Prosocial Analysis

Prosocial Parent + Child, All Attributes

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.2
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

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

```
library('miscTools')
cm_tb <- data.frame(matrix(ncol = 11, nrow = 0))
colnames(cm_tb) <- c("Sensitivity", "Specificity", "Pos Pred Value", "Neg Pred Value"
, "Precision", "Recall", "F1", "Prevalence", "Detection Rate", "Detection Prevalence"
, "Balanced Accuracy")

truedf <- df[df$y==1,]
truedf <- truedf[sample(1:nrow(truedf), 200, replace = TRUE), ]
falsedf <- df[df$y==0,]
falsedf <- falsedf[sample(1:nrow(falsedf), 200, replace = TRUE), ]

finaldf <- rbind(truedf, falsedf)
glm.fit <- glm(y ~ ., data = finaldf, family = binomial)


summary(glm.fit)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = finaldf)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.46203  -0.47470   0.00792   0.42255   2.54893
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -20.59969    7.79488  -2.643 0.008224 **
## asr_scr_perstr_t          0.06403    0.02290   2.796 0.005167 **
## asr_scr_anxdep_t         -0.10873    0.08617  -1.262 0.207019
## asr_scr_withdrawn_t      -0.02139    0.07689  -0.278 0.780911
## asr_scr_somatic_t        -0.28039    0.12327  -2.275 0.022932 *
## asr_scr_thought_t         0.10438    0.05715   1.826 0.067778 .
## asr_scr_attention_t      -0.09900    0.11034  -0.897 0.369573
## asr_scr_aggressive_t      0.11435    0.09477   1.207 0.227614
## asr_scr_rulebreak_t       0.07050    0.07312   0.964 0.334969
## asr_scr_intrusive_t       0.15219    0.06488   2.346 0.018990 *
## asr_scr_internal_t        0.08195    0.07938   1.032 0.301876
## asr_scr_external_t       -0.15935    0.07027  -2.268 0.023350 *
## asr_scr_totprob_t        -0.03393    0.11146  -0.304 0.760790
## asr_scr_depress_t         0.09531    0.08788   1.085 0.278141
## asr_scr_anxdisord_t       0.04146    0.06641   0.624 0.532421
## asr_scr_somaticpr_t       0.16956    0.10021   1.692 0.090655 .
## asr_scr_avoidant_t       -0.06946    0.06137  -1.132 0.257675
## asr_scr_adhd_t           -0.06132    0.21170  -0.290 0.772085
## asr_scr_antisocial_t      0.09308    0.08079   1.152 0.249233
```

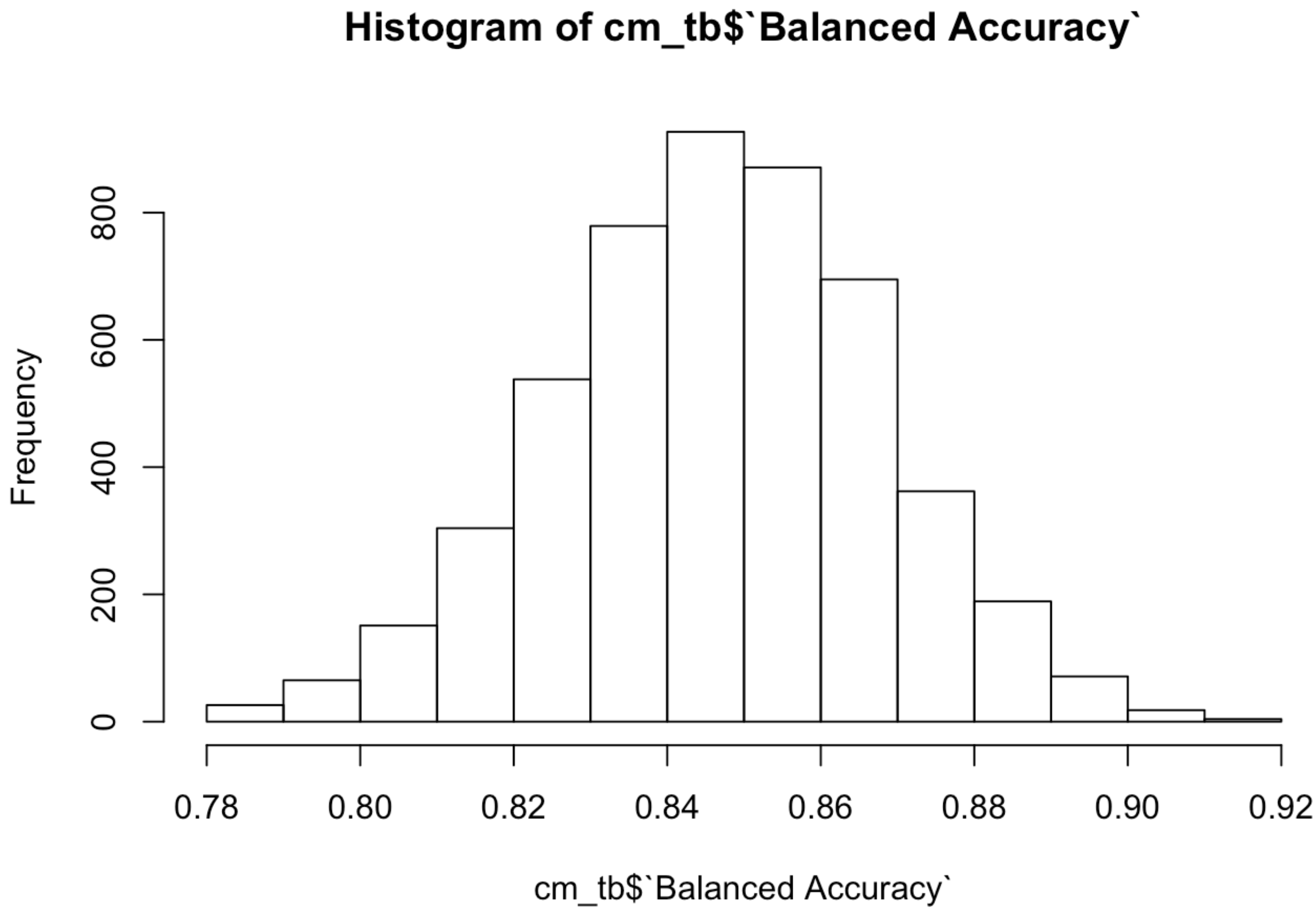
```
## asr_scr_inattention_t      0.07709    0.14200   0.543 0.587176    
## asr_scr_hyperactive_t      0.02589    0.13883   0.186 0.852076    
## crpbi_bothcare             0.01846    0.02798   0.660 0.509444    
## parent_monitor_y           1.06934    0.36280   2.947 0.003204 ** 
## kbi_p_conflict            -1.42586    0.41956  -3.398 0.000678 ***
## kbi_p_c_best_friend       -0.23850    0.20310  -1.174 0.240256    
## kbi_p_c_reg_friend_group  -0.87584    0.44689  -1.960 0.050013 .  
## kbi_p_c_bully             -0.36163    0.52446  -0.690 0.490485    
## kbi_p_c_mh_sa             -1.21295    0.47741  -2.541 0.011064 *  
## fes_youth                 -0.04578    0.09560  -0.479 0.632045    
## fes_p_ss_fc_pr            -0.03826    0.09960  -0.384 0.700894    
## macv_p_ss_fs               1.61283    0.45860   3.517 0.000437 ***
## macv_p_ss_fo              -0.78363    0.43374  -1.807 0.070811 .  
## macv_p_ss_isr             -0.78134    0.33933  -2.303 0.021300 *  
## macv_p_ss_fr               0.15248    0.34296   0.445 0.656612    
## macv_p_ss_r               -0.86698    0.26310  -3.295 0.000983 ***
## demo_prnt_age_v2           0.05382    0.03202   1.681 0.092811 .  
## demo_prnt_marital_v2       0.36297    0.14030   2.587 0.009680 ** 
## demo_comb_income_v2        0.13934    0.11311   1.232 0.217967    
## demo_fam_exp               1.81122    0.66476   2.725 0.006437 ** 
## demo_yrs_1                -0.44784    0.21837  -2.051 0.040284 *  
## demo_yrs_2                 1.47077    0.33609   4.376 1.21e-05 ***
## parent_rules_q1           -0.06633    0.31508  -0.211 0.833260    
## parent_rules_q4           -0.28319    0.41349  -0.685 0.493422    
## parent_rules_q7           -0.08896    0.35081  -0.254 0.799823    
## su_risk_p_1               -0.02480    0.15498  -0.160 0.872866    
## su_risk_p_2_3             -0.25335    0.25313  -1.001 0.316889    
## su_risk_p_4_5              0.01015    0.28731   0.035 0.971807    
## neighborhood1_2_3_p       -0.23022    0.22623  -1.018 0.308842    
## neighborhood_crime_y       0.47101    0.18954   2.485 0.012955 *  
## sexM                      -2.31467    0.41698  -5.551 2.84e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 554.52  on 399  degrees of freedom
## Residual deviance: 268.69  on 350  degrees of freedom
## AIC: 368.69
## 
## Number of Fisher Scoring iterations: 6
```

```
cm <- confusionMatrix(table(as.numeric(glm.fit$fitted.values>0.5), finaldf$y))
print(cm$byClass)
```

