Instagram fake user auditor

Code

Hide

```
library(rpart.plot)
```

```
Loading required package: rpart
```

Hide

```
test <- read.csv("test.csv")
train <- read.csv(("train.csv"))

dim(test)
```

```
[1] 120  12
```

Hide

```
dim(train)
```

```
[1] 576  12
```

Hide

```
summary(train)
```

```
  profile.pic     nums.length.username fullname.words  nums.length.fullname name..username    de
scription.length  external.URL
 Min.   :0.0000   Min.   :0.0000       Min.   : 0.00   Min.   :0.00000      Min.   :0.00000   Mi
n.   :  0.00     Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000       1st Qu.: 1.00   1st Qu.:0.00000      1st Qu.:0.00000   1s
t Qu.:  0.00     1st Qu.:0.0000
 Median :1.0000   Median :0.0000       Median : 1.00   Median :0.00000      Median :0.00000   Me
dian :  0.00     Median :0.0000
 Mean   :0.7014   Mean   :0.1638       Mean   : 1.46   Mean   :0.03609      Mean   :0.03472   Me
an   : 22.62     Mean   :0.1163
 3rd Qu.:1.0000   3rd Qu.:0.3100       3rd Qu.: 2.00   3rd Qu.:0.00000      3rd Qu.:0.00000   3r
d Qu.: 34.00     3rd Qu.:0.0000
 Max.   :1.0000   Max.   :0.9200       Max.   :12.00   Max.   :1.00000      Max.   :1.00000   Ma
x.   :150.00     Max.   :1.0000
    private          X.posts        X.followers         X.follows           fake
 Min.   :0.0000   Min.   :   0.0   Min.   :       0   Min.   :   0.0   Min.   :0.0
 1st Qu.:0.0000   1st Qu.:   0.0   1st Qu.:      39   1st Qu.:  57.5   1st Qu.:0.0
 Median :0.0000   Median :   9.0   Median :     150   Median : 229.5   Median :0.5
 Mean   :0.3819   Mean   : 107.5   Mean   :   85307   Mean   : 508.4   Mean   :0.5
 3rd Qu.:1.0000   3rd Qu.:  81.5   3rd Qu.:     716   3rd Qu.: 589.5   3rd Qu.:1.0
 Max.   :1.0000   Max.   :7389.0   Max.   :15338538   Max.   :7500.0   Max.   :1.0
```

Hide

```
colnames(test)
```

```
 [1] "profile.pic"          "nums.length.username" "fullname.words"       "nums.length.fullname"
"name..username"
 [6] "description.length"   "external.URL"         "private"              "X.posts"
"X.followers"
[11] "X.follows"            "fake"
```

Hide

```
colnames(train)
```

```
 [1] "profile.pic"          "nums.length.username" "fullname.words"       "nums.length.fullname"
"name..username"
 [6] "description.length"   "external.URL"         "private"              "X.posts"
"X.followers"
[11] "X.follows"            "fake"
```

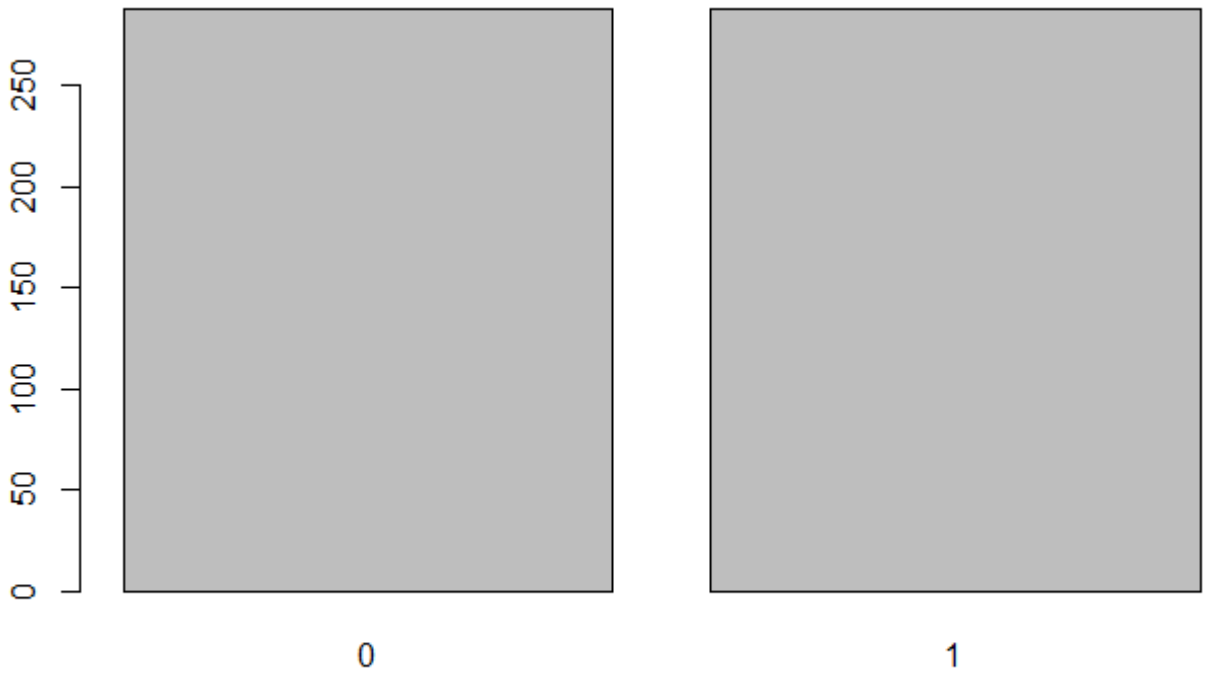
Hide

```
table(train['fake'])
```

```
  0   1
288 288
```

