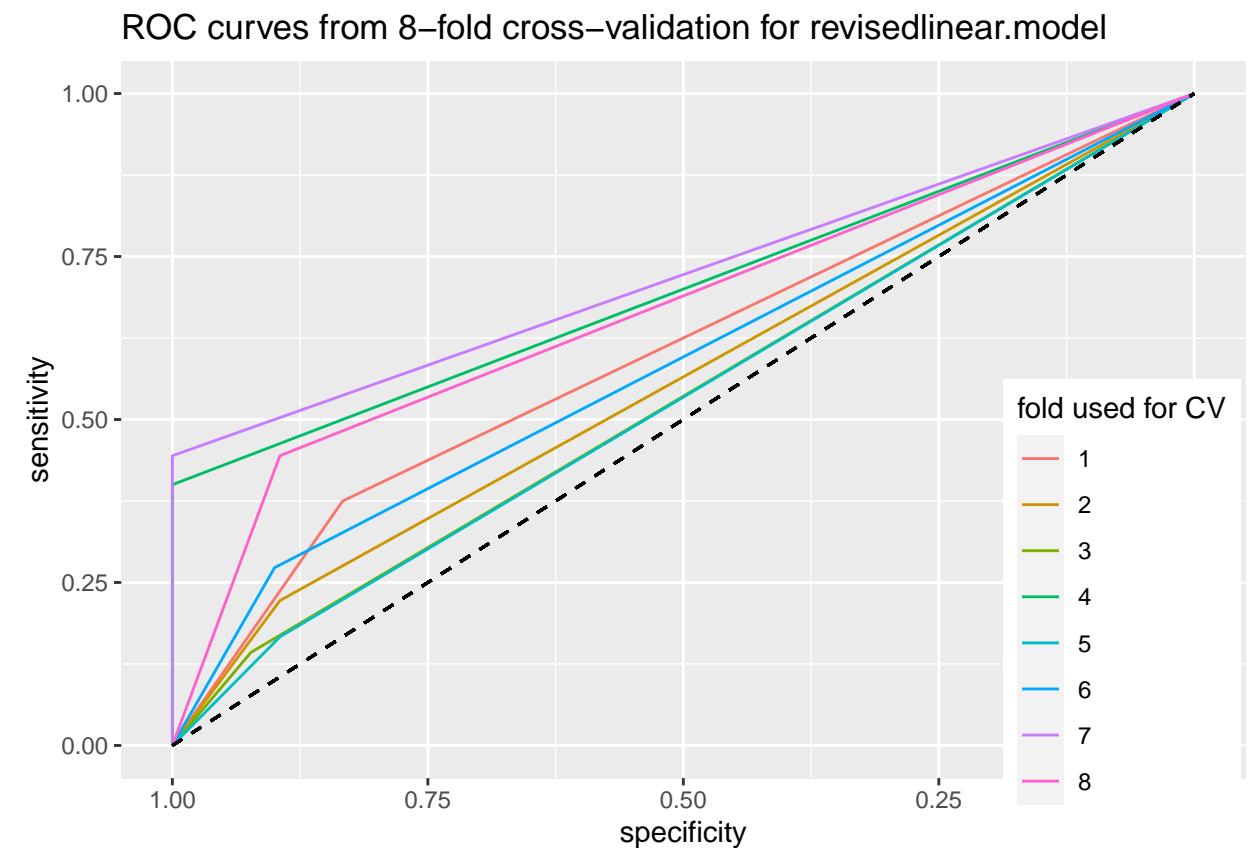
```
sum(revisedlinear.Fscore)/8
```

```
## [1] 0.8302935
```

```
sum(revisedlinear.AUCsc)/8
```

```
## [1] 0.6130614
```

```
ggroc(revisedlinear.ROCcurv, aes = c('colour', linetype = 1, size = 2))+  
  ggtitle('ROC curves from 8-fold cross-validation for revisedlinear.model')+  
  labs(colour = 'fold used for CV')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+  
  theme(legend.position = c(0.89, 0.25))
```



```
revisedlinear.ConfMat
```

```
## [[1]]
```

```
##      Reference
```

```
## Prediction 0 1
```

```
##      0 15 5
```

```
##      1 3 3
```

```
##
```

```
## [[2]]
##           Reference
## Prediction  0  1
##           0 17  7
##           1  2  2
##
```

```
## [[3]]
##           Reference
## Prediction  0  1
##           0 24  6
##           1  2  1
##
```

```
## [[4]]
##           Reference
## Prediction  0  1
##           0 21  6
##           1  0  4
##
```

```
## [[5]]
##           Reference
## Prediction  0  1
##           0 17  5
##           1  2  1
##
```

```
## [[6]]
##           Reference
## Prediction  0  1
##           0 18  8
##           1  2  3
##
```

```
## [[7]]
##           Reference
## Prediction  0  1
##           0 20  5
##           1  0  4
##
```

```
## [[8]]
##           Reference
## Prediction  0  1
##           0 17  5
##           1  2  4
##
```

```
sum(generlinear.RMSE)/8
```

```
## [1] 0.09966705
```

```
sum(generlinear accur)/8
```

```
## [1] 0.7131773
```

```
sum(generlinear.precis)/8
```

```
## [1] 0.7567957
```

```
sum(generlinear.recall)/8
```

```
## [1] 0.8755049
```

```
sum(generlinear.F1score)/8
```

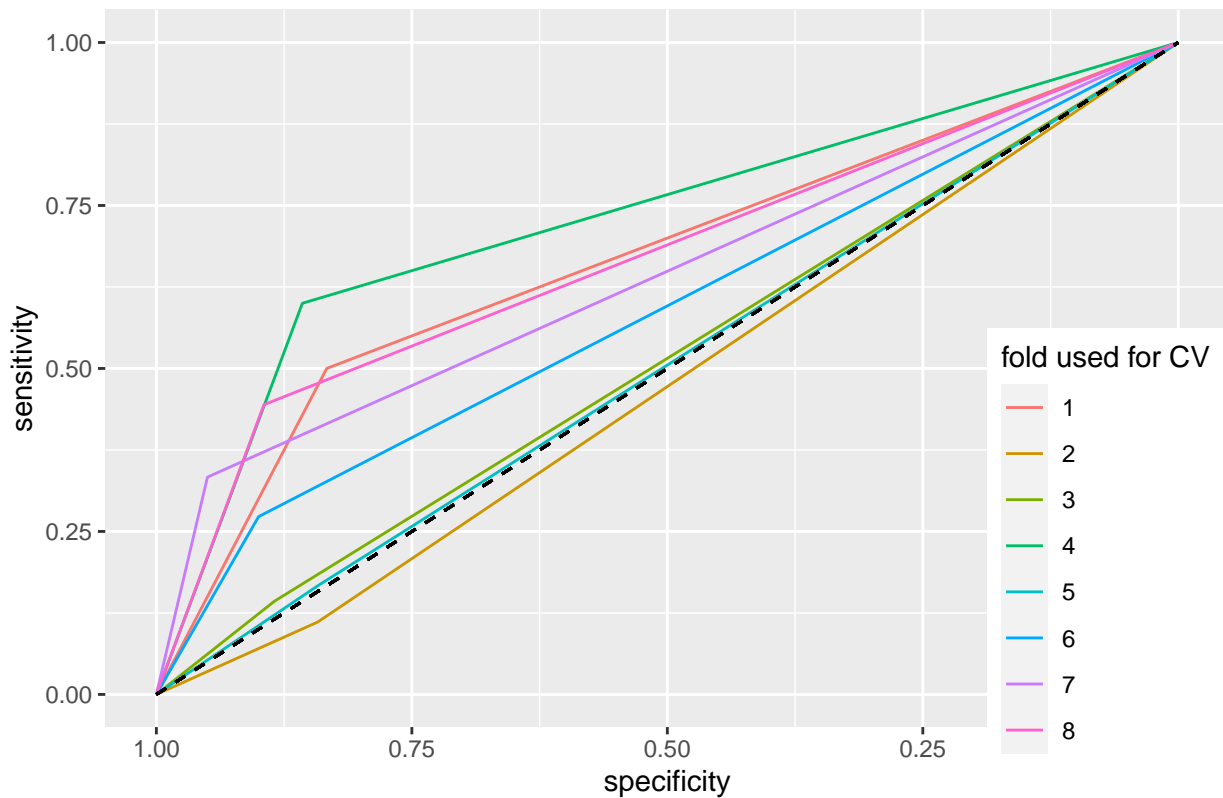
```
## [1] 0.8106114
```

```
sum(generlinear.AUCsc)/8
```

```
## [1] 0.5984487
```

```
ggroc(generlinear.ROCcurv, aes = c('colour', linetype = 1, size = 2))+  
  ggtitle('ROC curves from 8-fold cross-validation for generlinear.model')+  
  labs(colour = 'fold used for CV')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+  
  theme(legend.position = c(0.89, 0.25))
```

