

Response Features

W. Grossmann

16 Mai 2017

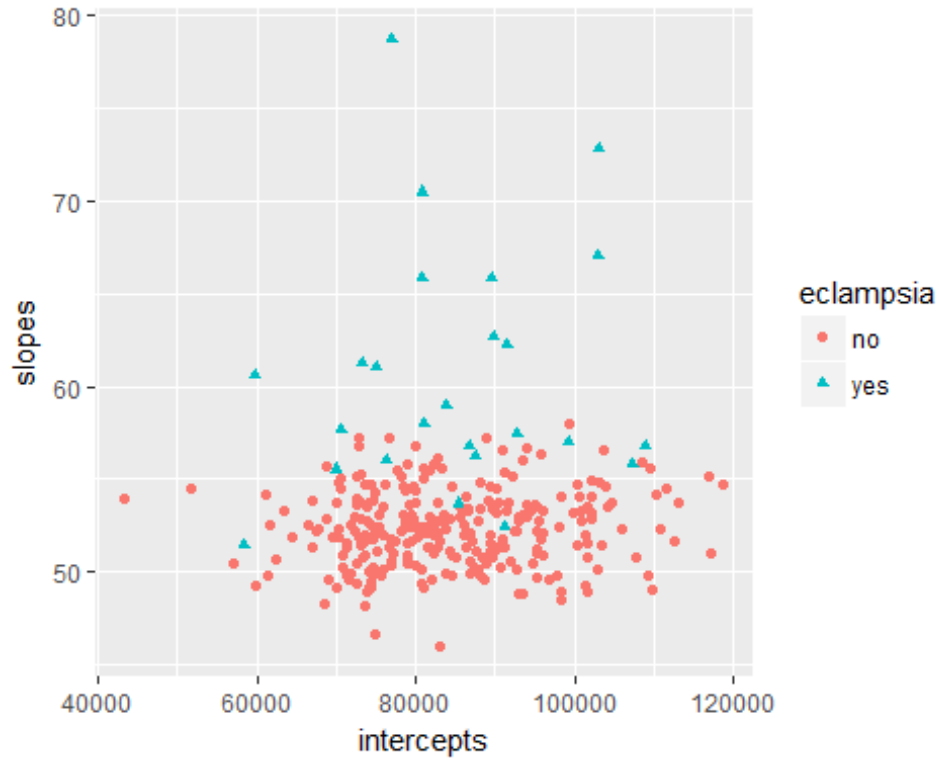
```
#####  
# Response Feature Analysis  
#####  
  
#-----  
# Libraries  
#-----  
library(lattice)  
library(ggplot2)  
library(Matrix)  
library(changepoint)  
  
## Loading required package: zoo  
  
##  
## Attaching package: 'zoo'  
  
## The following objects are masked from 'package:base':  
##  
##   as.Date, as.Date.numeric  
  
## Successfully loaded changepoint package version 2.2.2  
## NOTE: Predefined penalty values changed in version 2.2. Previous penalty  
## values with a postfix 1 i.e. SIC1 are now without i.e. SIC and previous penal  
## ties without a postfix i.e. SIC are now with a postfix 0 i.e. SIC0. See NEWS  
## and help files for further details.  
  
library(gridExtra)  
#-----  
# Data  
#-----  
pre.eclampsia <- read.csv("longitudinal2.csv", sep=";")  
head(pre.eclampsia)  
  
##   preEclampsia patient id day weightCheck bloodPressureCheck  
## 1         yes      8 9  0           5                3  
## 2         yes      8 9  1           4                2  
## 3         yes      8 9  2           3                1  
## 4         yes      8 9  3           2                4  
## 5         yes      8 9  4           1                3  
## 6         yes      8 9  5           7                2  
##   proteinuriaCheck weight weightChangeLast7Days systolicBloodPressure  
## 1                12  83964                34                119  
## 2                11  83977                47                119
```

```
## 3          10 84058          128          119
## 4           9 84110          180          120
## 5           8 84147          217          120
## 6           7 84204          274          123
##   diastolicBloodPressure proteinuria goToHospital
## 1                   71           98           yes
## 2                   72           93           yes
## 3                   72           94           yes
## 4                   71           95           yes
## 5                   69           92           yes
## 6                   69           89           yes
```

```
attach(pre.eclampsia)
ecl<-pre.eclampsia$preEclampsia[match(1:300,pre.eclampsia$id)]
#####
# Regression Coefficients for Weight
#####
slopes.weight<-numeric(max(id))
intercepts.weight<-numeric(max(id))
for(i in 1:max(id)){
  lmod.weight<-lm(weight~day,subset=(id==i))
  intercepts.weight[i]<-coef(lmod.weight)[1]
  slopes.weight[i]<-coef(lmod.weight)[2]
}
#-----
# Scatterplot for Groups
#-----

features1<-cbind(ecl, intercepts.weight,slopes.weight)
features1<-as.data.frame(features1)
eclampsia<-as.factor(ecl)

figure1<-ggplot(features1, aes(x=intercepts.weight, y = slopes.weight,
                             shape=eclampsia, colour=eclampsia)) + geom_point()
figure1 + labs(x = "intercepts", y = "slopes")
```

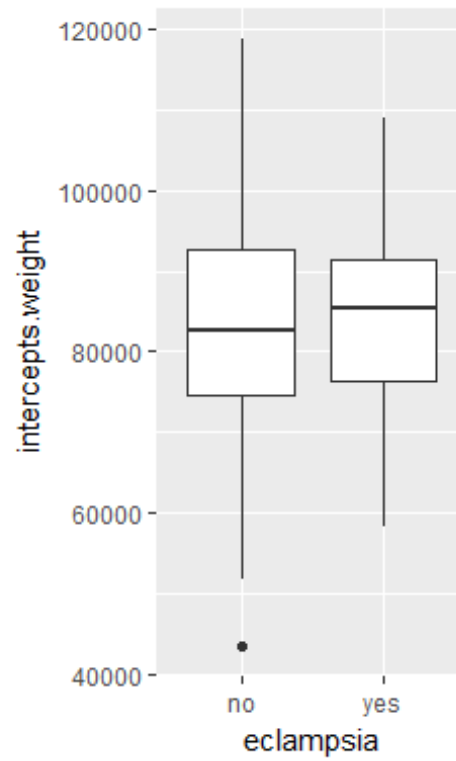
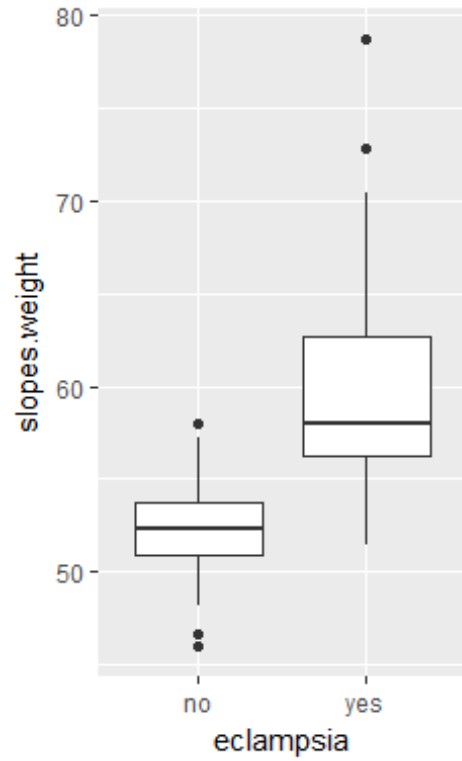


```

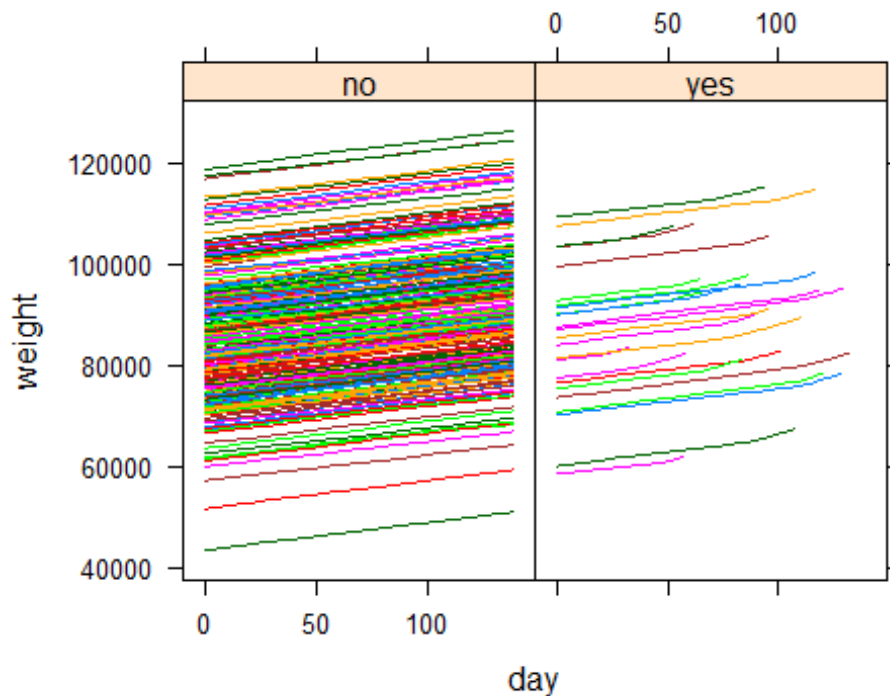
#-----
# Boxplots for groups
#-----

plot1<-ggplot(features1, aes(x=eclampsia, y = slopes.weight)) + geom_boxplot(
)
plot2<-ggplot(features1, aes(x=eclampsia, y = intercepts.weight)) + geom_boxp
lot()
grid.arrange(plot1, plot2, ncol=2)