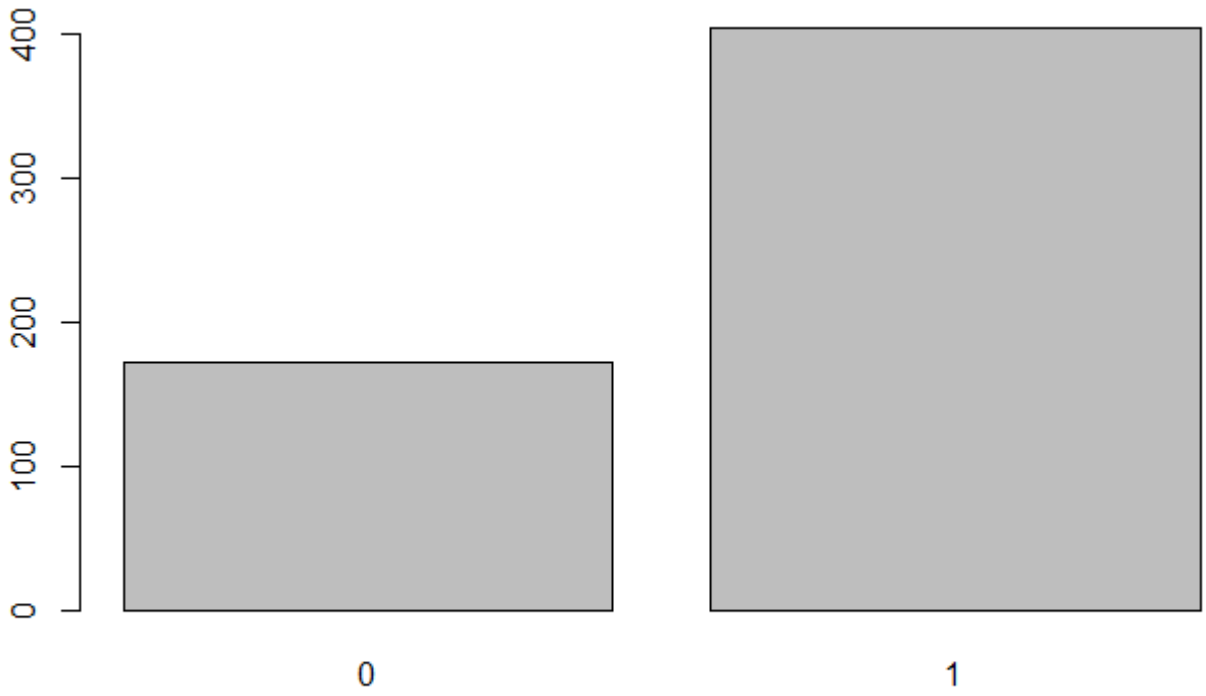
Hide

```
barplot(table(train['fake']))
```