ROC curves from 8-fold cross-validation for generlinear.model



```
generlinear.ConfMat
```

```
## [[1]]
```

```
##           Reference
```

```
## Prediction 0 1
```

```
##           0 15 4
```

```
##           1 3 4
```

```
##
```

```
## [[2]]
```

```
##           Reference
```

```
## Prediction 0 1
```

```
##           0 16 8
```

```
##           1 3 1
```

```
##
```



```
## [[3]]
##           Reference
## Prediction  0  1
##           0 23  6
##           1  3  1
##
```

```
## [[4]]
##           Reference
## Prediction  0  1
##           0 18  4
##           1  3  6
##
```

```
## [[5]]
##           Reference
## Prediction  0  1
##           0 16  5
##           1  3  1
##
```

```
## [[6]]
##           Reference
## Prediction  0  1
##           0 18  8
##           1  2  3
##
```

```
## [[7]]
##           Reference
## Prediction  0  1
##           0 19  6
##           1  1  3
##
```

```
## [[8]]
##           Reference
## Prediction  0  1
##           0 17  5
##           1  2  4
##
```

```
sum(generaddit.RMSE)/8
```

```
## [1] 0.09467038
```

```
sum(generaddit accur)/8
```

```
## [1] 0.7405892
```

```
sum(generaddit.precis)/8
```

```
## [1] 0.7770175
```

```
sum(generaddit.recall)/8
```

```
## [1] 0.8880362
```

```
sum(generaddit.F1score)/8
```

```
## [1] 0.8278937
```

```
sum(generaddit.AUCsc)/8
```

```
## [1] 0.6328439
```

```
ggroc(generaddit.ROCcurv, aes = c('colour', linetype = 1, size = 2))+
  ggtitle('ROC curves from 8-fold cross-validation for generaddit.model')+
  labs(colour = 'fold used for CV')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+
  theme(legend.position = c(0.89, 0.25))
```



```
generaddit.ConfMat
```

```
## [[1]]
##      Reference
## Prediction 0 1
##      0 15 4
##      1 3 4
##
## [[2]]
##      Reference
## Prediction 0 1
##      0 17 7
##      1 2 2
##
## [[3]]
##      Reference
## Prediction 0 1
##      0 23 7
##      1 3 0
```

```
##
## [[4]]
##           Reference
## Prediction  0  1
##           0 19  6
##           1  2  4
##
## [[5]]
##           Reference
## Prediction  0  1
##           0 16  4
##           1  3  2
##
## [[6]]
##           Reference
## Prediction  0  1
##           0 18  6
##           1  2  5
##
## [[7]]
##           Reference
## Prediction  0  1
##           0 19  5
##           1  1  4
##
## [[8]]
##           Reference
## Prediction  0  1
##           0 17  3
##           1  2  6

sum(lindiscr.RMSE)/8

## [1] 0.1036575

sum(lindiscr accur)/8

## [1] 0.6880544

sum(lindiscr.precis)/8

## [1] 0.7391981

sum(lindiscr.recall)/8

## [1] 0.8557315

sum(lindiscr.F1score)/8

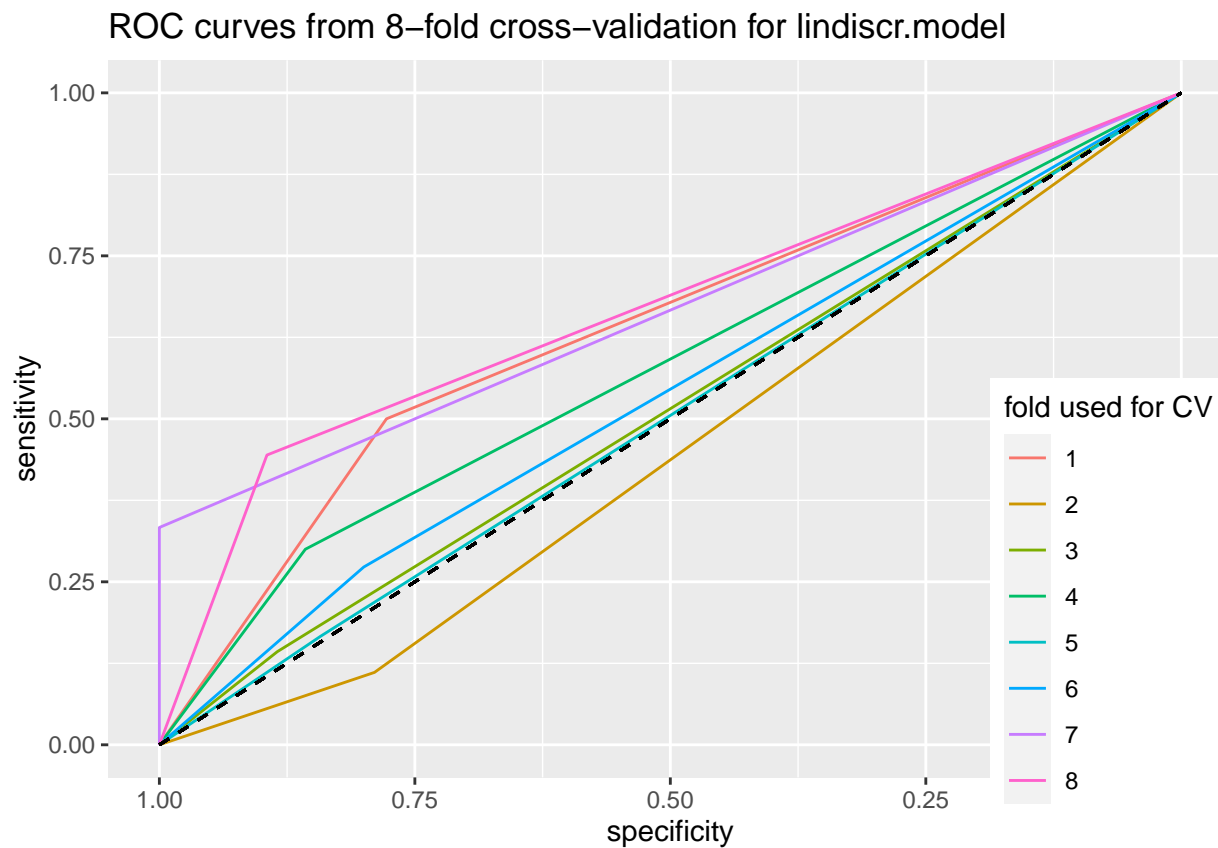
## [1] 0.7921428

sum(lindiscr.AUCsc)/8

## [1] 0.569812

ggroc(lindiscr.ROCcurv, aes = c('colour', linetype = 1, size = 2))+
  ggtitle('ROC curves from 8-fold cross-validation for lindiscr.model')+
  labs(colour = 'fold used for CV')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+
```

```
theme(legend.position = c(0.89, 0.25))
```



```
lindiscr.ConfMat
```

```
## [[1]]
##           Reference
## Prediction 0  1
##           0 14  4
##           1  4  4
##
## [[2]]
##           Reference
## Prediction 0  1
##           0 15  8
##           1  4  1
##
## [[3]]
##           Reference
## Prediction 0  1
##           0 23  6
##           1  3  1
##
## [[4]]
##           Reference
## Prediction 0  1
##           0 18  7
```

```

##           1  3  3
##
## [[5]]
##           Reference
## Prediction  0  1
##           0 16  5
##           1  3  1
##
## [[6]]
##           Reference
## Prediction  0  1
##           0 16  8
##           1  4  3
##
## [[7]]
##           Reference
## Prediction  0  1
##           0 20  6
##           1  0  3
##
## [[8]]
##           Reference
## Prediction  0  1
##           0 17  5
##           1  2  4

sum(quaddiscr.RMSE)/8

## [1] 0.1066421

sum(quaddiscr accur)/8

## [1] 0.6668111

sum(quaddiscr.precis)/8

## [1] 0.7142866

sum(quaddiscr.recall)/8

## [1] 0.867331

sum(quaddiscr.F1score)/8

## [1] 0.7821984

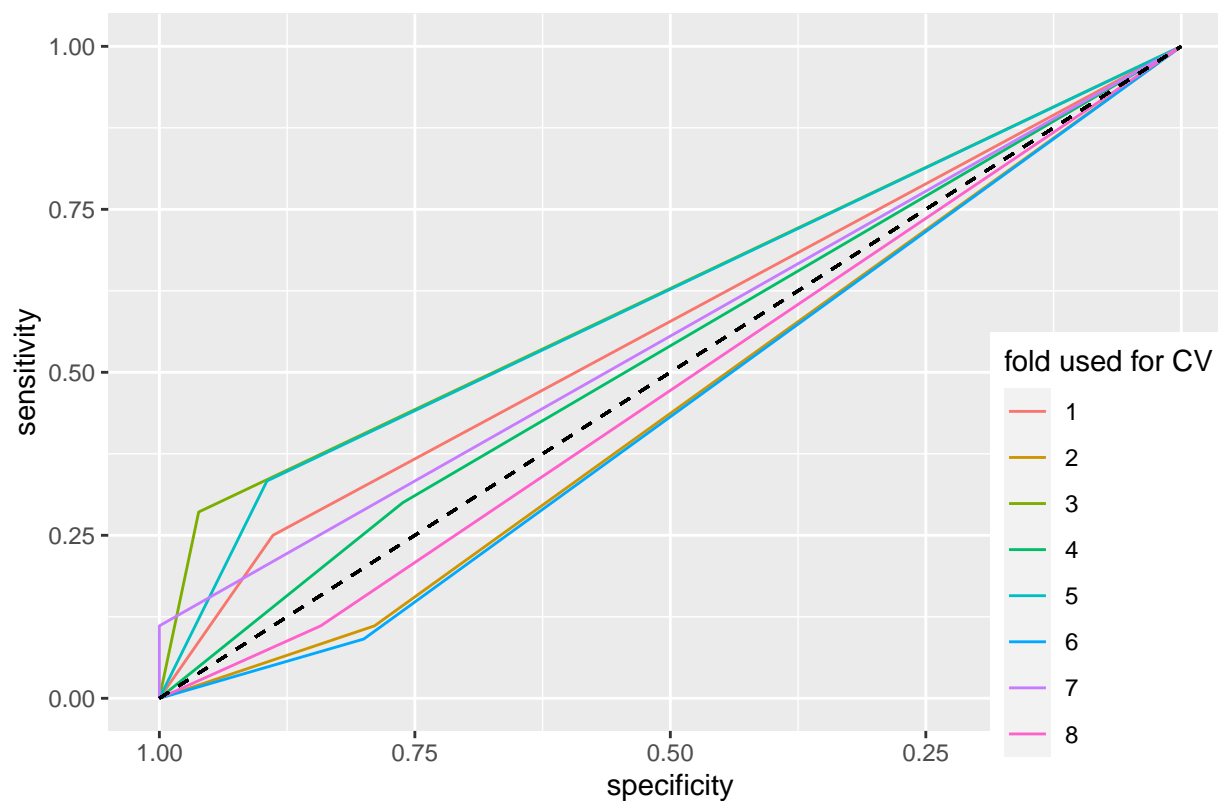
sum(quaddiscr.AUCsc)/8

## [1] 0.5332461

ggroc(quaddiscr.ROCcurv, aes = c('colour', linetype = 1, size = 2))+
  ggtitle('ROC curves from 8-fold cross-validation for quaddiscr.model')+
  labs(colour = 'fold used for CV')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+
  theme(legend.position = c(0.89, 0.25))

```

ROC curves from 8-fold cross-validation for quaddiscr.model



quaddiscr.ConfMat

```
## [[1]]
##           Reference
## Prediction 0  1
##           0 16  6
##           1  2  2
##
## [[2]]
##           Reference
## Prediction 0  1
##           0 15  8
##           1  4  1
##
## [[3]]
##           Reference
## Prediction 0  1
##           0 25  5
##           1  1  2
##
## [[4]]
##           Reference
## Prediction 0  1
##           0 16  7
##           1  5  3
##
```

```
## [[5]]
##           Reference
## Prediction  0  1
##           0 17  4
##           1  2  2
##
```

```
## [[6]]
##           Reference
## Prediction  0  1
##           0 16 10
##           1  4  1
##
```

```
## [[7]]
##           Reference
## Prediction  0  1
##           0 20  8
##           1  0  1
##
```

```
## [[8]]
##           Reference
## Prediction  0  1
##           0 16  8
##           1  3  1
##
```

```
sum(mixeddiscr.RMSE)/8
```

```
## [1] 0.1055745
```

```
sum(mixeddiscr accur)/8
```

```
## [1] 0.6767798
```

```
sum(mixeddiscr.precis)/8
```

```
## [1] 0.7269975
```

```
sum(mixeddiscr.recall)/8
```

```
## [1] 0.8608368
```

```
sum(mixeddiscr.F1score)/8
```

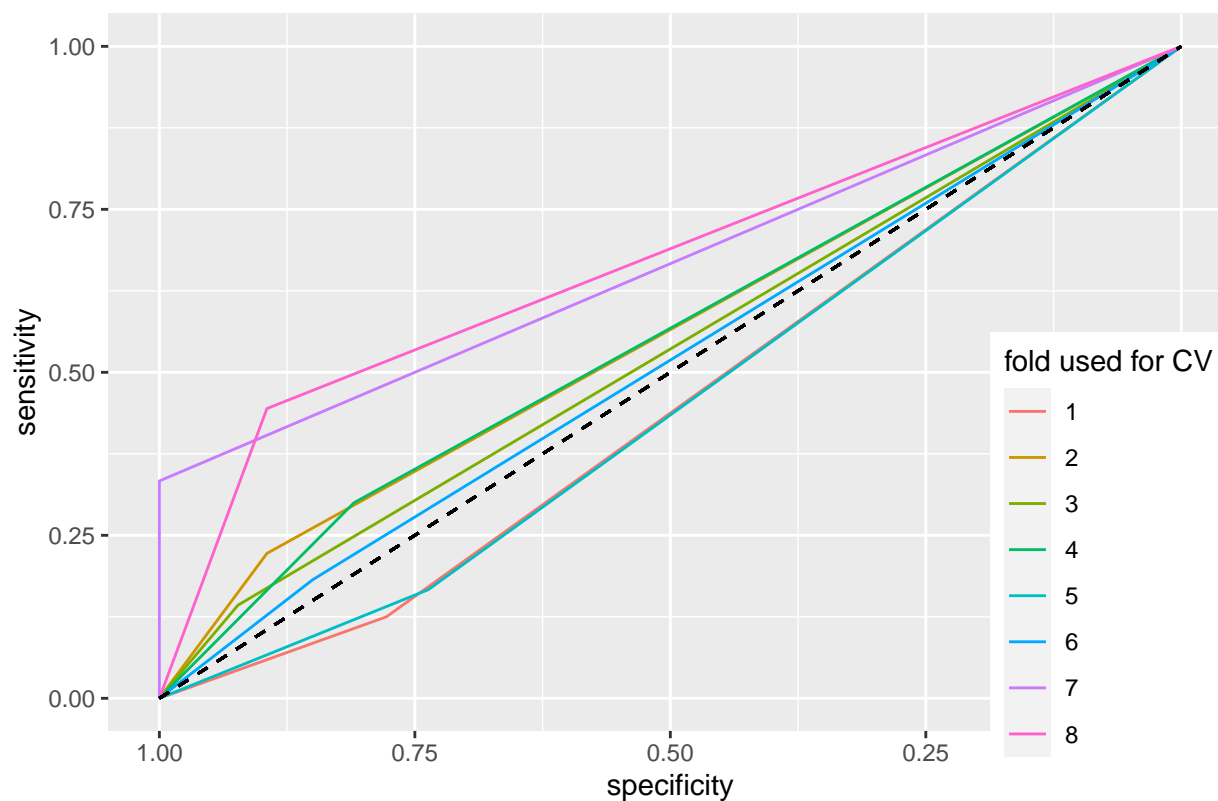
```
## [1] 0.7870189
```

```
sum(mixeddiscr.AUCsc)/8
```

```
## [1] 0.5501898
```

```
ggroc(mixeddiscr.ROCcurv, aes = c('colour', linetype = 1, size = 2))+
  ggtitle('ROC curves from 8-fold cross-validation for mixeddiscr.model')+
  labs(colour = 'fold used for CV')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+
  theme(legend.position = c(0.89, 0.25))
```

ROC curves from 8-fold cross-validation for mixeddiscr.model



```
mixeddiscr.ConfMat
```

```
## [[1]]
##           Reference
## Prediction 0  1
##           0 14  7
##           1  4  1
##
## [[2]]
##           Reference
## Prediction 0  1
##           0 17  7
##           1  2  2
##
## [[3]]
##           Reference
## Prediction 0  1
##           0 24  6
##           1  2  1
##
## [[4]]
##           Reference
## Prediction 0  1
##           0 17  7
##           1  4  3
##
## [[5]]
```



```
##           Reference
## Prediction  0  1
##           0 14  5
##           1  5  1
##
```

```
## [[6]]
##           Reference
## Prediction  0  1
##           0 17  9
##           1  3  2
##
```

```
## [[7]]
##           Reference
## Prediction  0  1
##           0 20  6
##           1  0  3
##
```

```
## [[8]]
##           Reference
## Prediction  0  1
##           0 17  5
##           1  2  4
##
```

```
sum(flexdiscr.RMSE)/8
```

```
## [1] 0.1036575
```

```
sum(flexdiscr accur)/8
```

```
## [1] 0.6880544
```

```
sum(flexdiscr.precis)/8
```

```
## [1] 0.7391981
```

```
sum(flexdiscr.recall)/8
```

```
## [1] 0.8557315
```

```
sum(flexdiscr.F1score)/8
```

