Prosocial Analysis

##function definition & library

```
GlobalT <- 5000
library(caret)
singlehistplot <- function(df,main)
{
  #sprintf("mean = %f ", mean(df))
  hist(df, main=main)
  qts <- quantile(df,probs=c(.025,.975))
  hist(df, main=main)
  abline(v=qts[1],col="red")
  abline(v=qts[2],col="red")
}

histplot <- function(cm_tb)
{
  singlehistplot(cm_tb$Accuracy, main="Accuracy")
  singlehistplot(cm_tb$`Test Accuracy`, main="Test Accuracy")
  singlehistplot(cm_tb$AIC, main="AIC")
  singlehistplot(cm_tb$`Balanced Accuracy`, main="Balanced Accuracy")
  singlehistplot(cm_tb$Sensitivity, main="Sensitivity")
  singlehistplot(cm_tb$Specificity, main="Specificity")
  singlehistplot(cm_tb$Precision, main="Precision")
  singlehistplot(cm_tb$Recall, main="Recall")
  singlehistplot(cm_tb$F1, main="F1")
}

prosocial_preprocessing <- function(df)
{
  df <- df[complete.cases(df), ]
  df<- df[(df$prosocial_parent  <= (3)) | (df$prosocial_parent  == 6), ]
  df<- df[(df$prosocial_child <= (3)) | (df$prosocial_child  == 6), ]
  df$prosocial_sum <- df$prosocial_parent+df$prosocial_child
  df <- df[df$prosocial_sum != 9, ]
  df <- df[df$prosocial_sum != 8, ]
  df <- df[df$prosocial_sum != 7, ]
  cut <- df[df$prosocial_sum == 6, ]
  cut <- cut[cut$prosocial_child != 3, ]
  df <- subset(df, !(subjectkey %in% cut$subjectkey))
  df$y <- ifelse(df$prosocial_sum <= (6), 0, 1)
  return(df)
}
```

```
twoclasssample <- function(df, n)
{
  truedf <- df[df$y==1,]
  truedf <- truedf[sample(1:nrow(truedf), n, replace = TRUE), ]
  falsedf <- df[df$y==0,]
  falsedf <- falsedf[sample(1:nrow(falsedf), n, replace = TRUE), ]
  finaldf <- rbind(truedf, falsedf)
  return(finaldf)
}

bootstrap_glm <- function(df, T, plot=TRUE, filesname)
{
  names <- c("Sensitivity", "Specificity", "Pos Pred Value", "Neg Pred Value", "Preci
sion", "Recall", "F1", "Prevalence", "Detection Rate", "Detection Prevalence", "Balan
ced Accuracy", 'Accuracy', 'AIC', 'Converged', 'Test Accuracy')

  cm_tb <- data.frame(matrix(ncol = length(names), nrow = 0))
  colnames(cm_tb) <- names

  finaldf <- twoclasssample(df, n=200)
  dt = sort(sample(nrow(finaldf), nrow(finaldf)*.75))
  train<-finaldf[dt,]
  test<-finaldf[-dt,]
  glm.fit <- glm(y ~ ., data = train, family = binomial)

  s_tb <- as.data.frame(t(coef(summary(glm.fit))[,4]))

  for (t in 1:T){
    finaldf <- twoclasssample(df, n=200)

    dt = sort(sample(nrow(finaldf), nrow(finaldf)*.75))
    train<-finaldf[dt,]
    test<-finaldf[-dt,]

    glm.fit <- glm(y ~ ., data = train, family = binomial)


    cm <- confusionMatrix(table(as.numeric(glm.fit$fitted.values>0.5),train$y))
    pred <- predict(glm.fit, test)
    cm2 <- confusionMatrix(table(as.numeric(pred>0.5), test$y))

    v <- t(cm$byClass)
    v<-append(v, cm$overall[1])
    v<-append(v, glm.fit$aic)
    v<-append(v, glm.fit$converged)
    v<-append(v, cm2$overall[1])
```

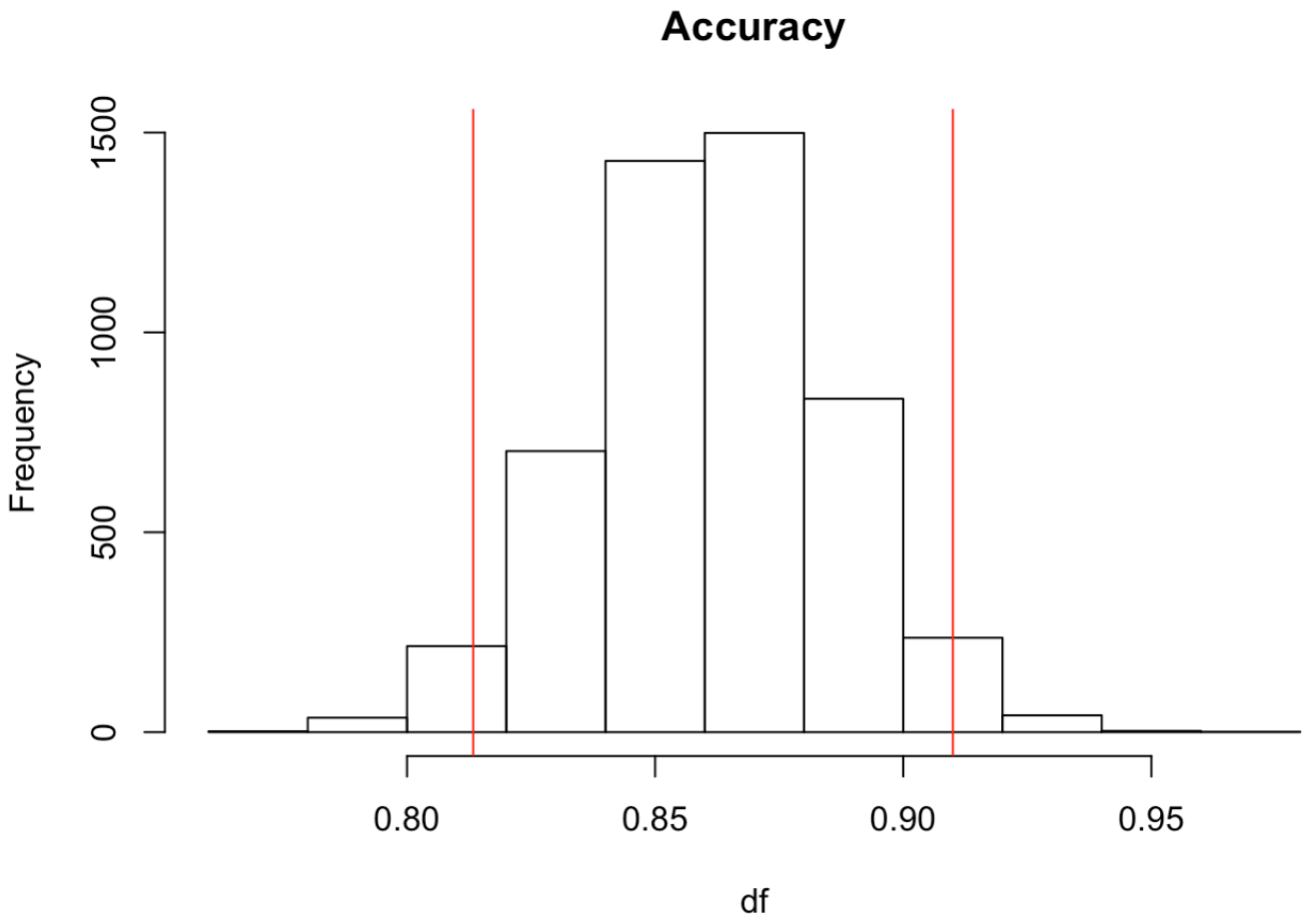
```
    cm_tb <- rbind(cm_tb, as.data.frame(t(v)))
    s_tb <- rbind(s_tb, as.data.frame(t(coef(summary(glm.fit))[,4])))
  }

  s_tb <- s_tb[2:nrow(s_tb),]
  write.csv(s_tb, file=filesname, row.names = FALSE)
  colnames(cm_tb) <- names
  if(plot){
    histplot(cm_tb)
  }
  return(cm_tb)
}

singleglm <- function(df)
{
  finaldf<-twoclasssample(df, n=200)

  glm.fit <- glm(y ~ ., data = finaldf, family = binomial)
  summary(glm.fit)
  cm <- confusionMatrix(table(as.numeric(glm.fit$fitted.values>0.5), finaldf$y))
  print(cm)
  print(cm$byClass)
}
```

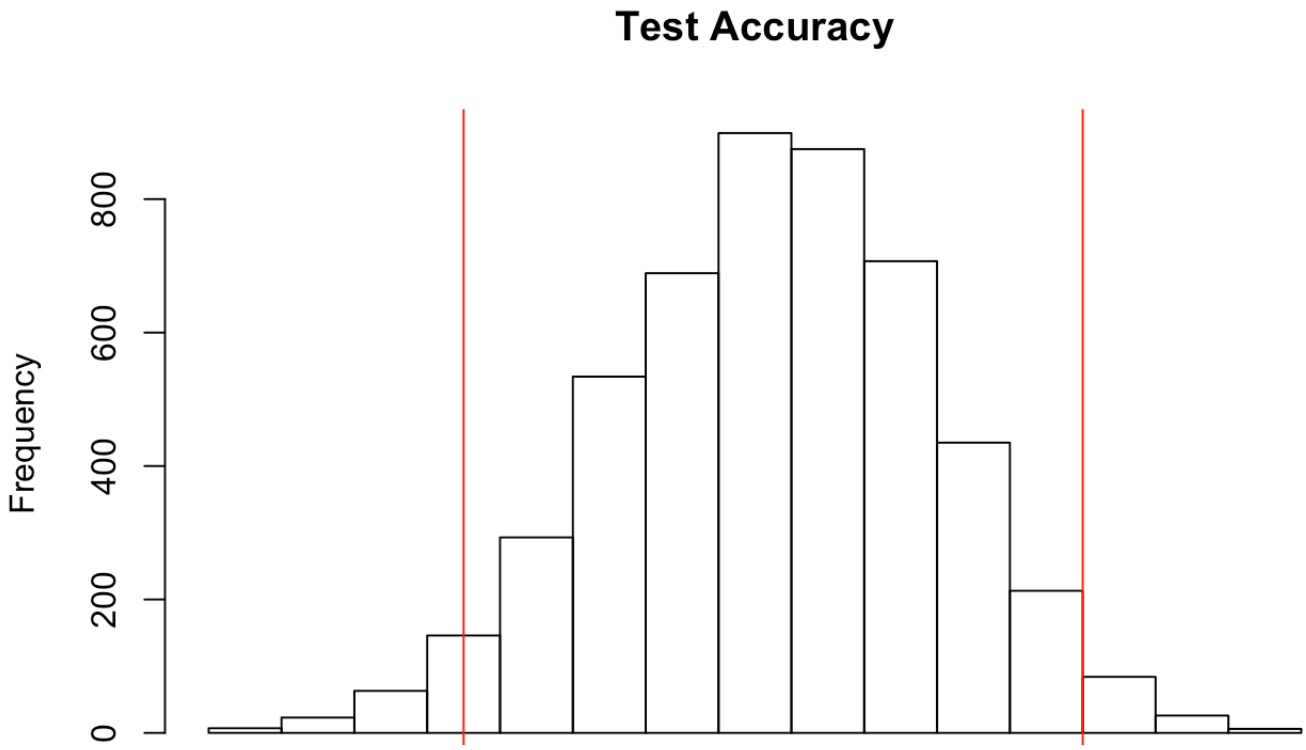
EPF

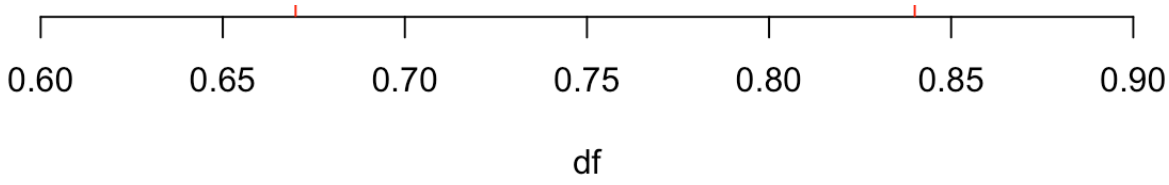
```
df<-prosocial_preprocessing(read.csv('cleandata.csv'))
df <- subset(df, select=-c(X,aggressive_sumscore , prosocial_child, prosocial_parent,
interview_date, interview_age, subjectkey, prosocial_sum))
```