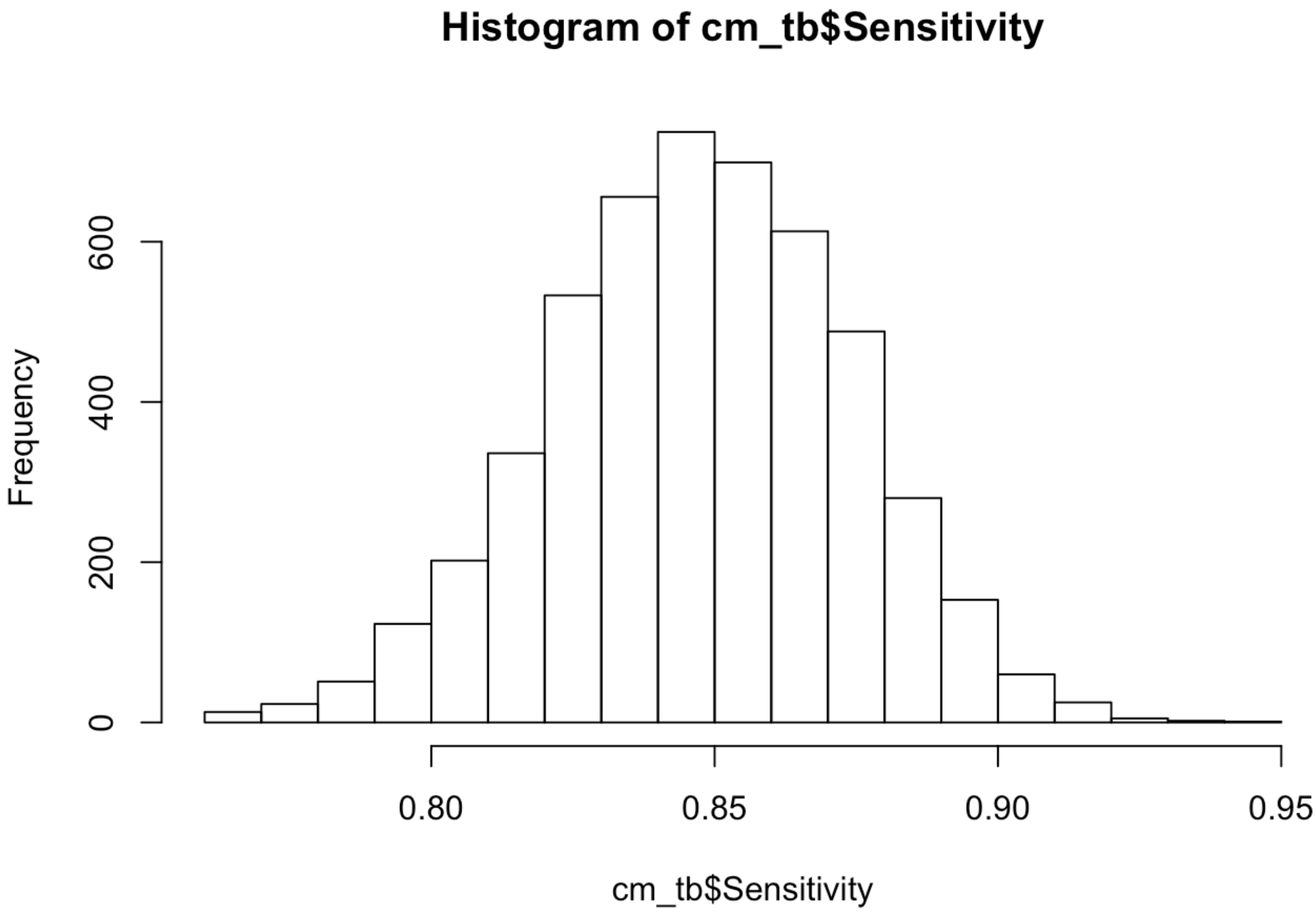
```
##          Sensitivity          Specificity       Pos Pred Value
##            0.8750000            0.8300000            0.8373206
##       Neg Pred Value            Precision               Recall
##            0.8691099            0.8373206            0.8750000
##                   F1           Prevalence       Detection Rate
##            0.8557457            0.5000000            0.4375000
## Detection Prevalence    Balanced Accuracy
##            0.5225000            0.8525000
```

```
for (t in 1:5000){
  truedf <- df[df$y==1,]
  truedf <- truedf[sample(1:nrow(truedf), 200, replace = TRUE), ]
  falsedf <- df[df$y==0,]
  falsedf <- falsedf[sample(1:nrow(falsedf), 200, replace = TRUE), ]

  finaldf <- rbind(truedf, falsedf)
  glm.fit <- glm(y ~ ., data = finaldf, family = binomial)


  cm <- confusionMatrix(table(as.numeric(glm.fit$fitted.values>0.5), finaldf$y))
  cm_tb <- rbind(cm_tb, as.data.frame(t(cm$byClass)))
}
```