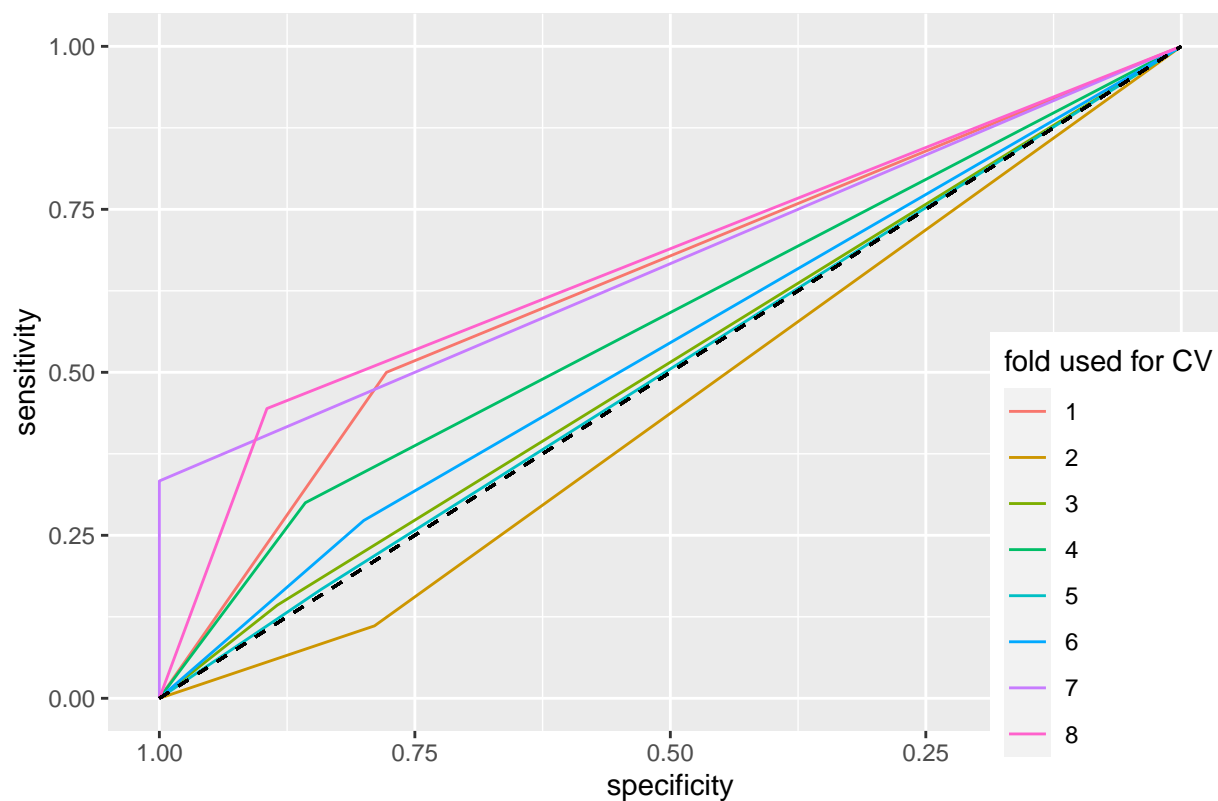
```
## [1] 0.7921428
```

```
sum(flexdiscr.AUCsc)/8
```

```
## [1] 0.569812
```

```
ggroc(flexdiscr.ROCcurv, aes = c('colour', linetype = 1, size = 2))+
  ggtitle('ROC curves from 8-fold cross-validation for flexdiscr.model')+
  labs(colour = 'fold used for CV')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+
  theme(legend.position = c(0.89, 0.25))
```

ROC curves from 8-fold cross-validation for flexdiscr.model



```
flexdiscr.ConfMat
```

```
## [[1]]
##           Reference
## Prediction 0  1
##           0 14  4
##           1  4  4
##
## [[2]]
##           Reference
## Prediction 0  1
##           0 15  8
##           1  4  1
##
## [[3]]
##           Reference
## Prediction 0  1
##           0 23  6
##           1  3  1
##
## [[4]]
##           Reference
## Prediction 0  1
##           0 18  7
##           1  3  3
##
## [[5]]
```

```
##           Reference
## Prediction  0  1
##           0 16  5
##           1  3  1
```

```
##
## [[6]]
##           Reference
## Prediction  0  1
##           0 16  8
##           1  4  3
```

```
##
## [[7]]
##           Reference
## Prediction  0  1
##           0 20  6
##           1  0  3
```

```
##
## [[8]]
##           Reference
## Prediction  0  1
##           0 17  5
##           1  2  4
```

```
sum(decisiontree.RMSE)/8
```

```
## [1] 0.09723775
```

```
sum(decisiontree accur)/8
```

```
## [1] 0.7230848
```

```
sum(decisiontree.precis)/8
```

```
## [1] 0.8048025
```

```
sum(decisiontree.recall)/8
```

```
## [1] 0.8015724
```

```
sum(decisiontree.F1score)/8
```

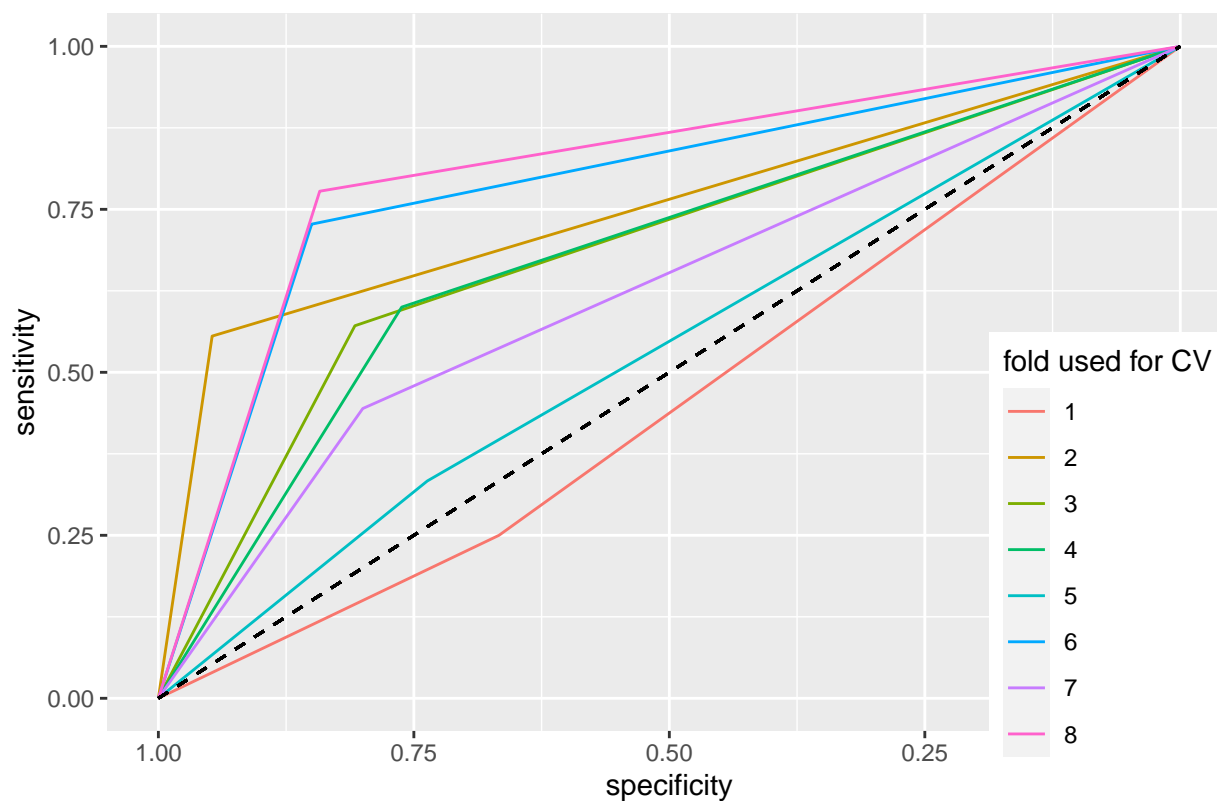
```
## [1] 0.8021641
```

```
sum(decisiontree.AUCsc)/8
```

```
## [1] 0.6670245
```

```
ggroc(decisiontree.ROCcurv, aes = c('colour', linetype = 1, size = 2))+
  ggtitle('ROC curves from 8-fold cross-validation for decisiontree.model')+
  labs(colour = 'fold used for CV')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+
  theme(legend.position = c(0.89, 0.25))
```

ROC curves from 8-fold cross-validation for decisiontree.model



```
decisiontree.ConfMat
```

```
## [[1]]
##           Reference
## Prediction 0  1
##           0 12  6
##           1  6  2
##
## [[2]]
##           Reference
## Prediction 0  1
##           0 18  4
##           1  1  5
##
## [[3]]
##           Reference
## Prediction 0  1
##           0 21  3
##           1  5  4
##
## [[4]]
##           Reference
## Prediction 0  1
##           0 16  4
##           1  5  6
##
## [[5]]
```

```
##           Reference
## Prediction  0  1
##           0 14  4
##           1  5  2
```

```
## [[6]]
##           Reference
## Prediction  0  1
##           0 17  3
##           1  3  8
```

```
## [[7]]
##           Reference
## Prediction  0  1
##           0 16  5
##           1  4  4
```

```
## [[8]]
##           Reference
## Prediction  0  1
##           0 16  2
##           1  3  7
```

```
sum(prunedectree.RMSE)/8
```

```
## [1] 0.09531263
```

```
sum(prunedectree accur)/8
```

```
## [1] 0.7355376
```

```
sum(prunedectree.precis)/8
```

```
## [1] 0.8165335
```

```
sum(prunedectree.recall)/8
```

```
## [1] 0.812353
```

```
sum(prunedectree.F1score)/8
```

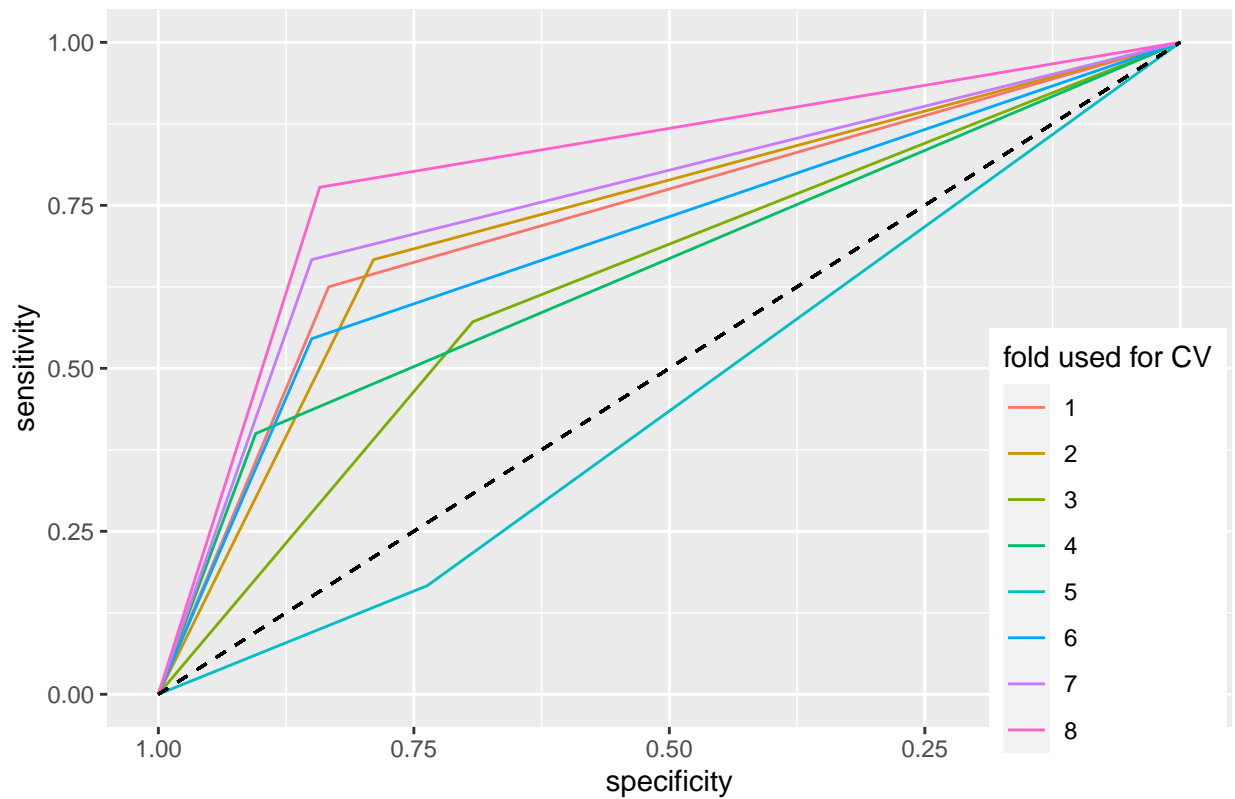
```
## [1] 0.8121774
```

```
sum(prunedectree.AUCsc)/8
```

```
## [1] 0.6824053
```

```
ggroc(prunedectree.ROCcurv, aes = c('colour', linetype = 1, size = 2))+
  ggtitle('ROC curves from 8-fold cross-validation for prunedectree.model')+
  labs(colour = 'fold used for CV')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)+
  theme(legend.position = c(0.89, 0.25))
```

ROC curves from 8-fold cross-validation for prunedectree.model



prunedectree.ConfMat

```
## [[1]]
##           Reference
## Prediction 0  1
##           0 15  3
##           1  3  5
##
## [[2]]
##           Reference
## Prediction 0  1
##           0 15  3
##           1  4  6
##
## [[3]]
##           Reference
## Prediction 0  1
##           0 18  3
##           1  8  4
##
## [[4]]
##           Reference
## Prediction 0  1
##           0 19  6
##           1  2  4
##
```

```
## [[5]]
##           Reference
## Prediction 0  1
##           0 14  5
##           1  5  1
##
```

```
## [[6]]
##           Reference
## Prediction 0  1
##           0 17  5
##           1  3  6
##
```

```
## [[7]]
##           Reference
## Prediction 0  1
##           0 17  3
##           1  3  6
##
```

```
## [[8]]
##           Reference
## Prediction 0  1
##           0 16  2
##           1  3  7
##
```