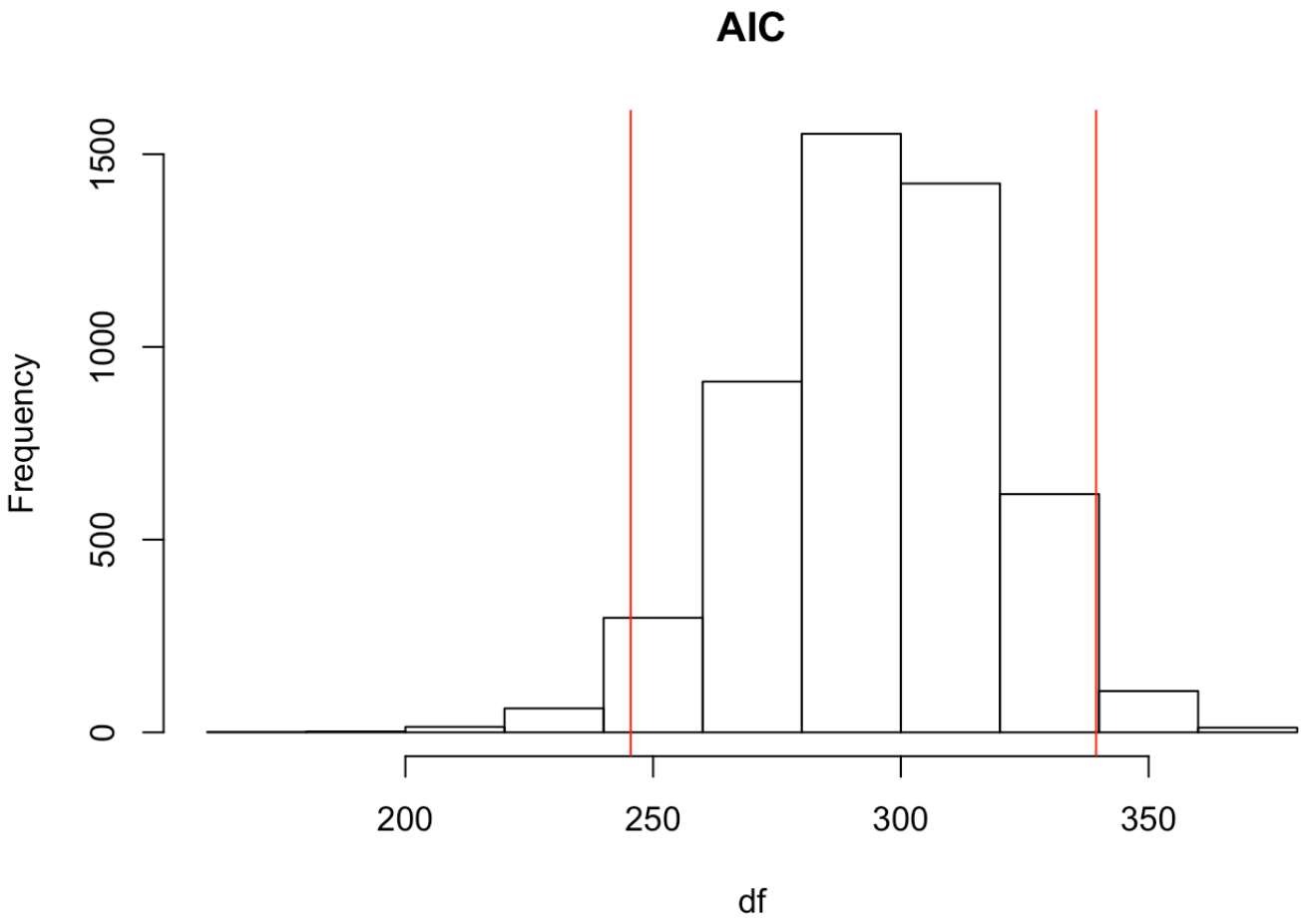
```
#singleglm(df)
```

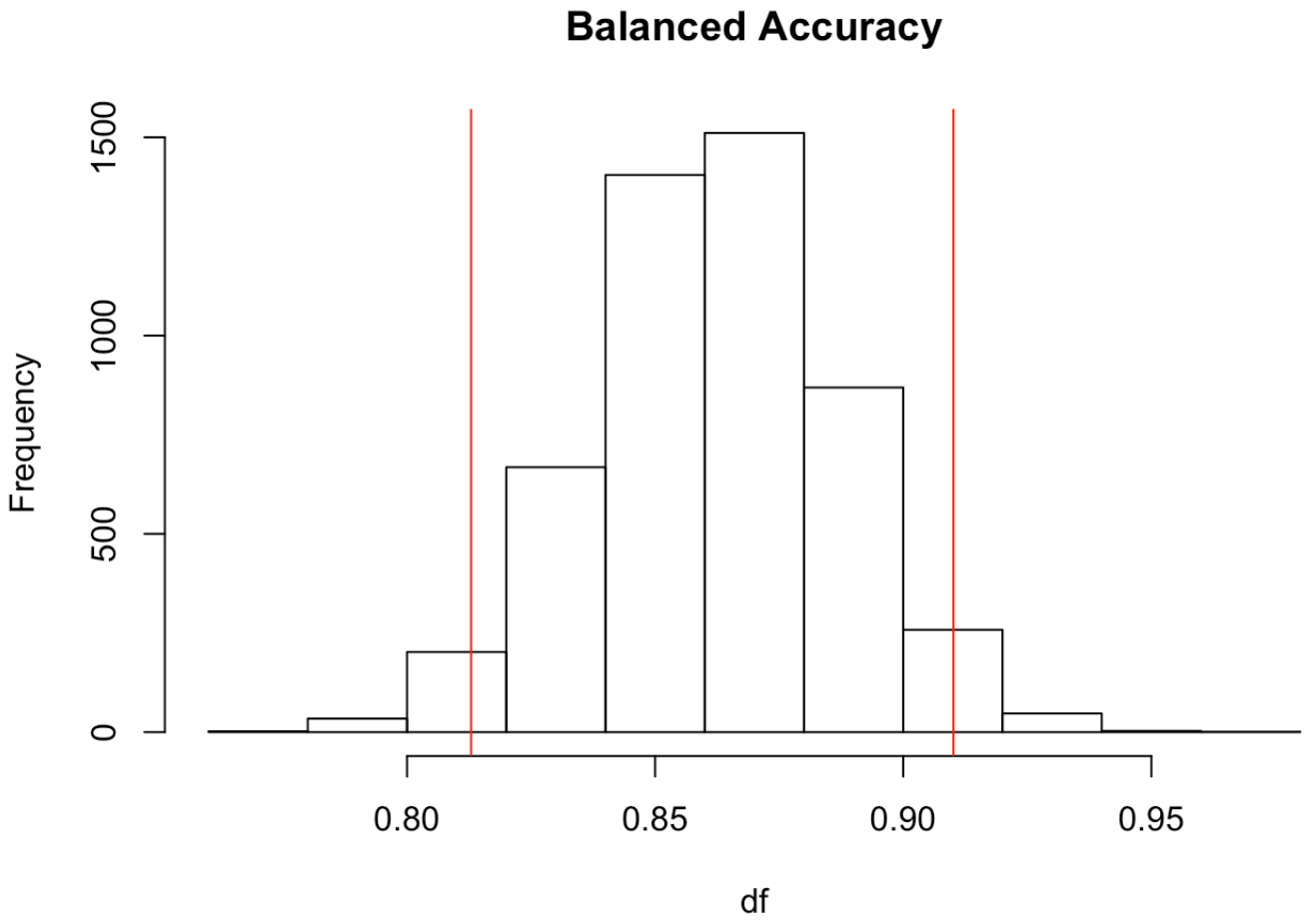
```
cm_tb<-bootstrap_glm(df, T=GlobalT, filesname = './final/Prosocial-Attributes-EPF.csv
')
```

