

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, filename, filename2, filename3)
{
  names <- c("Sensitivity", "Specificity", "Pos Pred Value", "Neg Pred Value", "Precision", "Recall", "F1", "Prevalence", "Detection Rate", "Detection Prevalence", "Balanced Accuracy", 'Accuracy', 'AIC', 'Converged', 'Test Accuracy')
  cm_tb <- data.frame(matrix(ncol = length(names), nrow = 0))

  names <- c("Test-Sensitivity", "Test-Specificity", "Test-Pos Pred Value", "Test-Neg Pred Value", "Test-Precision", "Test-Recall", "Test-F1", "Test-Prevalence", "Test-Detection Rate", "Test-Detection Prevalence", "Test-Balanced Accuracy", 'Test-Accuracy')
  tcm_tb <- data.frame(matrix(ncol = length(names), nrow = 0))

  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)
  b_tb <- as.data.frame(t(coef(summary(glm.fit))[,1]))
  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)
    tcm <- 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$converged)
v<-append(v, tcm$overall[1])

v2 <- t(tcm$byClass)
v2<-append(v2, tcm$overall[1])

cm_tb <- rbind(cm_tb, as.data.frame(t(v)))
tcm_tb <- rbind(tcm_tb, as.data.frame(t(v2)))
s_tb <- rbind(s_tb, as.data.frame(t(coef(summary(glm.fit))[,4])))
b_tb <- rbind(b_tb, as.data.frame(t(coef(summary(glm.fit))[,1])))

}
colnames(tcm_tb) <- names
s_tb <- s_tb[2:nrow(s_tb),]
b_tb <- b_tb[2:nrow(b_tb),]

write.csv(s_tb, file=filename, row.names = FALSE)
write.csv(b_tb, file=filename3, row.names = FALSE)
write.csv(tcm_tb, file=filename2, row.names = FALSE)

colnames(cm_tb) <- c("Sensitivity", "Specificity", "Pos Pred Value", "Neg Pred Value", "Precision", "Recall", "F1", "Prevalence", "Detection Rate", "Detection Prevalence", "Balanced Accuracy", 'Accuracy', 'AIC', 'Converged', 'Test Accuracy')
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)
}