##Conclusions from Model Comparision over Training Cross-Validation
##From what we see on our graphs, and the average AUC score over 8-fold
##Cross-Validation, the best models in this case are the Revised Linear
##model, Generalized Additive model, Decision Tree model, & Pruned Decision
##Tree model (who all have average AUC above 0.6). We must however note that
##these models are very far from perfect (a random guess model would have n
##AUC score near 0.5 while the perfect model would have a near or equal to 1
##AUC score).
##Thus, from our cross-validation, it doesn't seem like our classifications
##models are usable in any way from cross-validation results. Let us apply this
##on our test just to see if any significant changes appear

```
##Applied to test
trainset <- data.frame()
for (i in c(1:8)) {
  trainset <- rbind(trainset, groups[[i]])
}
testset <- data.frame()
for (i in c(9,10)) {
  testset <- rbind(testset, groups[[i]])
}

simplelinear.model <- lm(DEATH_EVENT ~ . -time, data = trainset)
simplelinear.model
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Coefficients:
##           (Intercept)                age                anaemia
```

```
##          1.423e+00          1.045e-02          3.230e-02
## creatinine_phosphokinase          diabetes          ejection_fraction
##          3.748e-05          4.680e-02          -9.334e-03
##          high_blood_pressure          platelets          serum_creatinine
##          8.462e-02          2.411e-08          8.338e-02
##          serum_sodium          sex          smoking
##          -1.159e-02          -7.086e-02          5.166e-02
```

```
summary(simplelinear.model)
```

```
##
## Call:
## lm(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7864 -0.2984 -0.1402  0.3740  0.9992
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.423e+00  9.716e-01   1.464 0.144615
## age            1.045e-02  2.402e-03   4.351 2.08e-05 ***
## anaemia        3.230e-02  5.771e-02   0.560 0.576254
## creatinine_phosphokinase 3.748e-05  2.946e-05   1.272 0.204550
## diabetes       4.680e-02  5.783e-02   0.809 0.419229
## ejection_fraction -9.334e-03  2.489e-03  -3.750 0.000226 ***
## high_blood_pressure  8.462e-02  5.780e-02   1.464 0.144597
## platelets       2.411e-08  2.879e-07   0.084 0.933333
## serum_creatinine  8.338e-02  2.979e-02   2.799 0.005587 **
## serum_sodium    -1.159e-02  6.980e-03  -1.660 0.098258 .
## sex            -7.086e-02  6.730e-02  -1.053 0.293563
## smoking         5.166e-02  6.781e-02   0.762 0.446944
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4161 on 219 degrees of freedom
## Multiple R-squared:  0.2165, Adjusted R-squared:  0.1772
## F-statistic: 5.502 on 11 and 219 DF,  p-value: 9.525e-08
```

```
anova(simplelinear.model)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: DEATH_EVENT
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age           1  4.096   4.0962  23.6620 2.193e-06 ***
## anaemia       1  0.018   0.0180   0.1040 0.747392
## creatinine_phosphokinase 1  0.294   0.2943   1.7001 0.193645
## diabetes      1  0.094   0.0936   0.5407 0.462940
## ejection_fraction 1  3.287   3.2874  18.9900 2.020e-05 ***
## high_blood_pressure 1  0.322   0.3220   1.8600 0.174026
## platelets     1  0.046   0.0457   0.2638 0.608027
## serum_creatinine 1  1.700   1.6999   9.8195 0.001963 **
## serum_sodium  1  0.410   0.4102   2.3696 0.125162
## sex           1  0.110   0.1097   0.6339 0.426781
```



```

## smoking          1 0.100 0.1005 0.5805 0.446944
## Residuals       219 37.912 0.1731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

simplelinear.test <- predict(simplelinear.model, testset)
for (j in 1:length(simplelinear.test)) {
  if(simplelinear.test[j] < 0.5){
    simplelinear.test[j] <- 0
  }
  if(simplelinear.test[j] >= 0.5){
    simplelinear.test[j] <- 1
  }
}
simplelinear.test

##      2      3      4      15      29      55      60      64      65      79      86      87      91      96      114      116      121      130      131      137
##      0      1      0      0      1      0      1      0      0      1      0      0      0      0      0      0      0      0      0      0
## 156 159 160 165 168 178 200 208 218 220 221 231 250 263 273 284 285 291  11  13
##      0      0      0      0      0      0      1      0      1      0      1      0      0      1      0      0      0      0      1      0
##  17  19  20  23  30  31  48  83  88  92  94 122 127 139 152 163 180 185 213 225
##      0      1      0      0      1      1      0      1      0      0      0      0      1      0      0      0      0      0      0      0
## 227 242 253 262 265 266 270 299
##      0      0      0      0      0      0      0      0
simplelinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - simplelinear.test)^2))/nrow(testset)
simplelinear.test.ConfMat <- confusionMatrix(factor(simplelinear.test), factor(testset$DEATH_EVENT))
simplelinear.test accur <- simplelinear.test.ConfMat$overall[1]
simplelinear.test.precis <- simplelinear.test.ConfMat$byClass[5]
simplelinear.test.recall <- simplelinear.test.ConfMat$byClass[6]
simplelinear.test.F1score <- simplelinear.test.ConfMat$byClass[7]
simplelinear.test.ConfMat <- simplelinear.test.ConfMat$table
simplelinear.test.ROCcurv <- roc(testset$DEATH_EVENT, simplelinear.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

simplelinear.test.AUCsc <- auc(testset$DEATH_EVENT, simplelinear.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

revisedlinear.model <- lm(DEATH_EVENT ~ age + ejection_fraction + serum_creatinine, data = trainset)
revisedlinear.model

##
## Call:
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Coefficients:
##      (Intercept)              age  ejection_fraction  serum_creatinine
##      -0.07880          0.01035          -0.00995           0.08979

summary(revisedlinear.model)

##
## Call:

```

```
## lm(formula = DEATH_EVENT ~ age + ejection_fraction + serum_creatinine,
##     data = trainset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7939 -0.2961 -0.1470  0.4008  1.0051
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.078798   0.160678  -0.490  0.62432
## age             0.010347   0.002339   4.423 1.51e-05 ***
## ejection_fraction -0.009950   0.002413  -4.123 5.25e-05 ***
## serum_creatinine  0.089788   0.029134   3.082 0.00231 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4162 on 227 degrees of freedom
## Multiple R-squared:  0.1875, Adjusted R-squared:  0.1768
## F-statistic: 17.47 on 3 and 227 DF, p-value: 3.08e-10
```

```
anova(revisedlinear.model)
```

```
## Analysis of Variance Table
##
## Response: DEATH_EVENT
##              Df Sum Sq Mean Sq F value    Pr(>F)
## age              1  4.096   4.0962  23.6514 2.157e-06 ***
## ejection_fraction  1  3.334   3.3338  19.2491 1.757e-05 ***
## serum_creatinine  1  1.645   1.6449   9.4977 0.002312 **
## Residuals       227 39.315   0.1732
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
revisedlinear.test <- predict(revisedlinear.model, testset)
for (j in 1:length(revisedlinear.test)) {
  if(revisedlinear.test[j] < 0.5){
    revisedlinear.test[j] <- 0
  }
  if(revisedlinear.test[j] >= 0.5){
    revisedlinear.test[j] <- 1
  }
}
revisedlinear.test
```

```
##      2      3      4      15      29      55      60      64      65      79      86      87      91      96      114      116      121      130      131      137
##      0      1      0      0      1      0      1      0      0      0      0      0      0      0      0      0      0      0      0      0
## 156 159 160 165 168 178 200 208 218 220 221 231 250 263 273 284 285 291  11  13
##      0      0      0      0      1      0      0      1      1      0      1      0      0      0      0      0      0      0      1      0
##      17      19      20      23      30      31      48      83      88      92      94      122      127      139      152      163      180      185      213      225
##      1      0      0      0      1      1      0      1      0      0      0      0      0      0      0      0      0      0      0      0
## 227 242 253 262 265 266 270 299
##      0      0      0      0      0      0      0      0
```

```
revisedlinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - revisedlinear.test)^2)/nrow(testset))
revisedlinear.test.ConfMat <- confusionMatrix(factor(revisedlinear.test), factor(testset$DEATH_EVENT))
revisedlinear.test accur <- revisedlinear.test.ConfMat$overall[1]
```

```

revisedlinear.test.precis <- revisedlinear.test.ConfMat$byClass[5]
revisedlinear.test.recall <- revisedlinear.test.ConfMat$byClass[6]
revisedlinear.test.F1score <- revisedlinear.test.ConfMat$byClass[7]
revisedlinear.test.ConfMat <- revisedlinear.test.ConfMat$table
revisedlinear.test.ROccurv <- roc(testset$DEATH_EVENT, revisedlinear.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
revisedlinear.test.AUCsc <- auc(testset$DEATH_EVENT, revisedlinear.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
generlinear.model <- glm(DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase + ejection_fraction + h
generlinear.model

##
## Call:  glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Coefficients:
##              (Intercept)                age                anaemia
##              3.6165550                0.0586927                0.2615002
## creatinine_phosphokinase      ejection_fraction      high_blood_pressure
##              0.0001998                -0.0580905                0.5620745
##              serum_creatinine      serum_sodium
##              0.5317051                -0.0528194
##
## Degrees of Freedom: 230 Total (i.e. Null);  223 Residual
## Null Deviance:      281.7
## Residual Deviance: 228.6      AIC: 244.6
summary(generlinear.model)

##
## Call:
## glm(formula = DEATH_EVENT ~ age + anaemia + creatinine_phosphokinase +
##      ejection_fraction + high_blood_pressure + serum_creatinine +
##      serum_sodium, family = binomial(link = "logit"), data = trainset)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9688  -0.7403  -0.4847   0.8420   2.4676
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.616550  5.5244721   0.655 0.512698
## age            0.0586927  0.0146583   4.004 6.23e-05 ***
## anaemia        0.2615002  0.3416900   0.765 0.444085
## creatinine_phosphokinase 0.0001998  0.0001631   1.225 0.220449
## ejection_fraction -0.0580905  0.0164650  -3.528 0.000419 ***
## high_blood_pressure  0.5620745  0.3365968   1.670 0.094944 .
## serum_creatinine    0.5317051  0.1975575   2.691 0.007115 **
## serum_sodium     -0.0528194  0.0404205  -1.307 0.191299

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 281.71  on 230  degrees of freedom
## Residual deviance: 228.57  on 223  degrees of freedom
## AIC: 244.57
##
## Number of Fisher Scoring iterations: 5
anova(generlinear.model)

## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: DEATH_EVENT
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev
## NULL                      230      281.71
## age                      1  19.8254      229      261.88
## anaemia                   1   0.1074      228      261.78
## creatinine_phosphokinase  1   1.4760      227      260.30
## ejection_fraction         1  18.0546      226      242.25
## high_blood_pressure        1   1.8036      225      240.44
## serum_creatinine           1  10.1363      224      230.31
## serum_sodium               1   1.7392      223      228.57

generlinear.test <- predict(generlinear.model, testset, type = 'response')
for (j in 1:length(generlinear.test)) {
  if(generlinear.test[j] < 0.5){
    generlinear.test[j] <- 0
  }
  if(generlinear.test[j] >= 0.5){
    generlinear.test[j] <- 1
  }
}
generlinear.test

##      2      3      4      15      29      55      60      64      65      79      86      87      91      96      114      116      121      130      131      137
##      0      1      0      0      1      0      1      0      0      1      0      0      0      0      0      0      0      0      0      0
## 156 159 160 165 168 178 200 208 218 220 221 231 250 263 273 284 285 291  11  13
##      0      0      0      0      1      0      1      0      1      0      1      0      0      1      0      0      0      0      1      0
##      17      19      20      23      30      31      48      83      88      92      94 122 127 139 152 163 180 185 213 225
##      0      1      0      0      1      1      0      1      0      0      0      0      1      0      0      0      0      0      0      0
## 227 242 253 262 265 266 270 299
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0

generlinear.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - generlinear.test)^2)/nrow(testset))
generlinear.test.ConfMat <- confusionMatrix(factor(generlinear.test), factor(testset$DEATH_EVENT))
generlinear.test accur <- generlinear.test.ConfMat$overall[1]
generlinear.test.precis <- generlinear.test.ConfMat$byClass[5]
```

```

generlinear.test.recall <- generlinear.test.ConfMat$byClass[6]
generlinear.test.F1score <- generlinear.test.ConfMat$byClass[7]
generlinear.test.ConfMat <- generlinear.test.ConfMat$table
generlinear.test.ROCcurv <- roc(testset$DEATH_EVENT, generlinear.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

generlinear.test.AUCsc <- auc(testset$DEATH_EVENT, generlinear.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

generaddit.model <- gam(DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) + s(se
generaddit.model

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Estimated degrees of freedom:
## 2.83 1.00 3.50 3.07 total = 11.4
##
## REML score: 124.8923
summary(generaddit.model)

##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEATH_EVENT ~ s(age) + s(creatinine_phosphokinase) + s(ejection_fraction) +
##      s(serum_creatinine)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.29870    0.02532   11.79  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##             edf Ref.df      F p-value
## s(age)          2.831   3.552 6.561 0.000115 ***
## s(creatinine_phosphokinase) 1.000   1.001 1.844 0.175952
## s(ejection_fraction)      3.499   4.320 9.410 2.57e-07 ***
## s(serum_creatinine)       3.069   3.802 4.310 0.002711 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) = 0.296 Deviance explained = 32.8%
## -REML = 124.89 Scale est. = 0.14814 n = 231

```

```

generaddit.test <- predict(generaddit.model, testset, type = 'response')
for (j in 1:length(generaddit.test)) {
  if(generaddit.test[j] < 0.5){
    generaddit.test[j] <- 0
  }
  if(generaddit.test[j] >= 0.5){
    generaddit.test[j] <- 1
  }
}
generaddit.test

##      2      3      4      15      29      55      60      64      65      79      86      87      91      96      114      116      121      130      131      137
##      0      1      1      0      0      0      1      0      0      0      0      0      0      0      0      0      0      0      0      0
## 156 159 160 165 168 178 200 208 218 220 221 231 250 263 273 284 285 291  11  13
##      1      1      0      0      1      0      0      0      1      0      1      0      0      0      0      0      0      0      0      0
##  17  19  20  23  30  31  48  83  88  92  94 122 127 139 152 163 180 185 213 225
##      0      0      0      0      1      1      0      1      0      0      1      0      1      0      0      0      0      0      0      0
## 227 242 253 262 265 266 270 299
##      0      0      0      0      0      0      0      0      0

generaddit.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - generaddit.test)^2)/nrow(testset))
generaddit.test.ConfMat <- confusionMatrix(factor(generaddit.test), factor(testset$DEATH_EVENT))
generaddit.test accur <- generaddit.test.ConfMat$overall[1]
generaddit.test precis <- generaddit.test.ConfMat$byClass[5]
generaddit.test recall <- generaddit.test.ConfMat$byClass[6]
generaddit.test F1score <- generaddit.test.ConfMat$byClass[7]
generaddit.test.ConfMat <- generaddit.test.ConfMat$table
generaddit.test.ROCcurv <- roc(testset$DEATH_EVENT, generaddit.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

generaddit.test.AUCsc <- auc(testset$DEATH_EVENT, generaddit.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.model <- lda(DEATH_EVENT ~ . -time, data = trainset)
lindiscr.model

## Call:
## lda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##      0      1
## 0.7012987 0.2987013
##
## Group means:
##      age      anaemia creatinine_phosphokinase      diabetes      ejection_fraction
## 0 58.63375 0.3950617          569.6481 0.4135802          39.54321
## 1 66.32851 0.4492754          680.5217 0.4202899          33.76812
## high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0      0.3333333 265940.7          1.205864      137.3827 0.6358025
## 1      0.4492754 265384.1          1.767826      135.7826 0.6231884
##      smoking
## 0 0.3209877