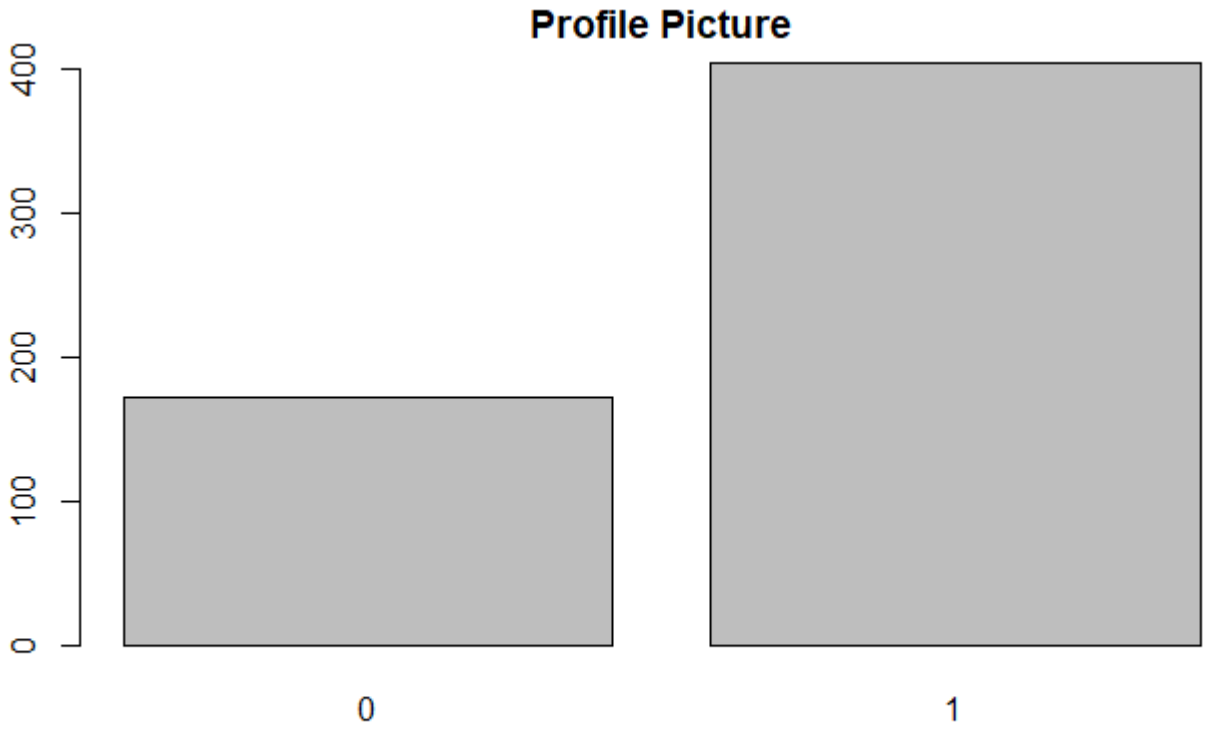
Hide

```
barplot(table(train['profile.pic']))
```