```



```
#-----
# Weight in the groups
#-----
xyplot(weight~day|preEclampsia,pre.eclampsia,type="l",groups=patient)
```



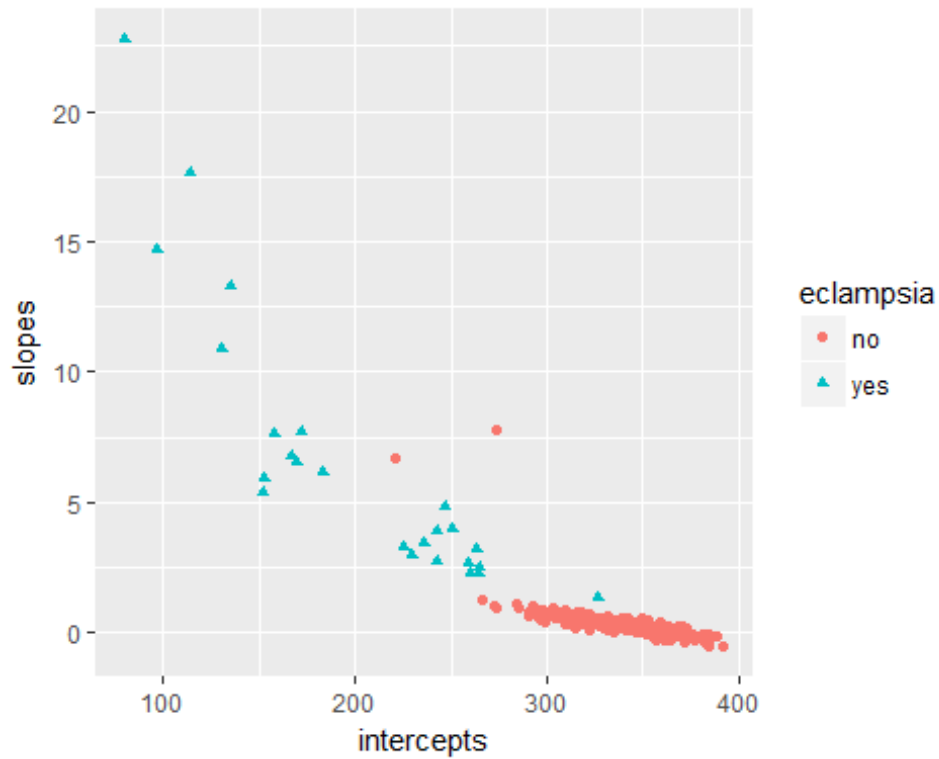
```

#-----
# Regression Coefficients for Weight gain Last 7 days
#-----
slopes.weightgain<-numeric(max(id))
intercepts.weightgain<-numeric(max(id))
for(i in 1:max(id)){
  lmod.weightg<-lm(weightChangeLast7Days~day,subset=(id==i))
  intercepts.weightgain[i]<-coef(lmod.weightg)[1]
  slopes.weightgain[i]<-coef(lmod.weightg)[2]
}
#-----
# Scatterplot for Groups
#-----

features12<-cbind(ecl, intercepts.weightgain,slopes.weightgain)
features12<-as.data.frame(features12)
eclampsia<-as.factor(ecl)

figure1<-ggplot(features12, aes(x=intercepts.weightgain, y = slopes.weightgain,
                               shape=eclampsia, colour=eclampsia)) + geom_point()
figure1 + labs(x = "intercepts", y = "slopes")

```

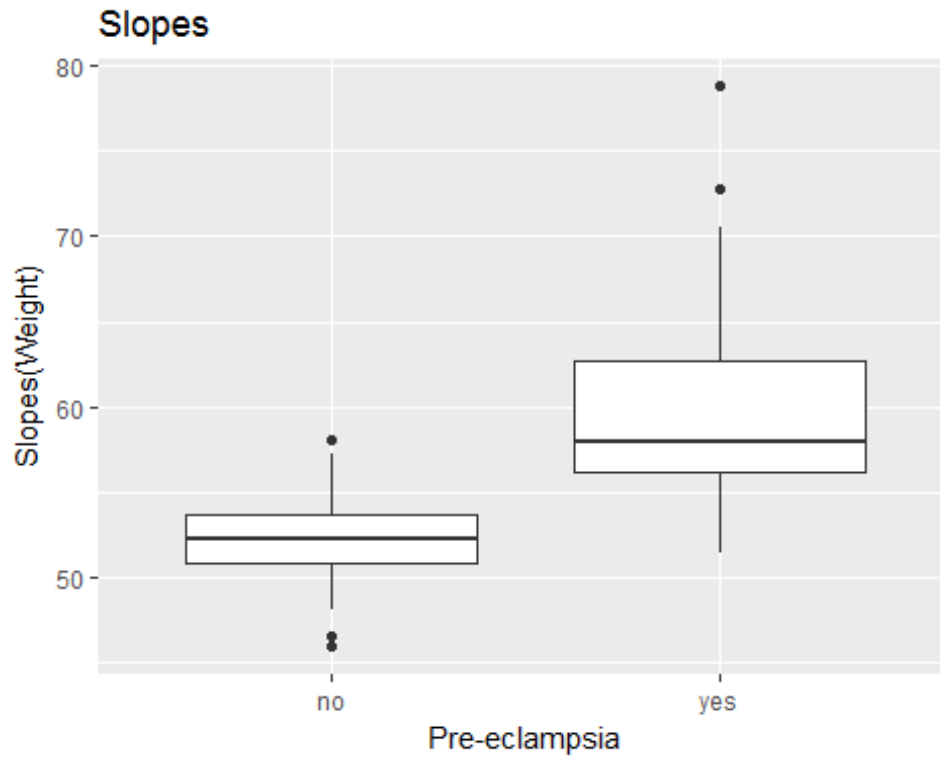


```

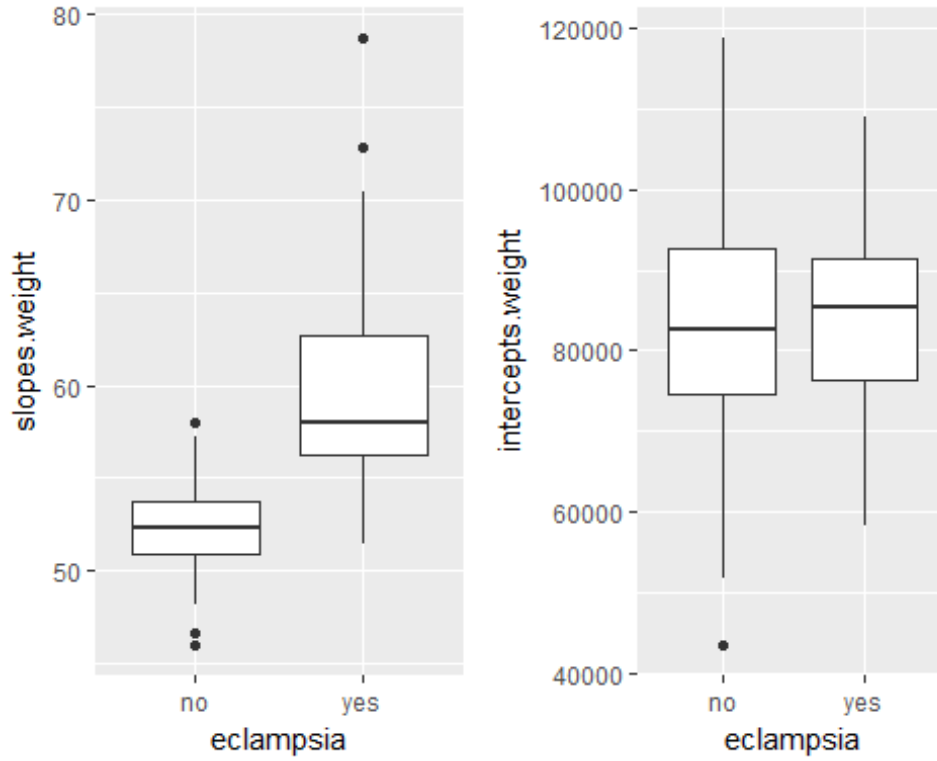
#-----
# Boxplots for groups
#-----

plot1<-ggplot(features1, aes(x=eclampsia, y = slopes.weight)) + geom_boxplot(
)
plot2<-ggplot(features1, aes(x=eclampsia, y = intercepts.weight)) + geom_boxp
lot()
plot3<- plot1+ labs(x = "Pre-eclampsia", y = "Slopes(Weight)")+
  ggtitle("Slopes")
plot3

```



```
grid.arrange(plot1, plot2, ncol=2)
```



```
#####  
# Response features Based on Quantiles for Proteinuria
```