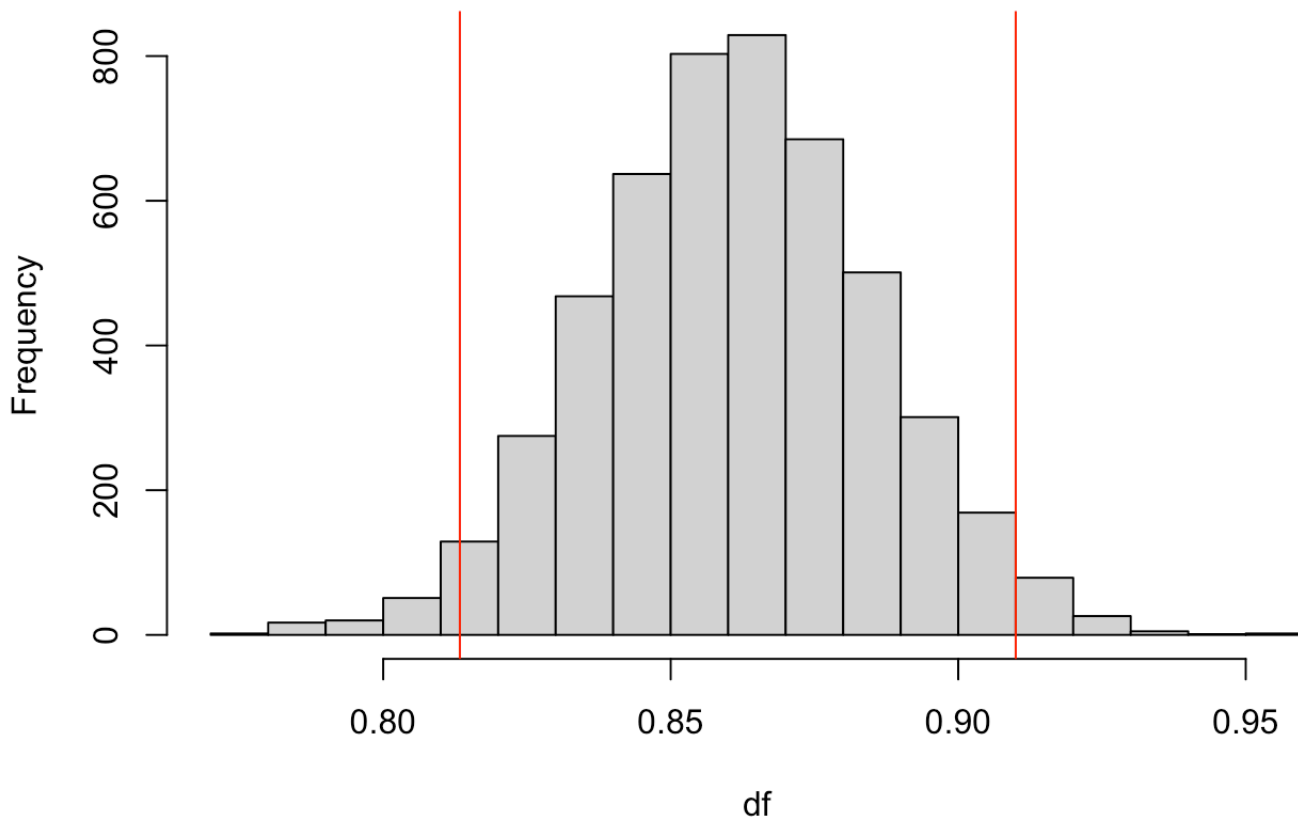
```

EPF

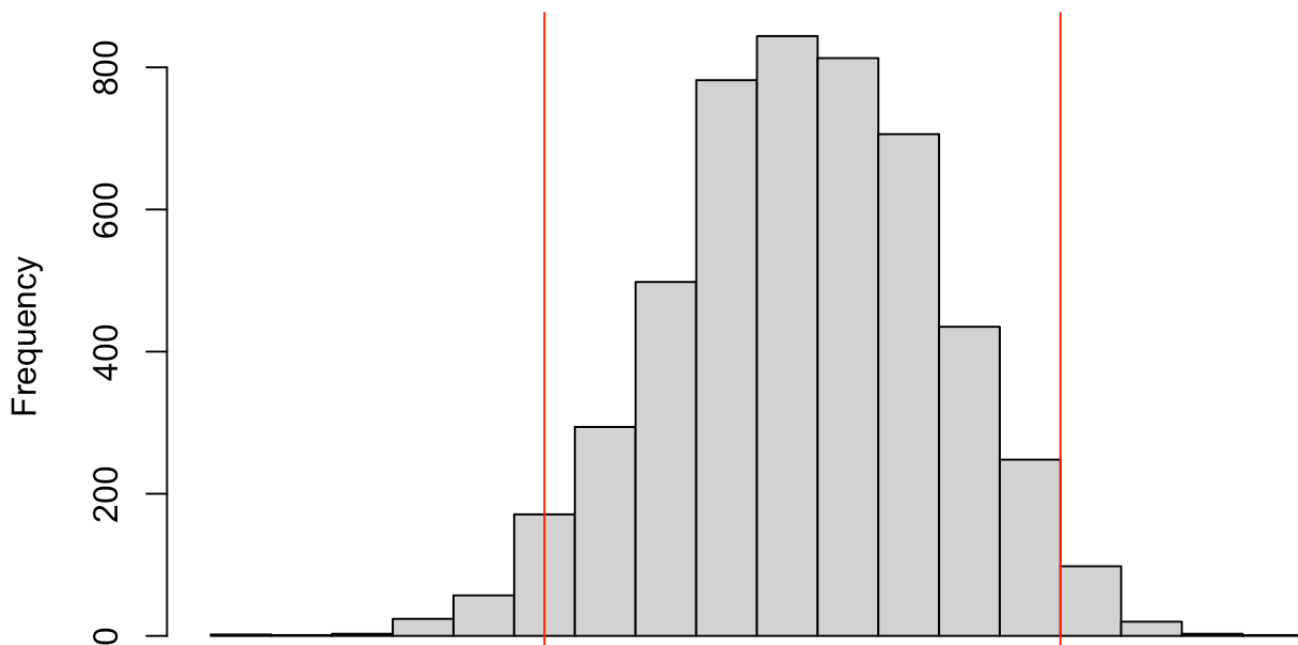
```
df<-prosocial_preprocessing(read.csv('cleandata.csv'))  
df <- subset(df, select=-c(X,aggressive_sumscore , prosocial_child, prosocial_parent,  
interview_date, interview_age, subjectkey, prosocial_sum))
```

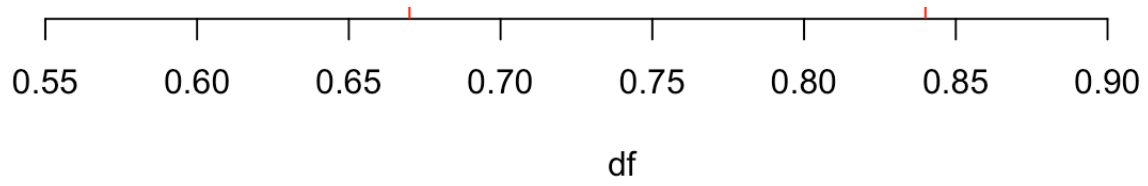
```