```

```

## 1 0.3188406
##
## Coefficients of linear discriminants:
##                               LD1
## age                          5.519907e-02
## anaemia                      1.705953e-01
## creatinine_phosphokinase     1.979781e-04
## diabetes                    2.471694e-01
## ejection_fraction           -4.930122e-02
## high_blood_pressure          4.469411e-01
## platelets                    1.273423e-07
## serum_creatinine             4.403657e-01
## serum_sodium                 -6.121693e-02
## sex                         -3.742431e-01
## smoking                      2.728574e-01

lindiscr.test <- predict(lindiscr.model, testset, type = 'response')
lindiscr.test <- as.numeric(lindiscr.test$class)-1
lindiscr.test

## [1] 0 1 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0
## [39] 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

lindiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - lindiscr.test)^2))/nrow(testset)
lindiscr.test.ConfMat <- confusionMatrix(factor(lindiscr.test), factor(testset$DEATH_EVENT))
lindiscr.test accur <- lindiscr.test.ConfMat$overall[1]
lindiscr.test.precis <- lindiscr.test.ConfMat$byClass[5]
lindiscr.test.recall <- lindiscr.test.ConfMat$byClass[6]
lindiscr.test.F1score <- lindiscr.test.ConfMat$byClass[7]
lindiscr.test.ConfMat <- lindiscr.test.ConfMat$table
lindiscr.test.ROCcurv <- roc(testset$DEATH_EVENT, lindiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

lindiscr.test.AUCsc <- auc(testset$DEATH_EVENT, lindiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.model <- qda(DEATH_EVENT ~ . -time, data = trainset)
quaddiscr.model

## Call:
## qda(DEATH_EVENT ~ . - time, data = trainset)
##
## Prior probabilities of groups:
##          0          1
## 0.7012987 0.2987013
##
## Group means:
##      age  anaemia creatinine_phosphokinase  diabetes ejection_fraction
## 0 58.63375 0.3950617          569.6481 0.4135802          39.54321
## 1 66.32851 0.4492754          680.5217 0.4202899          33.76812
##  high_blood_pressure platelets serum_creatinine serum_sodium      sex
## 0          0.3333333 265940.7          1.205864      137.3827 0.6358025
## 1          0.4492754 265384.1          1.767826      135.7826 0.6231884

```

```

##      smoking
## 0 0.3209877
## 1 0.3188406

quaddiscr.test <- predict(quaddiscr.model, testset, type = 'response')
quaddiscr.test <- as.numeric(quaddiscr.test$class)-1
quaddiscr.test

## [1] 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 0 0
## [39] 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

quaddiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - quaddiscr.test)^2)/nrow(testset))
quaddiscr.test.ConfMat <- confusionMatrix(factor(quaddiscr.test), factor(testset$DEATH_EVENT))
quaddiscr.test accur <- quaddiscr.test.ConfMat$overall[1]
quaddiscr.test.precis <- quaddiscr.test.ConfMat$byClass[5]
quaddiscr.test.recall <- quaddiscr.test.ConfMat$byClass[6]
quaddiscr.test.F1score <- quaddiscr.test.ConfMat$byClass[7]
quaddiscr.test.ConfMat <- quaddiscr.test.ConfMat$table
quaddiscr.test.ROCcurv <- roc(testset$DEATH_EVENT, quaddiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

quaddiscr.test.AUCsc <- auc(testset$DEATH_EVENT, quaddiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

mixeddiscr.model <- mda(DEATH_EVENT ~ . -time, data = trainset)
mixeddiscr.model

## Call:
## mda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 5
##
## Percent Between-Group Variance Explained:
##      v1      v2      v3      v4      v5
## 74.91 89.52 97.31 99.53 100.00
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.25108 ( N = 231 )
##
## Deviance: 229.971

mixeddiscr.test <- predict(mixeddiscr.model, testset, type = 'class')
mixeddiscr.test <- as.numeric(mixeddiscr.test)-1
mixeddiscr.test

## [1] 1 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0 1 0 0 0 0
## [39] 1 0 0 1 0 0 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

mixeddiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - mixeddiscr.test)^2)/nrow(testset))
mixeddiscr.test.ConfMat <- confusionMatrix(factor(mixeddiscr.test), factor(testset$DEATH_EVENT))
mixeddiscr.test accur <- mixeddiscr.test.ConfMat$overall[1]
mixeddiscr.test.precis <- mixeddiscr.test.ConfMat$byClass[5]
mixeddiscr.test.recall <- mixeddiscr.test.ConfMat$byClass[6]

```



```

mixeddiscr.test.F1score <- mixeddiscr.test.ConfMat$byClass[7]
mixeddiscr.test.ConfMat <- mixeddiscr.test.ConfMat$table
mixeddiscr.test.ROCcurv <- roc(testset$DEATH_EVENT, mixeddiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
mixeddiscr.test.AUCsc <- auc(testset$DEATH_EVENT, mixeddiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
flexdiscr.model <- fda(DEATH_EVENT ~ . -time, data = trainset)
flexdiscr.model

## Call:
## fda(formula = DEATH_EVENT ~ . - time, data = trainset)
##
## Dimension: 1
##
## Percent Between-Group Variance Explained:
##   v1
## 100
##
## Degrees of Freedom (per dimension): 12
##
## Training Misclassification Error: 0.24242 ( N = 231 )
flexdiscr.test <- predict(flexdiscr.model, testset, type = 'class')
flexdiscr.test <- as.numeric(flexdiscr.test)-1
flexdiscr.test

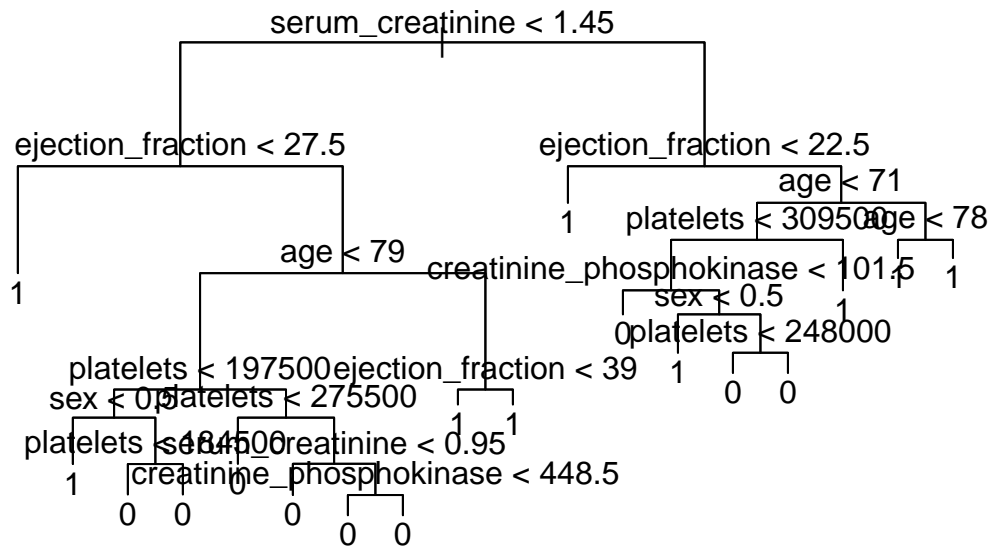
## [1] 0 1 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0
## [39] 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

flexdiscr.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - flexdiscr.test)^2))/nrow(testset)
flexdiscr.test.ConfMat <- confusionMatrix(factor(flexdiscr.test), factor(testset$DEATH_EVENT))
flexdiscr.test accur <- flexdiscr.test.ConfMat$overall[1]
flexdiscr.test.precis <- flexdiscr.test.ConfMat$byClass[5]
flexdiscr.test.recall <- flexdiscr.test.ConfMat$byClass[6]
flexdiscr.test.F1score <- flexdiscr.test.ConfMat$byClass[7]
flexdiscr.test.ConfMat <- flexdiscr.test.ConfMat$table
flexdiscr.test.ROCcurv <- roc(testset$DEATH_EVENT, flexdiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
flexdiscr.test.AUCsc <- auc(testset$DEATH_EVENT, flexdiscr.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
decisiontree.model <- tree(as.factor(DEATH_EVENT) ~ . -time, data = trainset)
plot(decisiontree.model)
text(decisiontree.model, pretty = 0)

```



```

decisiontree.test <- predict(decisiontree.model, testset, type = 'class')
decisiontree.test <- as.numeric(decisiontree.test)-1
decisiontree.test

## [1] 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 0 1 1 0 1 0 0 0 0
## [39] 1 0 1 1 1 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1

decisiontree.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - decisiontree.test)^2))/nrow(testset)
decisiontree.test.ConfMat <- confusionMatrix(factor(decisiontree.test), factor(testset$DEATH_EVENT))
decisiontree.test accur <- decisiontree.test.ConfMat$overall[1]
decisiontree.test.precis <- decisiontree.test.ConfMat$byClass[5]
decisiontree.test.recall <- decisiontree.test.ConfMat$byClass[6]
decisiontree.test.F1score <- decisiontree.test.ConfMat$byClass[7]
decisiontree.test.ConfMat <- decisiontree.test.ConfMat$table
decisiontree.test.ROCcurv <- roc(testset$DEATH_EVENT, decisiontree.test)

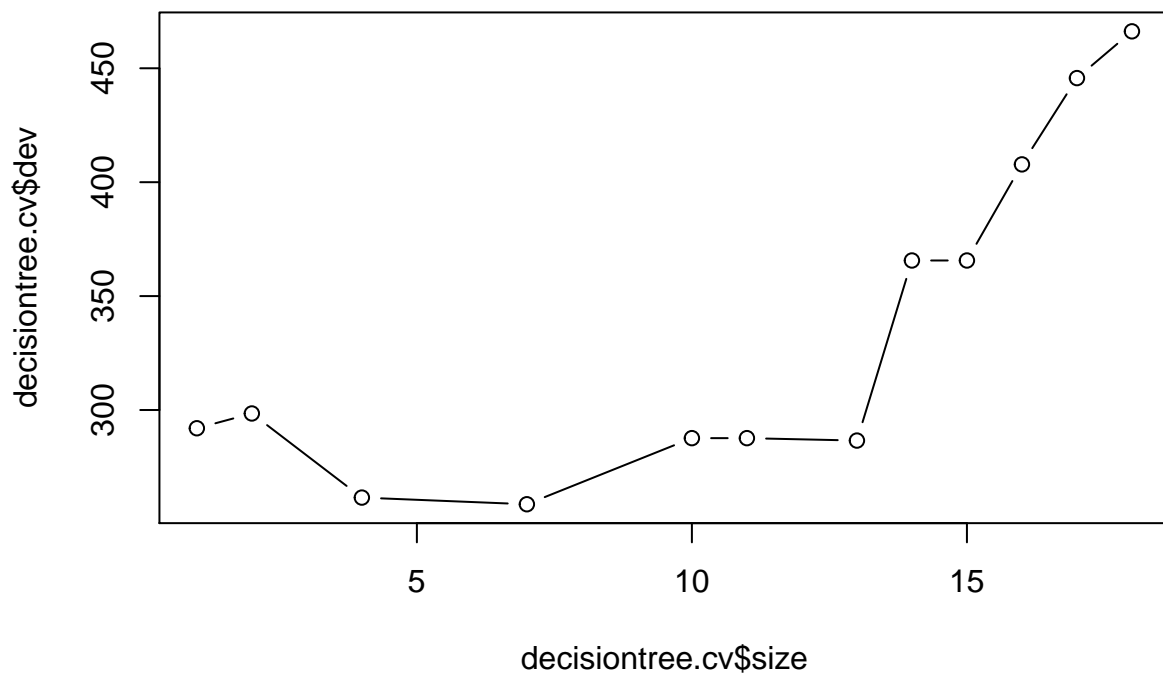
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

decisiontree.test.AUCsc <- auc(testset$DEATH_EVENT, decisiontree.test)

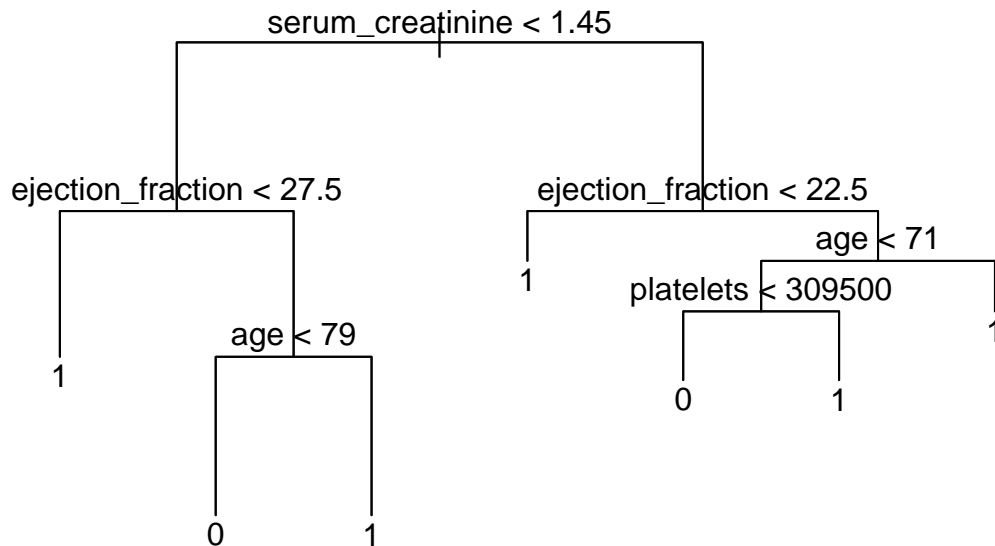
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

set.seed(5)
decisiontree.cv <- cv.tree(decisiontree.model)
plot(decisiontree.cv$size, decisiontree.cv$dev, type = 'b')