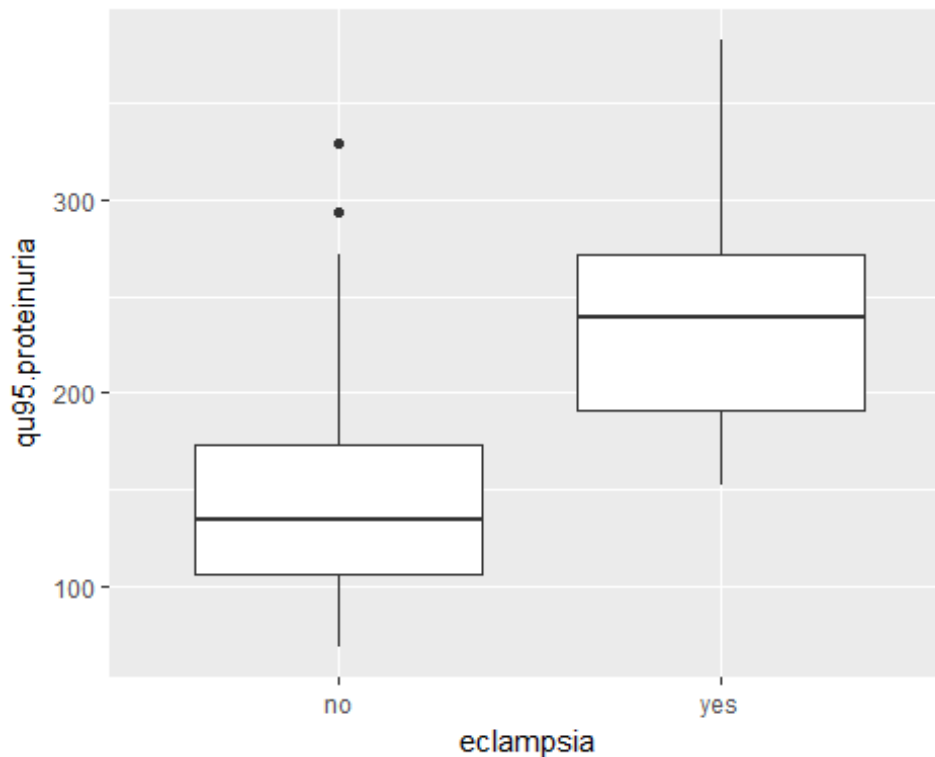
```

#=====
qu95.proteinuria<-numeric(max(id))

for(i in 1:max(id)){
  x<-subset(pre.eclampsia$proteinuria, subset =(id==i))
  qu95.proteinuria[i]<-quantile(x,probs = 0.95)
}

features2<-cbind(ecl, qu95.proteinuria)
features2<-as.data.frame(features2)
eclampsia<-as.factor(ecl)
ggplot(features2, aes(x=eclampsia, y = qu95.proteinuria)) + geom_boxplot()

```



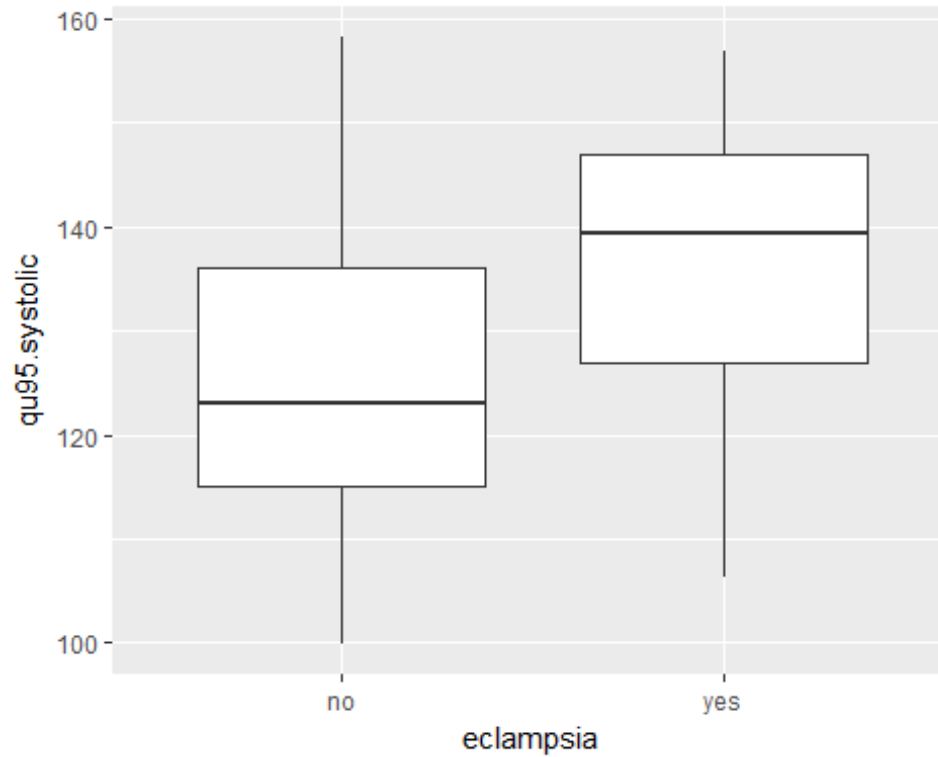
```

#=====
# Response features Based on Quantiles for SystolicBlood Pressure
#=====
qu95.systolic<-numeric(max(id))

for(i in 1:max(id)){
  x<-subset(pre.eclampsia$systolicBloodPressure, subset =(id==i))
  qu95.systolic[i]<-quantile(x,probs = 0.95)
}

features4<-cbind(ecl, qu95.systolic)
features4<-as.data.frame(features4)
eclampsia<-as.factor(ecl)
ggplot(features4, aes(x=eclampsia, y = qu95.systolic)) + geom_boxplot()

```

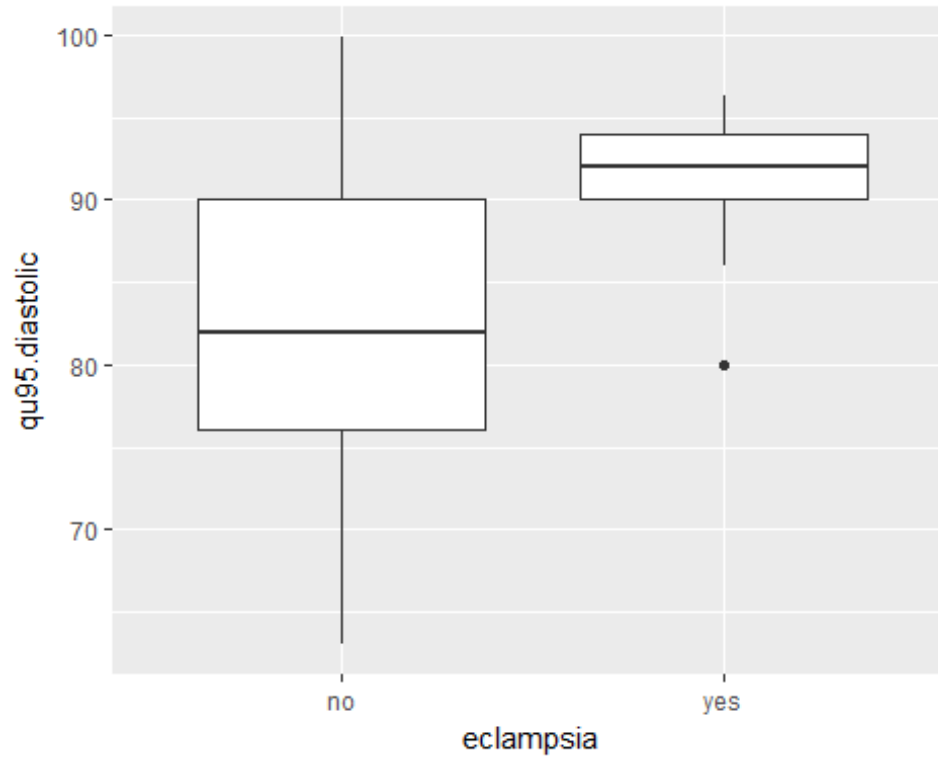
```

#####
# Response features Based on Quantiles for Diastolic Blood Pressure
#####
qu95.diastolic<-numeric(max(id))

for(i in 1:max(id)){
  x<-subset(pre.eclampsia$diastolicBloodPressure, subset =(id==i))
  qu95.diastolic[i]<-quantile(x,probs = 0.95)
}

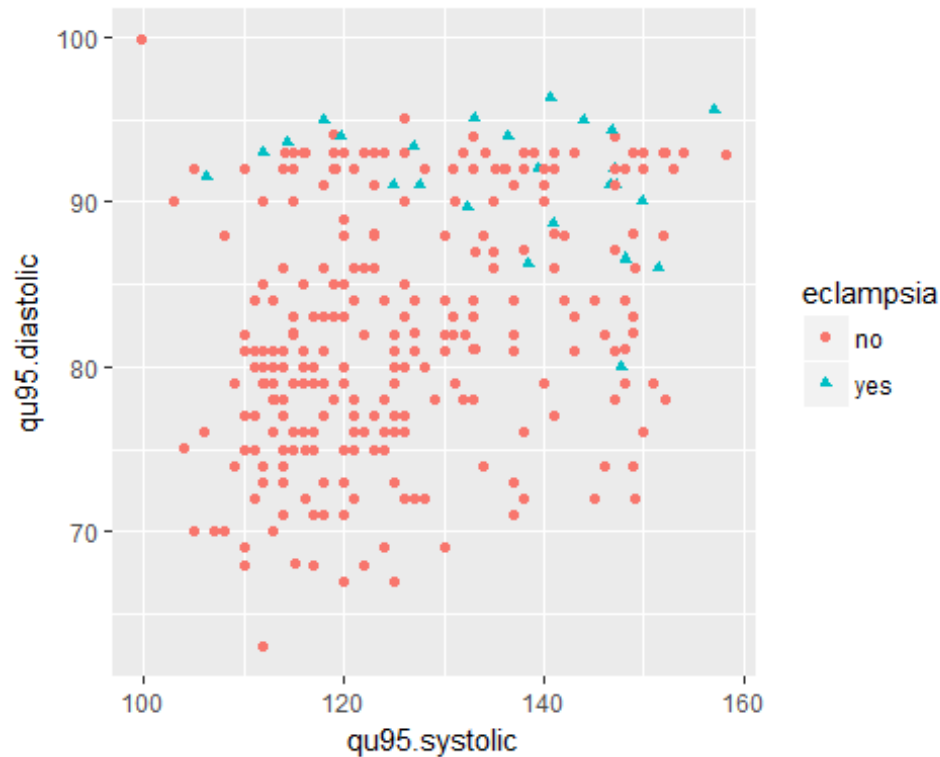
features5<-cbind(ecl, qu95.diastolic)
features5<-as.data.frame(features5)
eclampsia<-as.factor(ecl)
ggplot(features5, aes(x=eclampsia, y = qu95.diastolic)) + geom_boxplot()