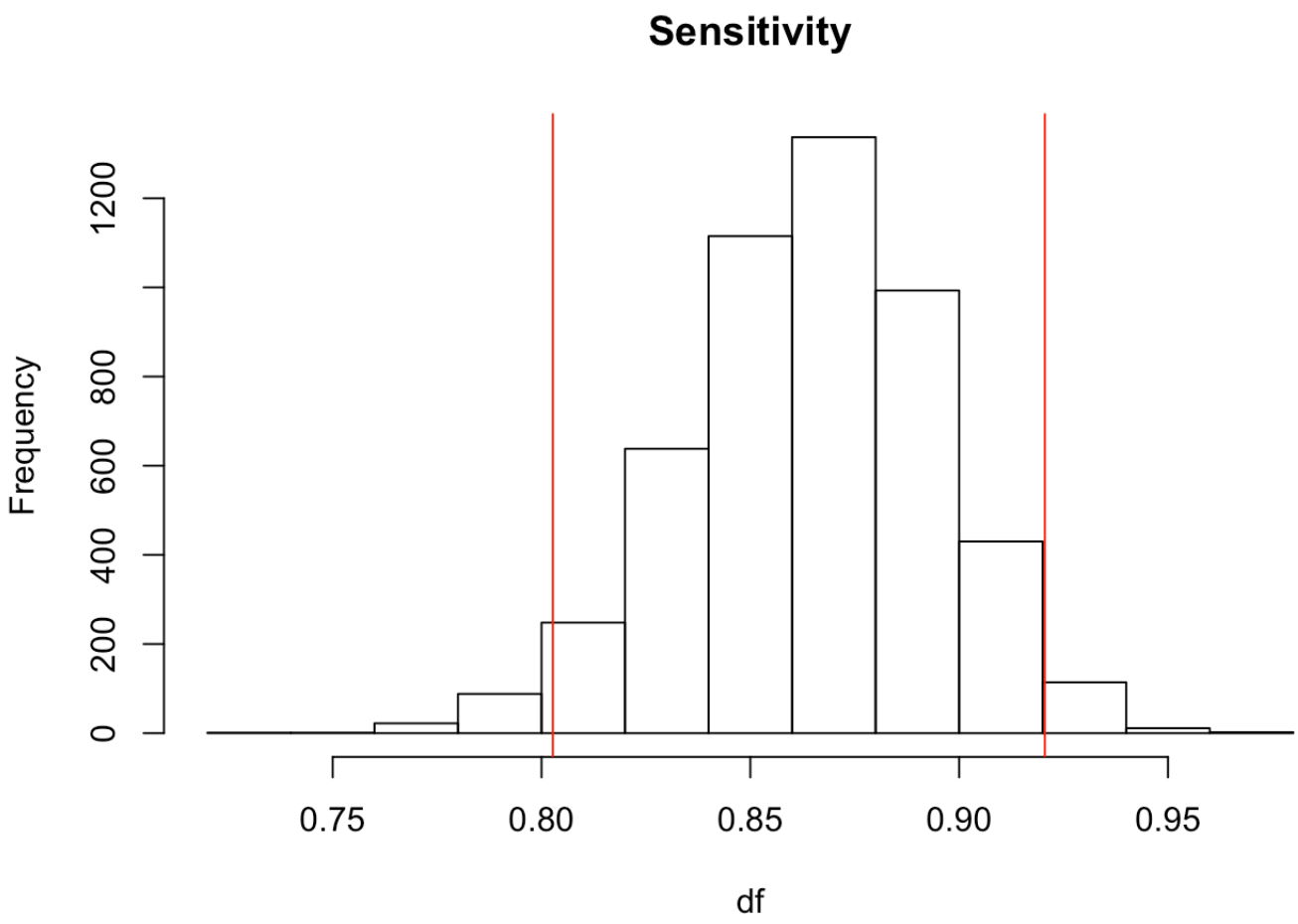

Accuracy
Frequency
0
500
1000
1500
0.80
0.85
0.90
0.95
df

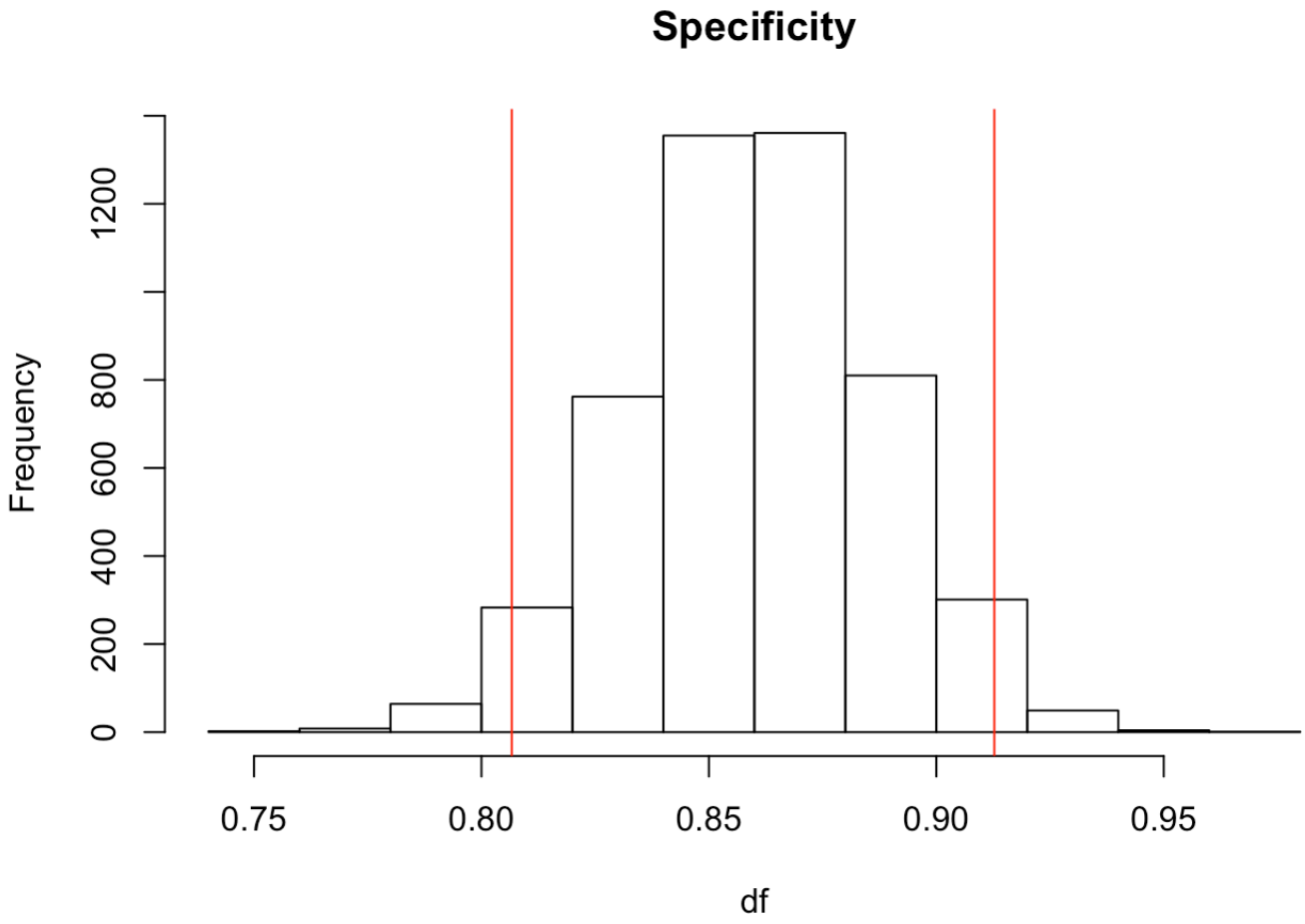

Test Accuracy
Frequency
0
200
400
600
800

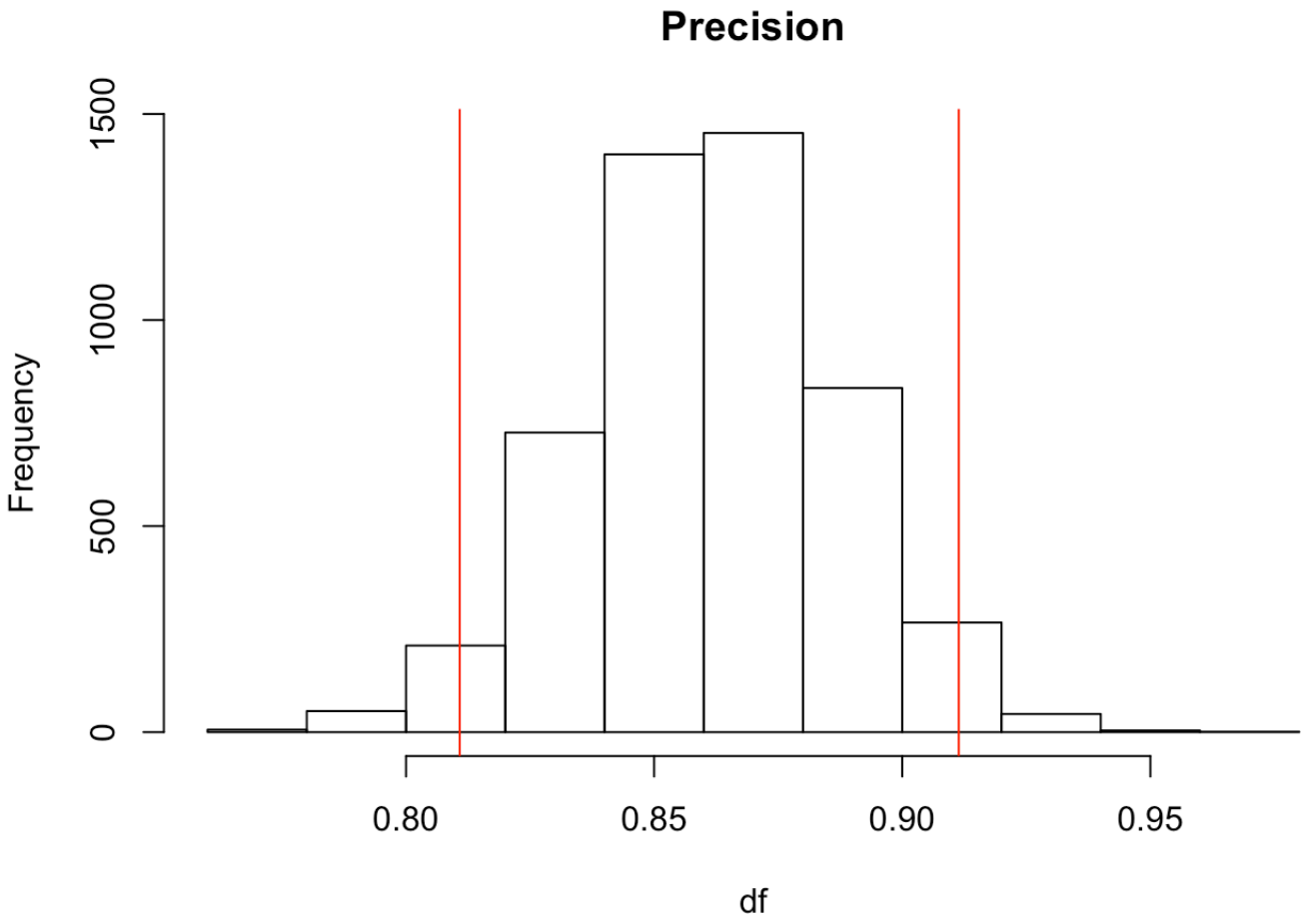

0.60
0.65
0.70
0.75
0.80
0.85
0.90
df

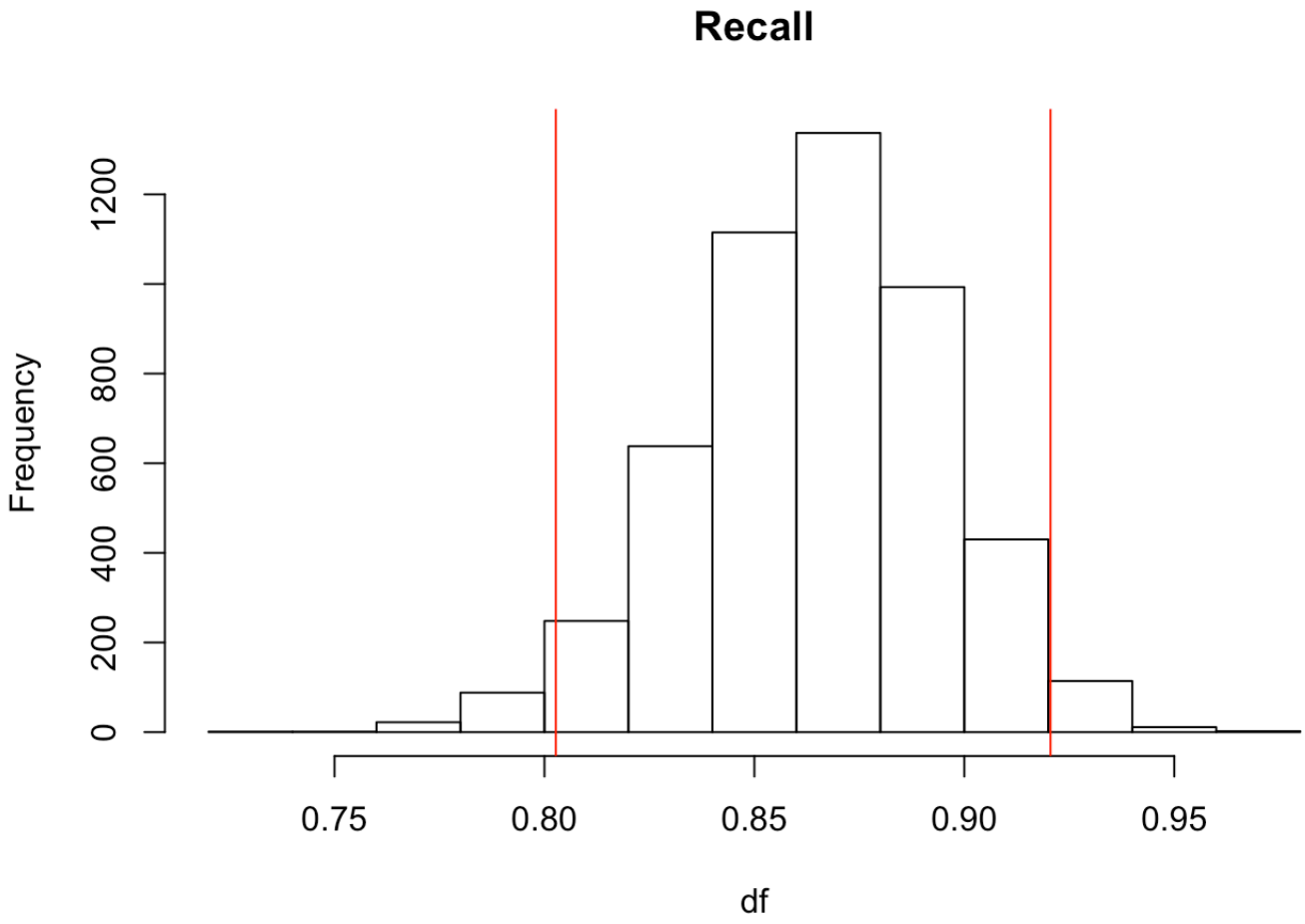

AIC
Frequency
0
500
1000
1500
200
250
300
350
df

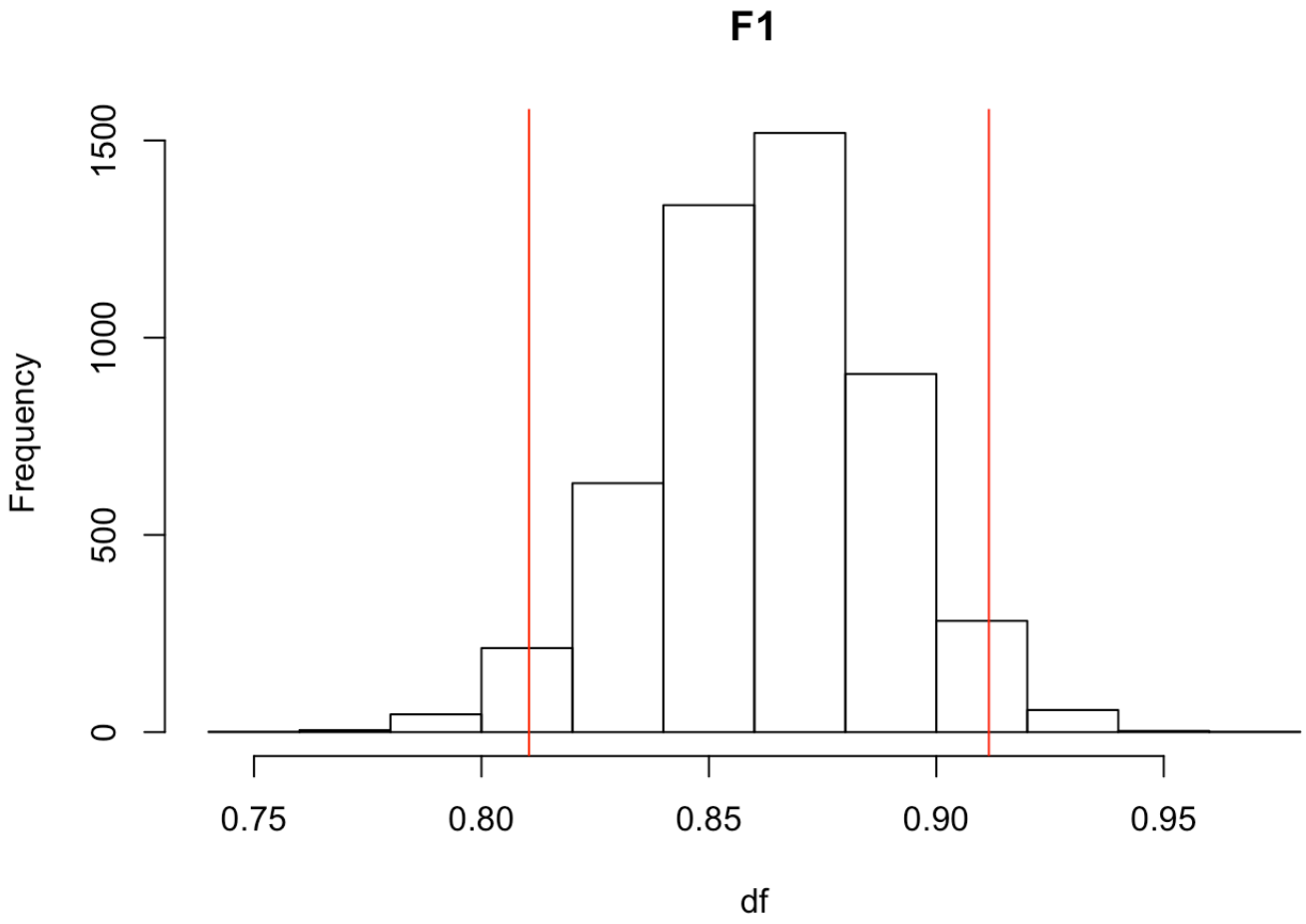

Balanced Accuracy
Frequency
0
500
1000
1500
0.80
0.85
0.90
0.95
df


Sensitivity
Frequency
0
200
400
600
800
1200
0.75
0.80
0.85
0.90
0.95
df


Specificity
Frequency
0
200
400
600
800
1200
0.75
0.80
0.85
0.90
0.95
df


Precision
Frequency
0
500
1000
1500
0.80
0.85
0.90
0.95
df


Recall
Frequency
0
200
400
600
800
1200
0.75
0.80
0.85
0.90
0.95
df




```
write.csv(cm_tb,"./final/Prosocial-Results-EPF.csv", row.names = FALSE)
```