```



```
prunedectree.model <- prune.tree(decisiontree.model, best = decisiontree.cv$size[which.min(decisiontree  
plot(prunedectree.model)  
text(prunedectree.model, pretty = 0)
```



```

prunedectree.test <- predict(prunedectree.model, testset, type = 'class')
prunedectree.test <- as.numeric(prunedectree.test)-1
prunedectree.test

## [1] 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0
## [39] 1 0 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1

prunedectree.test.RMSE <- sqrt(sum((testset$DEATH_EVENT - prunedectree.test)^2)/nrow(testset))
prunedectree.test.ConfMat <- confusionMatrix(factor(prunedectree.test), factor(testset$DEATH_EVENT))
prunedectree.test accur <- prunedectree.test.ConfMat$overall[1]
prunedectree.test.precis <- prunedectree.test.ConfMat$byClass[5]
prunedectree.test.recall <- prunedectree.test.ConfMat$byClass[6]
prunedectree.test.Fscore <- prunedectree.test.ConfMat$byClass[7]
prunedectree.test.ConfMat <- prunedectree.test.ConfMat$table
prunedectree.test.ROCcurv <- roc(testset$DEATH_EVENT, prunedectree.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases

prunedectree.test.AUCsc <- auc(testset$DEATH_EVENT, prunedectree.test)

## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
##Let us check results for our models once again

simplelinear.test.RMSE

## [1] 0.06063391

```

```
simplelinear.test accur
```

```
## Accuracy  
## 0.75
```

```
simplelinear.test precis
```

```
## Precision  
## 0.7222222
```

```
simplelinear.test recall
```

```
## Recall  
## 0.9512195
```

```
simplelinear.test Fscore
```

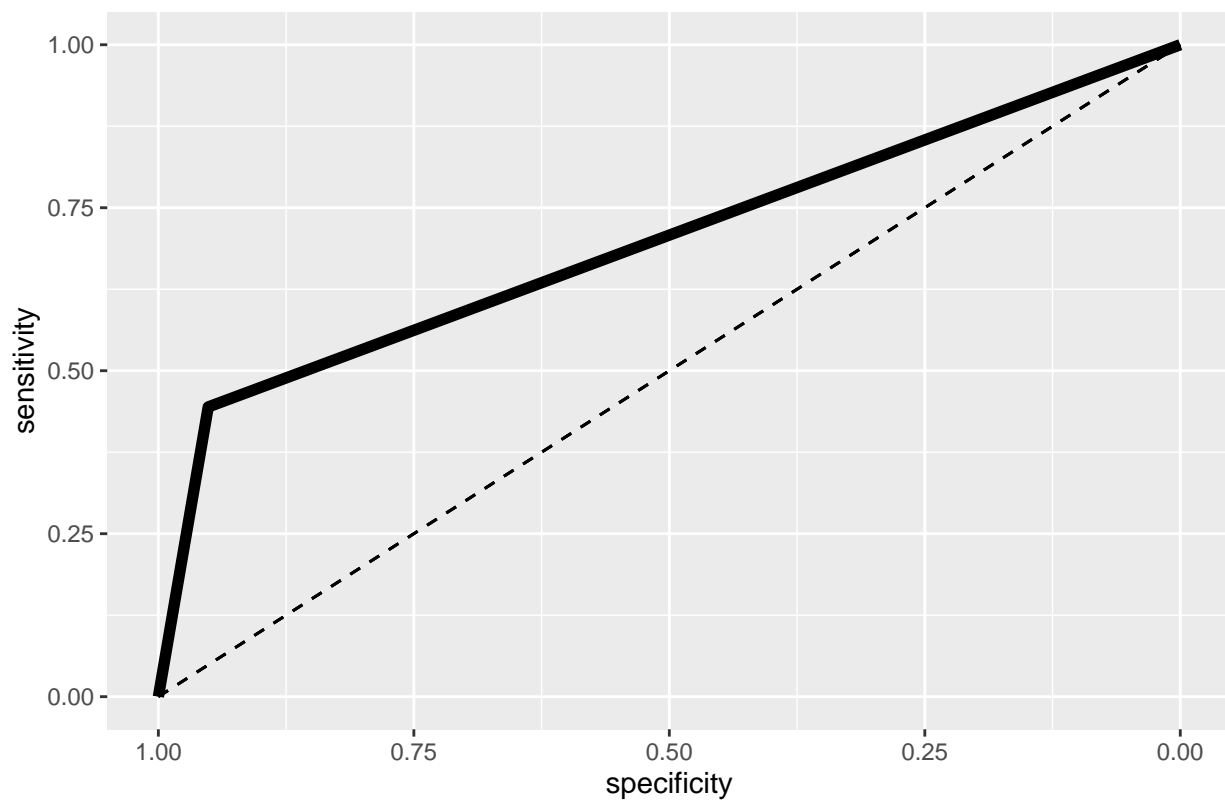
```
## F1  
## 0.8210526
```

```
simplelinear.test AUCsc
```

```
## Area under the curve: 0.6978
```

```
ggroc(simplelinear.test.ROCcurv, linetype = 1, size = 2)+  
  ggtitle('ROC curve from test set for simplelinear.model')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)
```

ROC curve from test set for simplelinear.model



```
simplelinear.test ConfMat
```

```

##           Reference
## Prediction  0  1
##           0 39 15
##           1  2 12
revisedlinear.test.RMSE

## [1] 0.06063391
revisedlinear.test accur

## Accuracy
##      0.75
revisedlinear.test.precis

## Precision
## 0.7142857
revisedlinear.test.recall

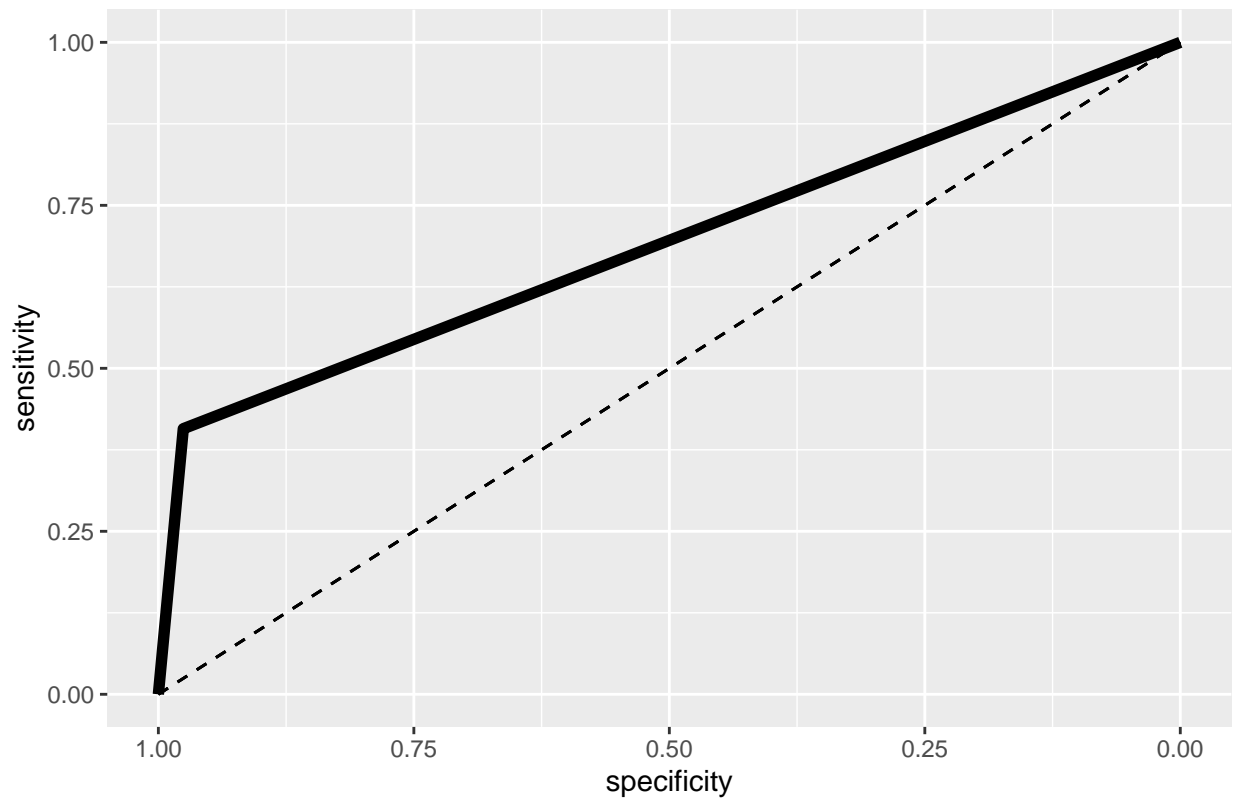
##      Recall
## 0.9756098
revisedlinear.test.F1score

##           F1
## 0.8247423
revisedlinear.test.AUCsc

## Area under the curve: 0.6915
ggroc(revisedlinear.test.ROCcurv, linetype = 1, size = 2)+
  ggtitle('ROC curve from test set for revisedlinear.model')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)

```

ROC curve from test set for revisedlinear.model



```
revisedlinear.test.ConfMat
```

```
##           Reference
## Prediction  0  1
##           0 40 16
##           1  1 11
```

```
generlinear.test.RMSE
```

```
## [1] 0.05882353
```

```
generlinear.test accur
```

```
## Accuracy
## 0.7647059
```

```
generlinear.test.precis
```

```
## Precision
## 0.7358491
```

```
generlinear.test.recall
```

```
## Recall
## 0.9512195
```

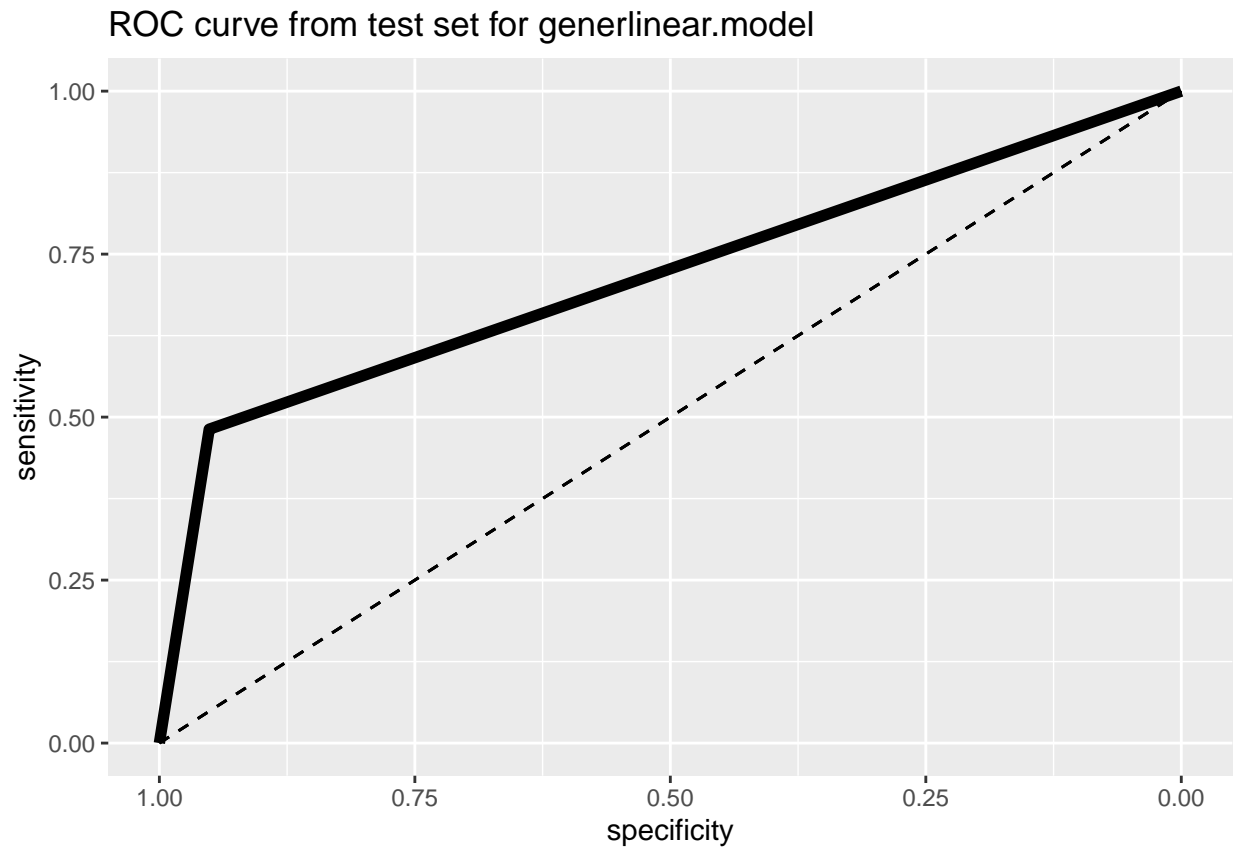
```
generlinear.test.F1score
```

```
## F1
## 0.8297872
```

```
generlinear.test.AUCsc
```

```
## Area under the curve: 0.7164
```

```
ggroc(generlinear.test.ROCcurv, linetype = 1, size = 2)+  
  ggtitle('ROC curve from test set for generlinear.model')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)
```



```
generlinear.test.ConfMat
```

```
##           Reference  
## Prediction  0  1  
##           0 39 14  
##           1  2 13
```

```
generaddit.test.RMSE
```

```
## [1] 0.06239177
```

```
generaddit.test accur
```

```
## Accuracy  
## 0.7352941
```

```
generaddit.test.precis
```

```
## Precision  
## 0.7090909
```



```
generaddit.test.recall
```

```
##      Recall  
## 0.9512195
```

```
generaddit.test.F1score
```