```



```
#-----
#Scatterplot q95.systolic and qu95.diastolic
#-----
features6<-cbind(ecl, qu95.systolic,qu95.diastolic)
features6<-as.data.frame(features6)
eclampsia<-as.factor(ecl)

figure9<-ggplot(features6, aes(x=qu95.systolic, y = qu95.diastolic,
                               shape=eclampsia, colour=eclampsia)) + geom_poi
nt()
figure9 + labs(x = "qu95.systolic", y = "qu95.diastolic")
```



```

=====
# Changepoint Analysis for Proteinuric Weight
=====

cp.resprot<-numeric(max(pre.eclampsia$id))
cp.resweight<-numeric(max(pre.eclampsia$id))
cp.proteinuria<-numeric(max(pre.eclampsia$id))
cp.weight<-numeric(max(pre.eclampsia$id))
for(i in 1:max(pre.eclampsia$id)){
  case<-subset(pre.eclampsia, subset =(id==i))
  modp<-lm(case$proteinuria~case$day)
  resp<-modp$residuals
  cpresprot<-cpt.mean(resp,penalty="SIC",pen.value=0,method="AMOC",Q=5,test.s
tat="Normal")
  prot<-cpt.mean(case$proteinuria,penalty="SIC",pen.value=0,method="AMOC",Q=5
,test.stat="Normal")
  modw<-lm(case$weight~case$day)
  resw<-modw$residuals
  cpresweight<-cpt.mean(resp,penalty="SIC",pen.value=0,method="AMOC",Q=5,test
.stat="Normal")
  wgt<-cpt.mean(case$weight,penalty="SIC",pen.value=0,method="AMOC",Q=5,test.
stat="Normal")

  cp.proteinuria[i]<-cpts(prot)/length(case$proteinuria)
  cp.resprot[i]<-cpts(cpresprot)/length(case$proteinuria)
  cp.weight[i]<-cpts(wgt)/length(case$weight)
  cp.resweight[i]<-cpts(cpresweight)/length(case$weight)
}

```

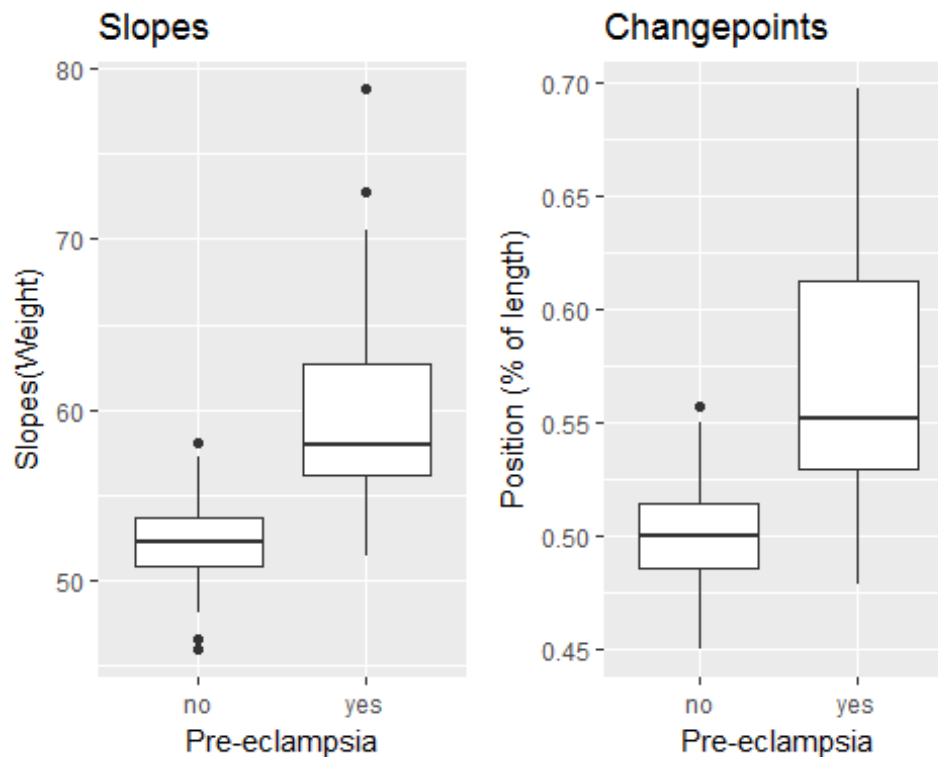
```

}
features3<-cbind(ecl, cp.proteinuria, cp.weight, cp.resprot,cp.resweight)
features3<-as.data.frame(features3)
eclampsia<-as.factor(ecl)

fig1<-ggplot(features3, aes(x=eclampsia, y = cp.weight)) + geom_boxplot()
Fig1<-fig1 + xlab("Pre-eclampsia") + ylab("Position (% of length)")
plot4<- Fig1 + ggtitle("Changepoints")

grid.arrange(plot3,plot4, ncol = 2)

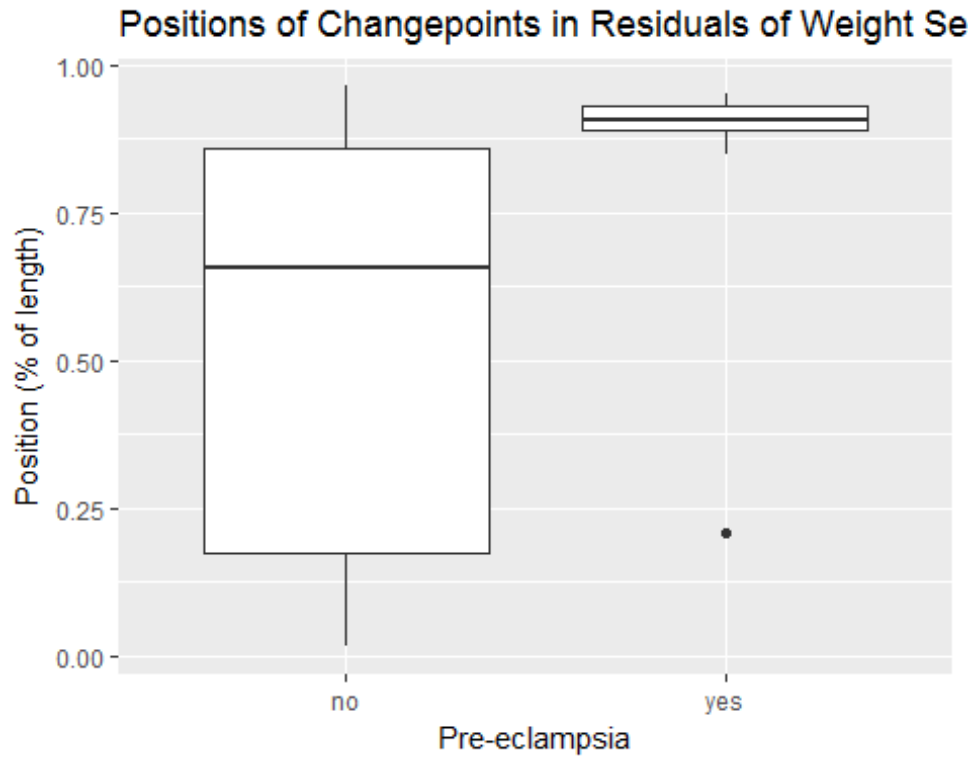
```