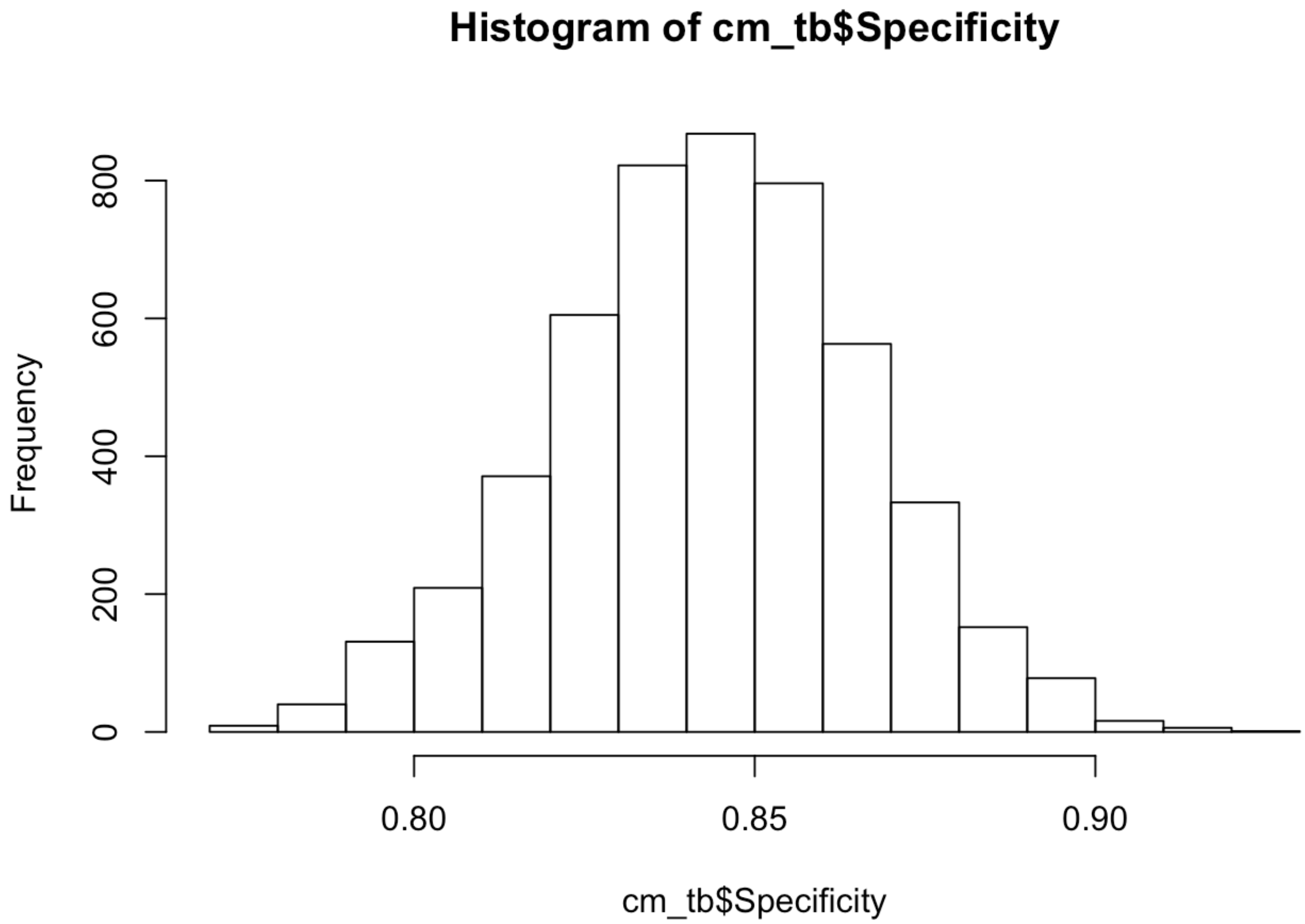
```
hist(cm_tb$`Balanced Accuracy`)
```



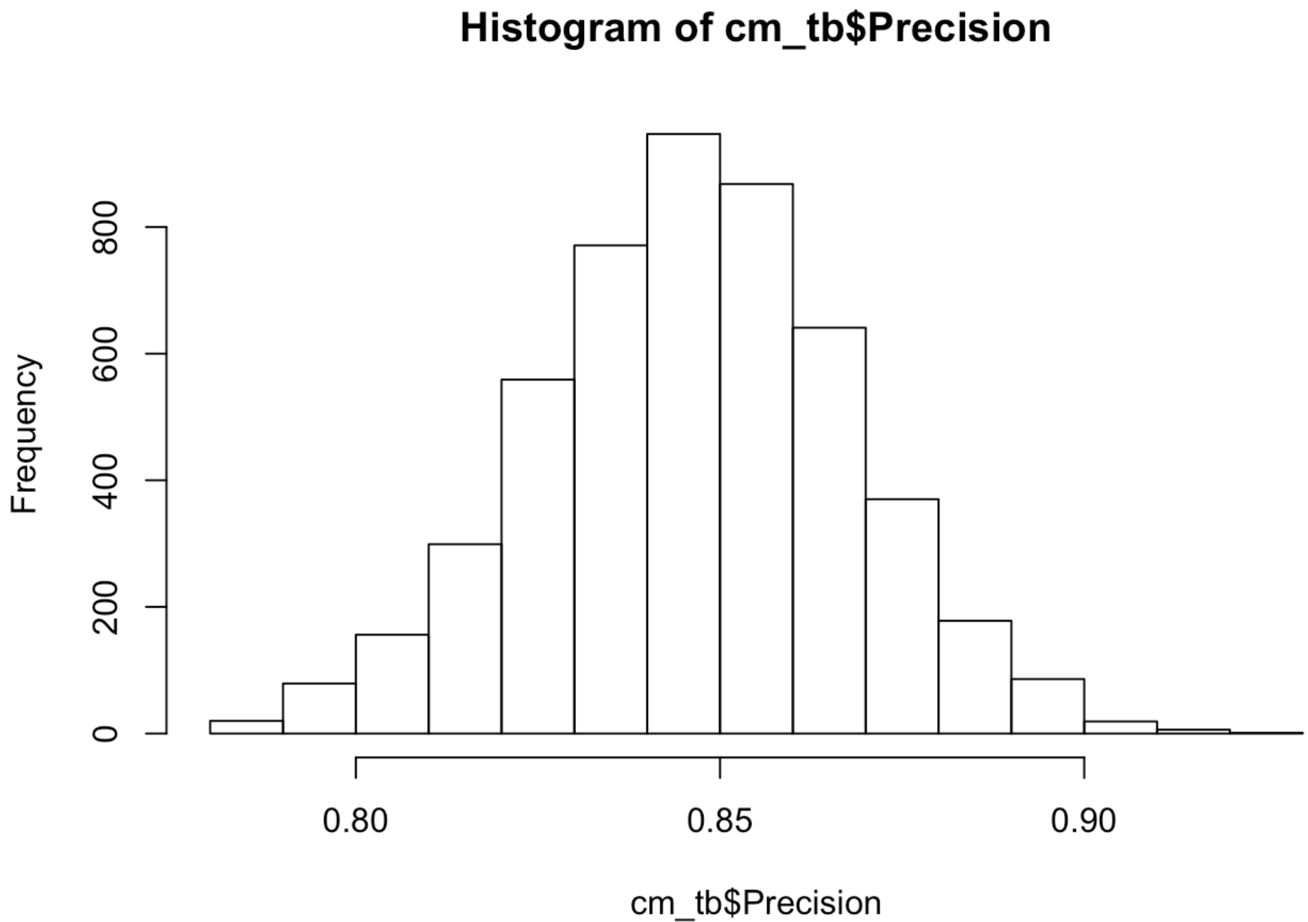

```
hist(cm_tb$Sensitivity)
```



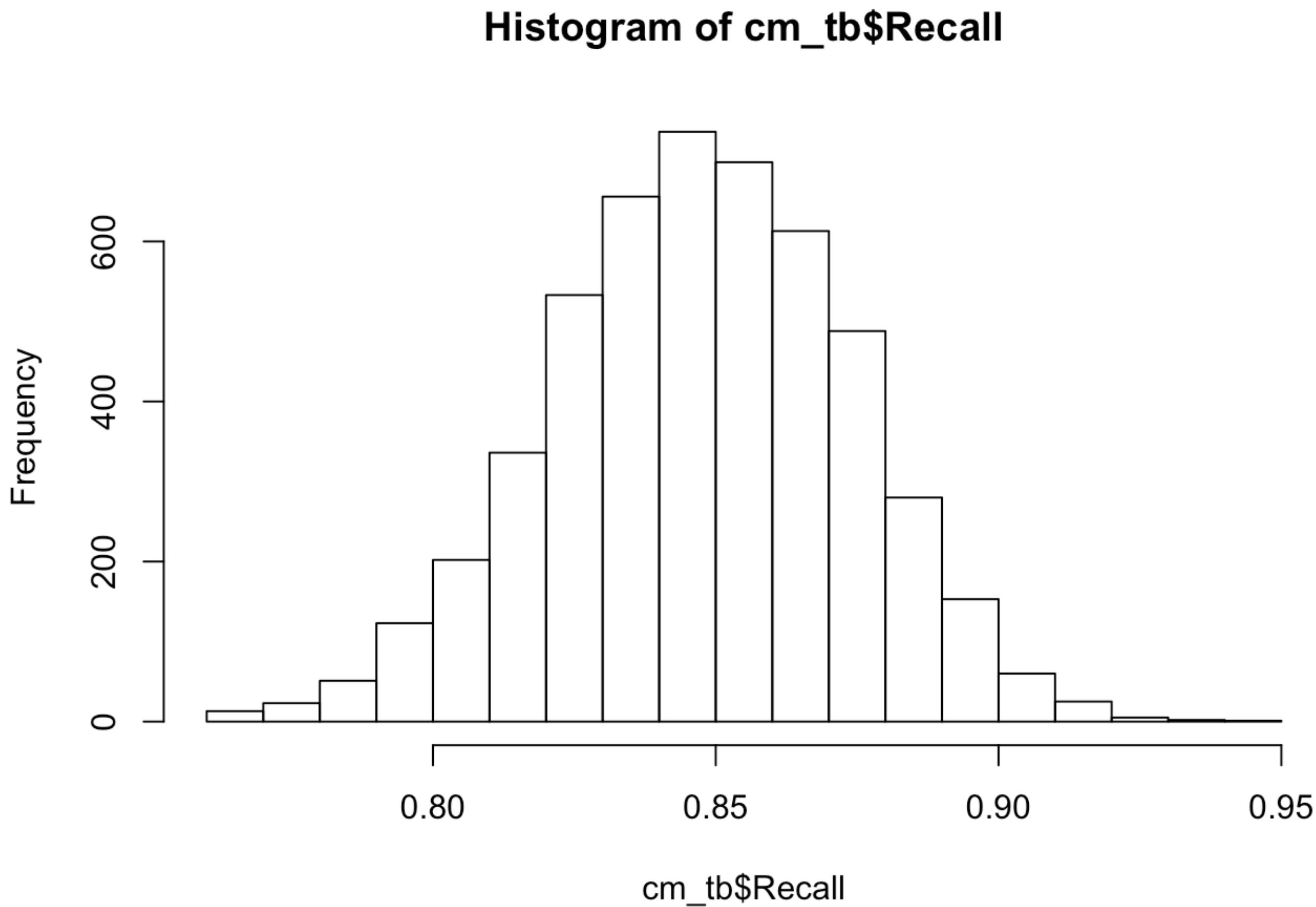

```
hist(cm_tb$Specificity)