cm_tb<-bootstrap_glm(df, T=GlobalT, filename = './Prosocial-A-EPF.csv', filename2 =  
 './T-Prosocial-A-EPF.csv', filename3 = './Prosocial-A-Beta-EPF.csv')
```

Accuracy

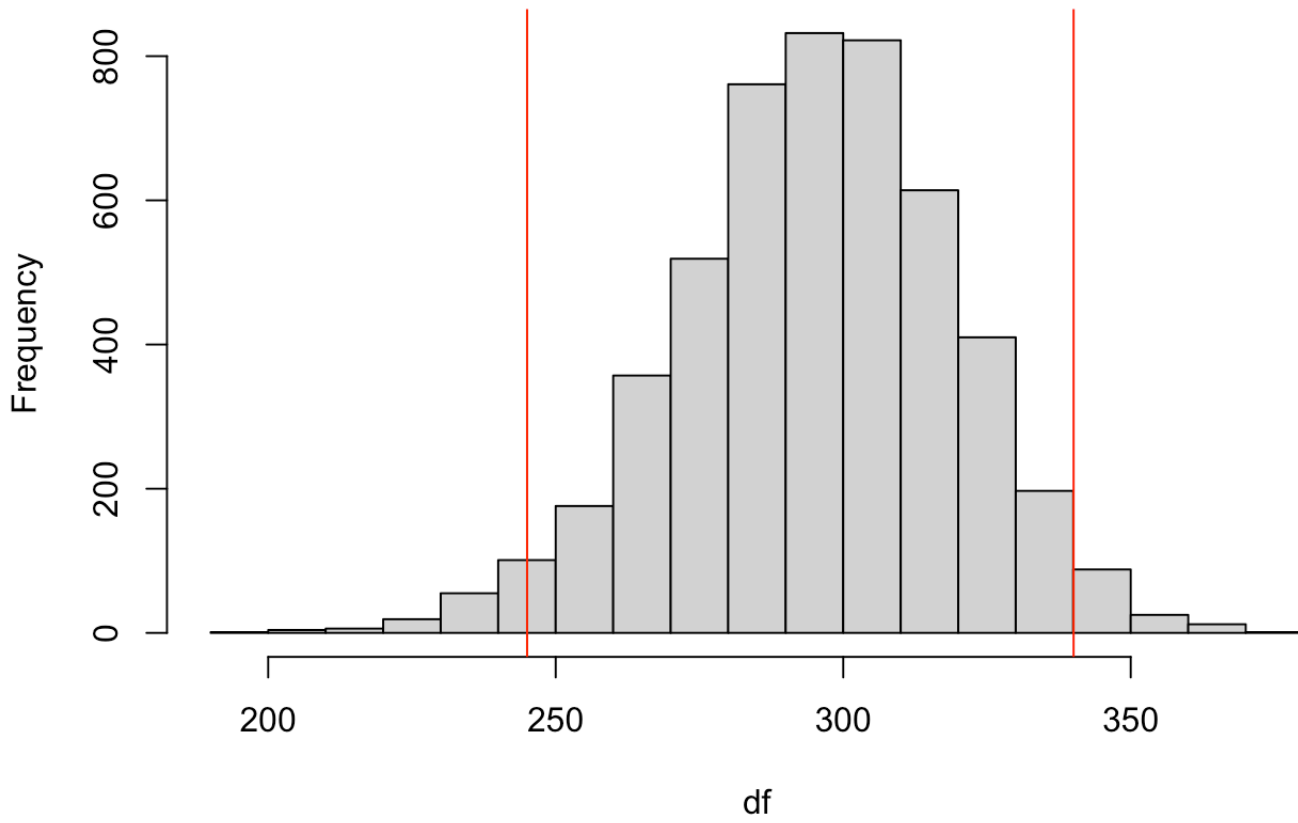


Test Accuracy

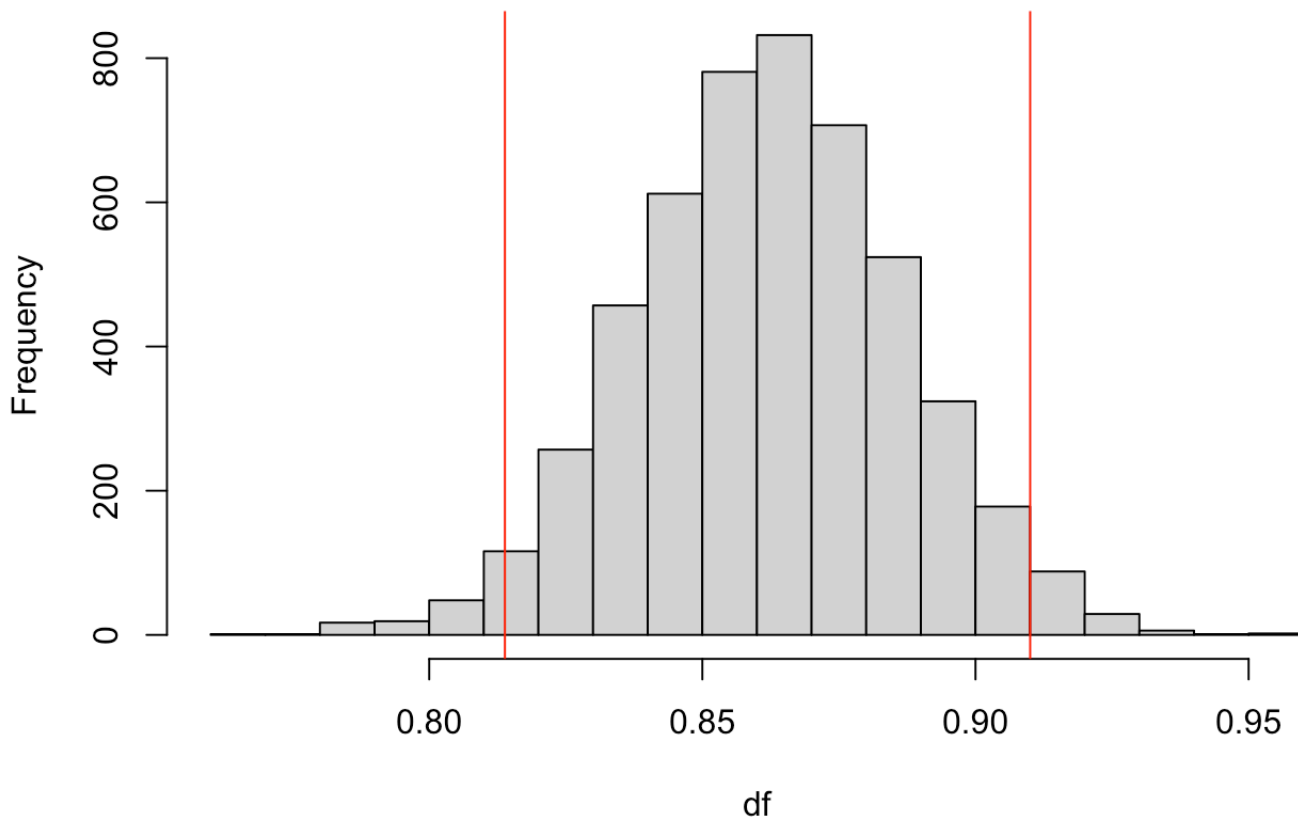




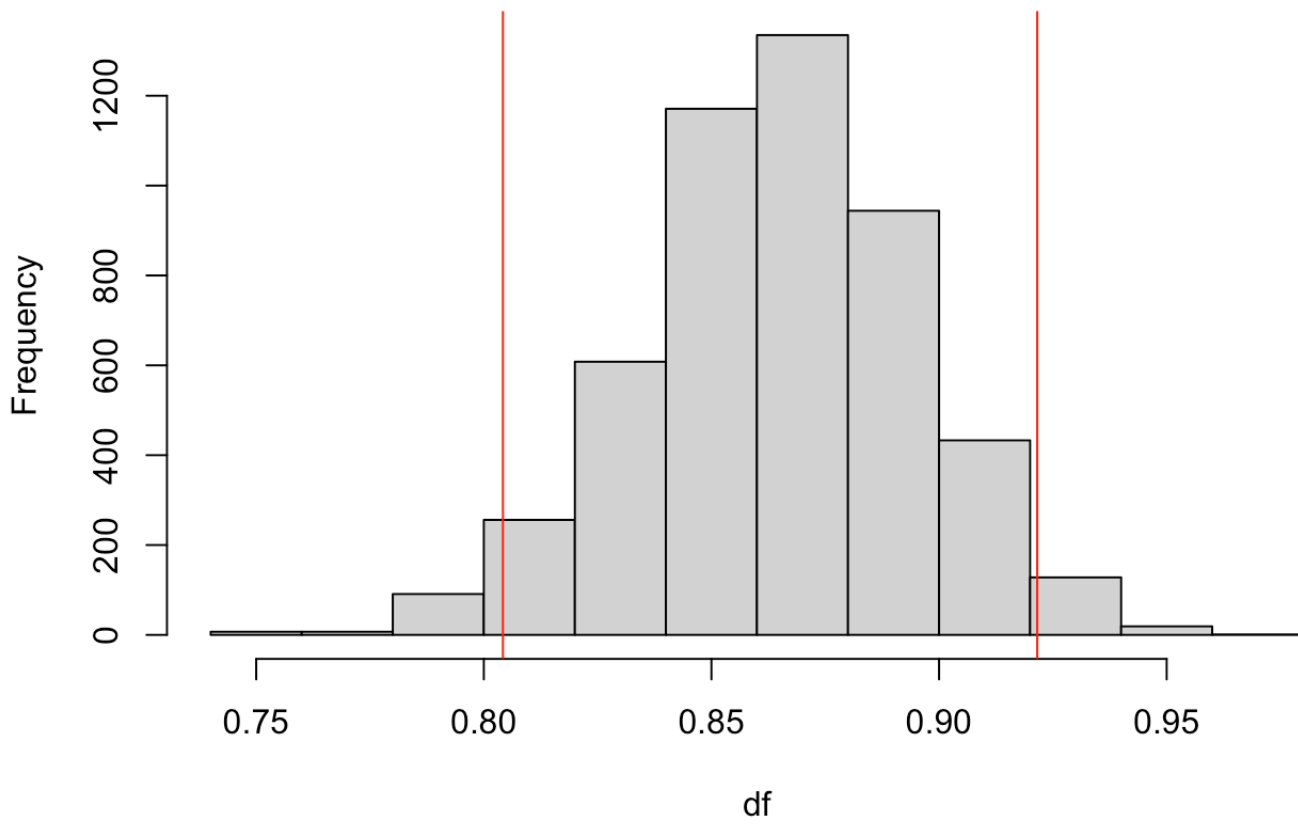
AIC



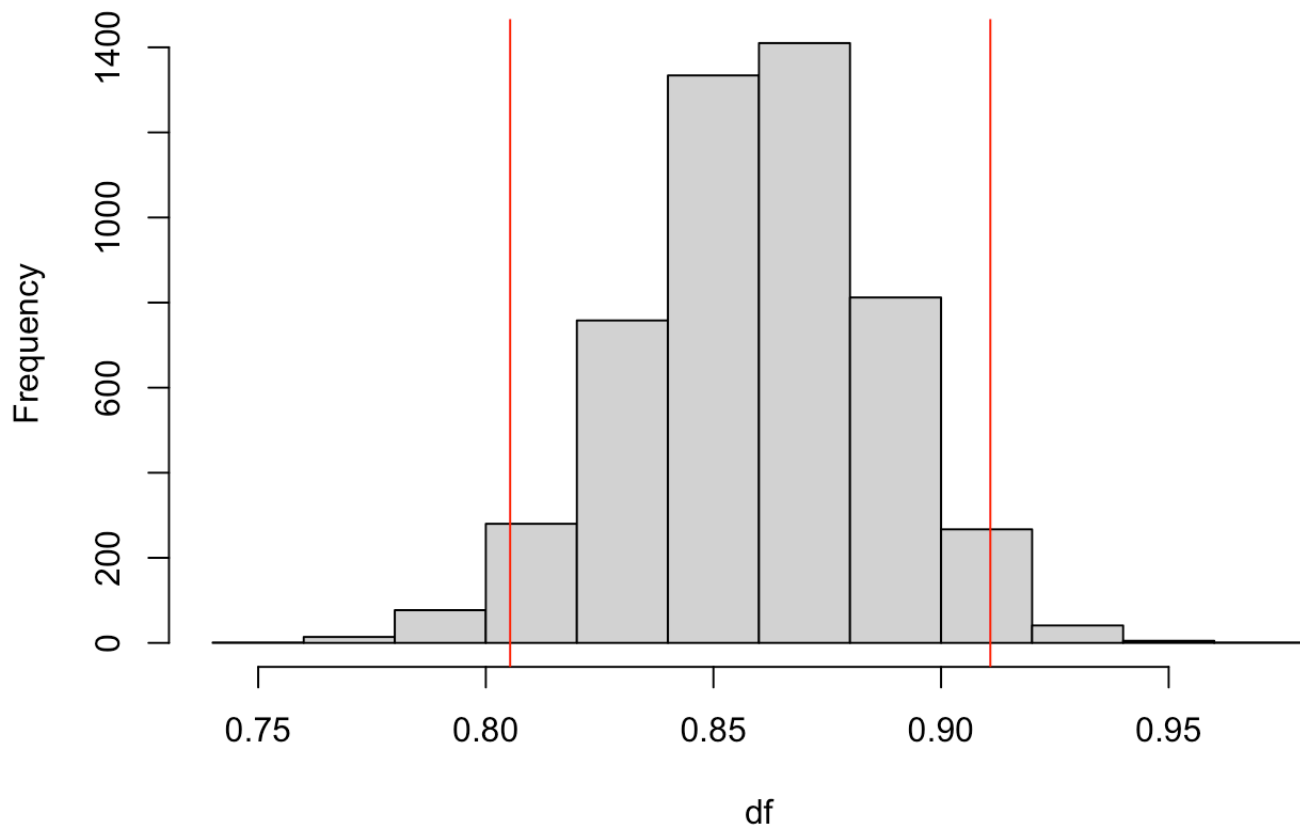
Balanced Accuracy