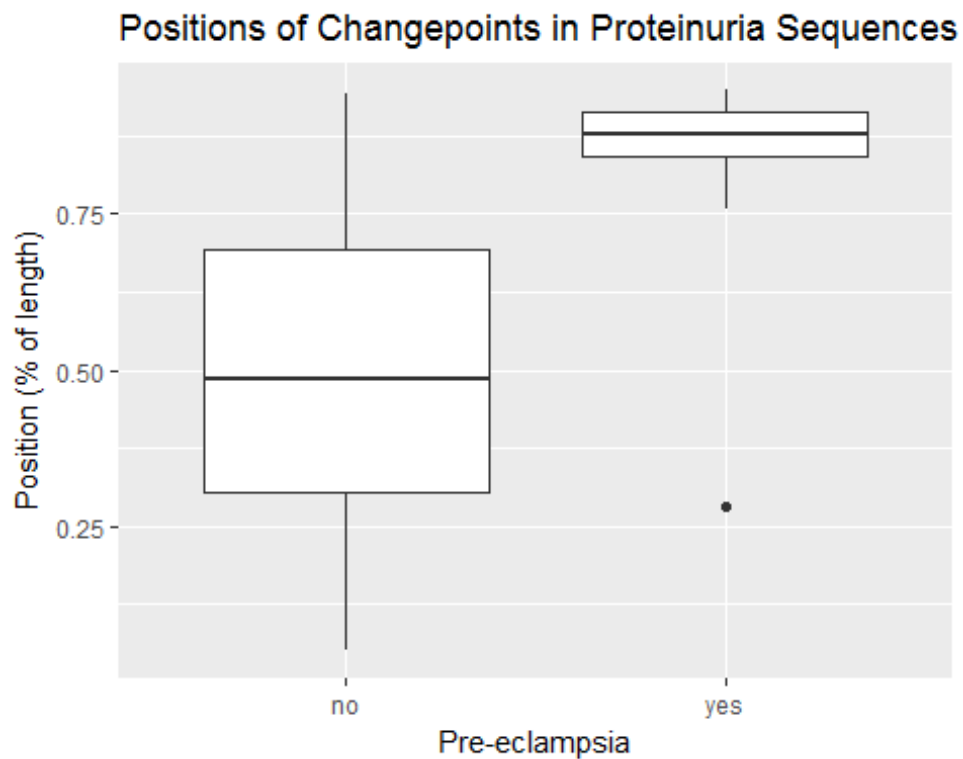
```

fig2<-ggplot(features3, aes(x=eclampsia, y = cp.resweight)) + geom_boxplot()
Fig2<-fig2 + xlab("Pre-eclampsia") + ylab("Position (% of length)")
Fig2 + ggtitle("Positions of Changepoints in Residuals of Weight Sequences")

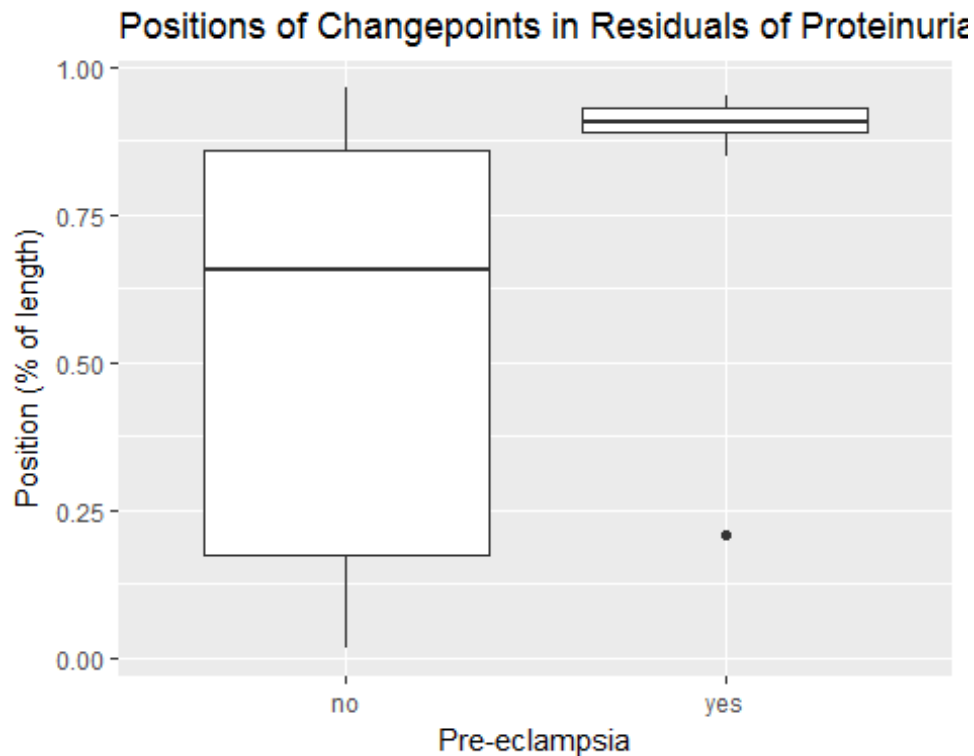
```



```
fig3<-ggplot(features3, aes(x=eclampsia, y = cp.proteinuria)) + geom_boxplot(
)
Fig3<-fig3 + xlab("Pre-eclampsia") + ylab("Position (% of length)")
Fig3 + ggtitle("Positions of Changepoints in Proteinuria Sequences")
```



```
fig4<-ggplot(features3, aes(x=eclampsia, y = cp.resprot)) + geom_boxplot()
Fig4<-fig4 + xlab("Pre-eclampsia") + ylab("Position (% of length)")
Fig4 + ggtitle("Positions of Changepoints in Residuals of Proteinuria")
```



```
#####
# Changepoint Analysis for weight change Last week
#####

cp.res.weightgain<-numeric(max(pre.eclampsia$id))
cp.weightgain<-numeric(max(pre.eclampsia$id))
for(i in 1:max(pre.eclampsia$id)){
  case<-subset(pre.eclampsia, subset =(id==i))
  modwg<-lm(case$weightChangeLast7Days~case$day)
  reswg<-modwg$residuals
  cpres.weightgain<-cpt.mean(reswg,penalty="SIC",pen.value=0,method="AMOC",Q=
5,test.stat="Normal")
  cpwgain<-cpt.mean(case$weight,penalty="SIC",pen.value=0,method="AMOC",Q=5,t
est.stat="Normal")

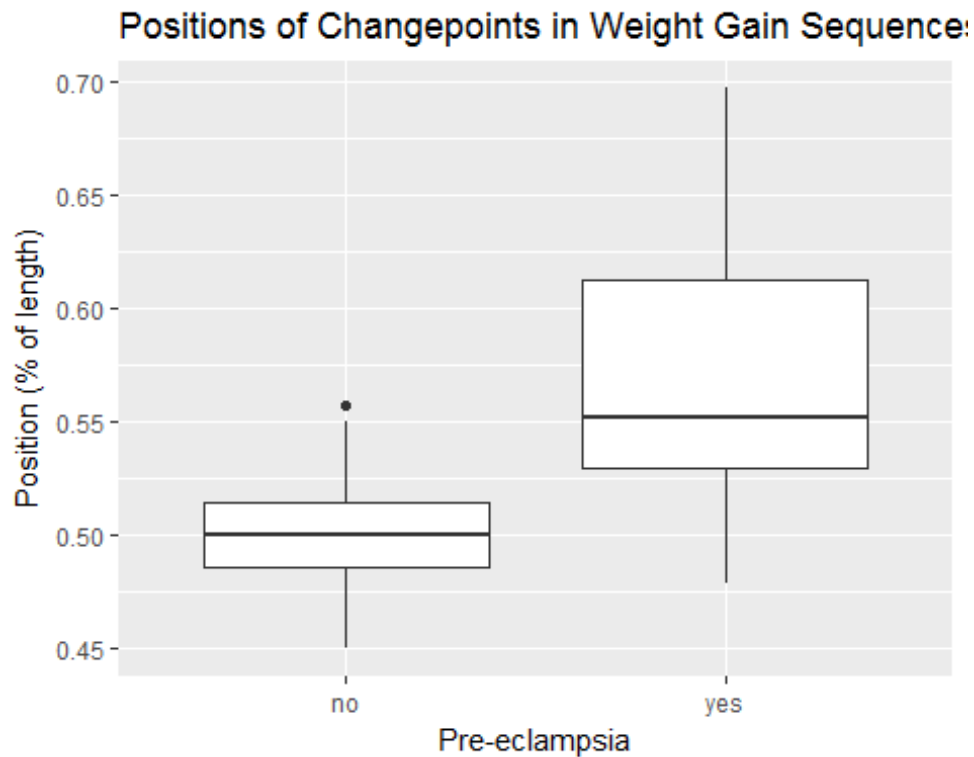
  cp.weightgain[i]<-cpts(cpwgain)/length(case$weight)
  cp.res.weightgain[i]<-cpts(cpres.weightgain)/length(case$weight)
}

features10<-cbind(ecl, cp.weightgain, cp.res.weightgain)
features10<-as.data.frame(features10)
eclampsia<-as.factor(ecl)
```

```

fig10<-ggplot(features10, aes(x=eclampsia, y = cp.weightgain)) + geom_boxplot
()
Fig10<-fig10 + xlab("Pre-eclampsia") + ylab("Position (% of length)")
Fig10 + ggtitle("Positions of Changepoints in Weight Gain Sequences")

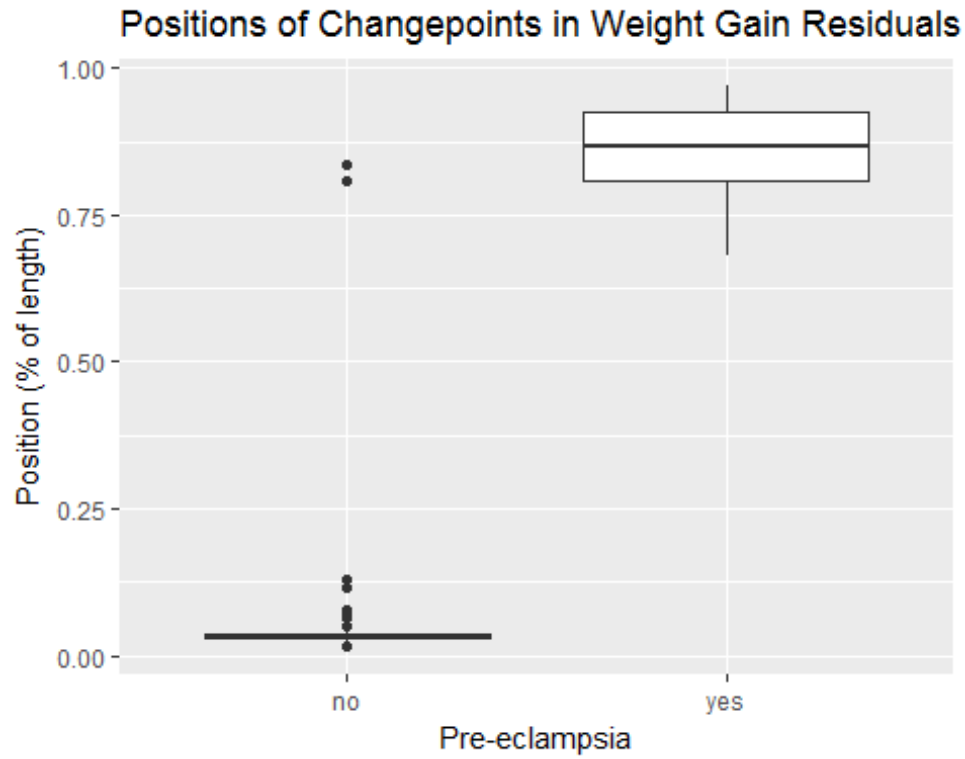
```