```



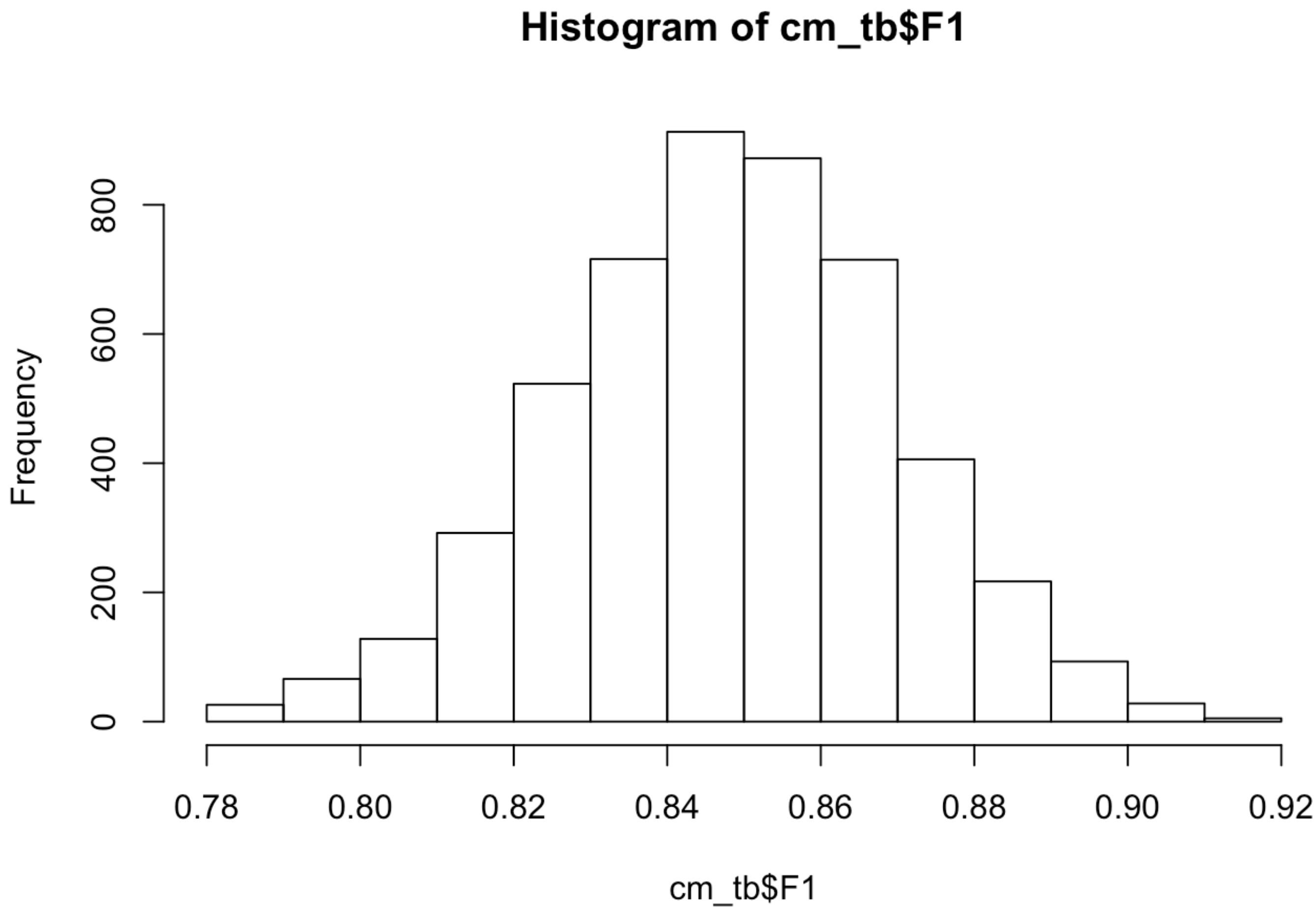

```
hist(cm_tb$Precision)
```




```
hist(cm_tb$Recall)
```


Histogram of cm_tb$Recall
Frequency
0
200
400
600
0.80
0.85
0.90
0.95
cm_tb$Recall

```
hist(cm_tb$F1)
```