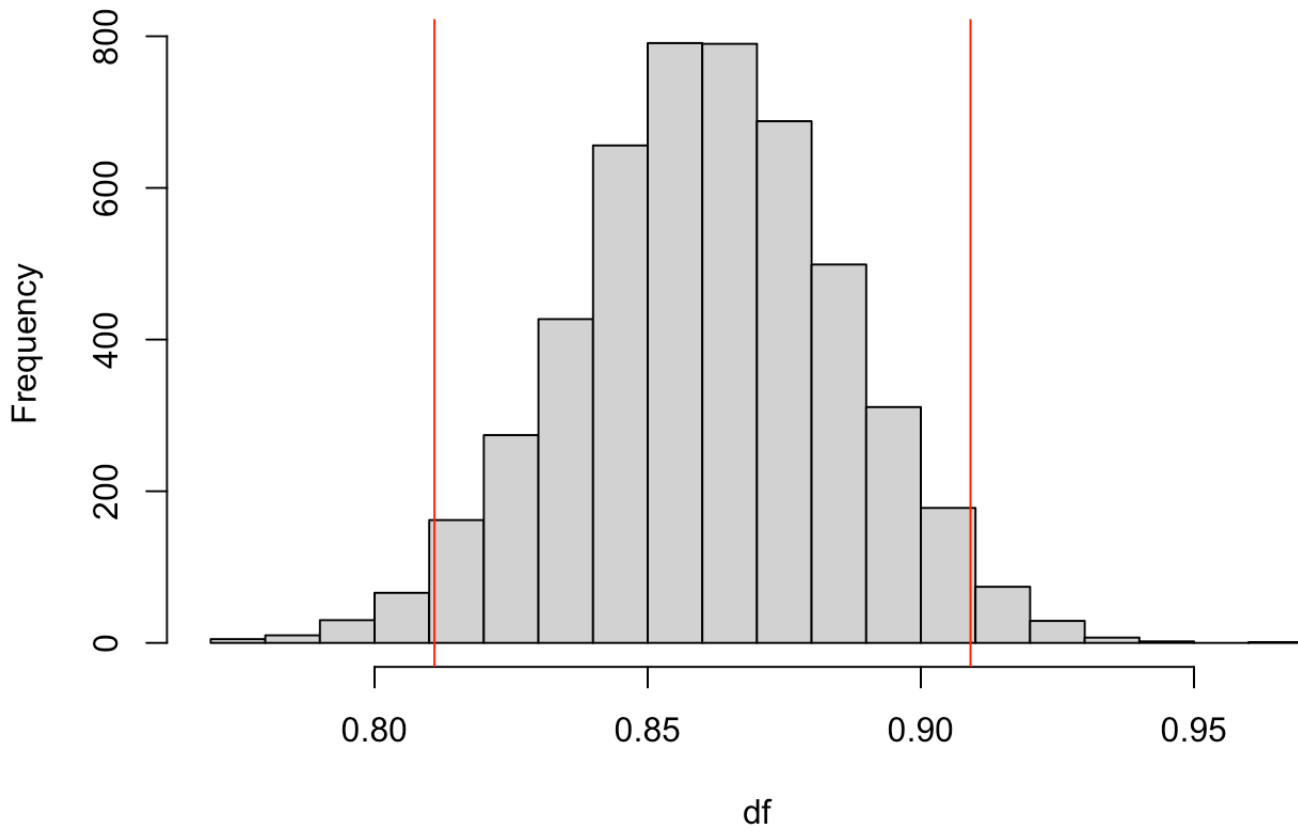
Sensitivity



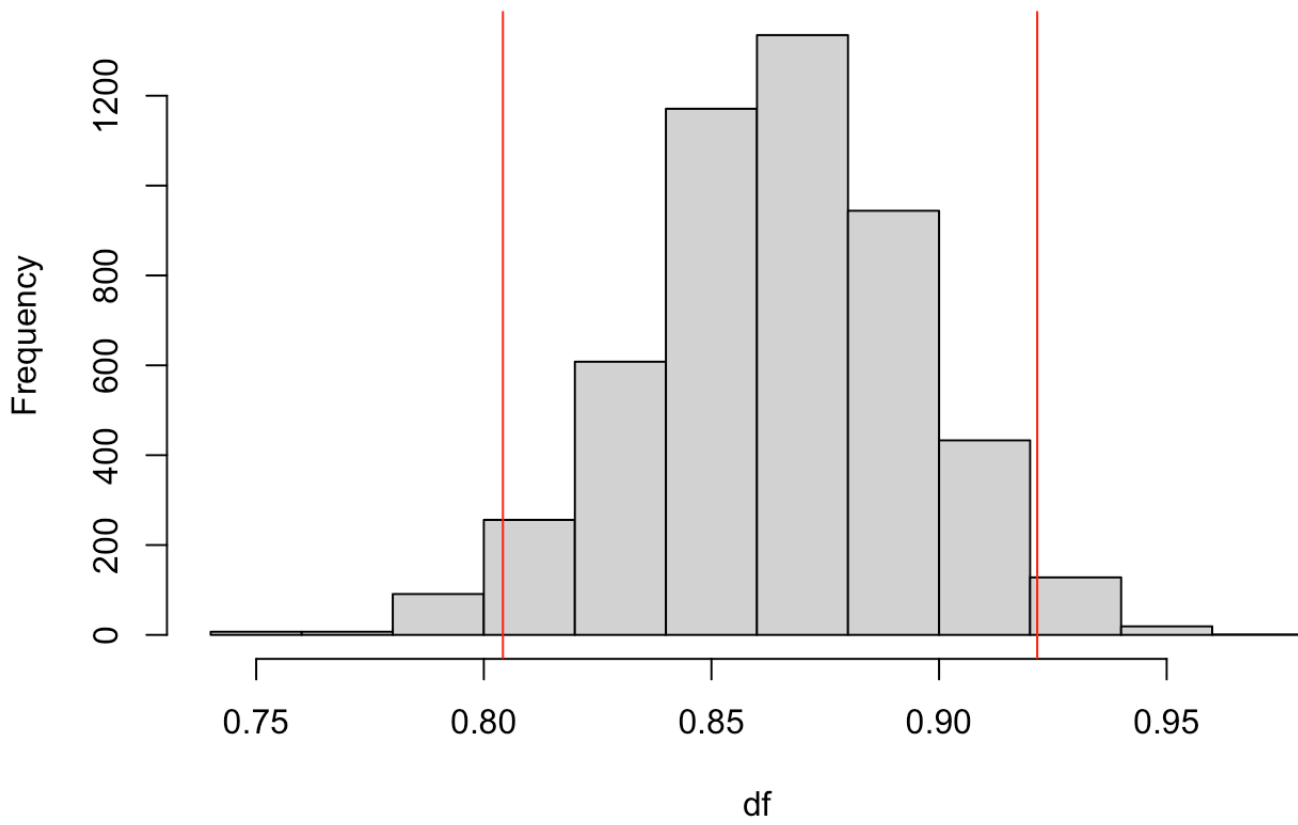
Specificity

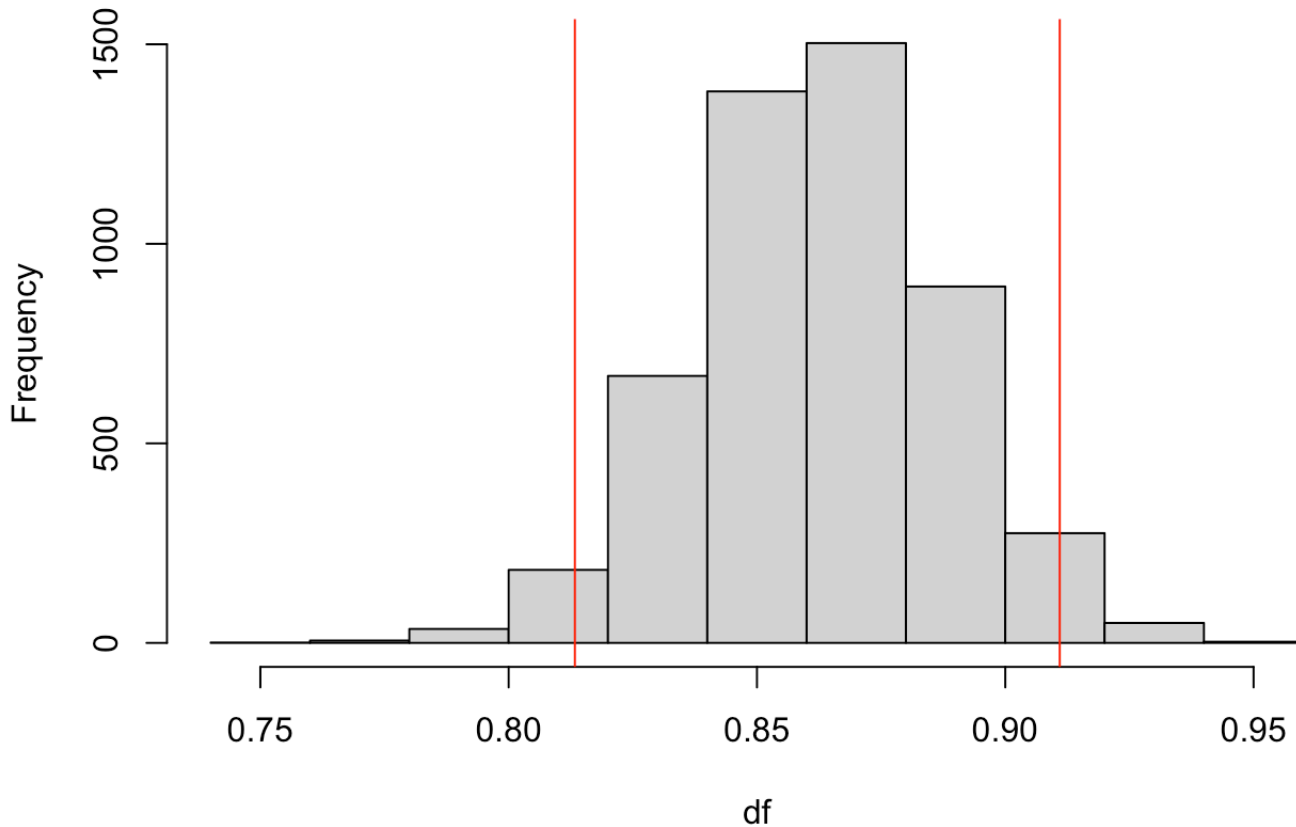


Precision



Recall



F1

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

Brain

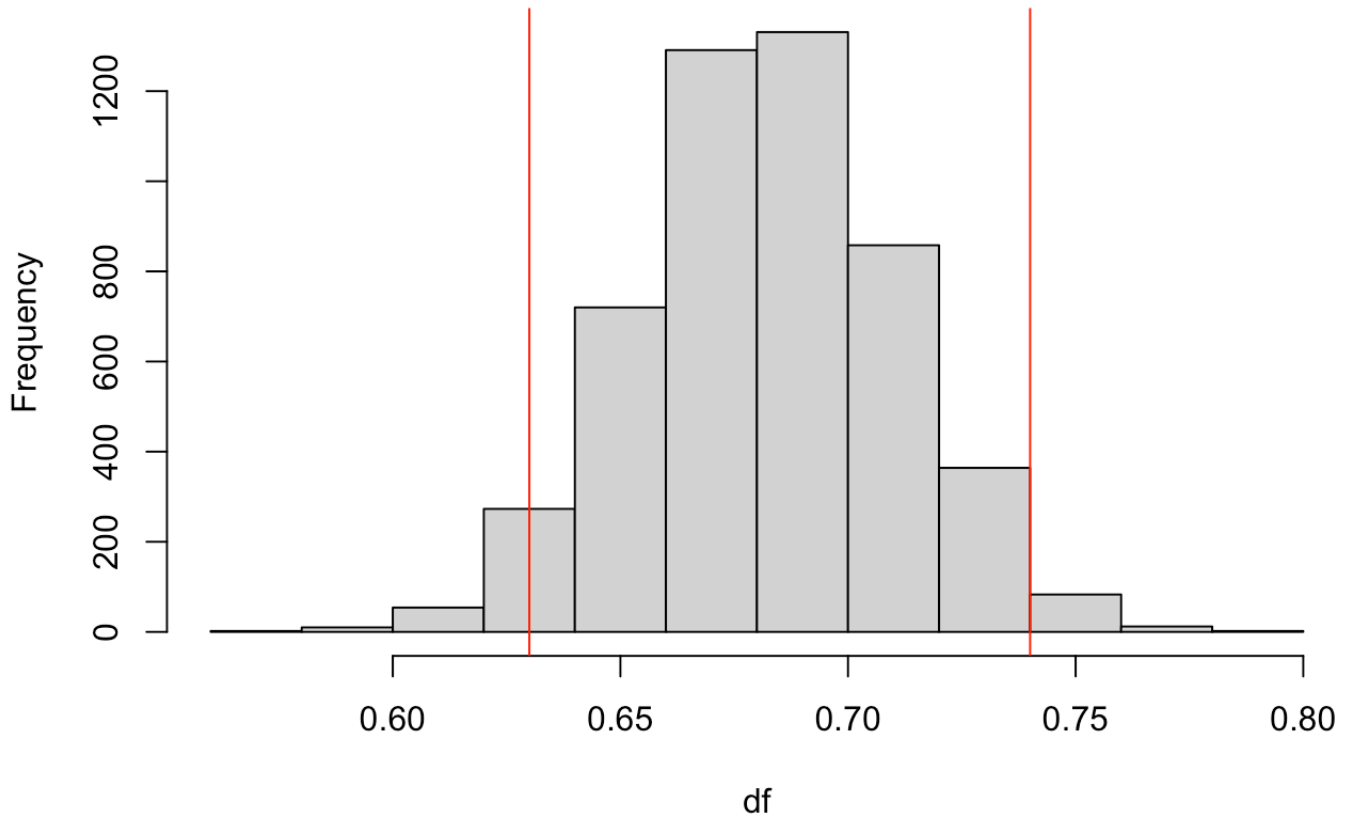
Attribute selection

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