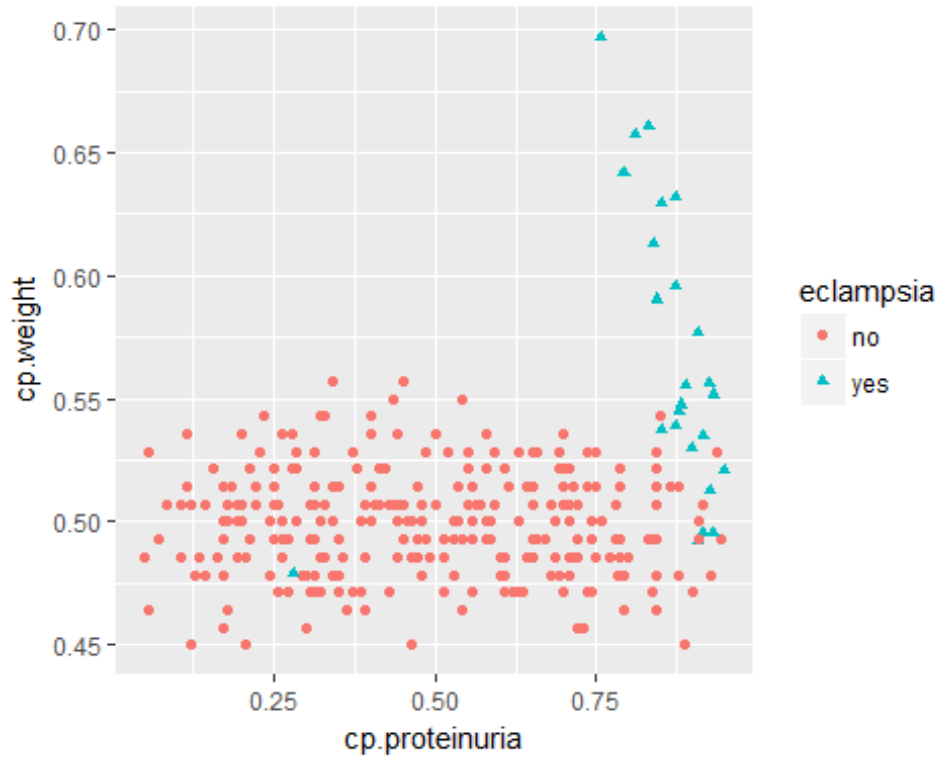
```

fig11<-ggplot(features10, aes(x=eclampsia, y = cp.res.weightgain)) + geom_box
plot()
Fig11<-fig11 + xlab("Pre-eclampsia") + ylab("Position (% of length)")
Fig11 + ggtitle("Positions of Changepoints in Weight Gain Residuals")

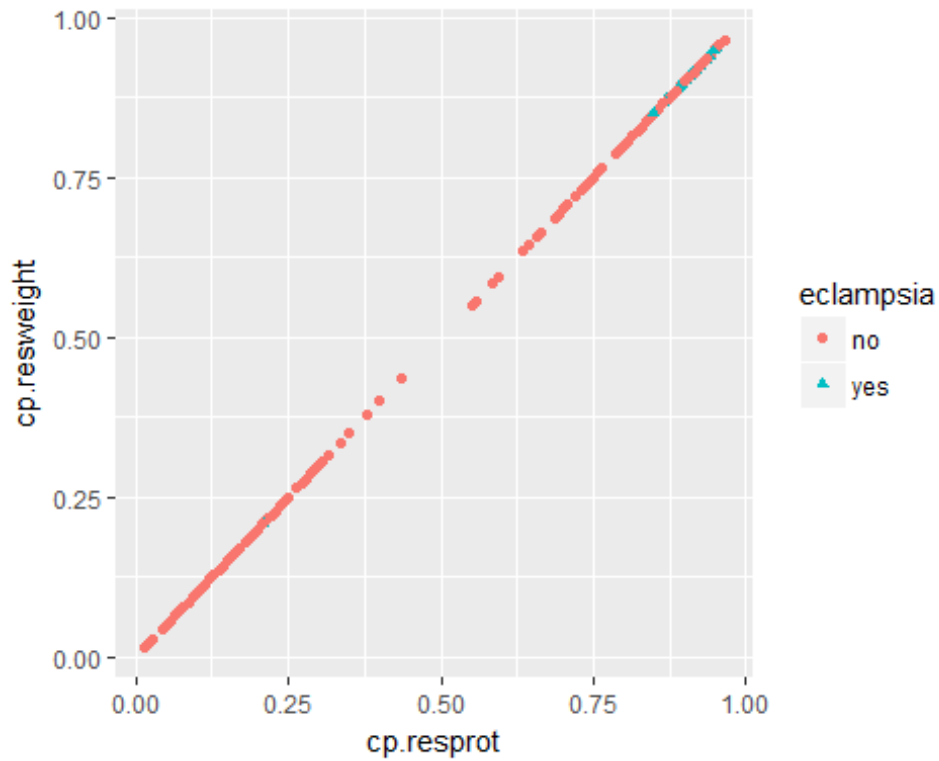
```



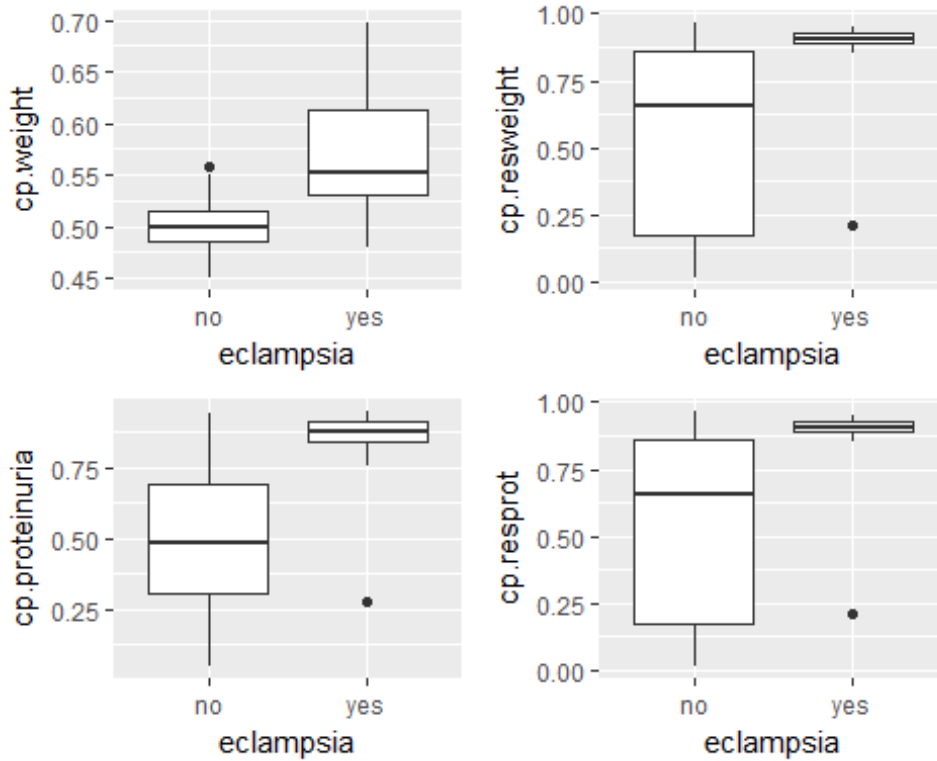
```
#-----
#other displays
#-----
Scatter1<-ggplot(features3, aes(x=cp.proteinuria, y = cp.weight,
                               shape=eclampsia, colour=eclampsia)) + geom_point()
Scatter1
```