Prosocial Child, brain

Attribute selection

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

```
cm_tb <- data.frame(matrix(ncol = 11, nrow = 0))
colnames(cm_tb) <- c("Sensitivity", "Specificity", "Pos Pred Value", "Neg Pred Value"
, "Precision", "Recall", "F1", "Prevalence", "Detection Rate", "Detection Prevalence"
, "Balanced Accuracy")

truedf <- df[df$y==1,]
truedf <- truedf[sample(1:nrow(truedf), 200, replace = TRUE), ]
falsedf <- df[df$y==0,]
falsedf <- falsedf[sample(1:nrow(falsedf), 200, replace = TRUE), ]

finaldf <- rbind(truedf, falsedf)
glm.fit <- glm(y ~ ., data = finaldf, family = binomial)


summary(glm.fit)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial, data = finaldf)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.08730  -1.05714   0.06183   1.01431   2.72349
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -0.02652    0.33057  -0.080  0.93605
## rsfmri_cor_ngd_cerc_scs_thp -0.54567    1.06612  -0.512  0.60877
```

```
## rsfmri_cor_ngd_cerc_scs_cde -2.30741    1.42031  -1.625  0.10425
## rsfmri_cor_ngd_cerc_scs_pt   3.57546    1.22387   2.921  0.00348 **
## rsfmri_cor_ngd_cerc_scs_hp  -1.89931    1.28600  -1.477  0.13970
## rsfmri_cor_ngd_cerc_scs_ag   1.02585    0.85389   1.201  0.22960
## rsfmri_cor_ngd_cerc_scs_aa   0.03424    0.83223   0.041  0.96718
## rsfmri_cor_ngd_cerc_scs_bs  -0.99262    0.90384  -1.098  0.27211
## rsfmri_cor_ngd_df_scs_thp   -0.58832    1.51702  -0.388  0.69816
## rsfmri_cor_ngd_df_scs_cde   -2.16270    1.12590  -1.921  0.05475 .
## rsfmri_cor_ngd_df_scs_pt    -0.62134    1.28794  -0.482  0.62950
## rsfmri_cor_ngd_df_scs_hp     0.21545    0.91573   0.235  0.81400
## rsfmri_cor_ngd_df_scs_ag     2.51557    1.33588   1.883  0.05969 .
## rsfmri_cor_ngd_df_scs_aa     0.12558    0.77297   0.162  0.87094
## rsfmri_cor_ngd_df_scs_bs    -0.05538    0.63796  -0.087  0.93083
## rsfmri_cor_ngd_dsa_scs_thp  -3.04228    1.04373  -2.915  0.00356 **
## rsfmri_cor_ngd_dsa_scs_cde  -0.36941    1.00900  -0.366  0.71428
## rsfmri_cor_ngd_dsa_scs_pt   -1.30430    1.30096  -1.003  0.31607
## rsfmri_cor_ngd_dsa_scs_hp    1.78239    1.55883   1.143  0.25286
## rsfmri_cor_ngd_dsa_scs_ag   -2.70473    1.67357  -1.616  0.10606
## rsfmri_cor_ngd_dsa_scs_aa   -0.06285    1.11981  -0.056  0.95524
## rsfmri_cor_ngd_dsa_scs_bs    2.82898    1.06666   2.652  0.00800 **
## rsfmri_cor_ngd_fopa_scs_thp  0.69106    0.89831   0.769  0.44172
## rsfmri_cor_ngd_fopa_scs_cde -3.56373    1.78703  -1.994  0.04613 *
## rsfmri_cor_ngd_fopa_scs_pt  -2.17060    1.09186  -1.988  0.04681 *
## rsfmri_cor_ngd_fopa_scs_hp  -0.84753    1.03024  -0.823  0.41071
## rsfmri_cor_ngd_fopa_scs_ag   0.06168    1.00645   0.061  0.95113
## rsfmri_cor_ngd_fopa_scs_aa  -1.14919    1.35387  -0.849  0.39598
## rsfmri_cor_ngd_fopa_scs_bs   1.03553    0.70233   1.474  0.14036
## rsfmri_cor_ngd_sa_scs_thp    1.16960    1.07106   1.092  0.27483
## rsfmri_cor_ngd_sa_scs_cde   -1.00591    0.89002  -1.130  0.25839
## rsfmri_cor_ngd_sa_scs_pt     1.63495    1.38240   1.183  0.23693
## rsfmri_cor_ngd_sa_scs_hp     1.27627    1.05554   1.209  0.22662
## rsfmri_cor_ngd_sa_scs_ag     0.82689    1.11242   0.743  0.45729
## rsfmri_cor_ngd_sa_scs_aa     2.56698    1.84823   1.389  0.16487
## rsfmri_cor_ngd_sa_scs_bs    -0.24077    0.84449  -0.285  0.77557
## rsfmri_cor_ngd_vta_scs_thp   1.58686    1.02339   1.551  0.12100
## rsfmri_cor_ngd_vta_scs_cde  -0.95752    1.35063  -0.709  0.47836
## rsfmri_cor_ngd_vta_scs_pt   -1.81284    1.54868  -1.171  0.24177
## rsfmri_cor_ngd_vta_scs_hp    2.54633    1.18001   2.158  0.03094 *
## rsfmri_cor_ngd_vta_scs_ag    1.43461    1.51570   0.946  0.34389
## rsfmri_cor_ngd_vta_scs_aa    0.14835    1.11851   0.133  0.89448
## rsfmri_cor_ngd_vta_scs_bs   -3.33302    1.40120  -2.379  0.01737 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 554.52  on 399  degrees of freedom
```

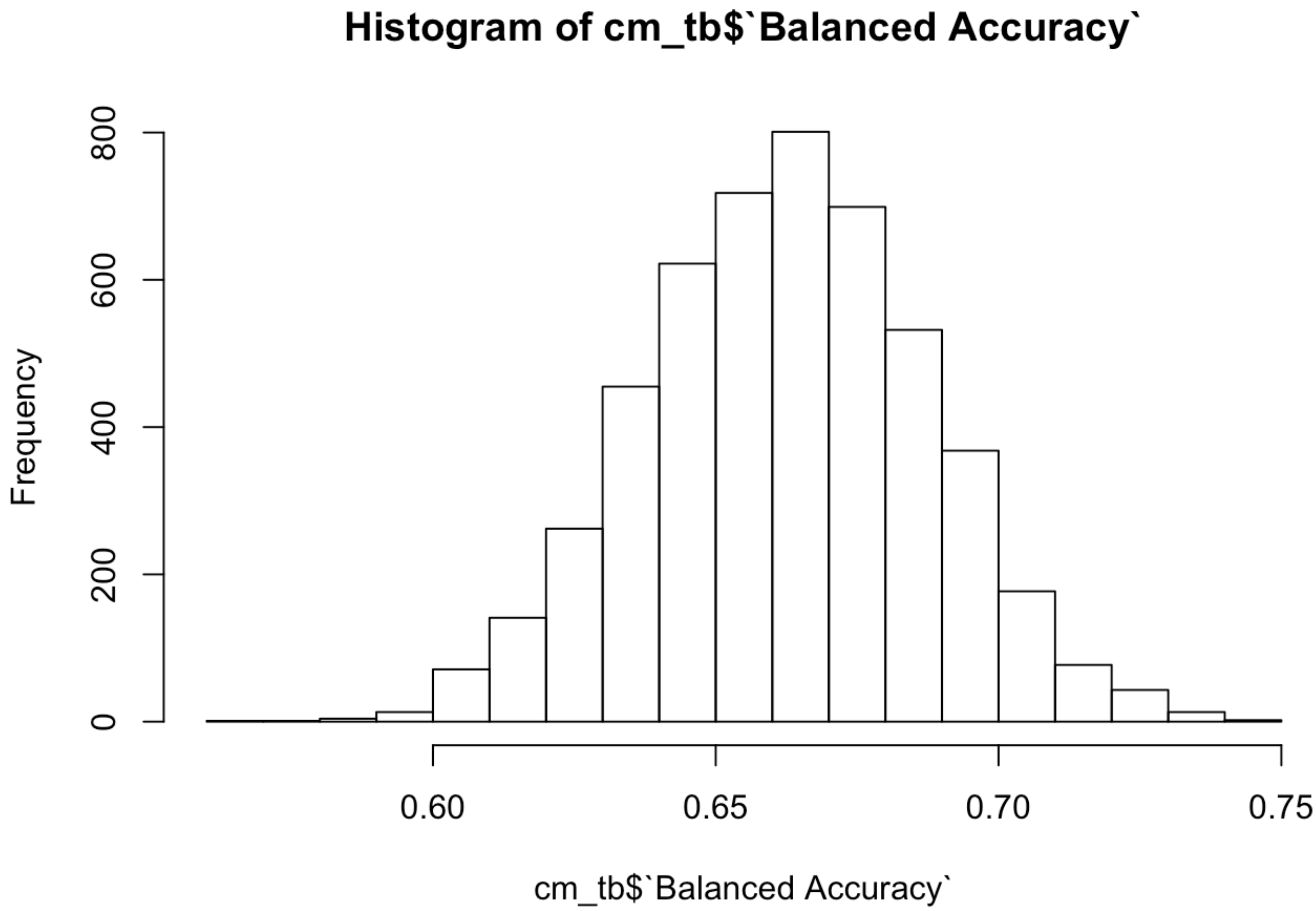
```
## Residual deviance: 484.86  on 357  degrees of freedom
## AIC: 570.86
##
## Number of Fisher Scoring iterations: 5
```

```
cm <- confusionMatrix(table(as.numeric(glm.fit$fitted.values>0.5), finaldf$y))
print(cm$byClass)
```