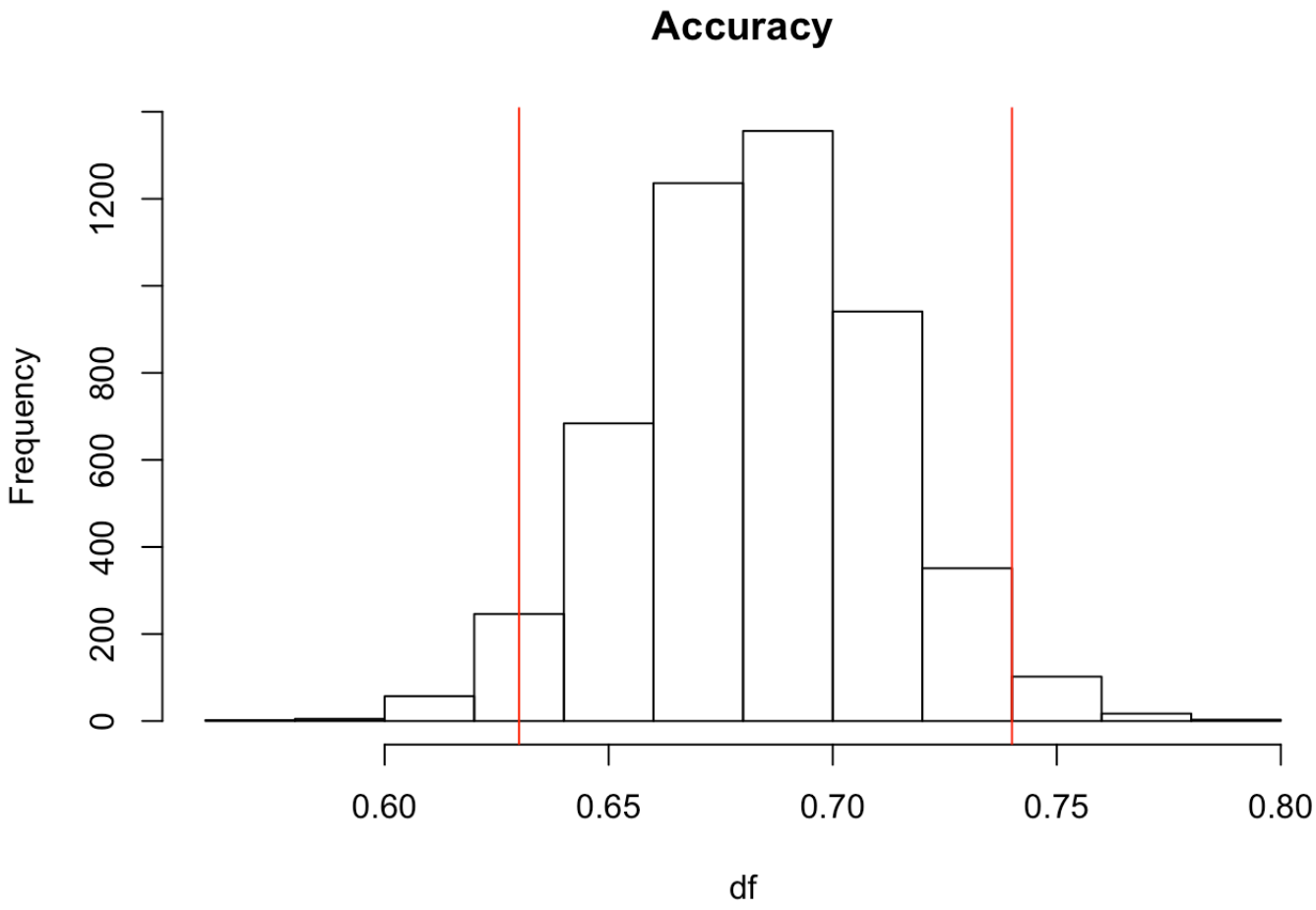
Brain

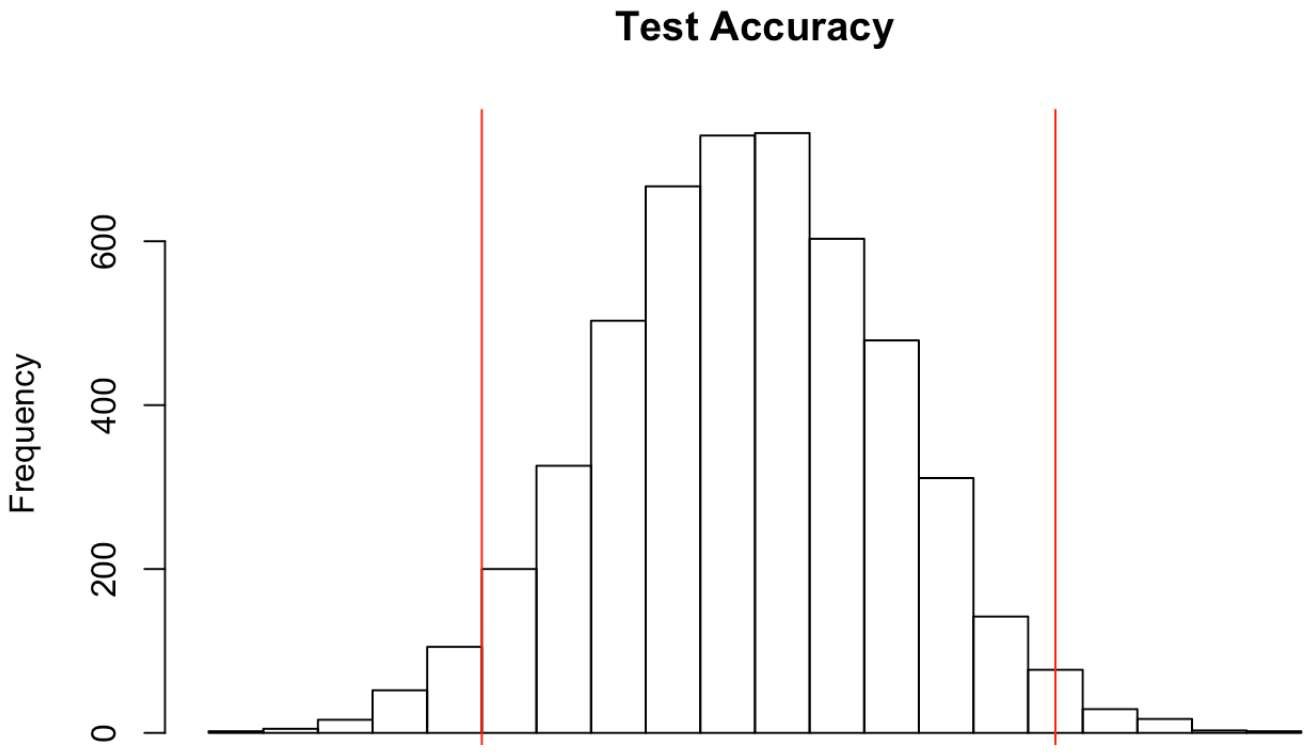
Attribute selection

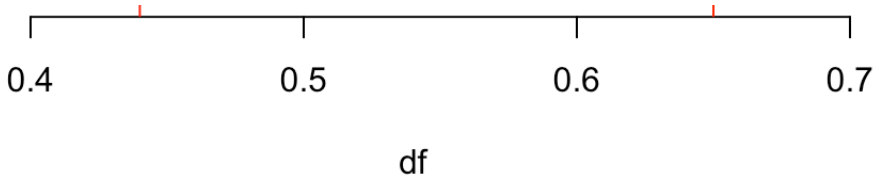
```
df<-prosocial_preprocessing(read.csv('brain_cb.csv'))
df <- subset(df, select=-c(aggressive_sumscore, prosocial_child, prosocial_parent, su
bjectkey,prosocial_sum))
```

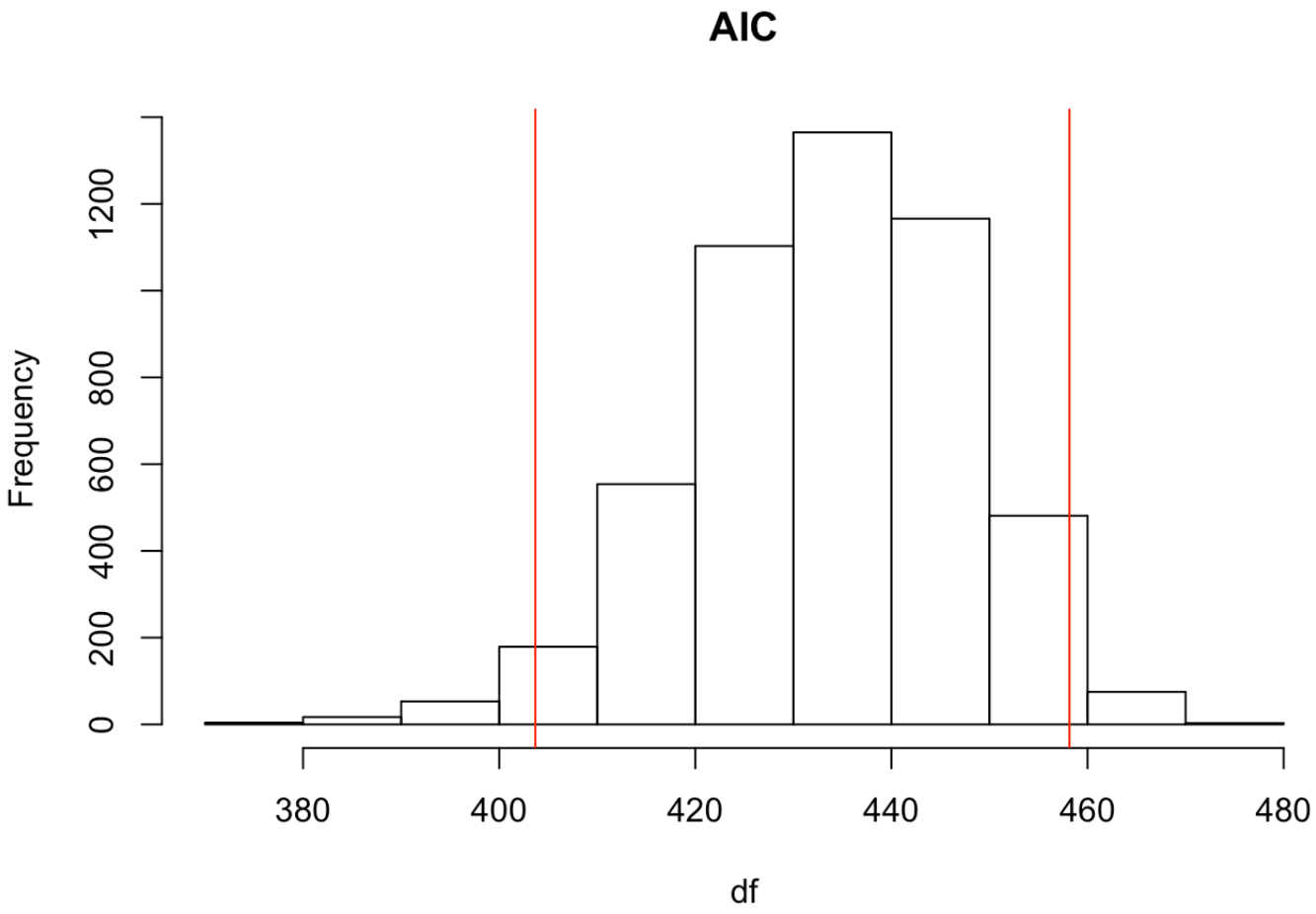
```
#singleglm(df)
```

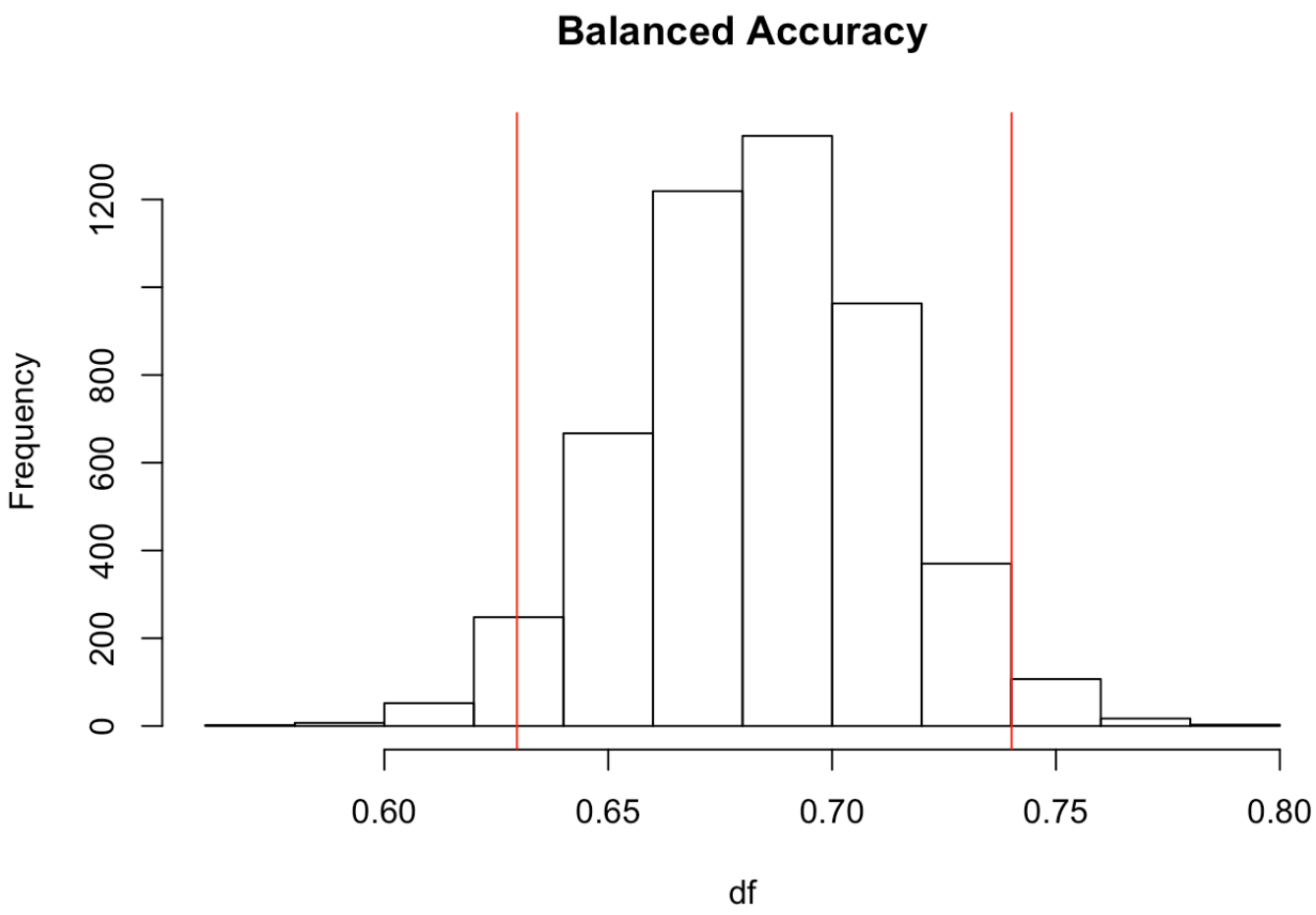
```
cm_tb<-bootstrap_glm(df, T=GlobalT, filesname = './final/Prosocial-Attributes-brain.c
sv')
```