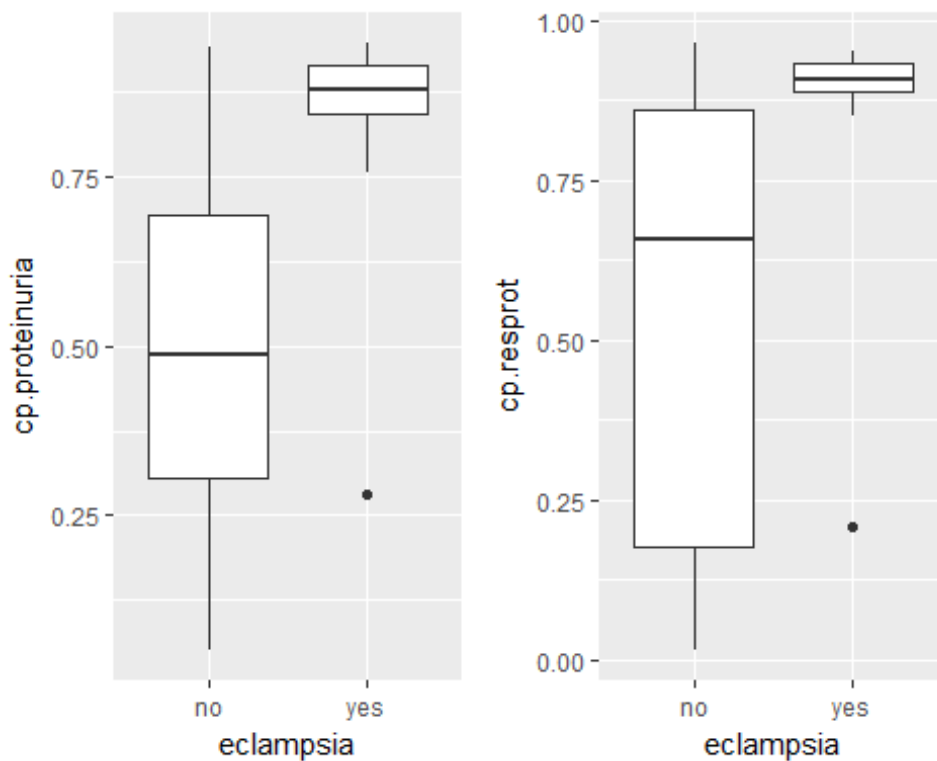
```
Scatter2<-ggplot(features3, aes(x=cp.resprot, y = cp.resweight,
                               shape=eclampsia, colour=eclampsia)) + geom_point()
Scatter2
```



```
grid.arrange(fig1, fig2,fig3,fig4, ncol=2,nrow =2 )
```



```
grid.arrange(fig3,fig4, ncol=2)
```



```

#=====
# Data for classification
#=====
features<-cbind(features3,features2[,-1],features1[,-1],
                features6[,-1],features10[,-1])
#-----
#Libraries for clasification
#-----
library("e1071")      # naiveBayes , SVM
library("klaR")       # NaiveBayes

## Loading required package: MASS

library("class")
library("MASS")
library("foreign")
library(car)
library(verification) # for ROC

## Loading required package: fields

## Loading required package: spam

## Loading required package: grid

## Spam version 1.4-0 (2016-08-29) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##   backsolve, forwardsolve

## Loading required package: maps

## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:car':
##
##   logit

## The following object is masked from 'package:lattice':
##
##   melanoma

## Loading required package: CircStats

```

```

## Loading required package: dtw
## Loading required package: proxy
##
## Attaching package: 'proxy'
## The following object is masked from 'package:spam':
##
##   as.matrix
## The following object is masked from 'package:Matrix':
##
##   as.matrix
## The following objects are masked from 'package:stats':
##
##   as.dist, dist
## The following object is masked from 'package:base':
##
##   as.matrix
## Loaded dtw v1.18-1. See ?dtw for help, citation("dtw") for use in publicat
ion.
library(rpart)          # fror CART
library(adabag)        # Boosting
## Loading required package: mlbench
## Loading required package: caret
Data.Tree<-as.data.frame(features)
attach(Data.Tree)
## The following objects are masked _by_ .GlobalEnv:
##
##   cp.proteinuria, cp.res.weightgain, cp.resprot, cp.resweight,
##   cp.weight, cp.weightgain, ecl, intercepts.weight,
##   qu95.diastolic, qu95.systolic, slopes.weight
head(Data.Tree)
##   ecl cp.proteinuria cp.weight cp.resprot cp.resweight features2[, -1]
## 1  1      0.2071429 0.4500000 0.9357143 0.9357143      217.00
## 2  1      0.1642857 0.4857143 0.1142857 0.1142857      256.00
## 3  1      0.3857143 0.5000000 0.8142857 0.8142857      145.05
## 4  1      0.3142857 0.4928571 0.8928571 0.8928571      107.00
## 5  1      0.5285714 0.4785714 0.9428571 0.9428571      126.00
## 6  1      0.7214286 0.5071429 0.2214286 0.2214286      120.00
##   intercepts.weight slopes.weight qu95.systolic qu95.diastolic
## 1      81810.00      52.99789      150.00      76.00

```

```
## 2      73520.60      54.71281      113.00      79.00
## 3      78687.72      52.64745      118.00      71.05
## 4      72480.88      53.95470      126.05      77.00
## 5      72863.77      53.83778      115.05      68.05
## 6      85057.45      50.74176      126.00      83.00
## cp.weightgain cp.res.weightgain
## 1      0.4500000      0.02857143
## 2      0.4857143      0.04285714
## 3      0.5000000      0.03571429
## 4      0.4928571      0.03571429
## 5      0.4785714      0.02142857
## 6      0.5071429      0.02857143
```

```
#=====
#Classification Trees
```

```
modT.rpart<-rpart(factor(ecl)~., data = Data.Tree)
modT.rpart
```

```
## n= 300
##
## node), split, n, loss, yval, (yprob)
##      * denotes terminal node
##
## 1) root 300 25 1 (0.91666667 0.08333333)
## 2) cp.res.weightgain< 0.40484 272 0 1 (1.00000000 0.00000000) *
## 3) cp.res.weightgain>=0.40484 28 3 2 (0.10714286 0.89285714) *
```

```
pred.tr<-predict(modT.rpart, Data.Tree, type = "class")
table(pred.tr,ecl)
```

```
##      ecl
## pred.tr no yes
##      1 272  0
##      2  3 25
```

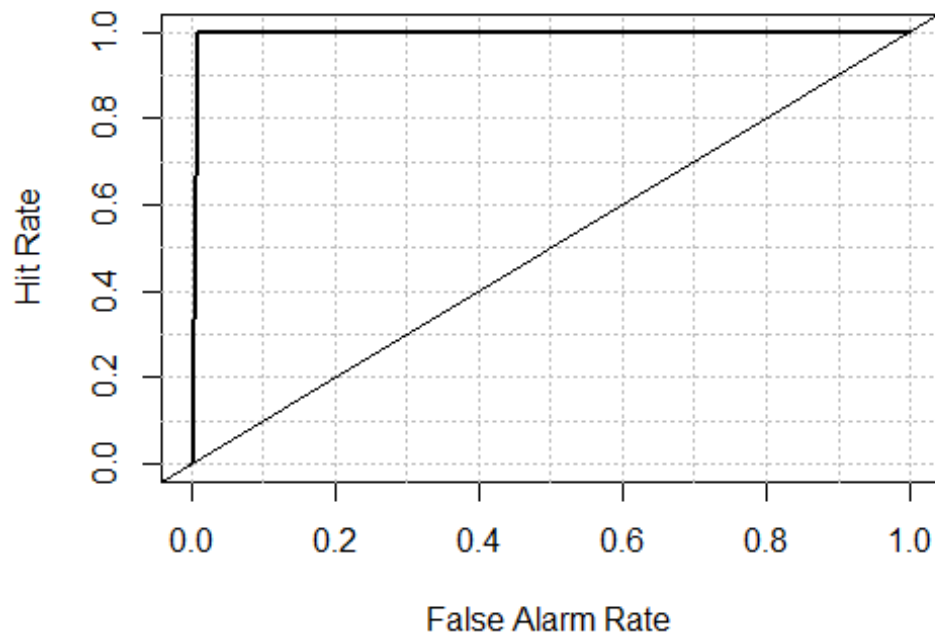
```
plot
```

```
## standardGeneric for "plot" defined from package "graphics"
##
## function (x, y, ...)
## standardGeneric("plot")
## <environment: 0x000000001222e870>
## Methods may be defined for arguments: x, y
## Use showMethods("plot") for currently available ones.
```

```
#-----
#ROC
```

```
ecl<-as.numeric(ecl)-1
pred.tr1<-predict(modT.rpart, Data.Tree)
roc.plot(ecl, pred.tr1[,2], plot.thres = NULL)
```

ROC Curve



```
roc.area(ecl,pred.tr1[,2])
```

```
## $A
## [1] 0.9945455
##
## $n.total
## [1] 300
##
## $n.events
## [1] 25
##
## $n.noevents
## [1] 275
##
## $p.value
## [1] 1.167375e-59
```

```
#-----
```

```
# Classification Tree with Cross validation
```

```
mod.cv<-rpart(factor(ecl)~., data = Data.Tree,
               control=rpart.control(xval=10))
summary(mod.cv)
```

```
## Call:
## rpart(formula = factor(ecl) ~ ., data = Data.Tree, control = rpart.control
## (xval = 10))
## n= 300
```

```

##
##      CP nsplit rel error xerror      xstd
## 1 0.88      0      1.00   1.00 0.19148542
## 2 0.01      1      0.12   0.12 0.06893475
##
## Variable importance
## cp.res.weightgain      slopes.weight      cp.weight      cp.weightgain
##              37              19              15              15
##      cp.proteinuria      qu95.diastolic
##              8              7
##
## Node number 1: 300 observations,      complexity param=0.88
## predicted class=1 expected loss=0.08333333 P(node) =1
## class counts: 275 25
## probabilities: 0.917 0.083
## left son=2 (272 obs) right son=3 (28 obs)
## Primary splits:
##      cp.res.weightgain < 0.40484 to the left, improve=40.47619, (0 mis
sing)
##      slopes.weight      < 57.33591 to the left, improve=24.66256, (0 mis
sing)
##      cp.weight      < 0.5436508 to the left, improve=20.22929, (0 mis
sing)
##      cp.weightgain      < 0.5436508 to the left, improve=20.22929, (0 mis
sing)
##      cp.proteinuria      < 0.8431153 to the left, improve=18.66449, (0 mis
sing)
## Surrogate splits:
##      slopes.weight      < 57.33591 to the left, agree=0.953, adj=0.500, (0
split)
##      cp.weight      < 0.5436508 to the left, agree=0.943, adj=0.393, (0
split)
##      cp.weightgain      < 0.5436508 to the left, agree=0.943, adj=0.393, (0
split)
##      cp.proteinuria      < 0.8431153 to the left, agree=0.927, adj=0.214, (0
split)
##      qu95.diastolic      < 93.225 to the left, agree=0.923, adj=0.179, (0
split)
##
## Node number 2: 272 observations
## predicted class=1 expected loss=0 P(node) =0.9066667
## class counts: 272 0
## probabilities: 1.000 0.000
##
## Node number 3: 28 observations
## predicted class=2 expected loss=0.1071429 P(node) =0.09333333
## class counts: 3 25
## probabilities: 0.107 0.893

pred.cv<-predict(mod.cv, Data.Tree, type = "class")
table(pred.cv,ecl)