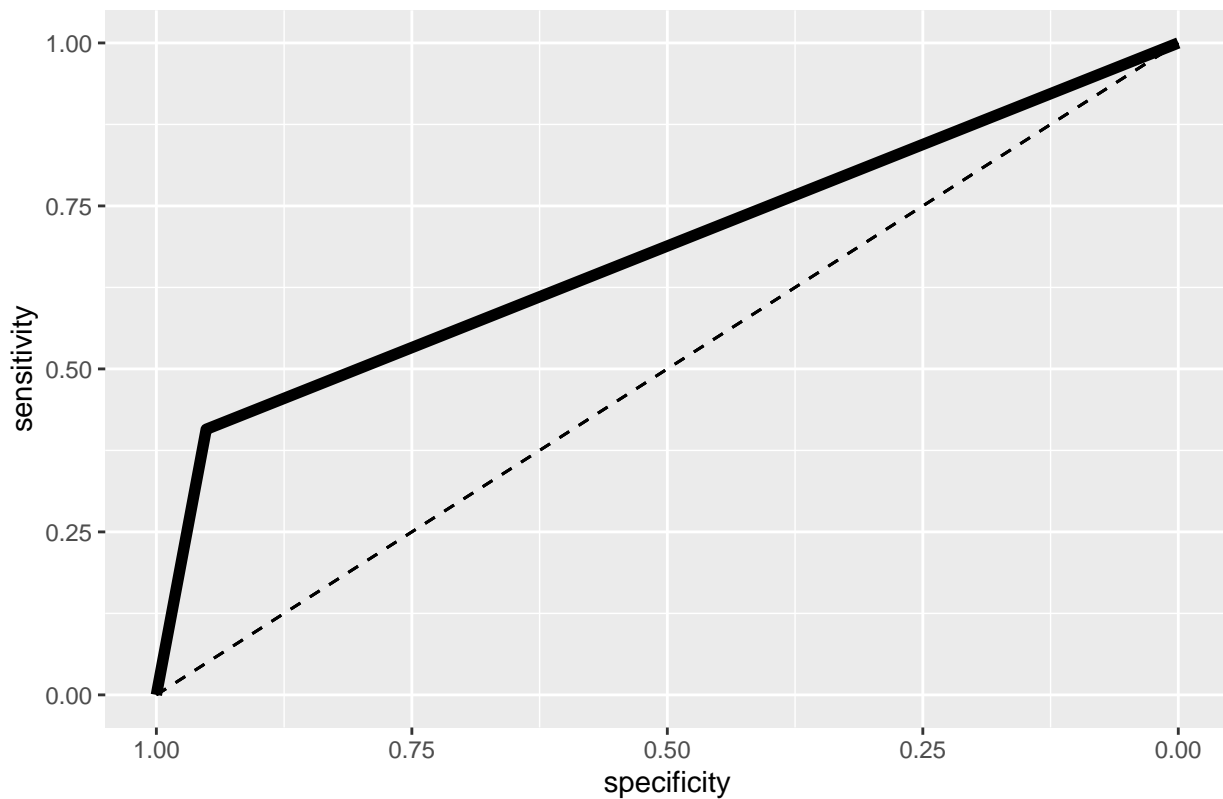
```
##      F1  
## 0.8125
```

```
generaddit.test.AUCsc
```

```
## Area under the curve: 0.6793
```

```
ggroc(generaddit.test.ROCcurv, linetype = 1, size = 2)+  
  ggtitle('ROC curve from test set for generaddit.model')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)
```

ROC curve from test set for generaddit.model



```
generaddit.test.ConfMat
```

```
##      Reference  
## Prediction 0 1  
##      0 39 16  
##      1  2 11
```

```
lindiscr.test.RMSE
```

```
## [1] 0.05882353
```

```
lindiscr.test accur
```

```
## Accuracy  
## 0.7647059
```

```
lindiscr.test precis
```

```
## Precision  
## 0.7358491
```

```
lindiscr.test recall
```

```
## Recall  
## 0.9512195
```

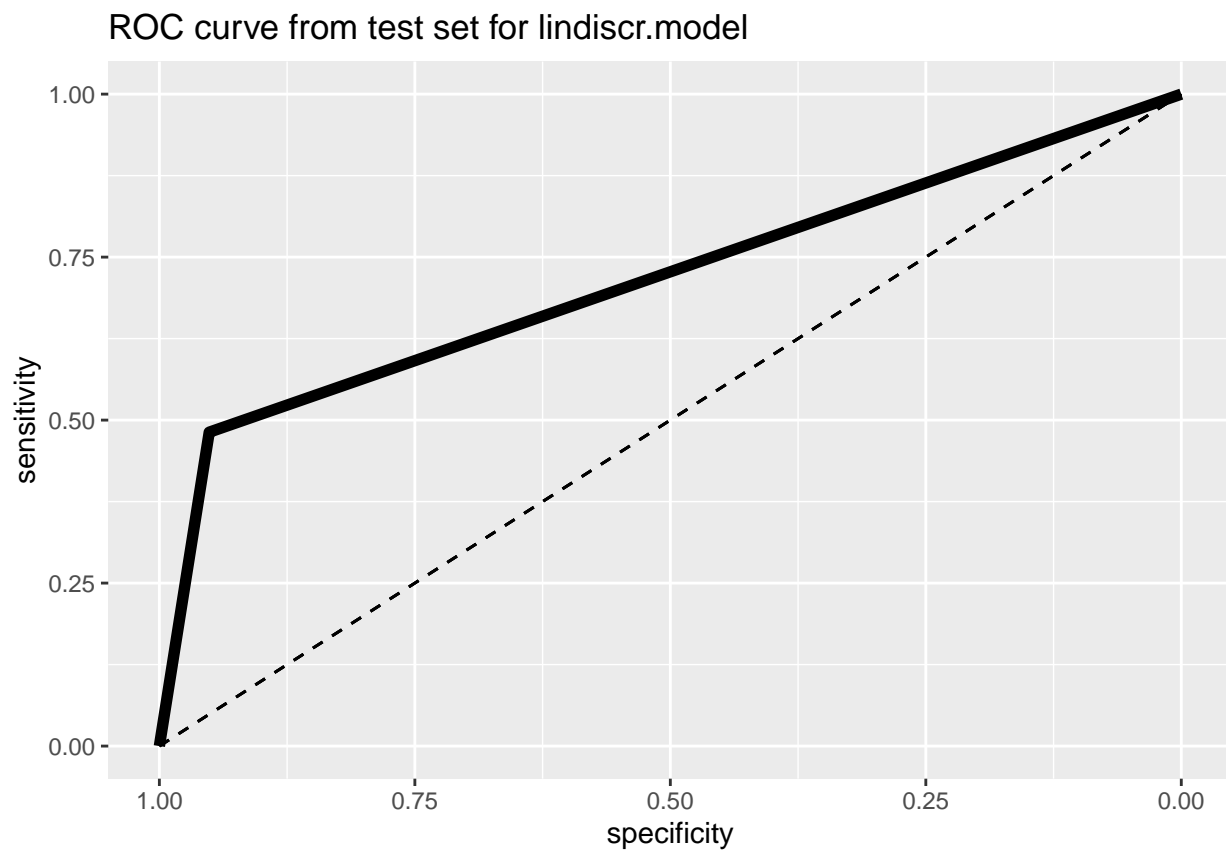
```
lindiscr.test F1score
```

```
## F1  
## 0.8297872
```

```
lindiscr.test AUCsc
```

```
## Area under the curve: 0.7164
```

```
ggroc(lindiscr.test.ROCcurv, linetype = 1, size = 2)+  
  ggtitle('ROC curve from test set for lindiscr.model')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)
```



```
lindiscr.test.ConfMat
```

```
## Reference
```

```

## Prediction  0  1
##           0 39 14
##           1  2 13
quaddiscr.test.RMSE

## [1] 0.06576671
quaddiscr.test.accur

## Accuracy
## 0.7058824
quaddiscr.test.precis

## Precision
## 0.6981132
quaddiscr.test.recall

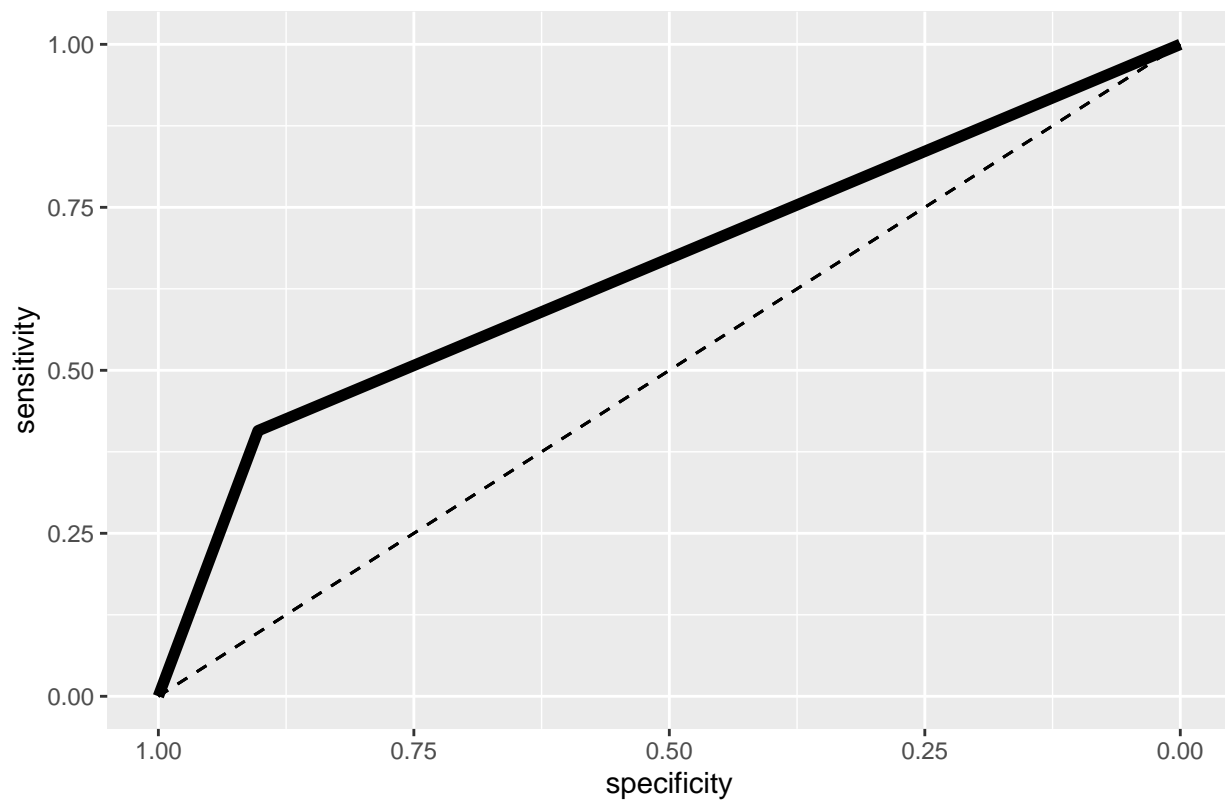
## Recall
## 0.902439
quaddiscr.test.F1score

## F1
## 0.787234
quaddiscr.test.AUCsc

## Area under the curve: 0.6549
ggroc(quaddiscr.test.ROCcurv, linetype = 1, size = 2)+
  ggtitle('ROC curve from test set for quaddiscr.model')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)

```

ROC curve from test set for quaddiscr.model



```
quaddiscr.test.ConfMat
```

```
##           Reference
## Prediction  0  1
##           0 37 16
##           1  4 11
```

```
mixeddiscr.test.RMSE
```

```
## [1] 0.05882353
```

```
mixeddiscr.test accur
```

```
## Accuracy
## 0.7647059
```

```
mixeddiscr.test.precis
```

```
## Precision
## 0.745098
```

```
mixeddiscr.test.recall
```

```
## Recall
## 0.9268293
```

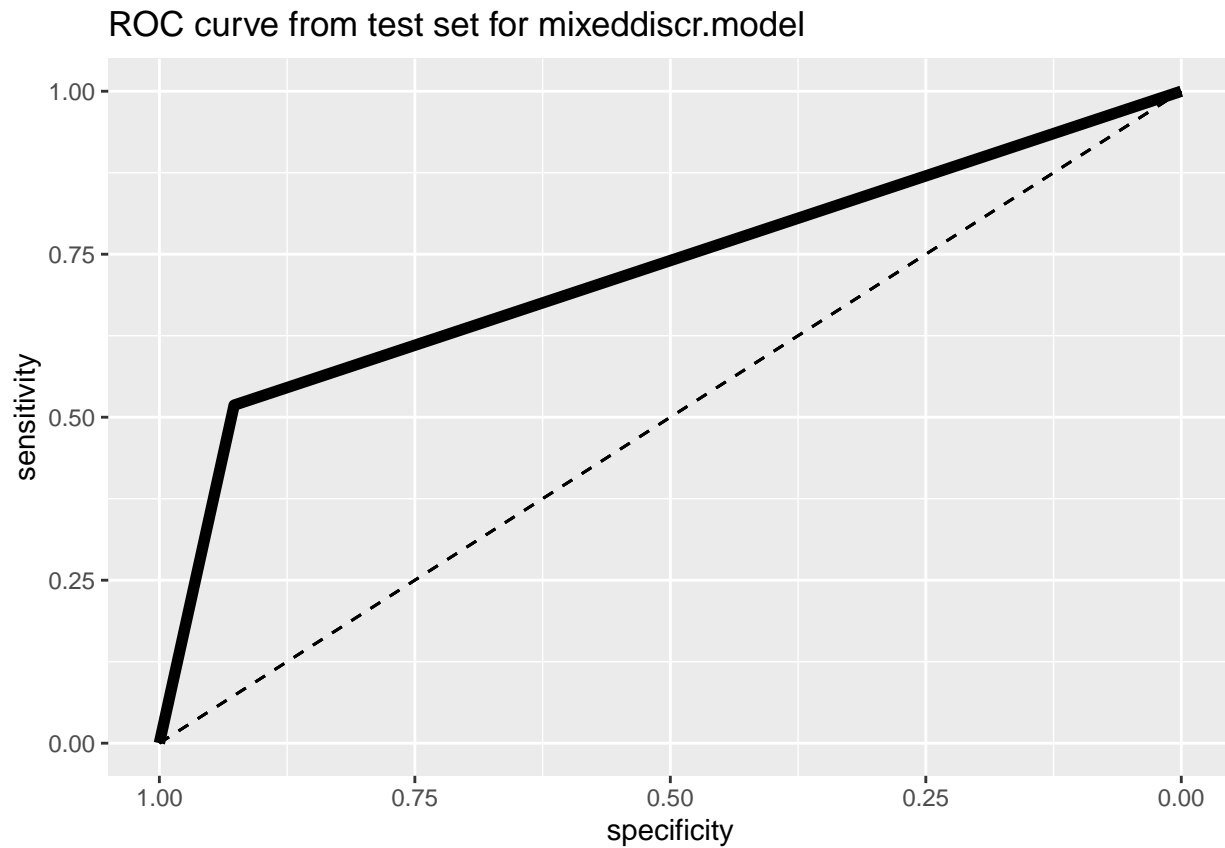
```
mixeddiscr.test.F1score
```

```
## F1
## 0.826087
```

```
mixeddiscr.test.AUCsc
```

```
## Area under the curve: 0.7227
```

```
ggroc(mixeddiscr.test.ROCcurv, linetype = 1, size = 2)+  
  ggtitle('ROC curve from test set for mixeddiscr.model')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)
```



```
mixeddiscr.test.ConfMat
```

```
##           Reference  
## Prediction  0  1  
##           0 38 13  
##           1  3 14
```

```
flexdiscr.test.RMSE
```

```
## [1] 0.05882353
```

```
flexdiscr.test accur
```

```
## Accuracy  
## 0.7647059
```

```
flexdiscr.test.precis
```

```
## Precision  
## 0.7358491
```

```
flexdiscr.test.recall
```

```
## Recall
## 0.9512195
```