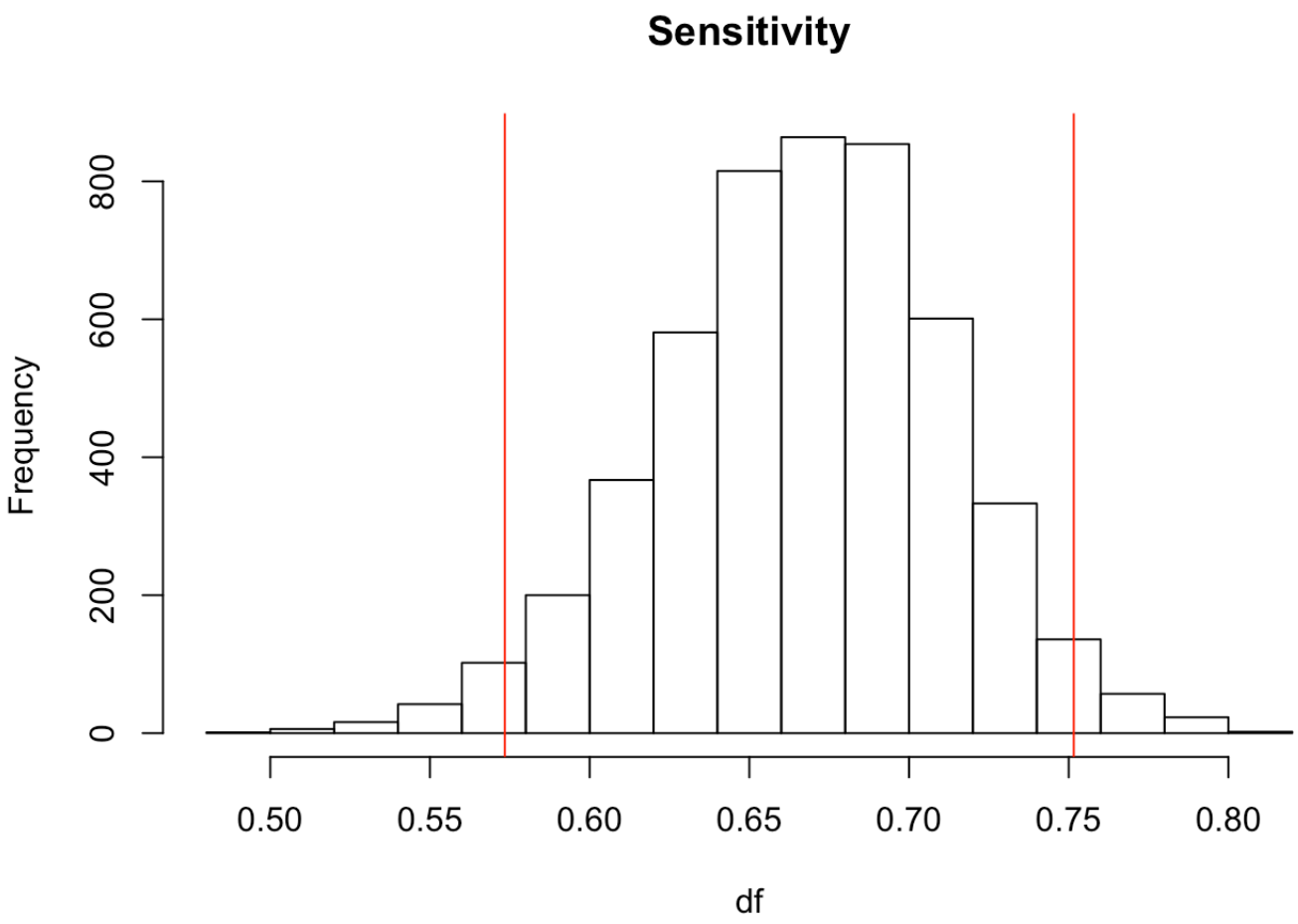

Accuracy
Frequency
0
200
400
600
800
1200
0.60
0.65
0.70
0.75
0.80
df

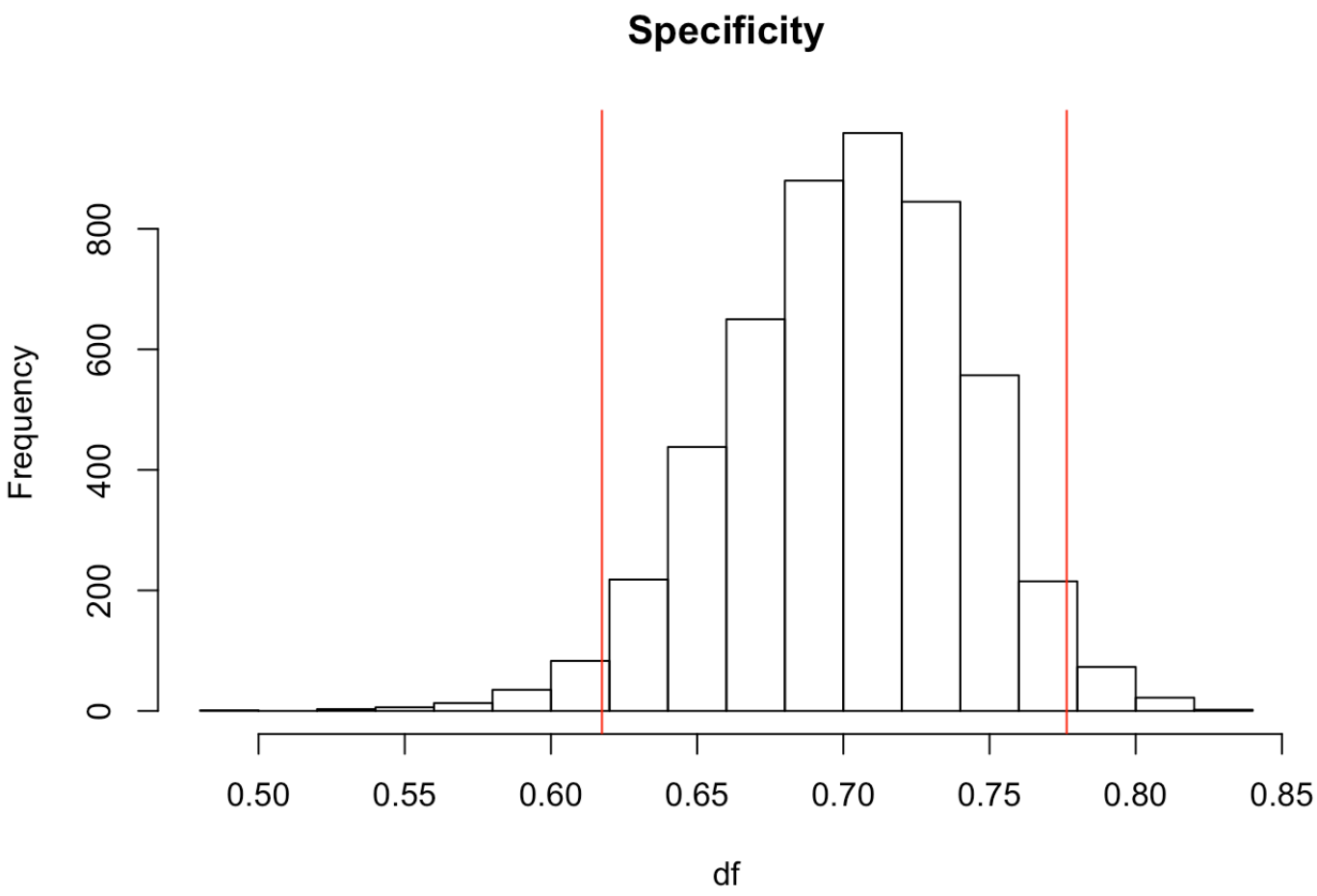

Test Accuracy
Frequency
0
200
400
600

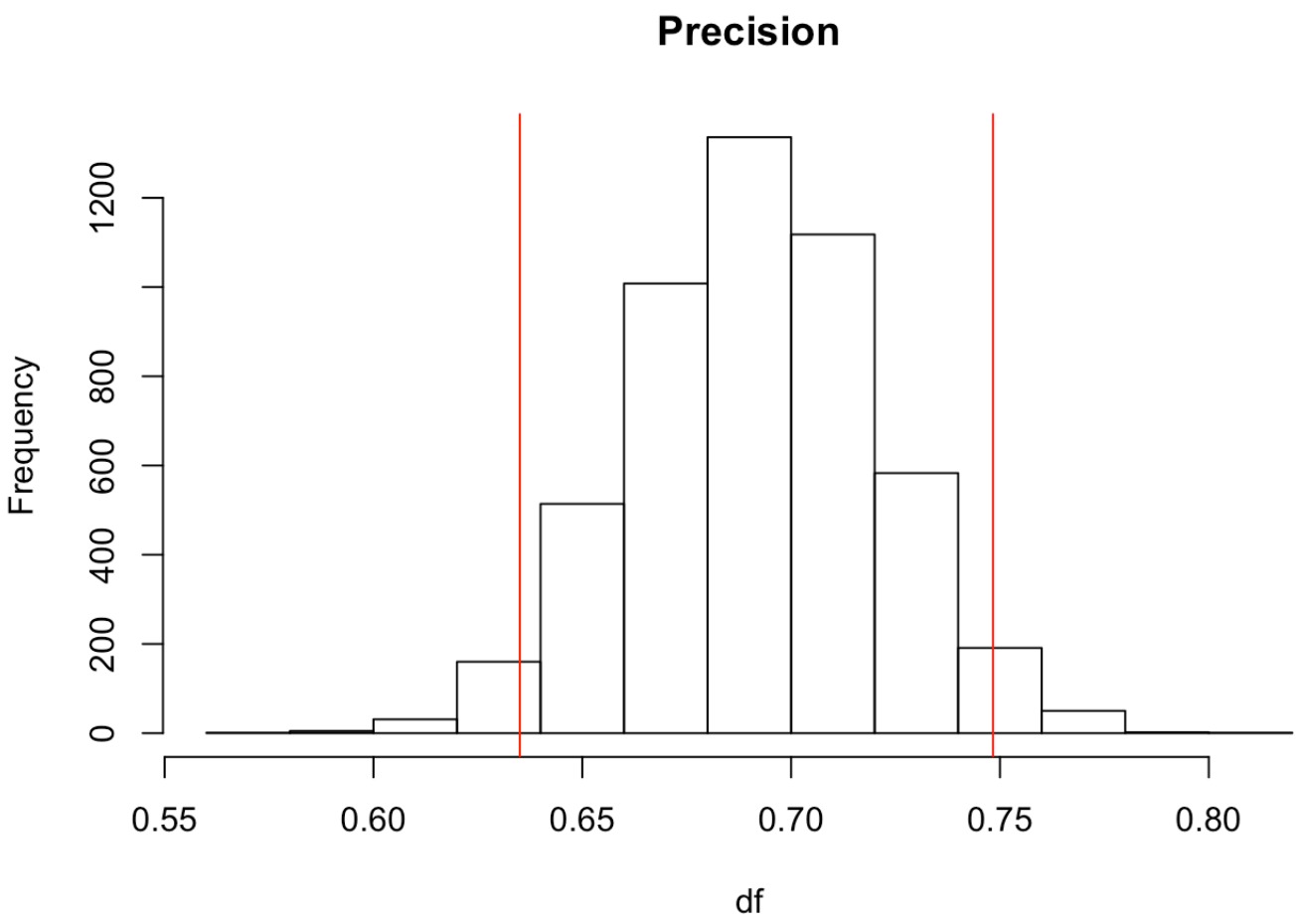

0.4
0.5
0.6
0.7
df

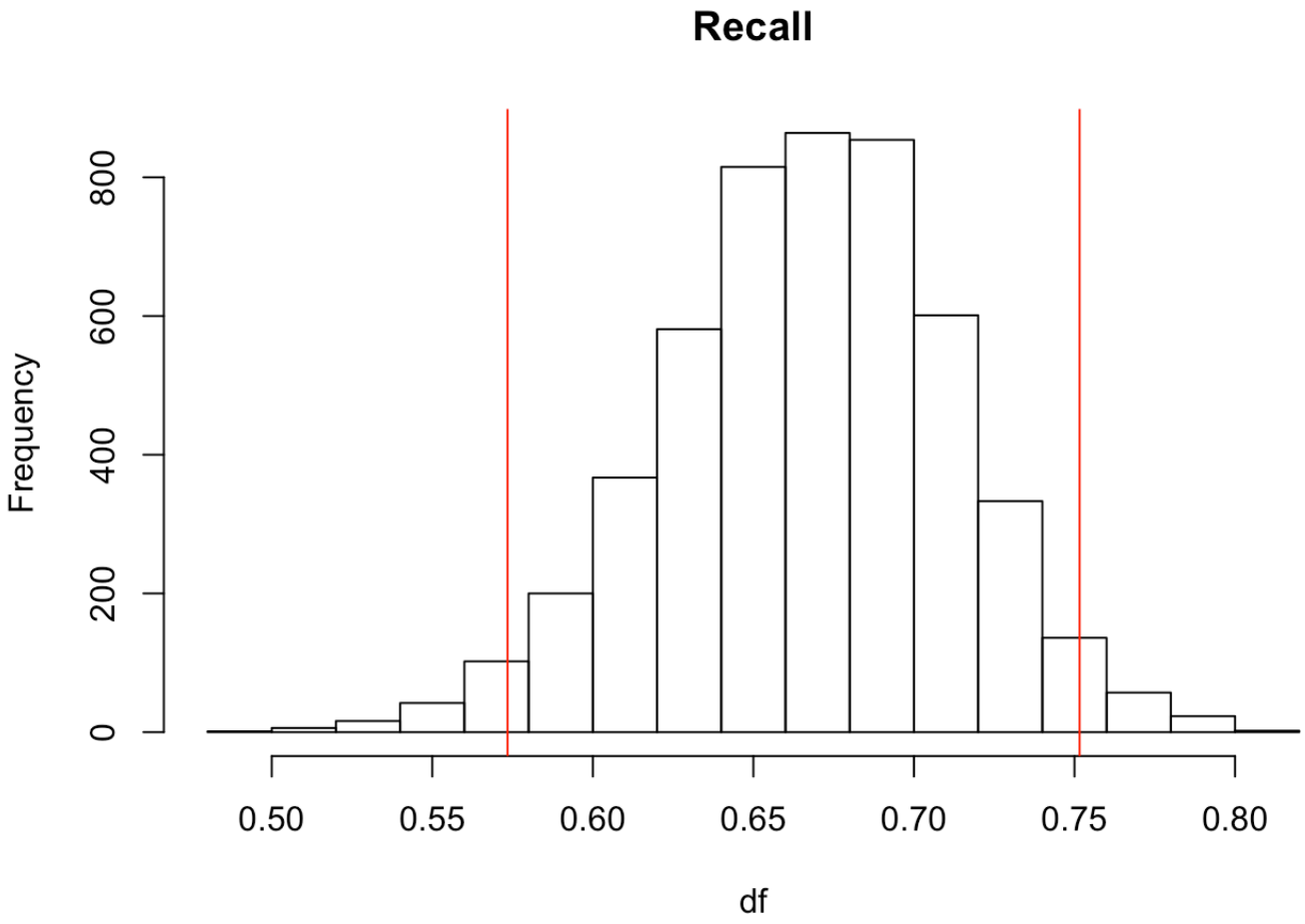

AIC
Frequency
0
200
400
600
800
1200
380
400
420
440
460
480
df

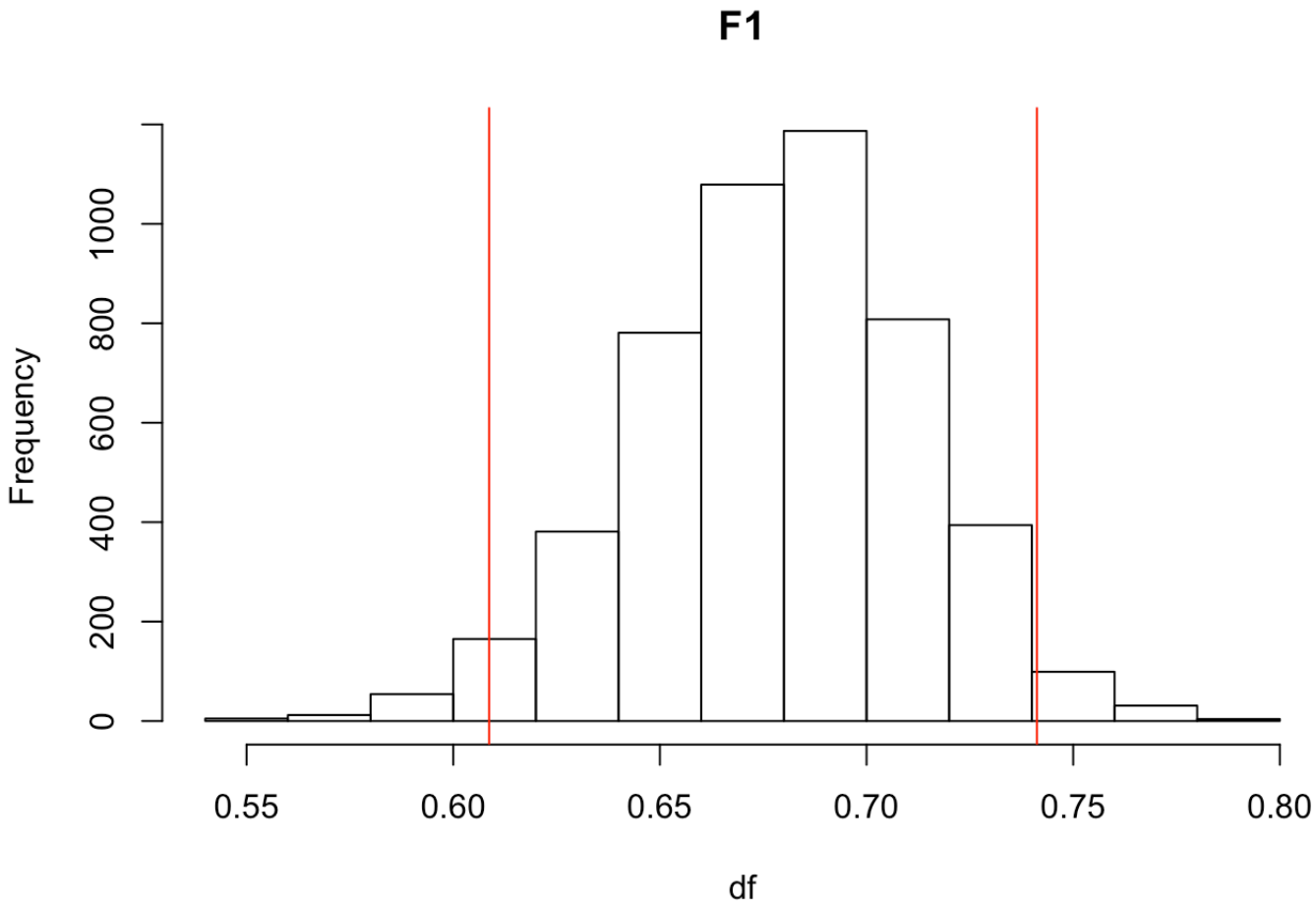

Balanced Accuracy
Frequency
0
200
400
600
800
1200
0.60
0.65
0.70
0.75
0.80
df


Sensitivity