```

```

##      ecl
## pred.cv  0  1
##      1 272  0
##      2  3 25

# same as first solution

mod.cv3<-rpart(factor(ecl)~., data = Data.Tree,
               control=rpart.control(xval=100))
summary(mod.cv3)

## Call:
## rpart(formula = factor(ecl) ~ ., data = Data.Tree, control = rpart.control
## (xval = 100))
##   n= 300
##
##      CP nsplit rel error xerror      xstd
## 1 0.88      0      1.00  1.00 0.19148542
## 2 0.01      1      0.12  0.12 0.06893475
##
## Variable importance
## cp.res.weightgain      slopes.weight      cp.weight      cp.weightgain
##                37                19                15                15
## cp.proteinuria      qu95.diastolic
##                8                7
##
## Node number 1: 300 observations,      complexity param=0.88
## predicted class=1 expected loss=0.08333333 P(node) =1
## class counts: 275 25
## probabilities: 0.917 0.083
## left son=2 (272 obs) right son=3 (28 obs)
## Primary splits:
##      cp.res.weightgain < 0.40484 to the left, improve=40.47619, (0 mis
## sing)
##      slopes.weight < 57.33591 to the left, improve=24.66256, (0 mis
## sing)
##      cp.weight < 0.5436508 to the left, improve=20.22929, (0 mis
## sing)
##      cp.weightgain < 0.5436508 to the left, improve=20.22929, (0 mis
## sing)
##      cp.proteinuria < 0.8431153 to the left, improve=18.66449, (0 mis
## sing)
## Surrogate splits:
##      slopes.weight < 57.33591 to the left, agree=0.953, adj=0.500, (0
## split)
##      cp.weight < 0.5436508 to the left, agree=0.943, adj=0.393, (0
## split)
##      cp.weightgain < 0.5436508 to the left, agree=0.943, adj=0.393, (0
## split)
##      cp.proteinuria < 0.8431153 to the left, agree=0.927, adj=0.214, (0
## split)

```



```

##      qu95.diastolic < 93.225   to the left, agree=0.923, adj=0.179, (0
split)
##
## Node number 2: 272 observations
## predicted class=1 expected loss=0 P(node) =0.9066667
##   class counts:  272   0
##   probabilities: 1.000 0.000
##
## Node number 3: 28 observations
## predicted class=2 expected loss=0.1071429 P(node) =0.09333333
##   class counts:    3   25
##   probabilities: 0.107 0.893

pred.cv3<-predict(mod.cv3, Data.Tree, type = "class")
table(pred.cv3,ecl)

##      ecl
## pred.cv3  0   1
##          1 272  0
##          2   3 25

#=====
#SVM with cross validation
#=====
x<-Data.Tree[, -1]
y<-as.factor(ecl)
mod.svm5<-svm(x,y, probability = TRUE, cross = 10,
              data=Data.Tree, kernel="radial")

summary(mod.svm5)

##
## Call:
## svm.default(x = x, y = y, kernel = "radial", cross = 10, probability = TRU
E,
##   data = Data.Tree)
##
##
## Parameters:
##   SVM-Type:  C-classification
##   SVM-Kernel: radial
##     cost:    1
##   gamma:    0.09090909
##
## Number of Support Vectors:  50
##
## ( 35 15 )
##
##
## Number of Classes:  2
##

```

```

## Levels:
## 0 1
##
## 10-fold cross-validation on training data:
##
## Total Accuracy: 99
## Single Accuracies:
## 96.66667 100 100 100 100 93.33333 100 100 100 100

pred5<-fitted(mod.svm5)
length(pred5)

## [1] 300

table(pred5, Data.Tree$ecl)

##
## pred5  1  2
##      0 275  0
##      1  0  25

#-----
#Probabilities

pred2<-predict(mod.svm5,x, decision.values= TRUE, probability = TRUE)
summary(pred2)

##  0  1
## 275 25

attr(pred2, "decision.values")[1:10,]

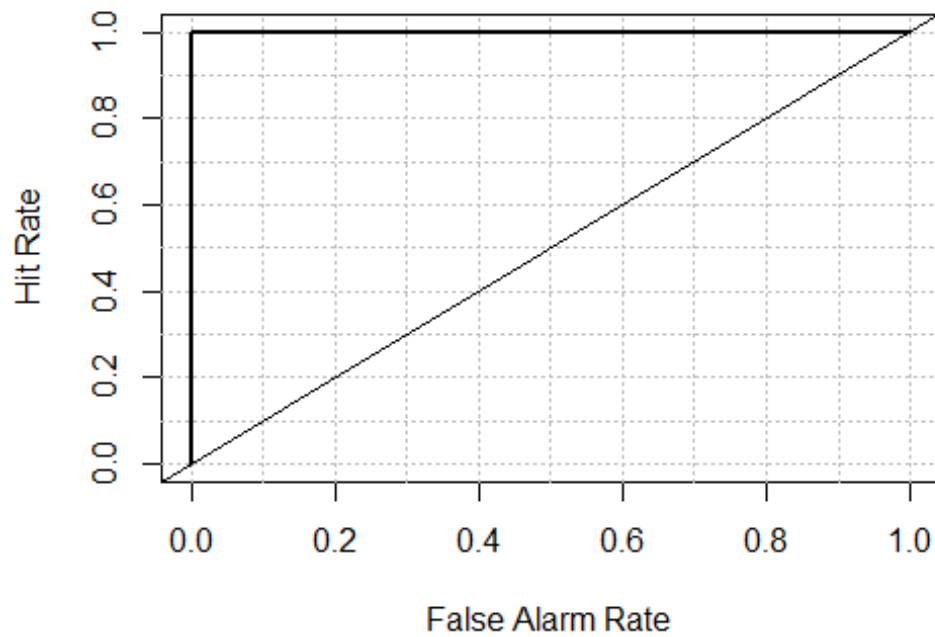
##          1          2          3          4          5          6
## 0.9997982 0.9996295 1.5673141 1.5091316 1.2488552 1.5828134
##          7          8          9         10
## 1.2953438 1.2736077 -1.1929344 1.4089791

prob<-attr(pred2, "probabilities")

roc.plot(ecl, prob[,2],plot.thres = NULL )

```

ROC Curve



```
roc.area(ecl,prob[,2])
```

```
## $A  
## [1] 1  
##  
## $n.total  
## [1] 300  
##  
## $n.events  
## [1] 25  
##  
## $n.noevents  
## [1] 275  
##  
## $p.value  
## [1] 6.338534e-17
```

```
#####  
# Boosting  
#####
```

```
ecl<-as.factor(ecl)  
levels(ecl)
```

```
## [1] "0" "1"
```

```
levels(ecl)[levels(ecl)=="0"] <- "no"  
levels(ecl)[levels(ecl)=="1"] <- "yes"  
levels(ecl)
```

```

## [1] "no" "yes"

data.boost<-cbind (Data.Tree[, -1],ecl)
data.boost<-as.data.frame(data.boost)

attach(data.boost)

## The following objects are masked _by_ .GlobalEnv:
##
## cp.proteinuria, cp.res.weightgain, cp.resprot, cp.resweight,
## cp.weight, cp.weightgain, ecl, intercepts.weight,
## qu95.diastolic, qu95.systolic, slopes.weight

## The following objects are masked from Data.Tree:
##
## cp.proteinuria, cp.res.weightgain, cp.resprot, cp.resweight,
## cp.weight, cp.weightgain, ecl, features2[, -1],
## intercepts.weight, qu95.diastolic, qu95.systolic,
## slopes.weight

mod.boost<-boosting(ecl~ .,
                    data=data.boost, mfinal = 10)
summary(mod.boost)

##          Length Class  Mode
## formula          3  formula call
## trees            10  -none- list
## weights          10  -none- numeric
## votes           600  -none- numeric
## prob            600  -none- numeric
## class           300  -none- character
## importance      11  -none- numeric
## terms           3    terms  call
## call            4    -none- call

pred<- predict(mod.boost, data.boost)
ls(pred)

## [1] "class"      "confusion" "error"      "formula"    "prob"       "votes"

pred$confusion

##          Observed Class
## Predicted Class no yes
##          no 275  0
##          yes  0 25

#-----
#Cross Validation
#-----
mod.boostCV<-boosting.cv(ecl~., data=data.boost, v=10, mfinal = 10)

```

```
## i: 1 Tue May 16 20:13:16 2017
## i: 2 Tue May 16 20:13:17 2017
## i: 3 Tue May 16 20:13:17 2017
## i: 4 Tue May 16 20:13:18 2017
## i: 5 Tue May 16 20:13:18 2017
## i: 6 Tue May 16 20:13:19 2017
## i: 7 Tue May 16 20:13:19 2017
## i: 8 Tue May 16 20:13:20 2017
## i: 9 Tue May 16 20:13:20 2017
## i: 10 Tue May 16 20:13:21 2017
```

```
summary(mod.boostCV)
```

```
##           Length Class  Mode
## class      300    -none- character
## confusion   4     table  numeric
## error       1     -none- numeric
```

```
mod.boostCV$error
```

```
## [1] 0.003333333
```

```
mod.boostCV$confusion
```

```
##           Observed Class
## Predicted Class no yes
##           no 274  0
##           yes  1 25
```

```
ls(mod.boostCV)
```

```
## [1] "class"      "confusion" "error"
```

```
# predT.cv<- predict(mod.boostCV, data.boostT)
```