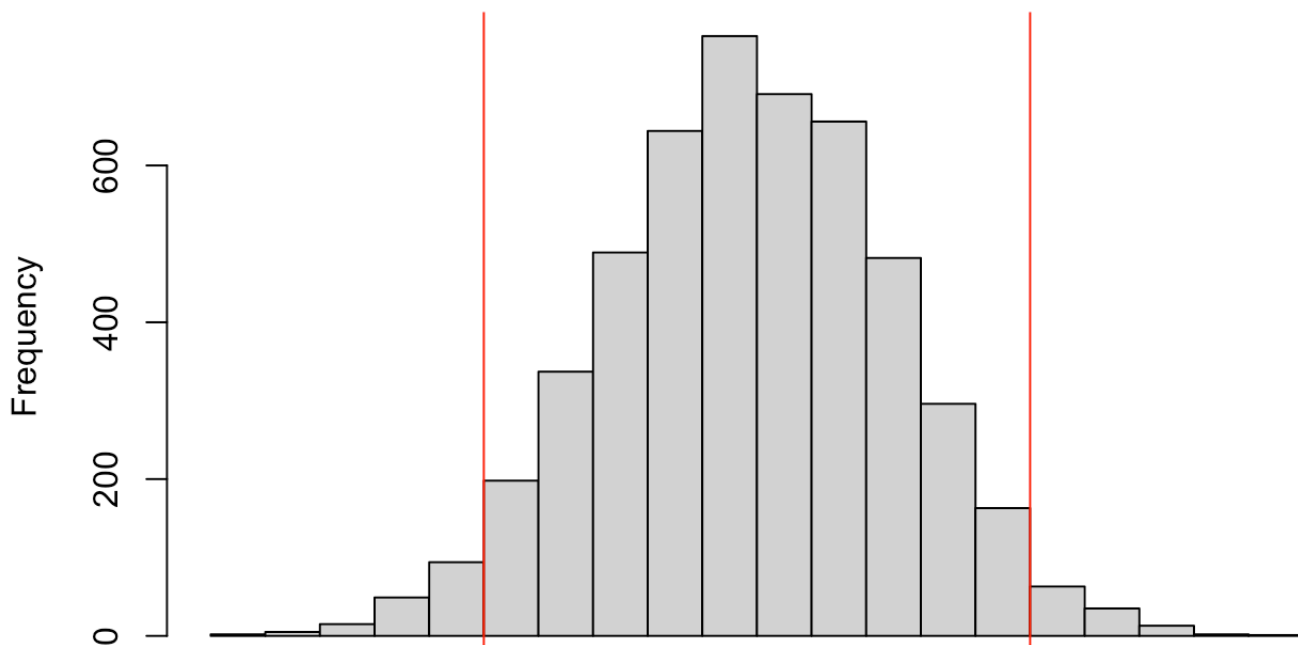
```
#singleglm(df)
```

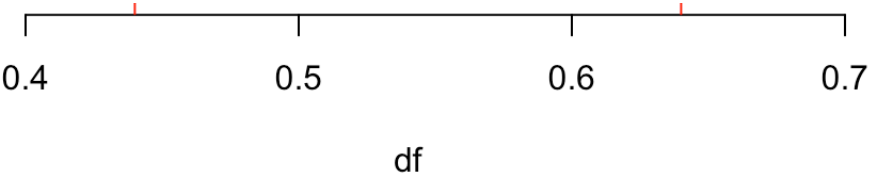
```
cm_tb<-bootstrap_glm(df, T=GlobalT, filename = './Prosocial-A-brain.csv', filename2 = './T-Prosocial-A-brain.csv', filename3 = './Prosocial-A-Beta-brain.csv')
```


Accuracy

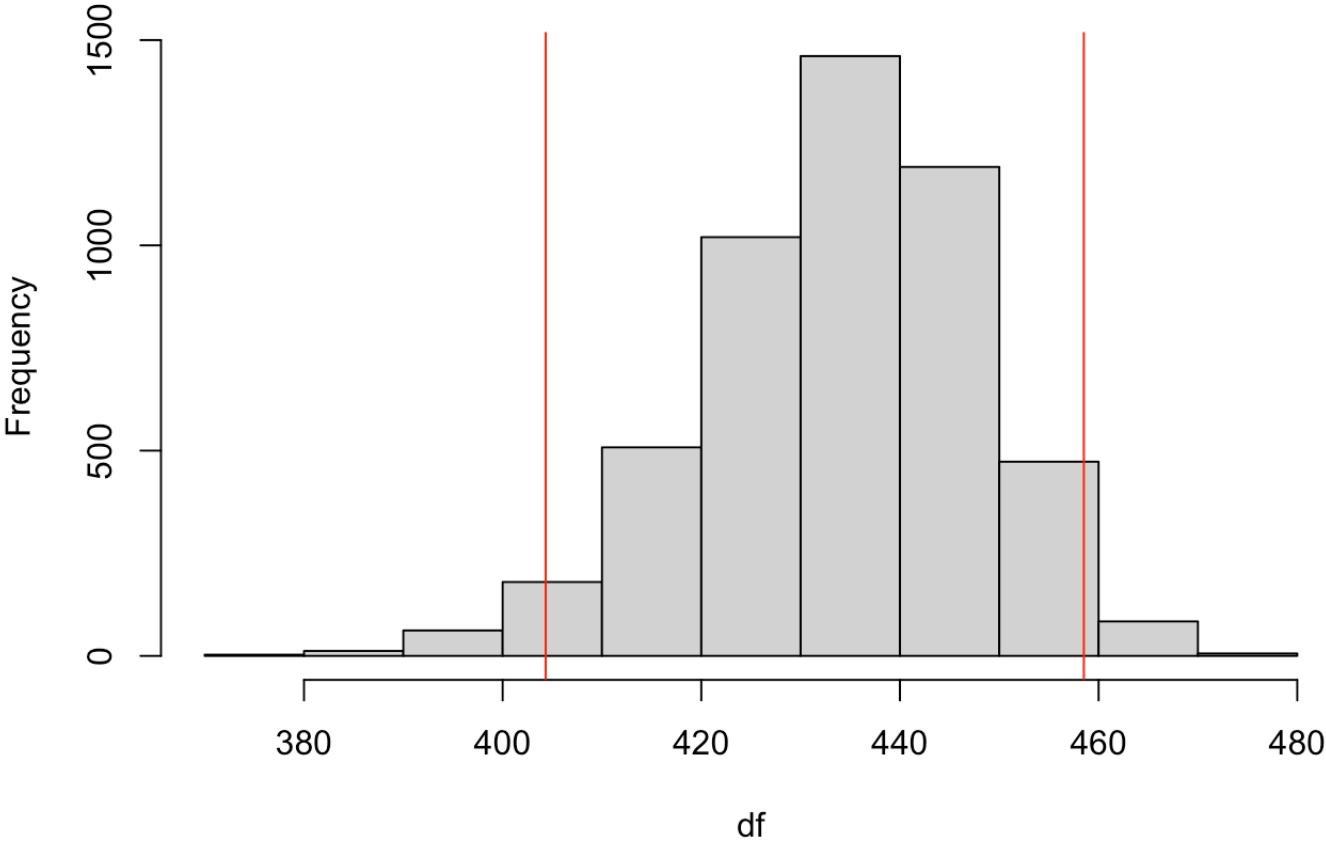


Test Accuracy

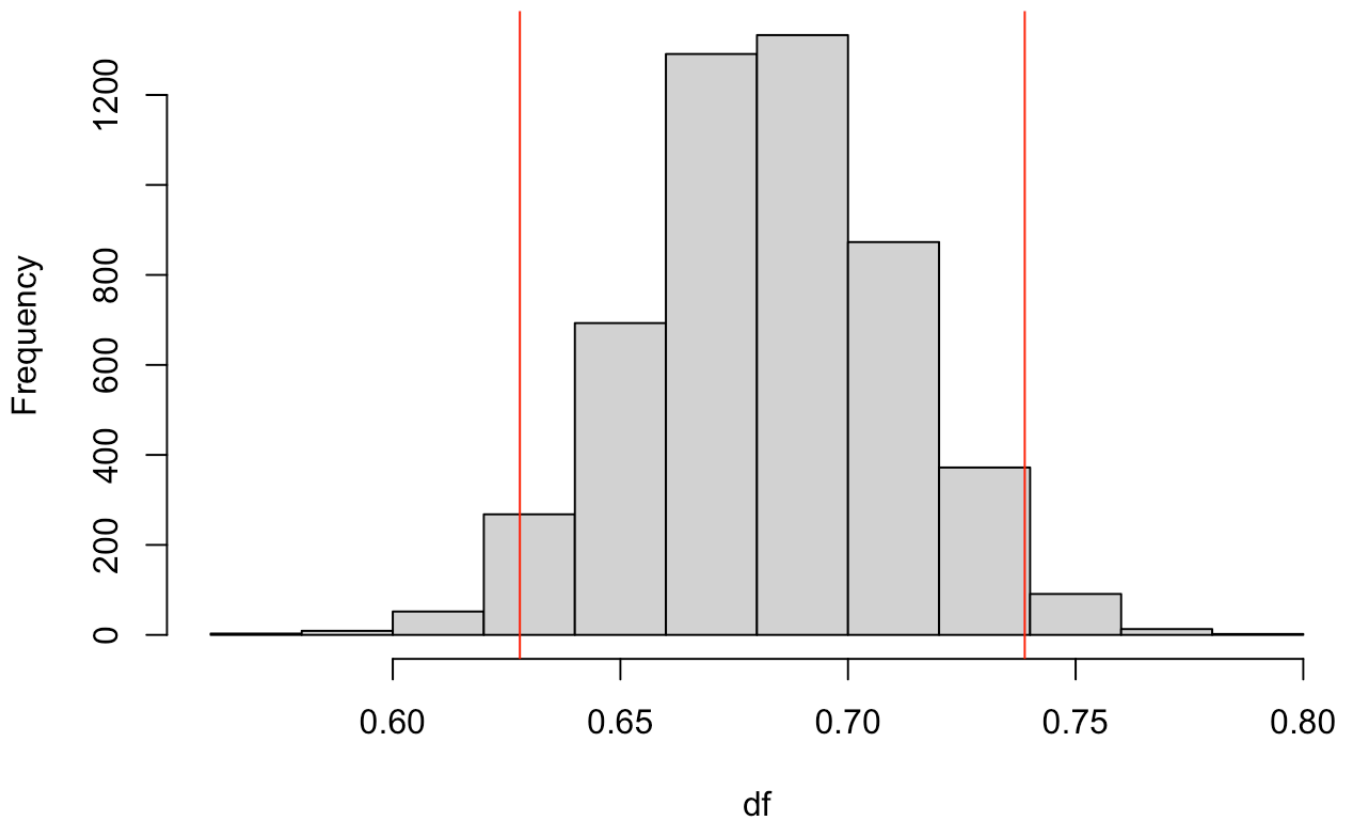