Frequency
0
200
400
600
800
0.50
0.55
0.60
0.65
0.70
0.75
0.80
df


Specificity
Frequency
0
200
400
600
800
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
df


Precision
Frequency
0
200
400
600
800
1200
0.55
0.60
0.65
0.70
0.75
0.80
df


Recall
Frequency
0
200
400
600
800
0.50
0.55
0.60
0.65
0.70
0.75
0.80
df

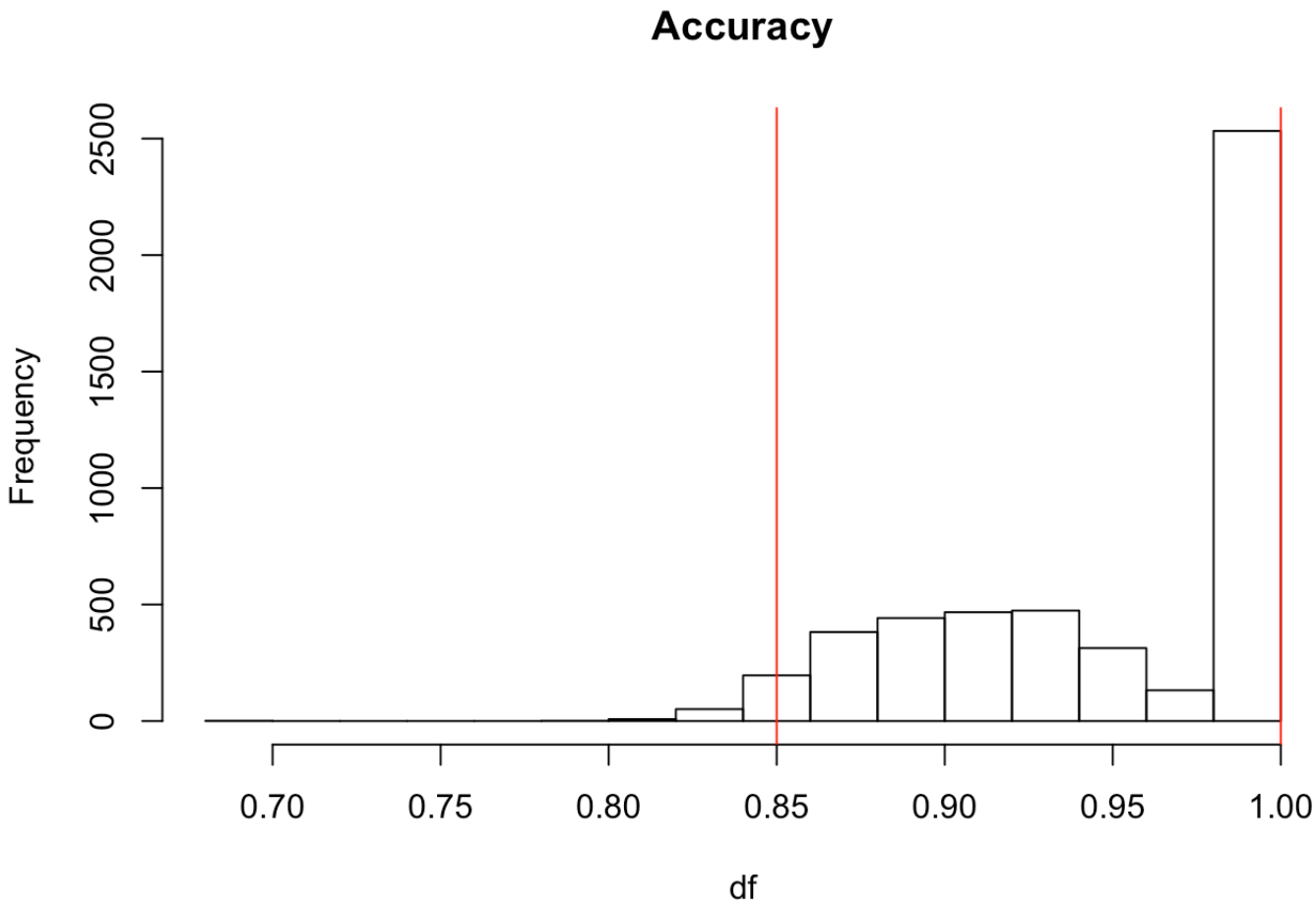


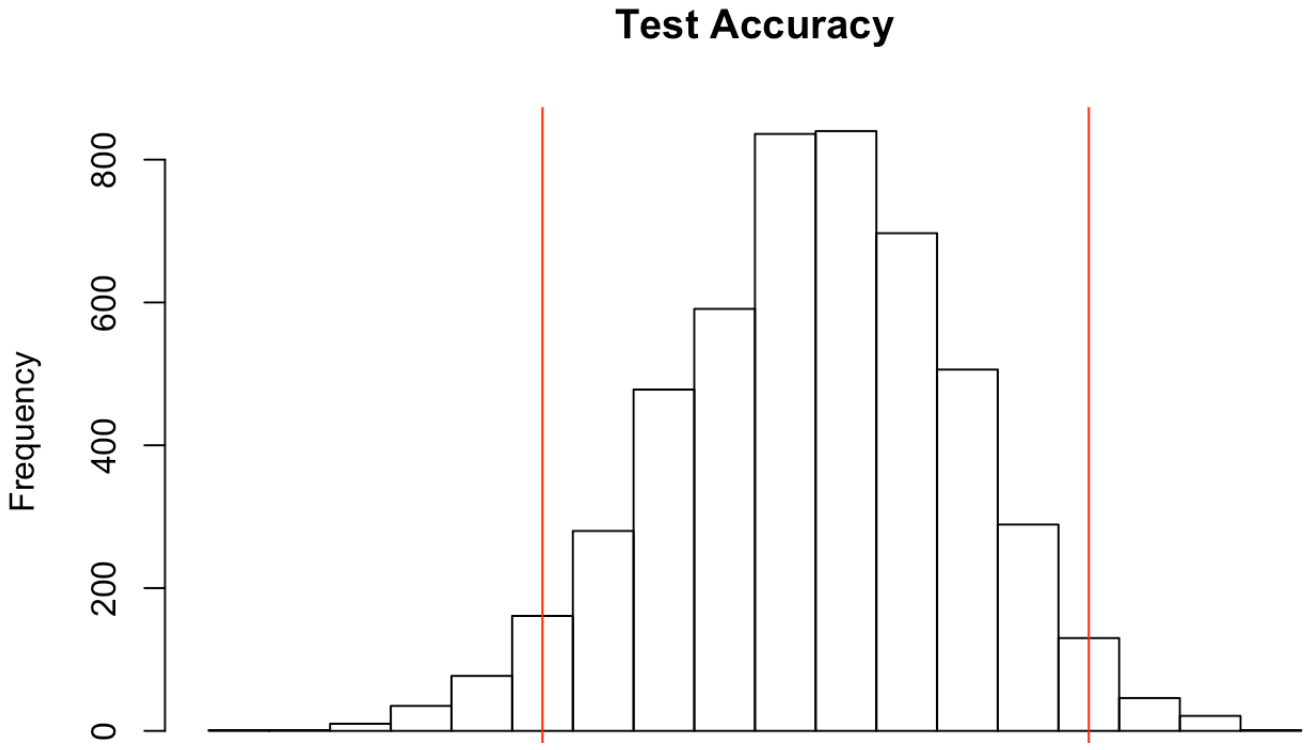

```
write.csv(cm_tb,"./final/Prosocial-Results-brain.csv", row.names = FALSE)
```