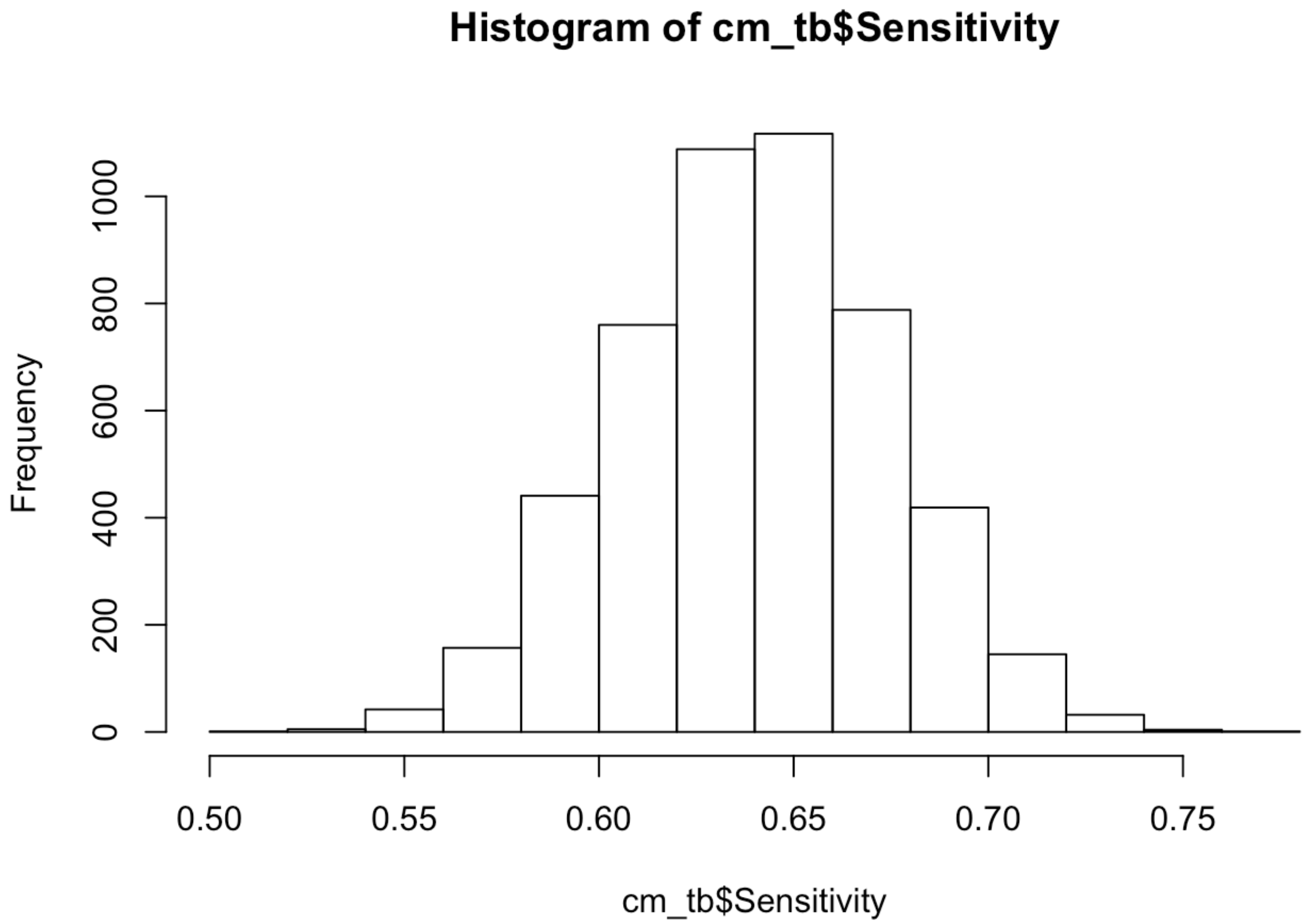
```
##          Sensitivity          Specificity       Pos Pred Value
##            0.6600000            0.7250000            0.7058824
##       Neg Pred Value            Precision               Recall
##            0.6807512            0.7058824            0.6600000
##                   F1           Prevalence       Detection Rate
##            0.6821705            0.5000000            0.3300000
## Detection Prevalence    Balanced Accuracy
##            0.4675000            0.6925000
```

```
for (t in 1:5000){
  truedf <- df[df$y==1,]
  truedf <- truedf[sample(1:nrow(truedf), 200, replace = TRUE), ]
  falsedf <- df[df$y==0,]
  falsedf <- falsedf[sample(1:nrow(falsedf), 200, replace = TRUE), ]

  finaldf <- rbind(truedf, falsedf)
  glm.fit <- glm(y ~ ., data = finaldf, family = binomial)


  summary(glm.fit)

  cm <- confusionMatrix(table(as.numeric(glm.fit$fitted.values>0.5), finaldf$y))
  cm_tb <- rbind(cm_tb, as.data.frame(t(cm$byClass)))
}
```