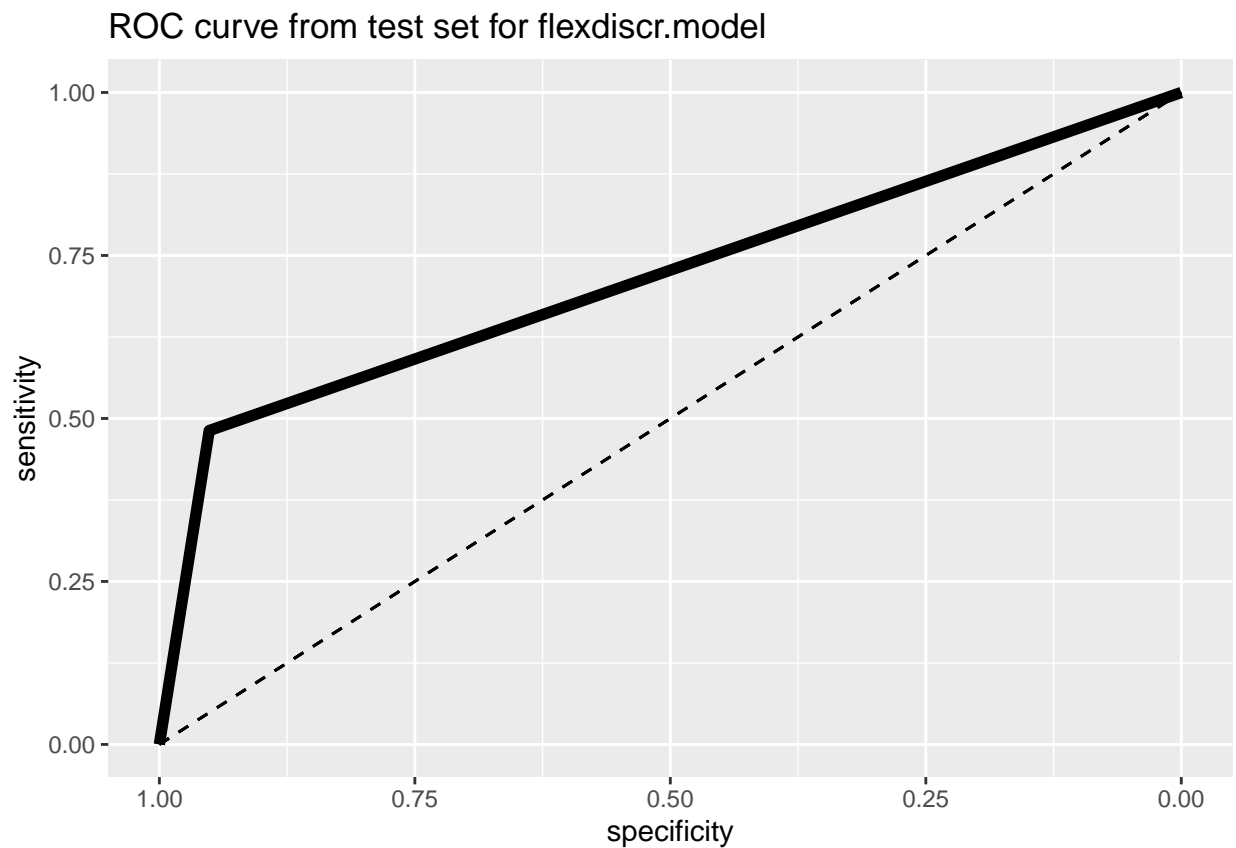
```
flexdiscr.test.F1score
```

```
## F1
## 0.8297872
```

```
flexdiscr.test.AUCsc
```

```
## Area under the curve: 0.7164
```

```
ggroc(flexdiscr.test.ROCcurv, linetype = 1, size = 2)+
  ggtitle('ROC curve from test set for flexdiscr.model')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)
```



```
flexdiscr.test.ConfMat
```

```
##           Reference
## Prediction  0  1
##           0 39 14
##           1  2 13
```

```
decisiontree.test.RMSE
```

```
## [1] 0.06410146
```

```
decisiontree.test accur
```

```
## Accuracy
## 0.7205882
```

```
decisiontree.test.precis
```

```
## Precision  
## 0.7619048
```

```
decisiontree.test.recall
```

```
## Recall  
## 0.7804878
```

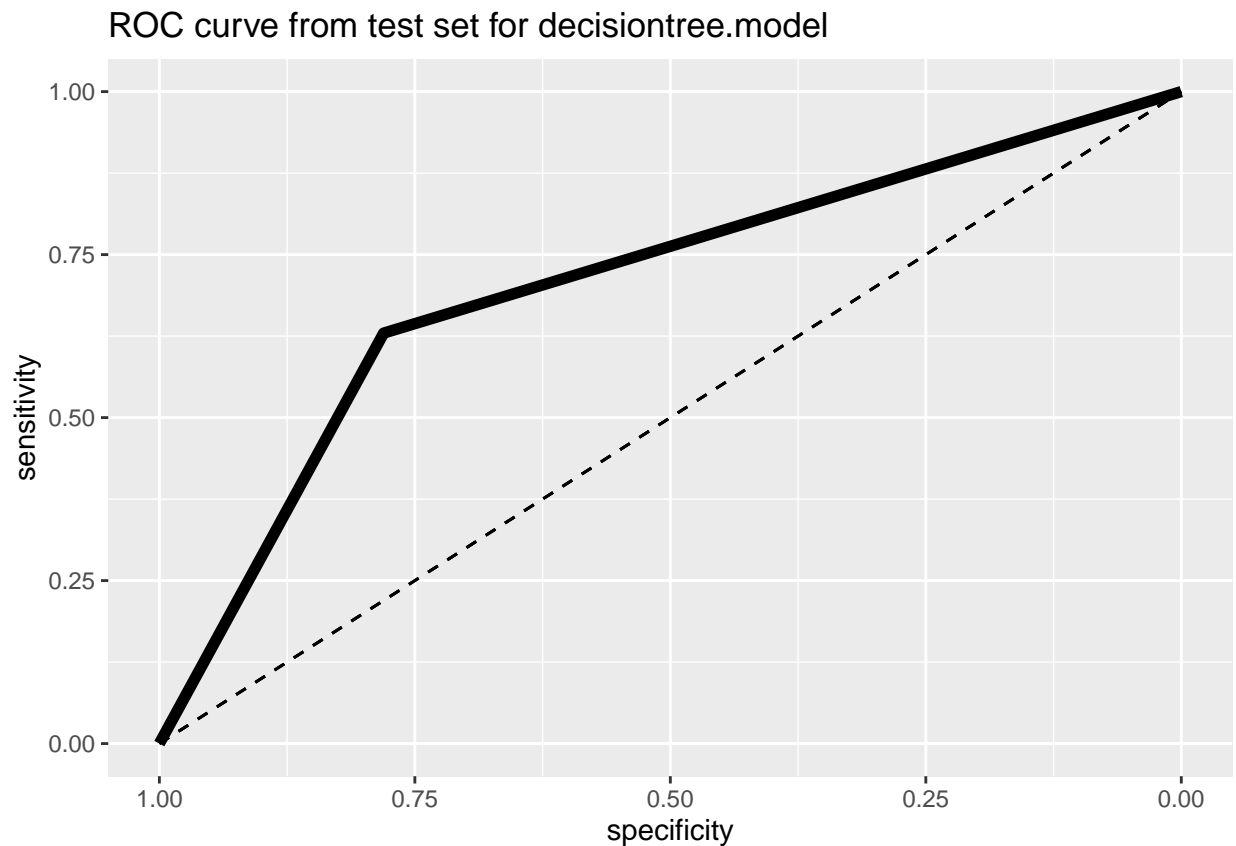
```
decisiontree.test.F1score
```

```
## F1  
## 0.7710843
```

```
decisiontree.test.AUCsc
```

```
## Area under the curve: 0.7051
```

```
ggroc(decisiontree.test.ROCcurv, linetype = 1, size = 2)+  
  ggtitle('ROC curve from test set for decisiontree.model')+  
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)
```



```
decisiontree.test.ConfMat
```

```
##           Reference  
## Prediction  0  1  
##           0 32 10  
##           1  9 17
```

```

prunedectree.test.RMSE

## [1] 0.06410146
prunedectree.test accur

## Accuracy
## 0.7205882
prunedectree.test.precis

## Precision
## 0.72
prunedectree.test.recall

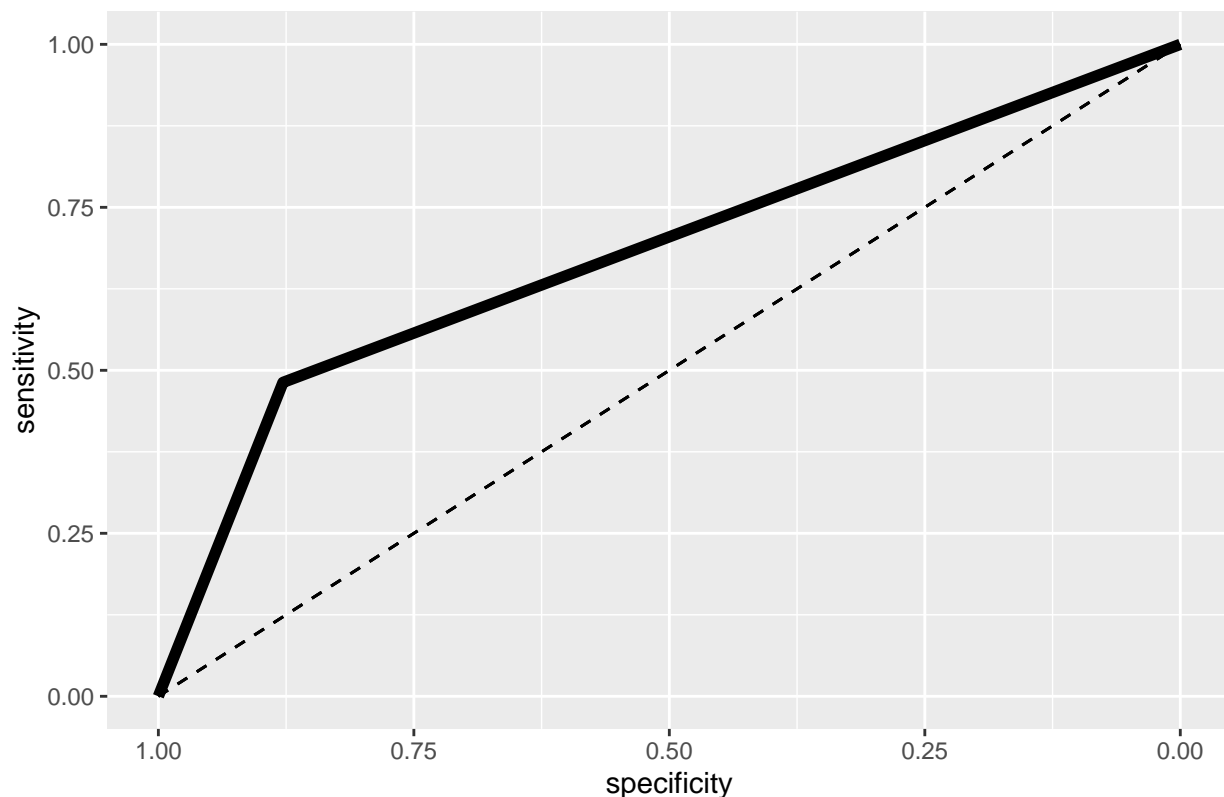
## Recall
## 0.8780488
prunedectree.test.F1score

## F1
## 0.7912088
prunedectree.test.AUCsc

## Area under the curve: 0.6798
ggroc(prunedectree.test.ROCcurv, linetype = 1, size = 2)+
  ggtitle('ROC curve from test set for prunedectree.model')+
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), colour = 'black', linetype = 2)

```


ROC curve from test set for prunedectree.model



```
prunedectree.test.ConfMat
```

```
##           Reference
## Prediction  0  1
##           0 36 14
##           1  5 13
```

##Conclusions from Tet Set

##As discussed above in the Cross Validation conclusions, it seems that there are no truly appropriate models here to classify the event of a death from heart failure. We can however note that all models have a F1 (harmonic) mean near 80% which is decent, but not amazing; and AUC scores 0.7 which gives the same conclusions. Since these methods rely on the same mathematical machine learning principals used for our probability regression, we can most likely assume that while our predictive models are satisfactorily appropriate, they may also not be perfect probability predictors

##Ideally, this dataset would be enlarged in the future with more data being sporadically added (as this might help improve the models). It may also be interesting to attempt any kind of clustering analysis to see if there significant differences between clusters (whether it be predictor values or actual death event outcomes) before using our machine learning models (this may lead to better improved, but would also depend on cluster sizes; where a cluster being too small outweighs model improvement)