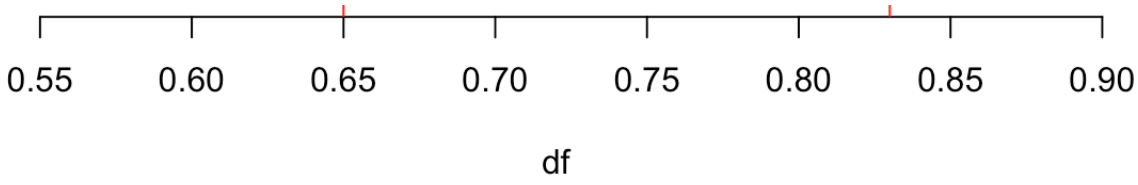
Brain + EPF

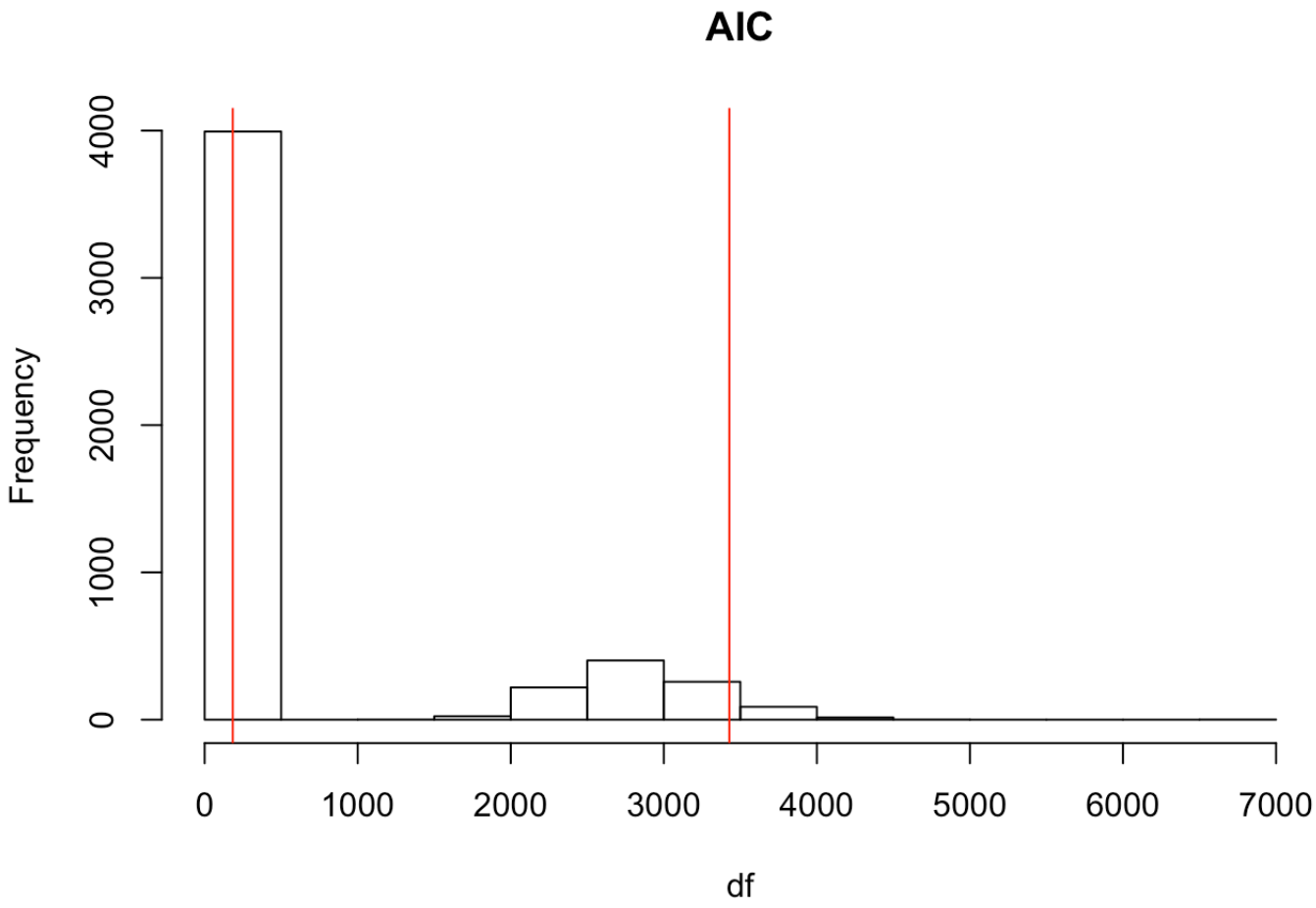
```
df <- merge(x = read.csv('cleandata.csv'), y = read.csv('brain_cb.csv'), by = "subjec
tkey", all.x = TRUE)
df <- df[complete.cases(df), ]

df<- df[(df$prosocial_parent.x  <= (3)) | (df$prosocial_parent.x  == 6), ]
df<- df[(df$prosocial_child.x <= (3)) | (df$prosocial_child.x  == 6), ]
df$prosocial_sum <- df$prosocial_parent.x+df$prosocial_child.x
df <- df[df$prosocial_sum != 9, ]
df <- df[df$prosocial_sum != 8, ]
df <- df[df$prosocial_sum != 7, ]
cut <- df[df$prosocial_sum == 6, ]
cut <- cut[cut$prosocial_child.x != 3, ]
df <- subset(df, !(subjectkey %in% cut$subjectkey))
df$y <- ifelse(df$prosocial_sum <= (6), 0, 1)

df <- subset(df, select=-c(X, prosocial_parent.x, prosocial_child.x, interview_date,s
ubjectkey, aggressive_sumscore.x, interview_age, aggressive_sumscore.y, prosocial_chi
ld.y, prosocial_parent.y,prosocial_sum))
#write.csv(df, './final/prosocialEPFbrain.csv', row.names = FALSE)
```

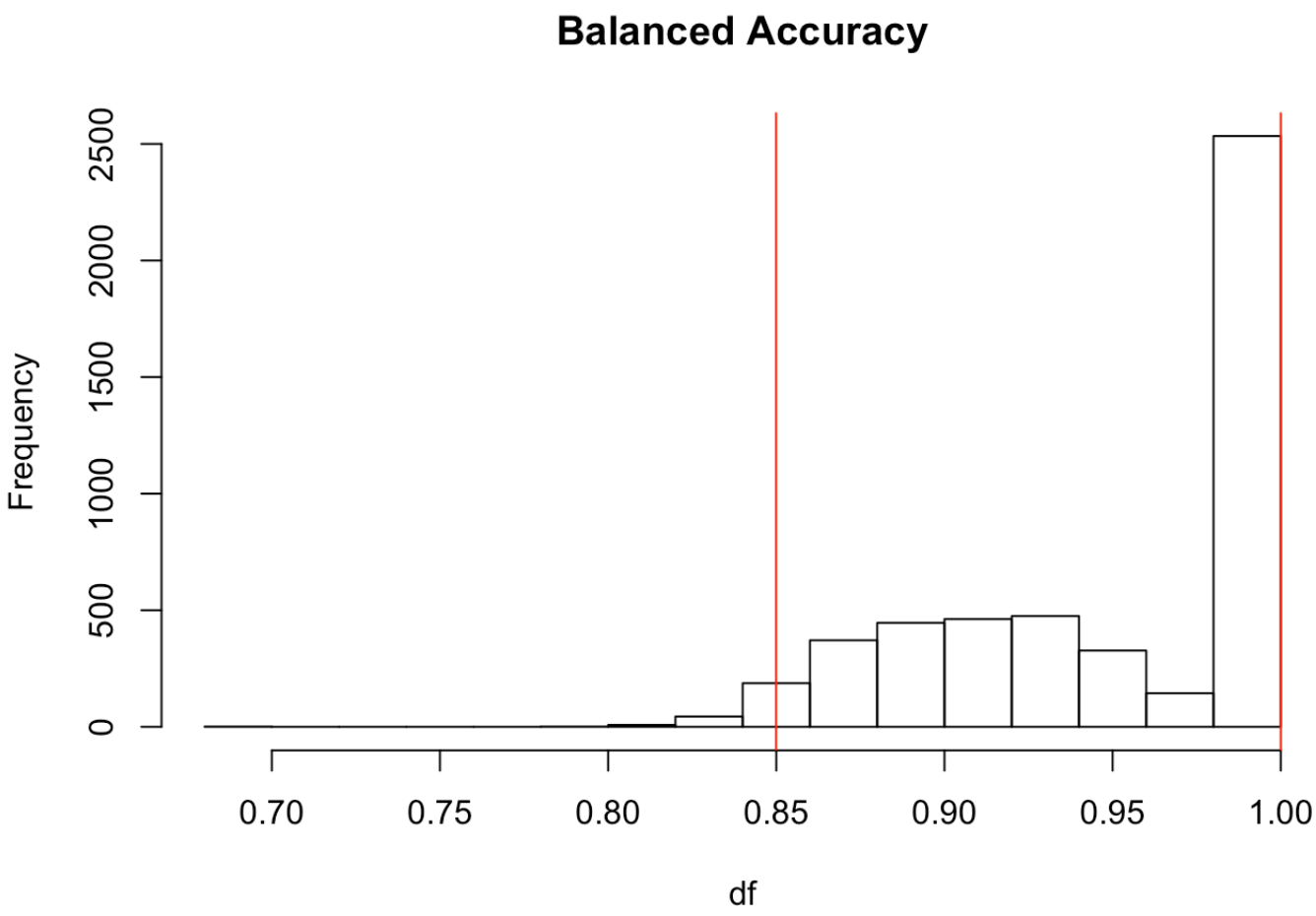
```
cm_tb<-bootstrap_glm(df, T=GlobalT, filesname = './final/Prosocial-Attributes-brain+E
PF.csv')
```

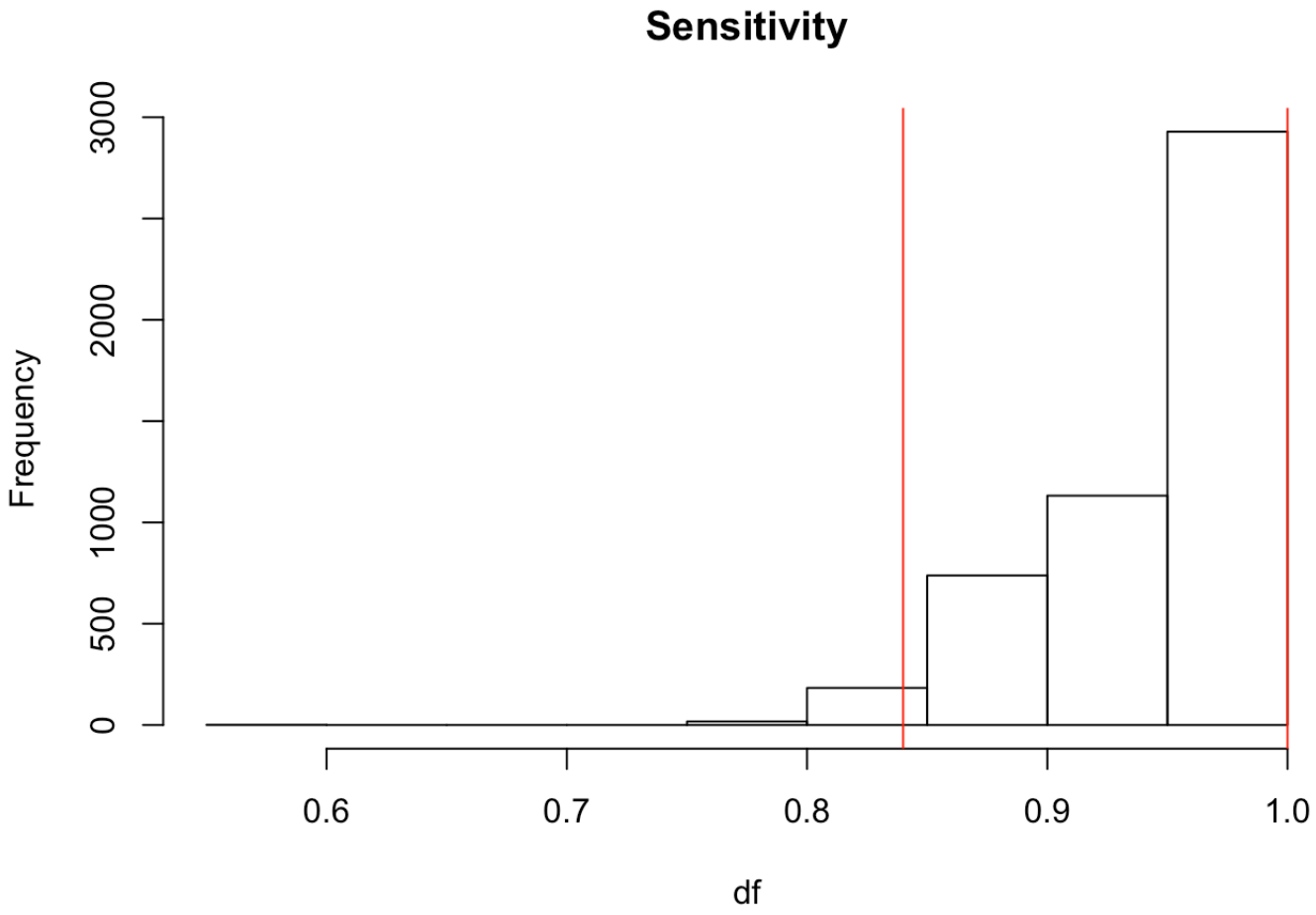

Accuracy
Frequency
0
500
1000
1500
2000
2500
0.70
0.75
0.80
0.85
0.90
0.95
1.00
df