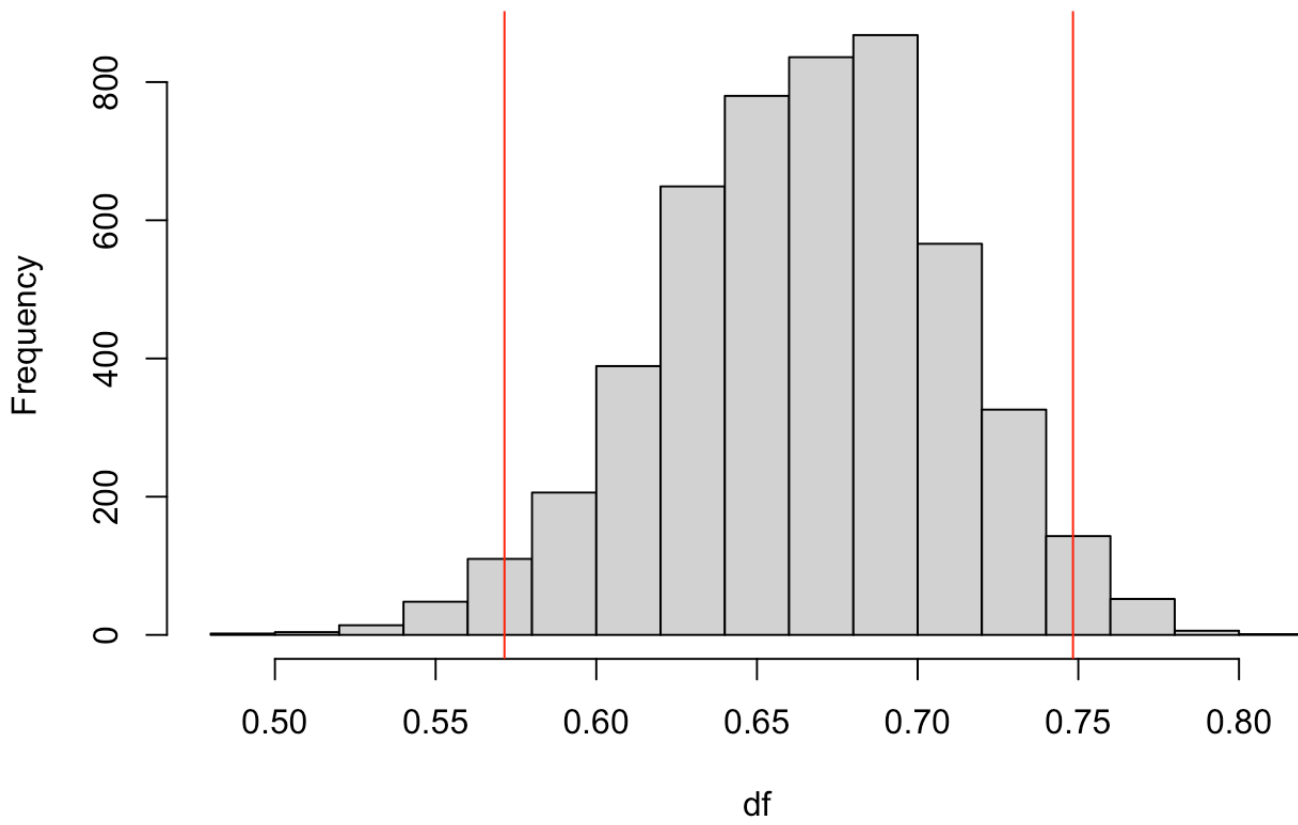
AIC



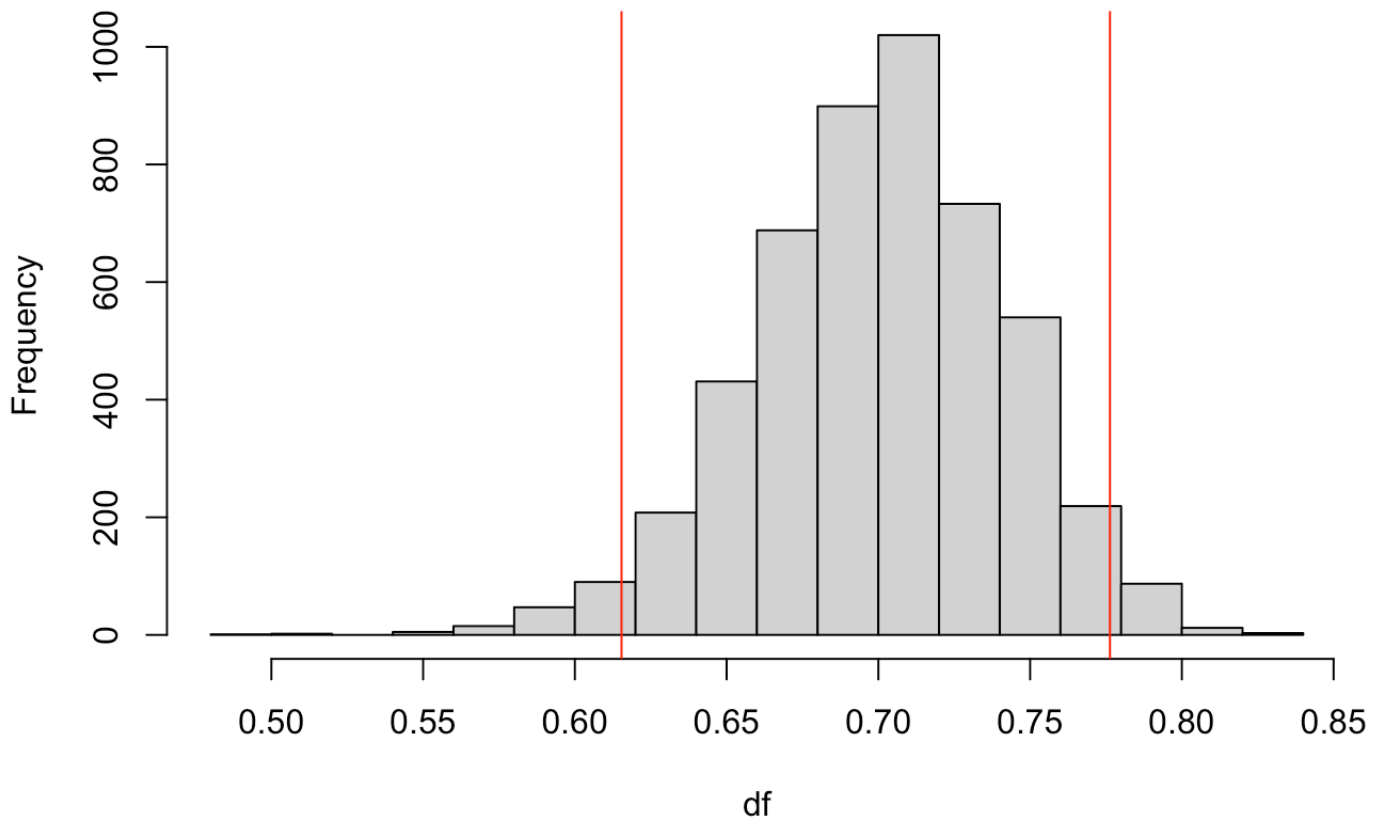
Balanced Accuracy



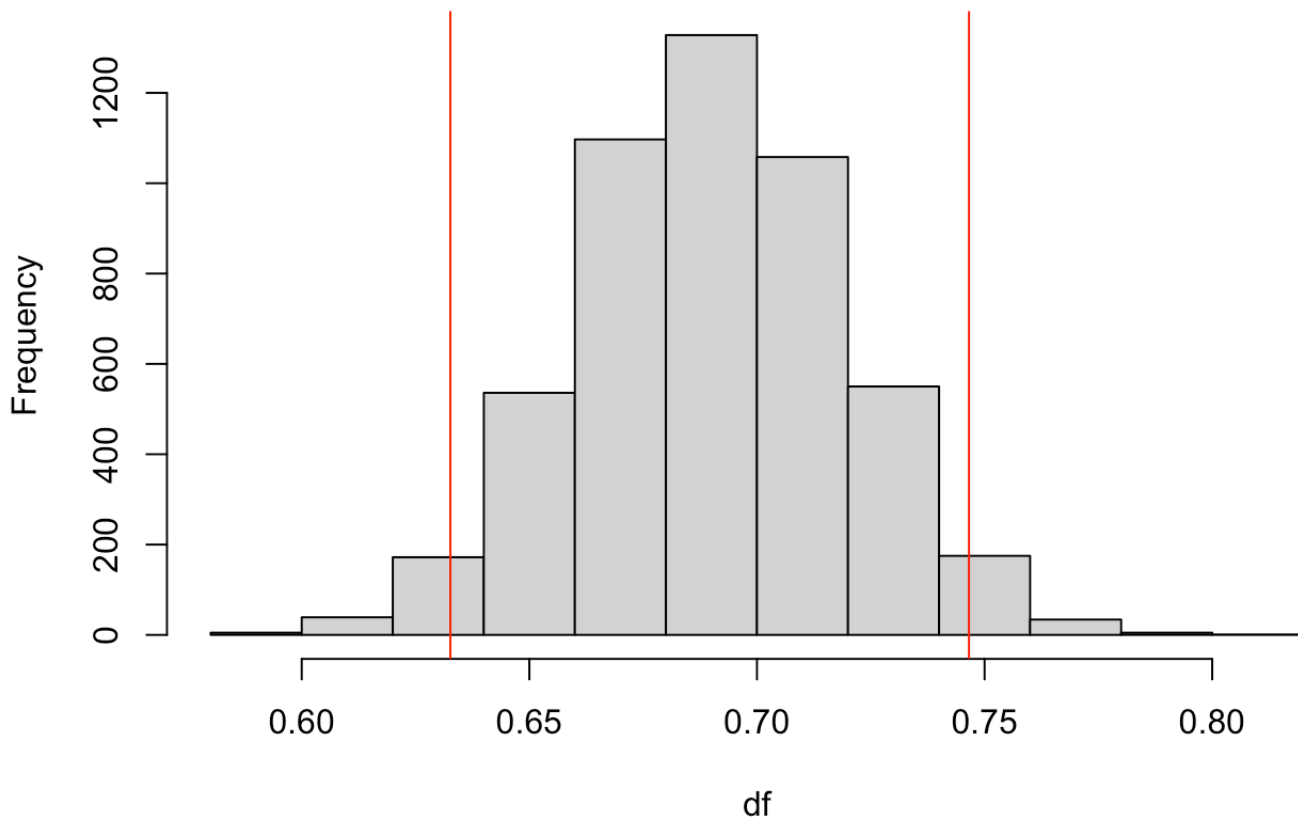
Sensitivity



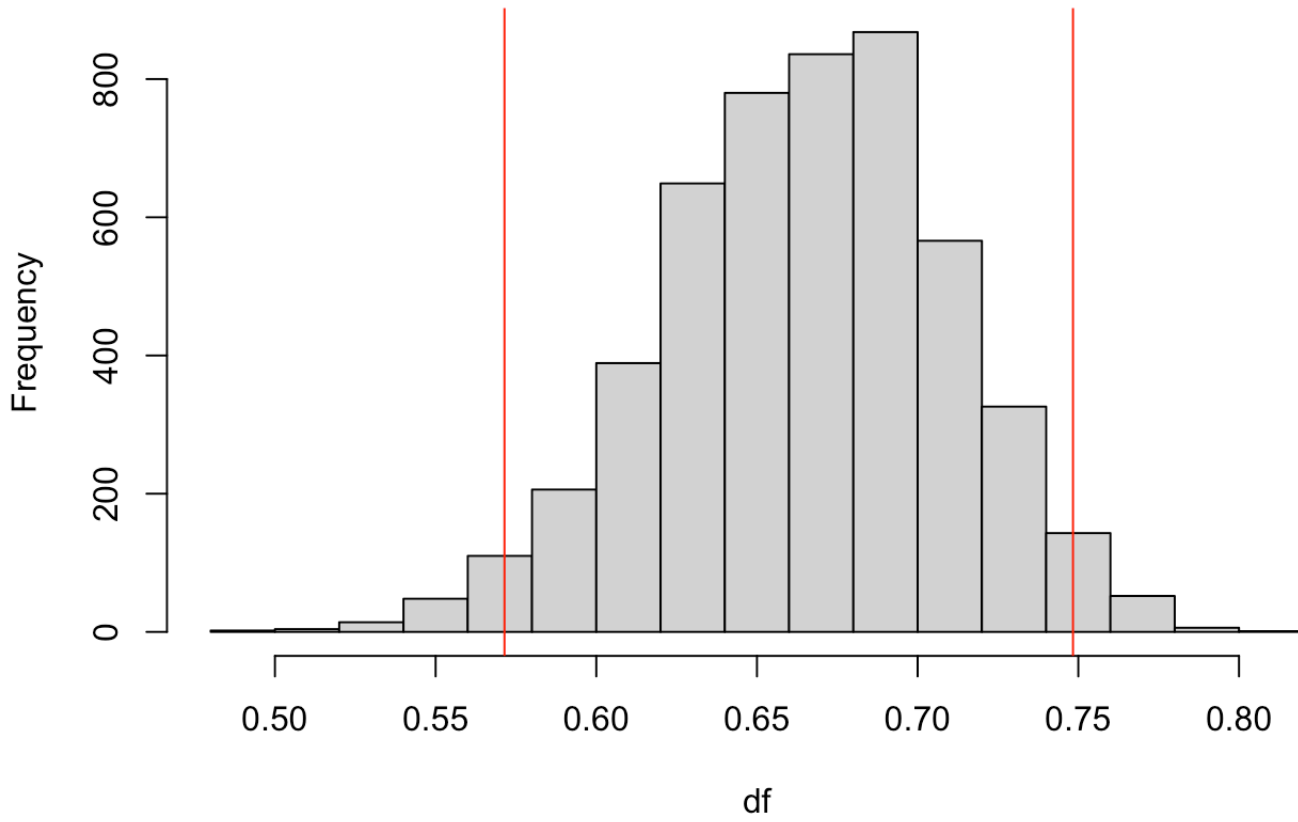
Specificity

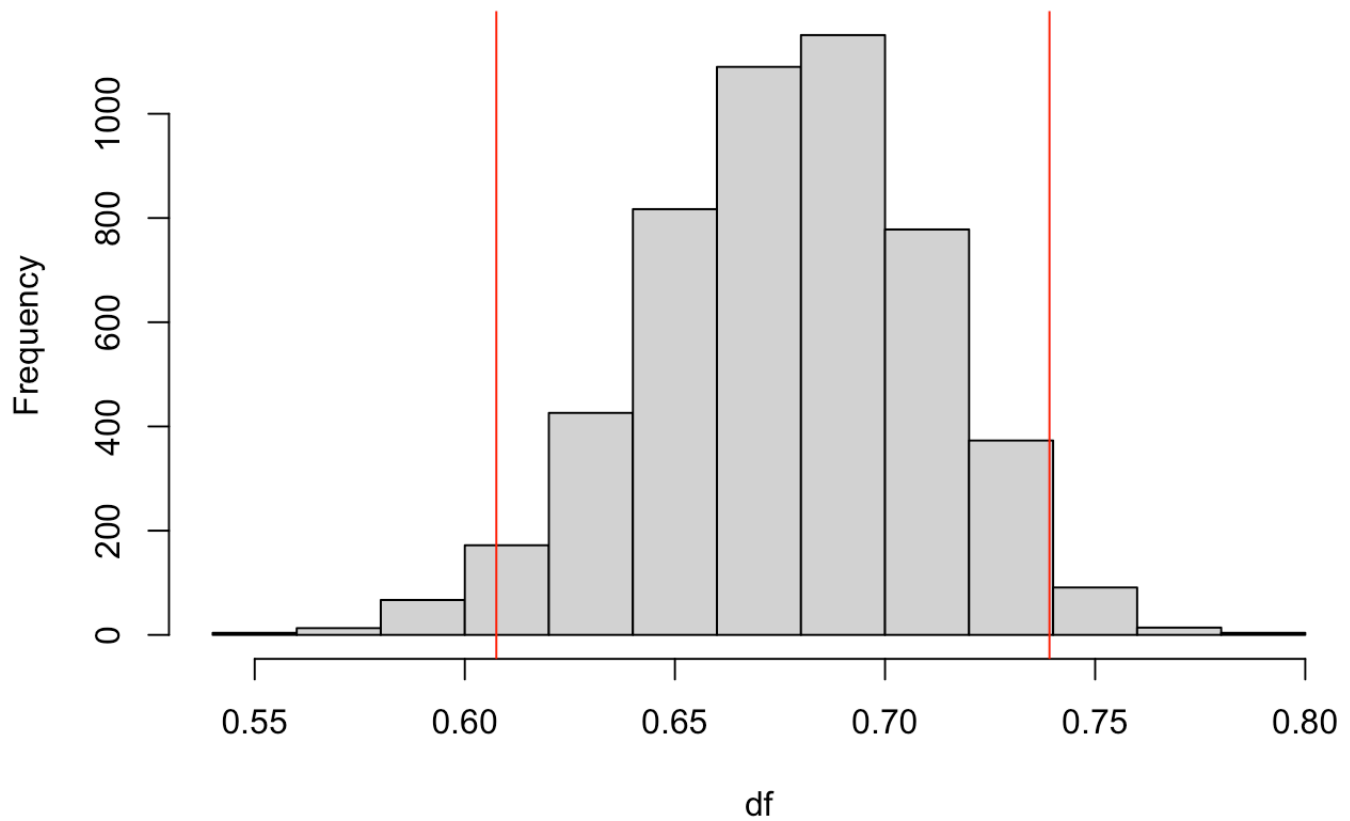


Precision



Recall



F1

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

Brain + EPF