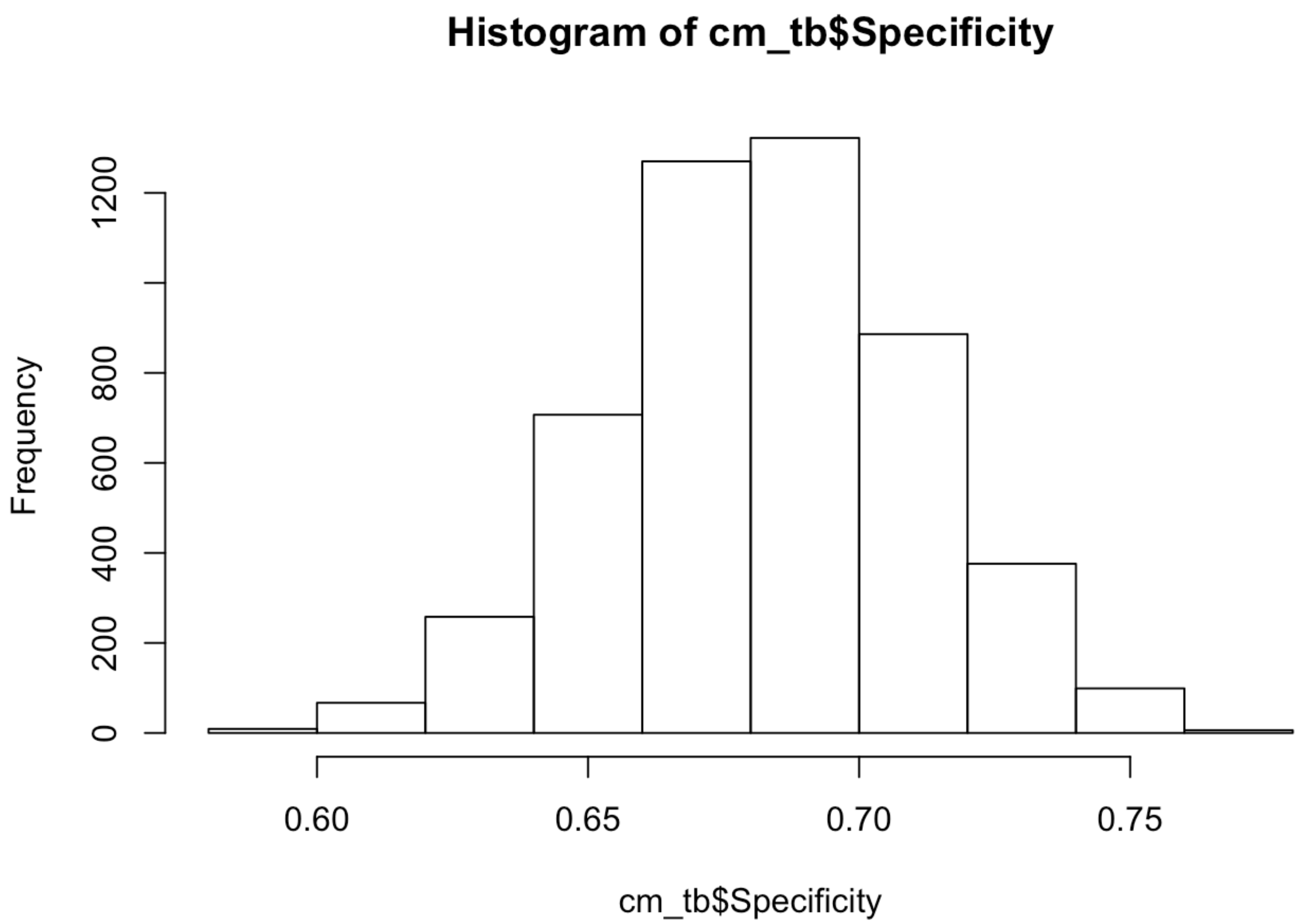
```
hist(cm_tb$`Balanced Accuracy`)
```



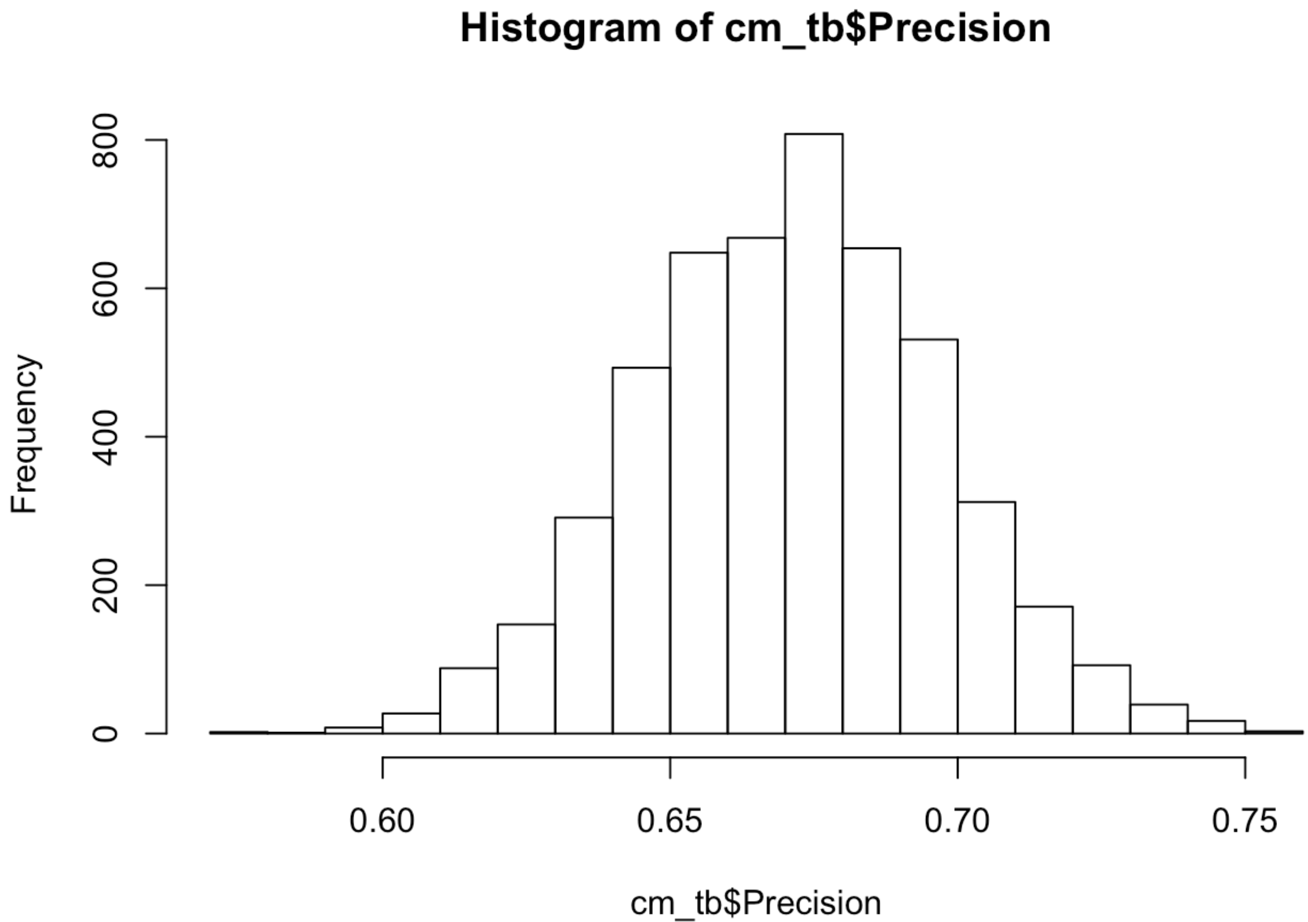

```
hist(cm_tb$Sensitivity)
```



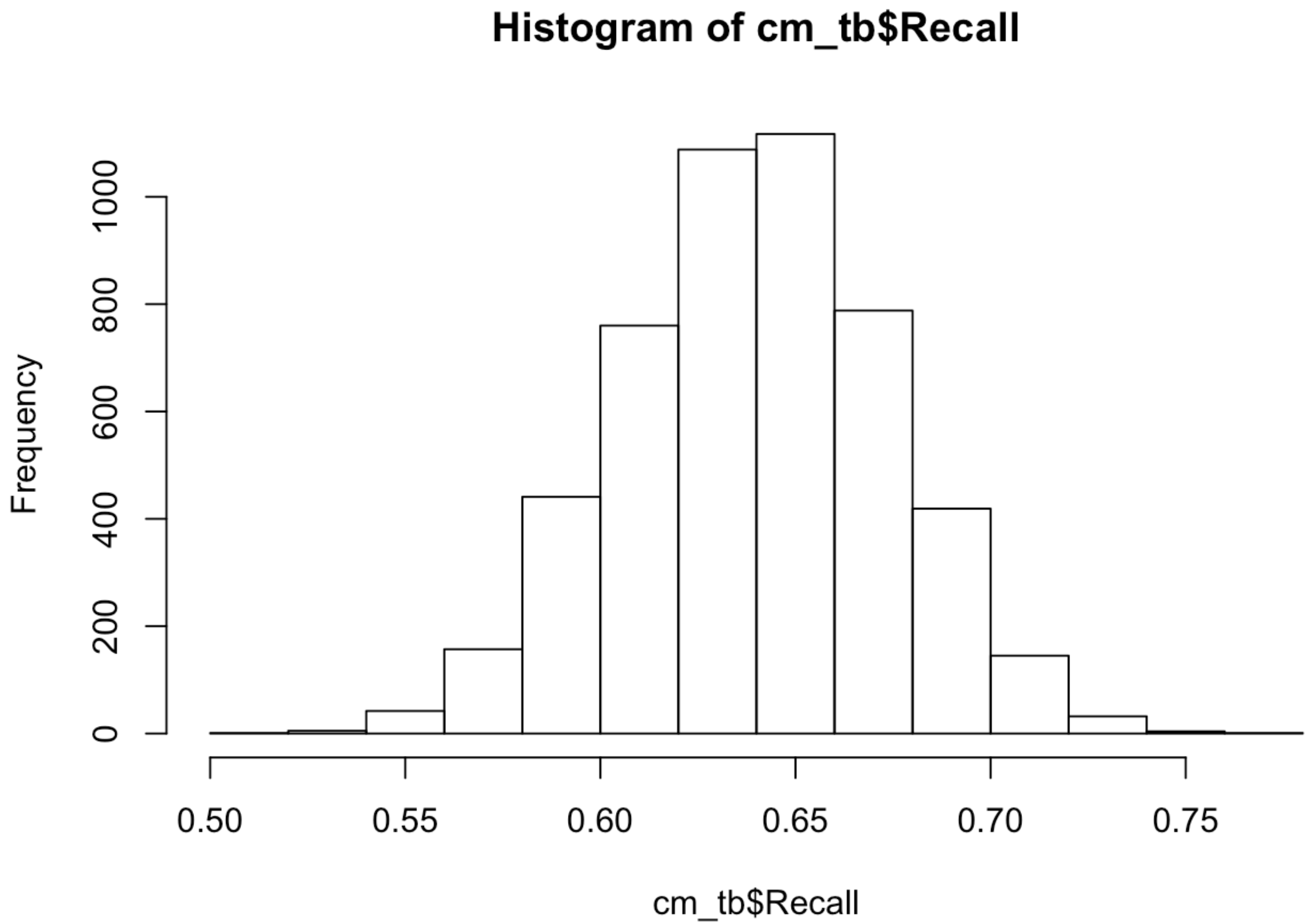

```
hist(cm_tb$Specificity)
```



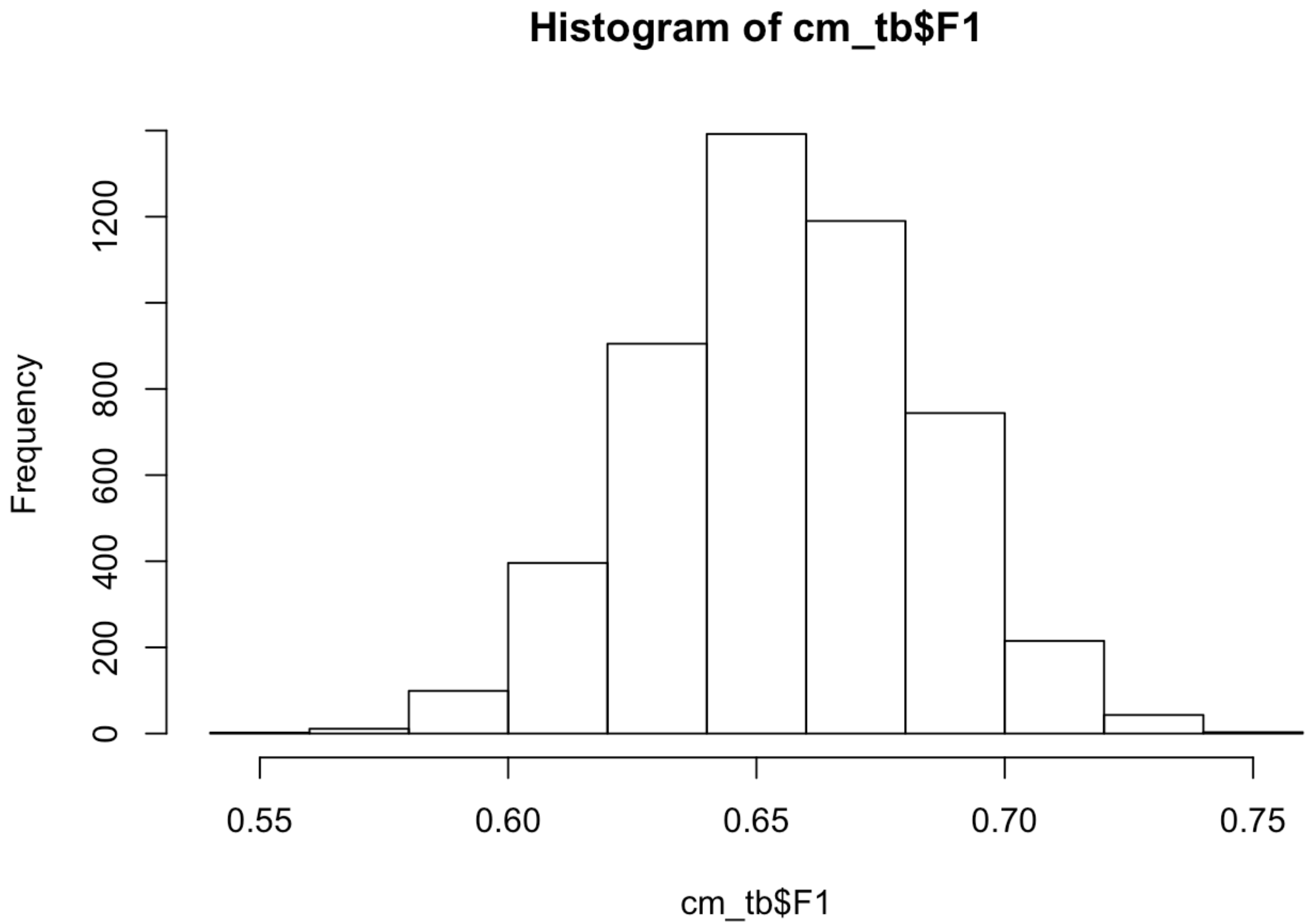

```
hist(cm_tb$Precision)
```




```
hist(cm_tb$Recall)
```


Histogram of cm_tb$Recall
Frequency
0
200
400
600
800
1000
0.50
0.55
0.60
0.65
0.70
0.75
cm_tb$Recall

```
hist(cm_tb$F1)
```


Histogram of cm_tb$F1
Frequency
0
200
400
600
800
1200
0.55
0.60
0.65
0.70
0.75
cm_tb$F1