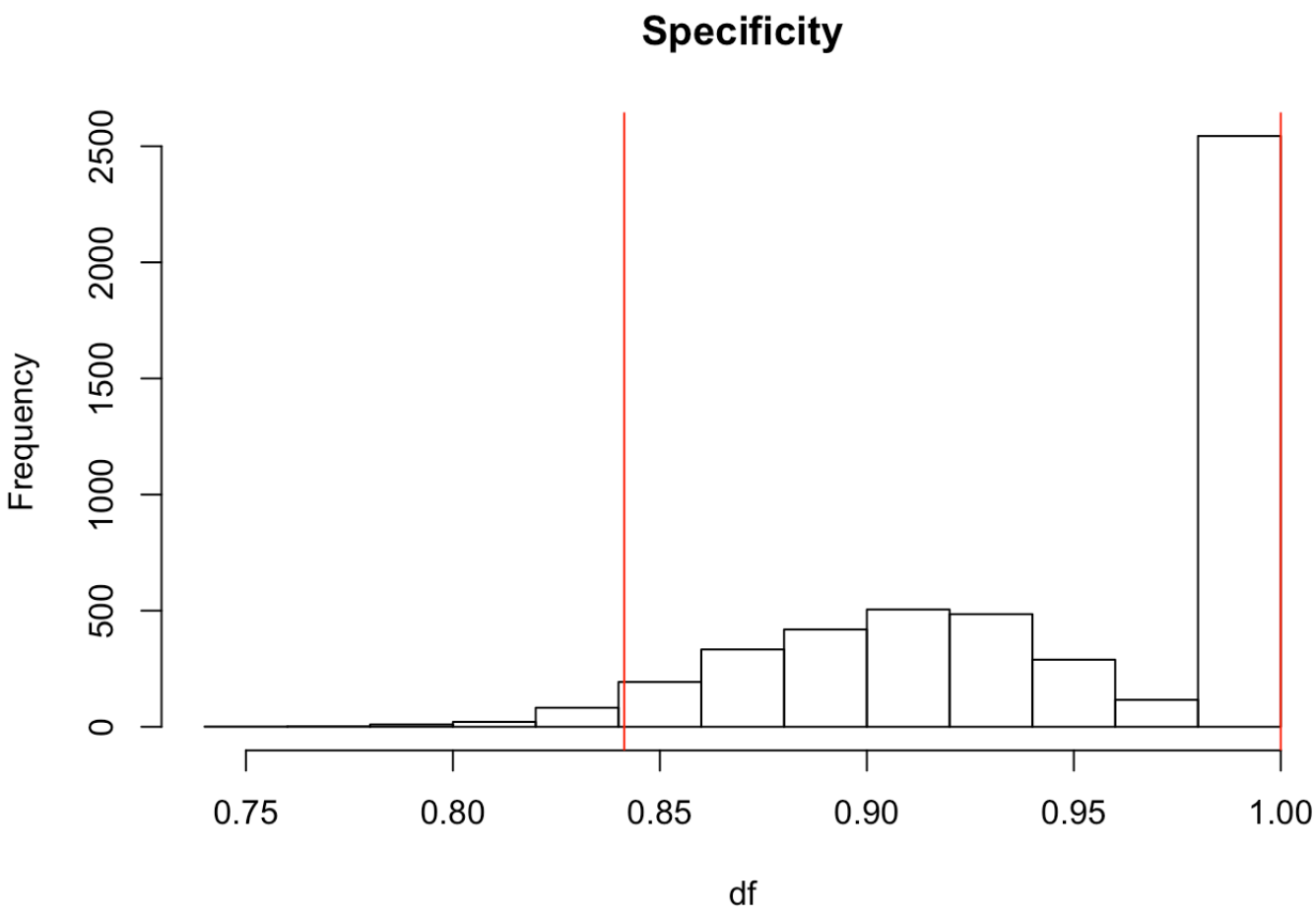

Test Accuracy
Frequency
0
200
400
600
800

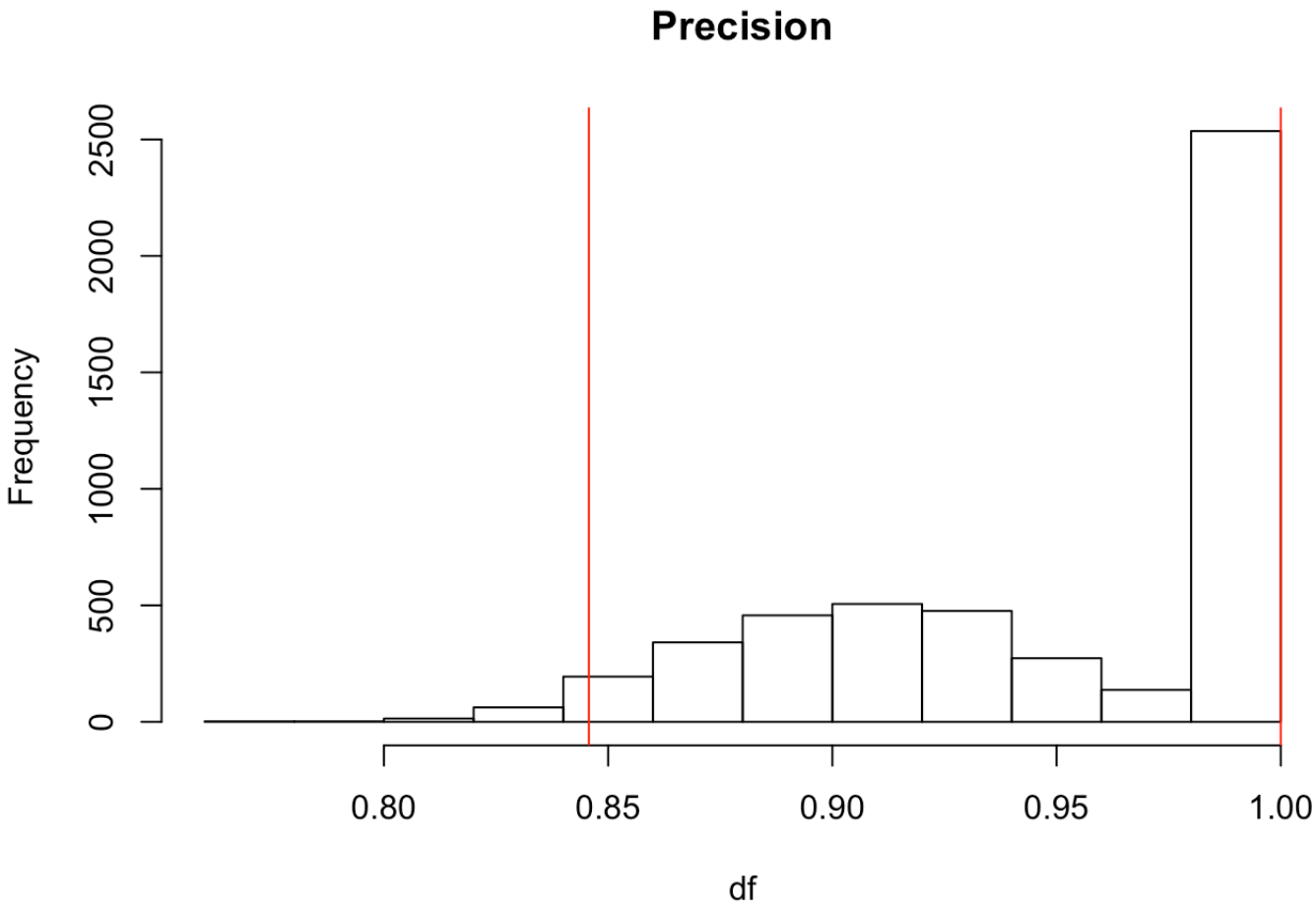

0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
df

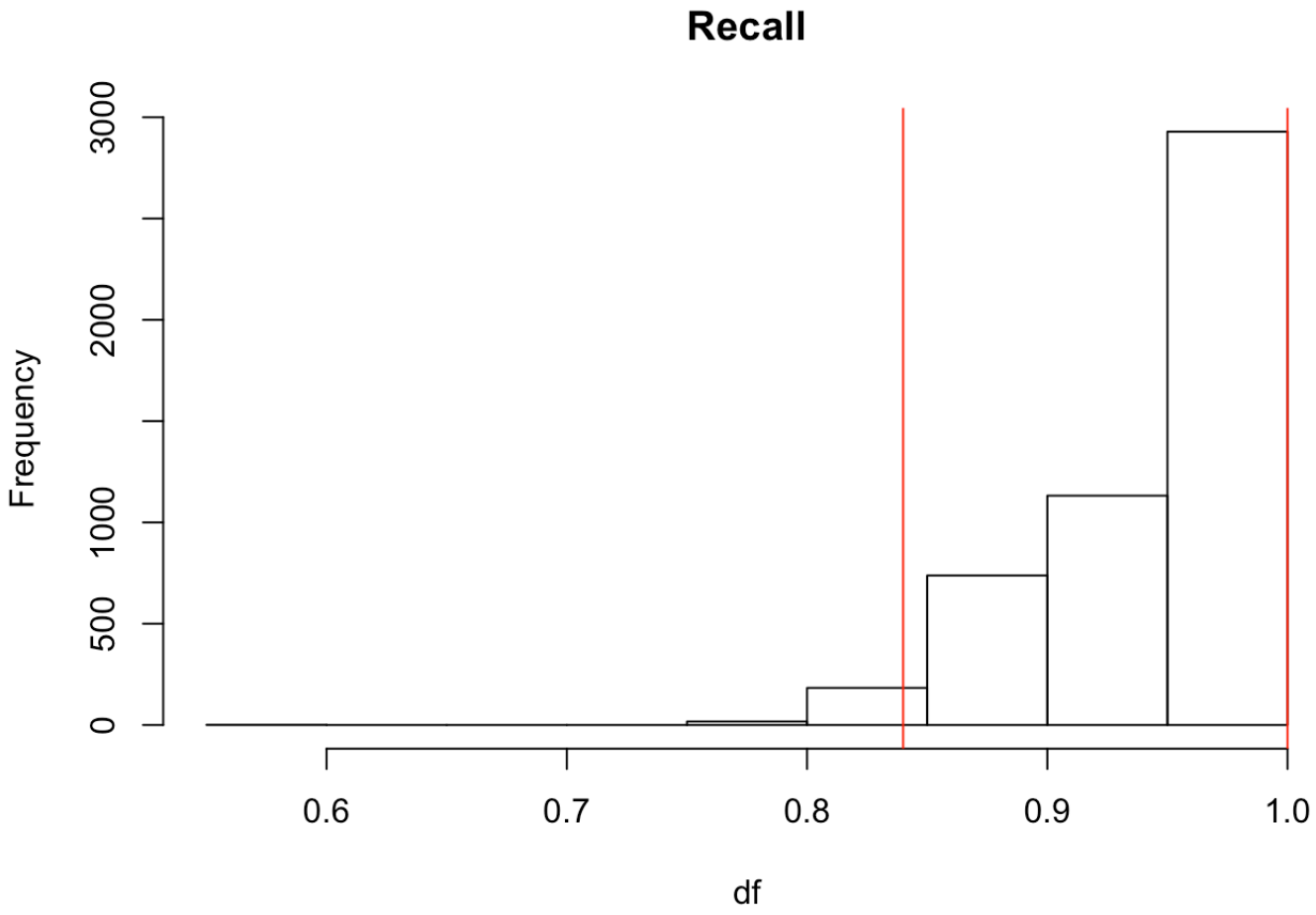

AIC
Frequency
0
1000
2000
3000
4000
0
1000
2000
3000
4000
5000
6000
7000
df

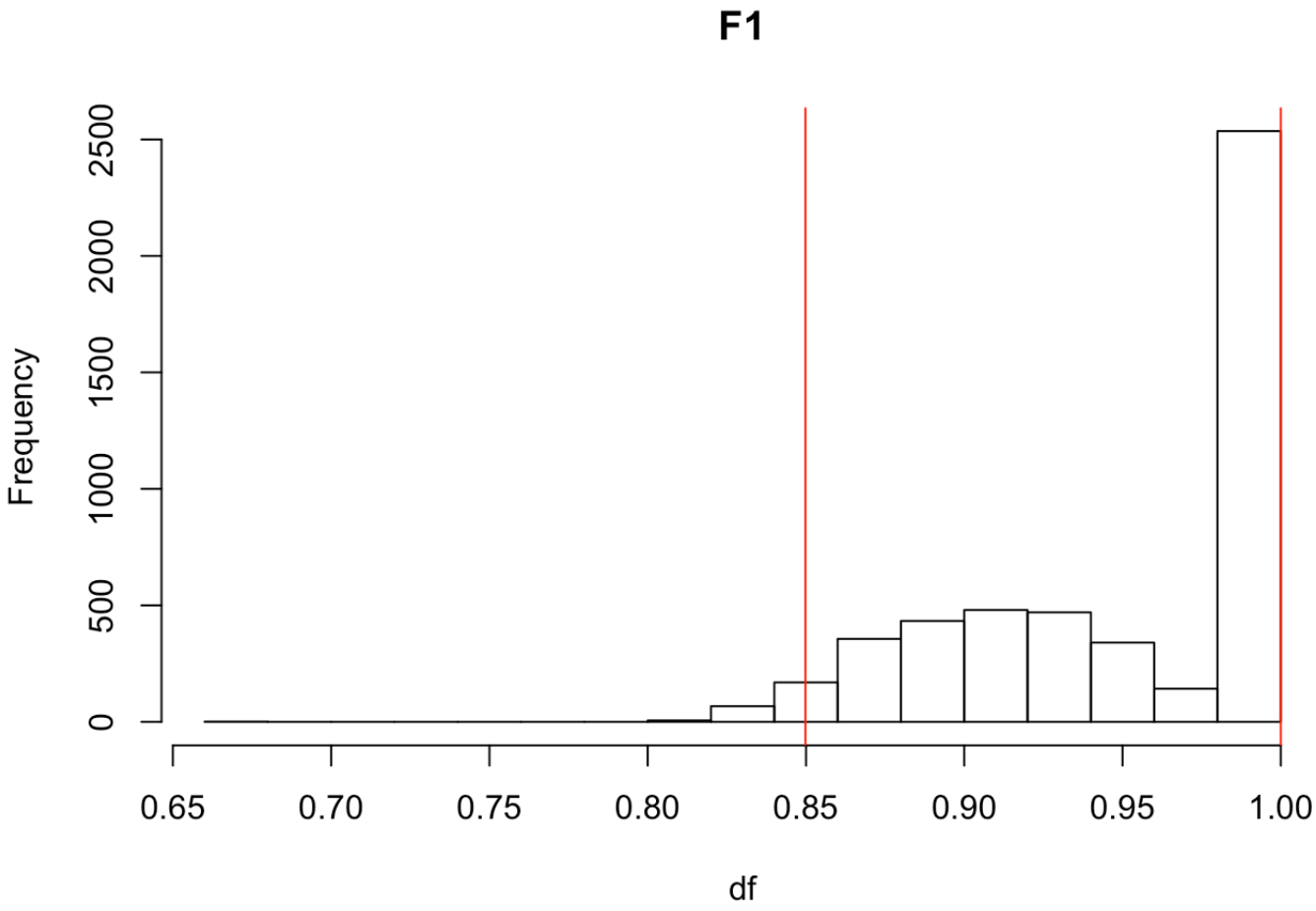

Balanced Accuracy
Frequency
0
500
1000
1500
2000
2500
0.70
0.75
0.80
0.85
0.90
0.95
1.00
df


Sensitivity
Frequency
0
500
1000
2000
3000
0.6
0.7
0.8
0.9
1.0
df


Specificity
Frequency
0
500
1000
1500
2000
2500
0.75
0.80
0.85
0.90
0.95
1.00
df


Precision
Frequency
0
500
1000
1500
2000
2500
0.80
0.85
0.90
0.95
1.00
df


Recall
Frequency
0
500
1000
2000
3000
0.6
0.7
0.8
0.9
1.0
df




```
write.csv(cm_tb,"./final/Prosocial-Results-brain+EPF.csv", row.names = FALSE)
```