```

df <- merge(x = read.csv('cleandata.csv'), y = read.csv('brain_cb.csv'), by = "subjectkey", 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,subjectkey, aggressive_sumscore.x, interview_age, aggressive_sumscore.y, prosocial_child.y, prosocial_parent.y,prosocial_sum))

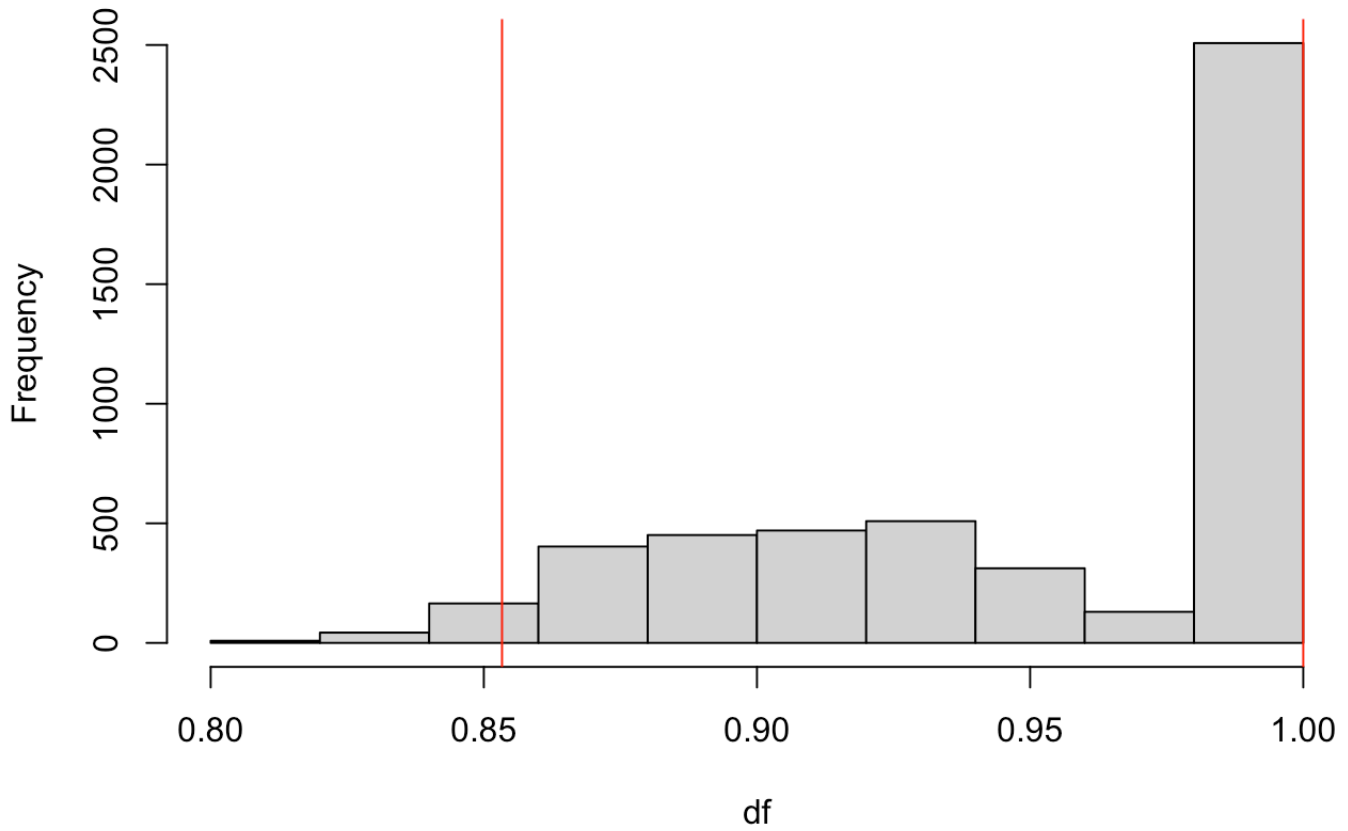
```

```

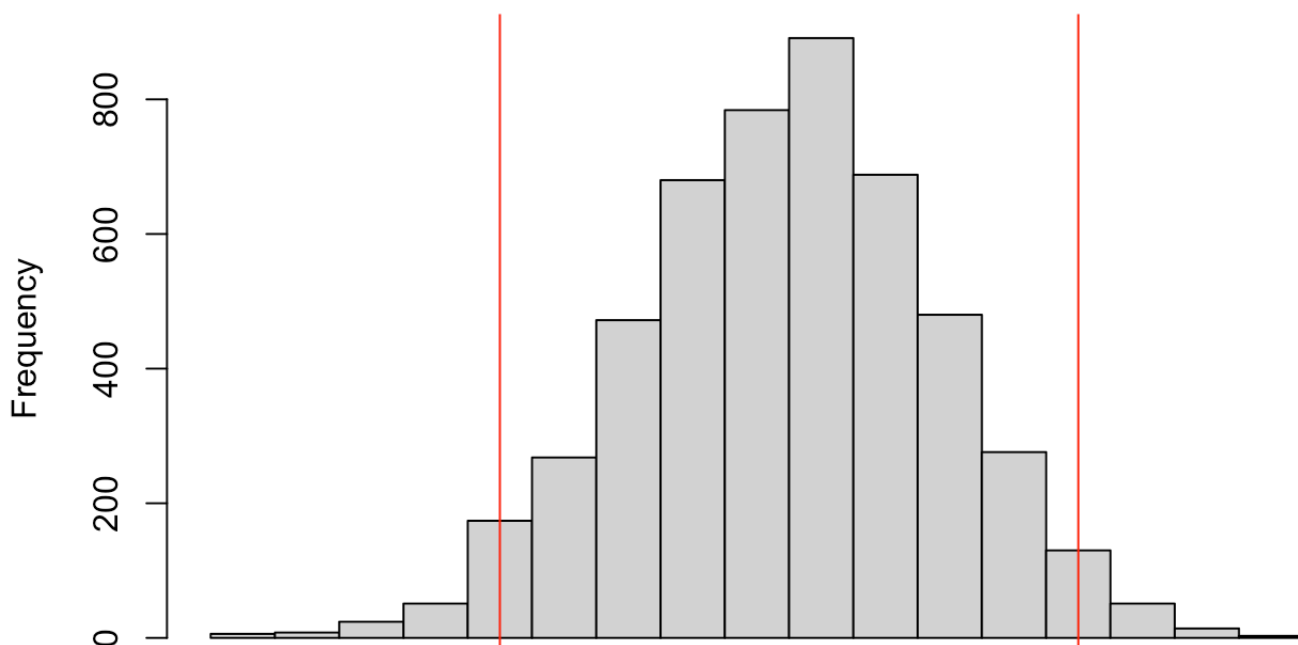
cm_tb<-bootstrap_glm(df, T=GlobalT, filename = './Prosocial-A-brain+EPF.csv', filename2 = './T-Prosocial-R-brain+EPF.csv', filename3 = './Prosocial-A-Beta-brain+EPF.csv')

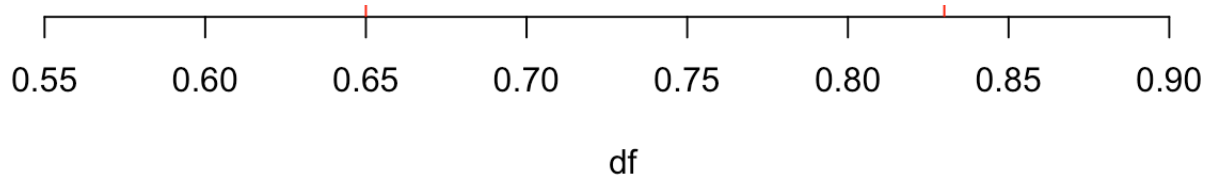
```


Accuracy

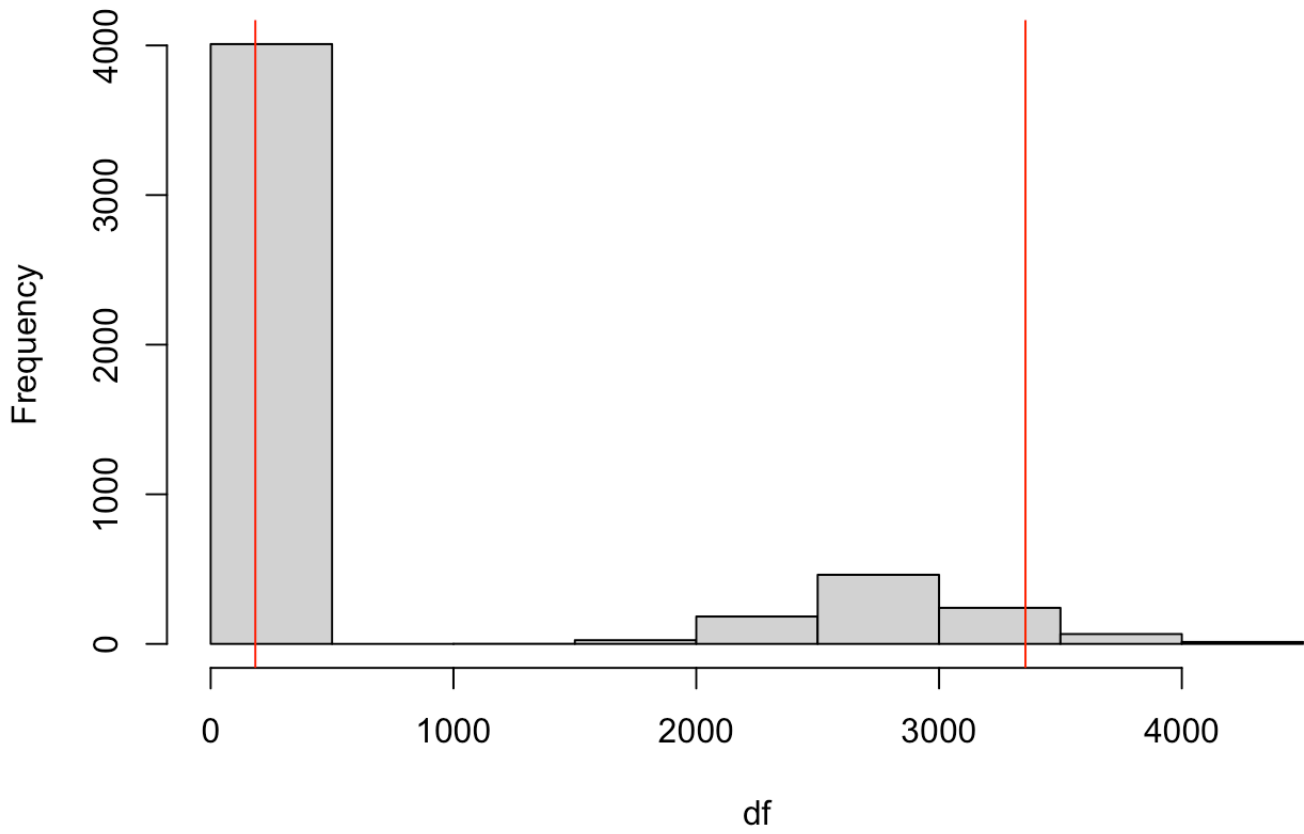


Test Accuracy

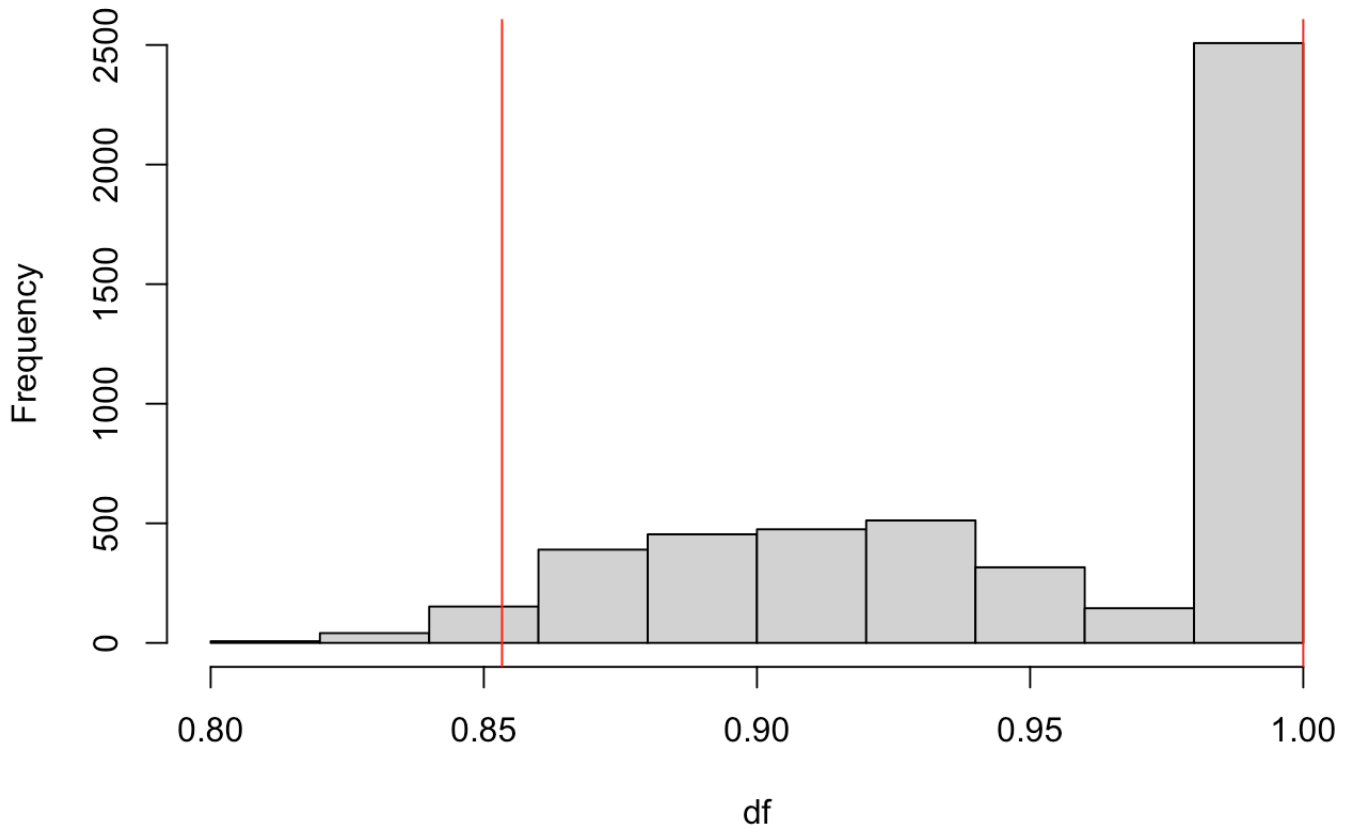




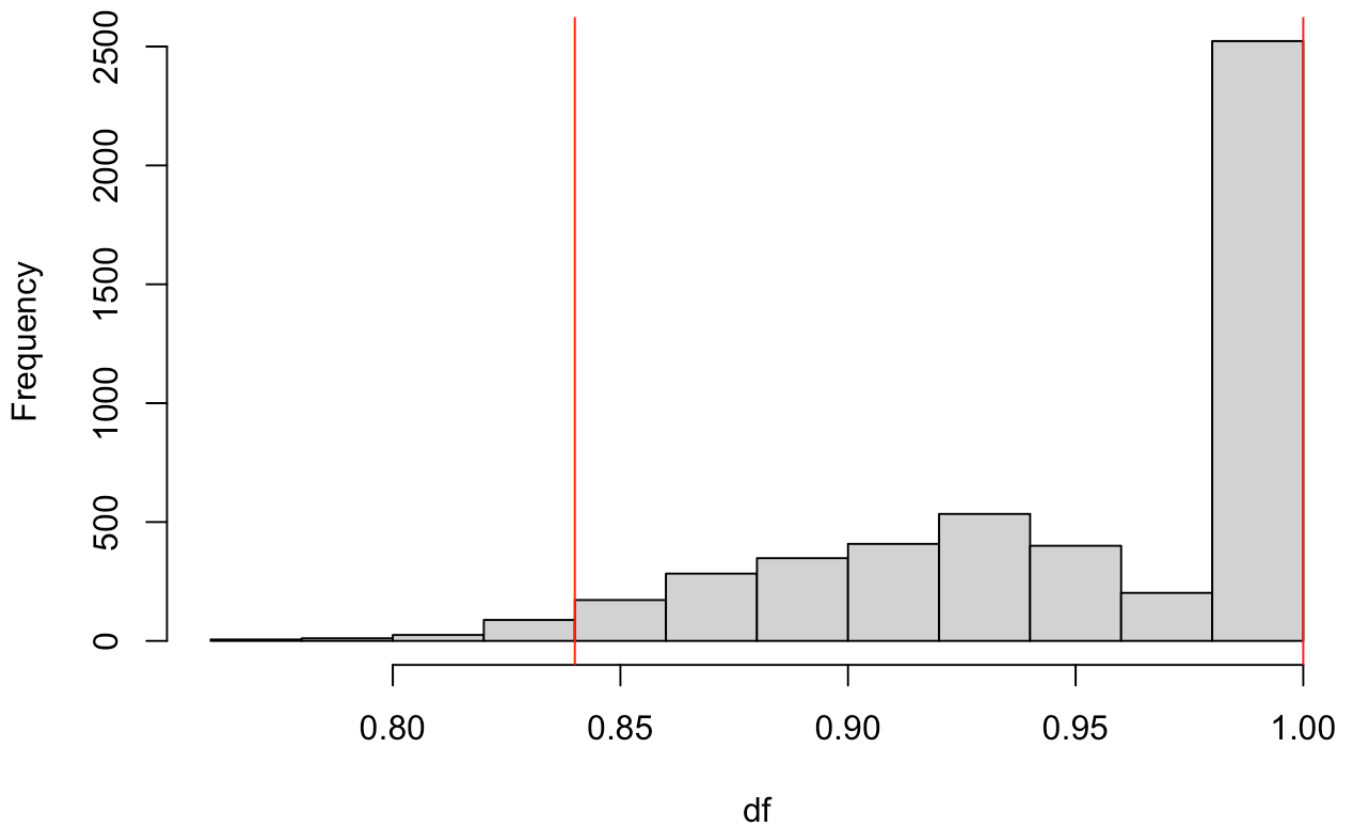
AIC



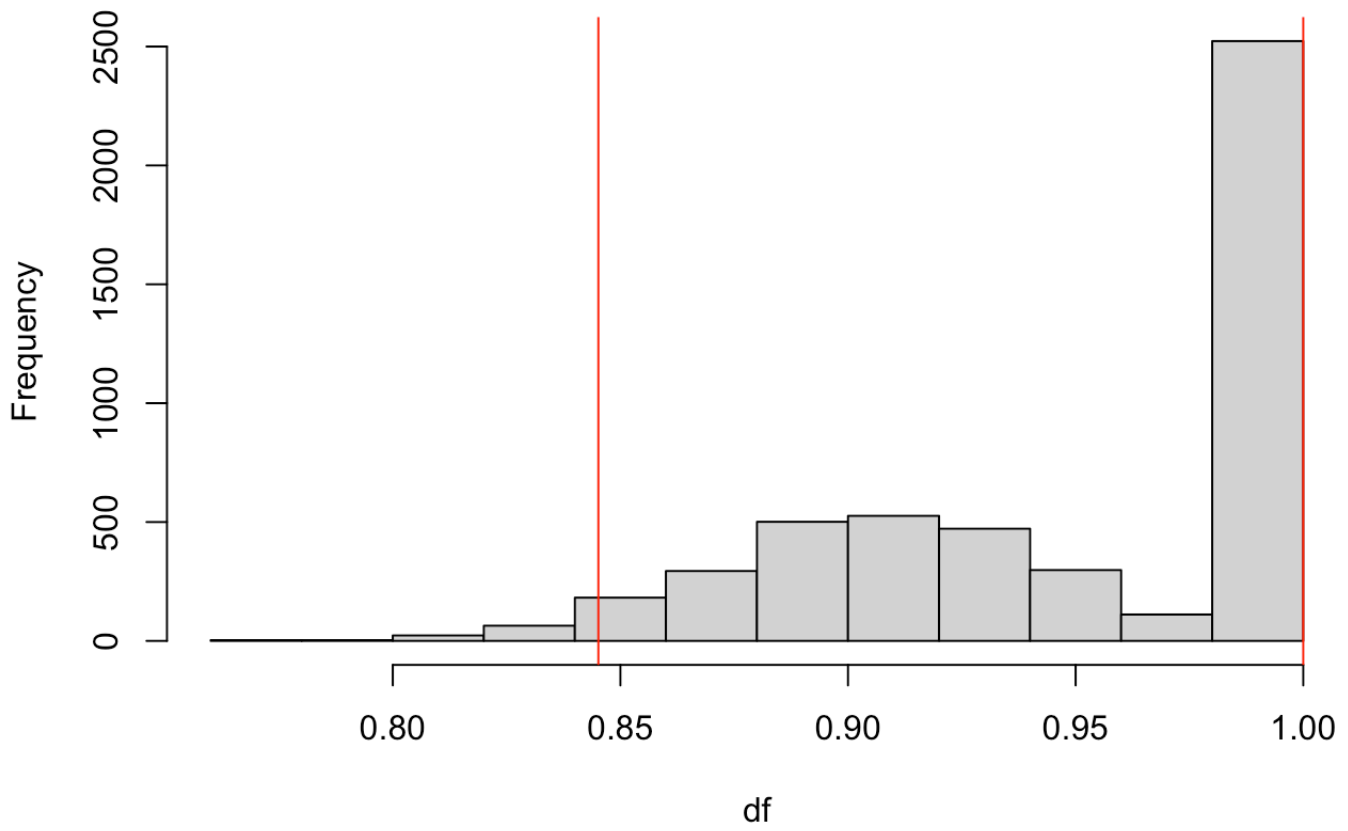
Balanced Accuracy



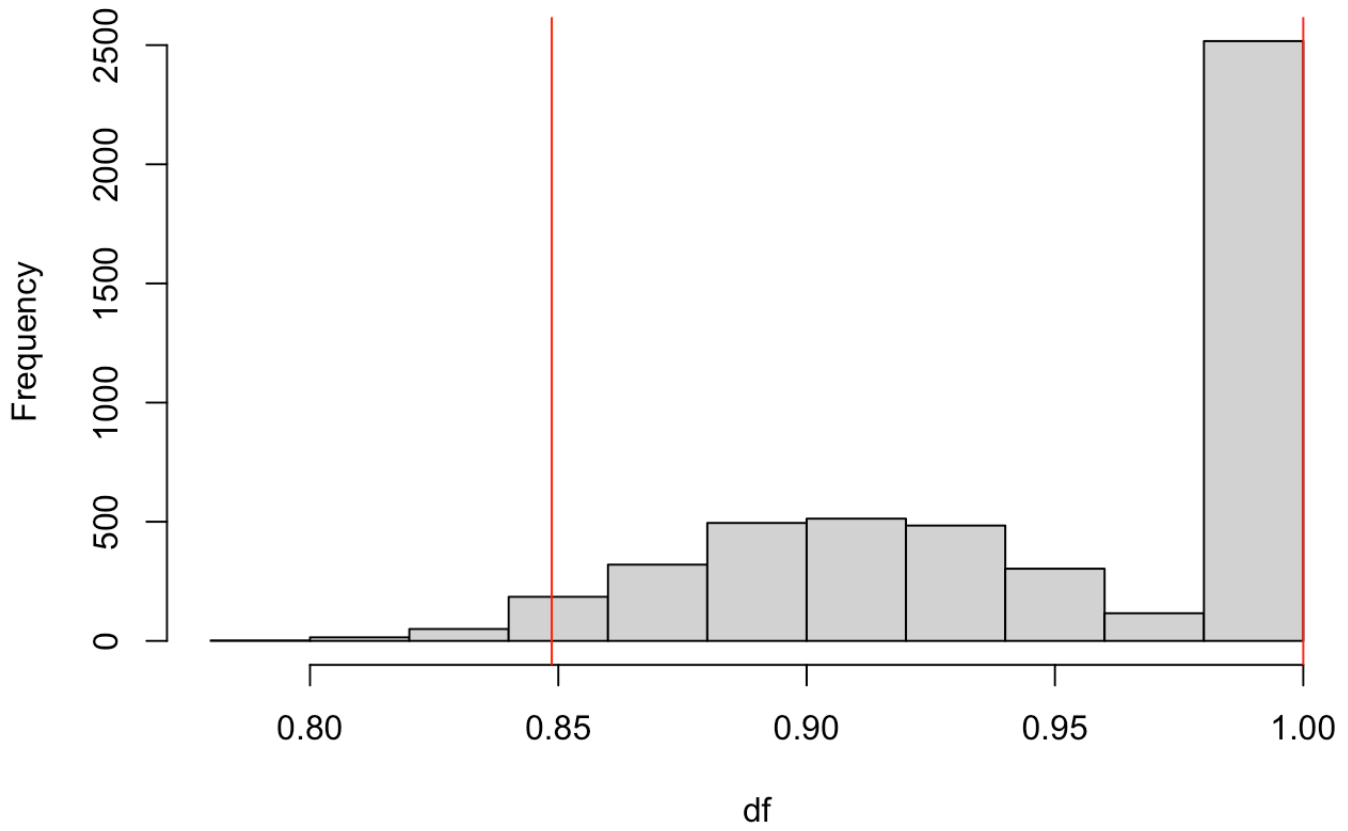
Sensitivity



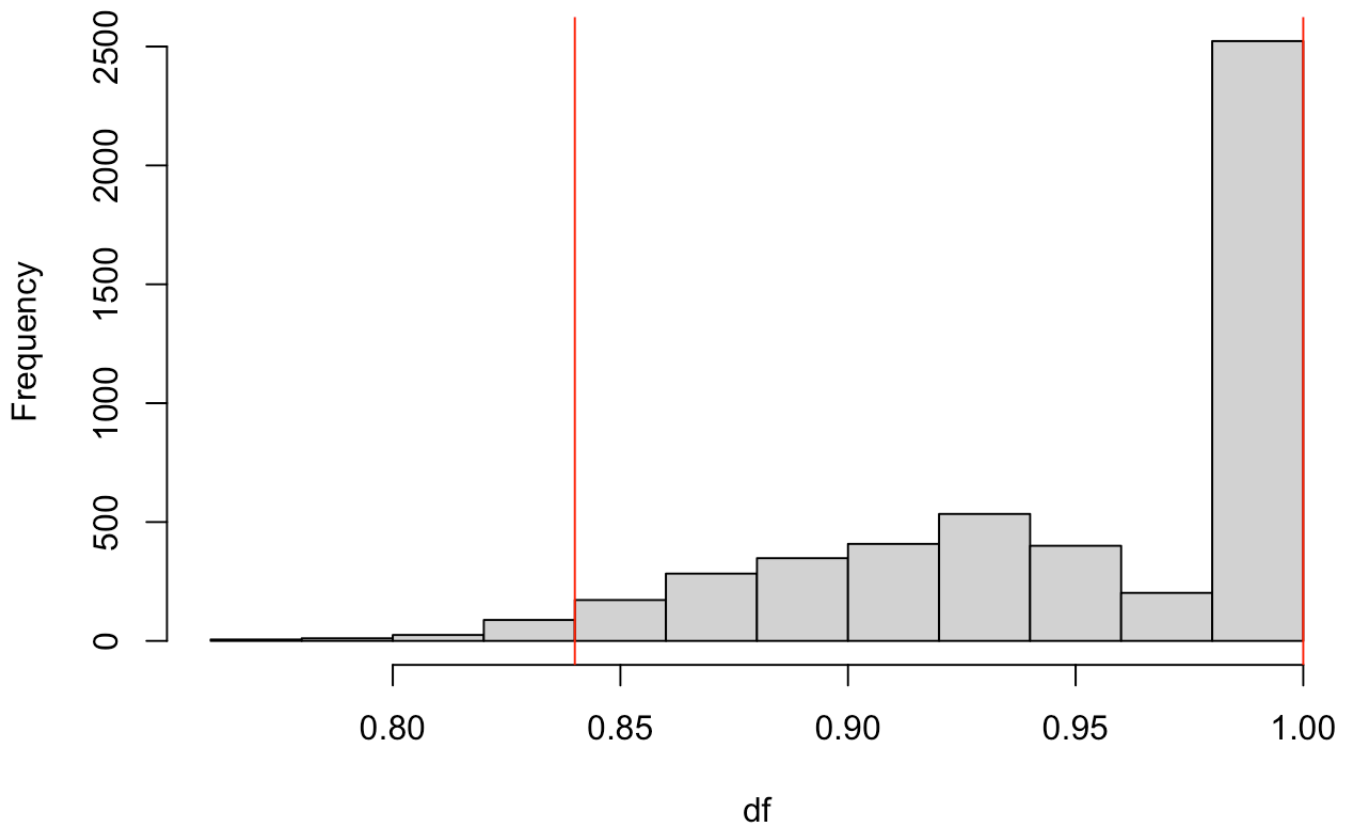
Specificity



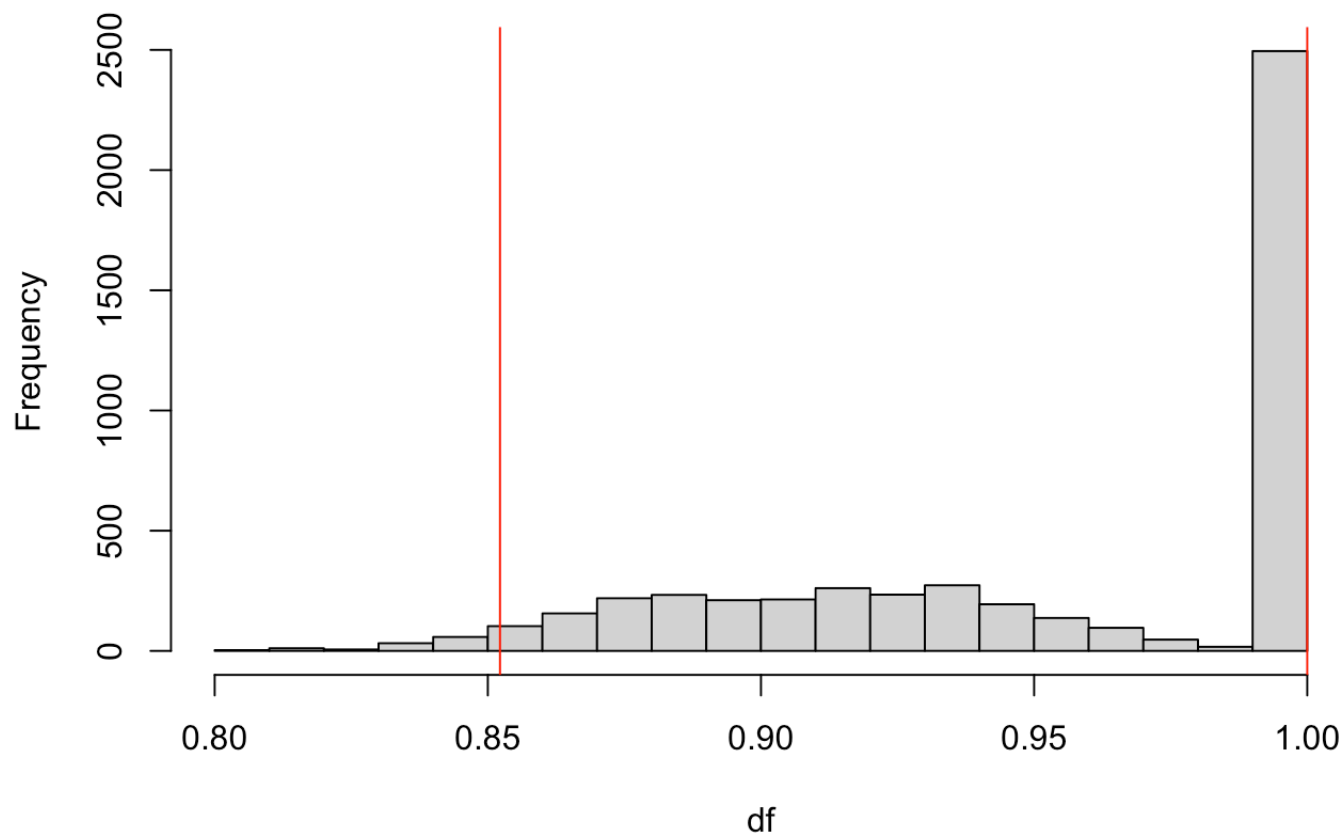
Precision



Recall



F1



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