Hide

```
barplot(table(train['profile.pic']), main  = "Profile Picture")
```

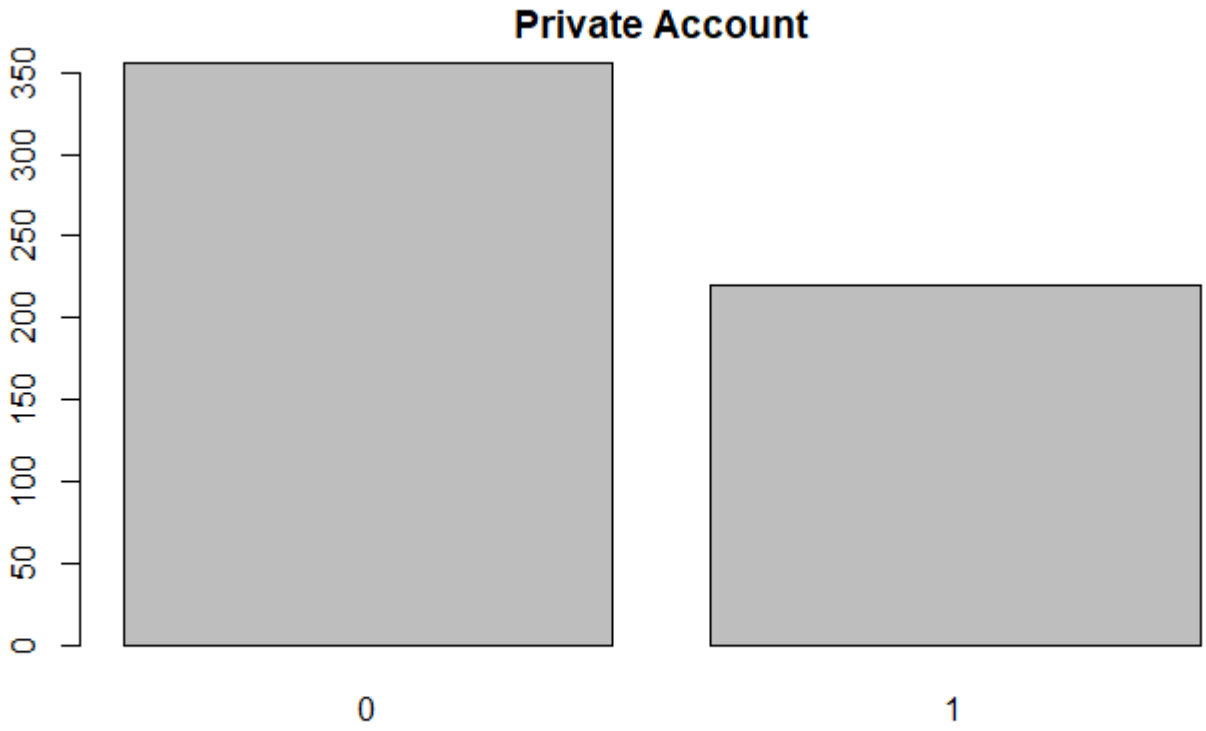


Hide

```
barplot(table(train['external.URL']), main  = "External URL in profile")
```

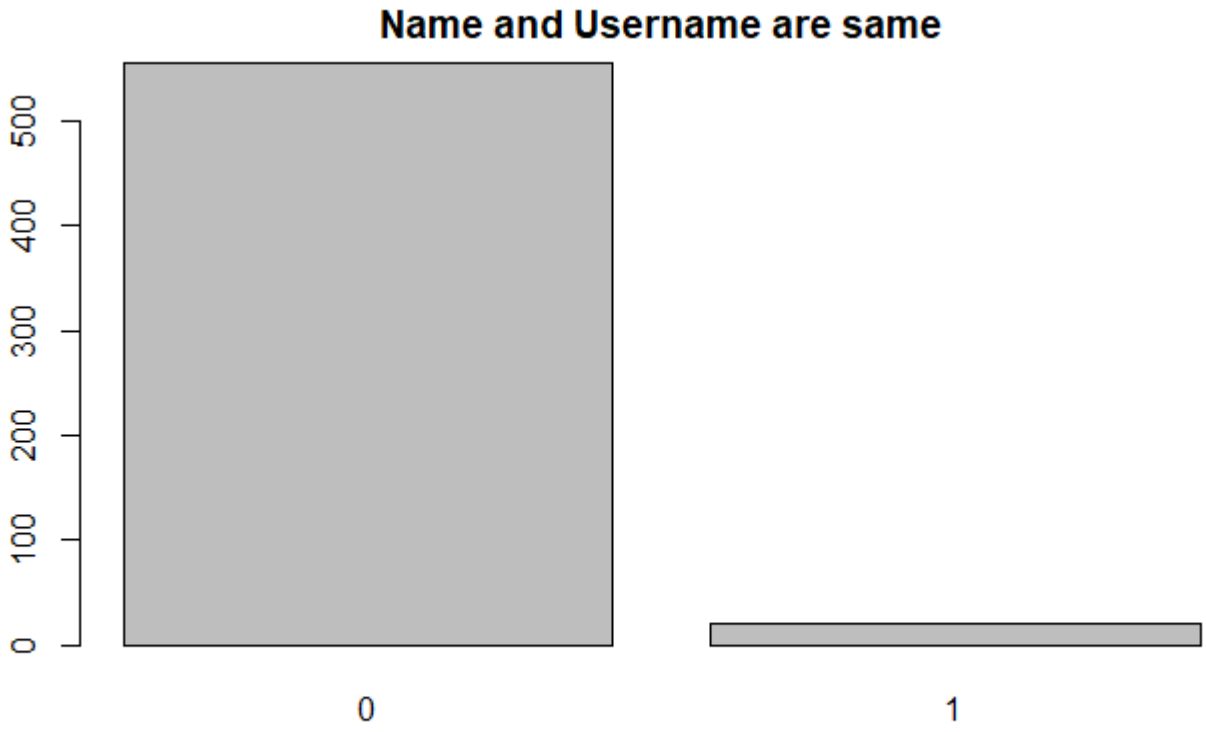
Hide

```
barplot(table(train['private']), main  = "Private Account")
```



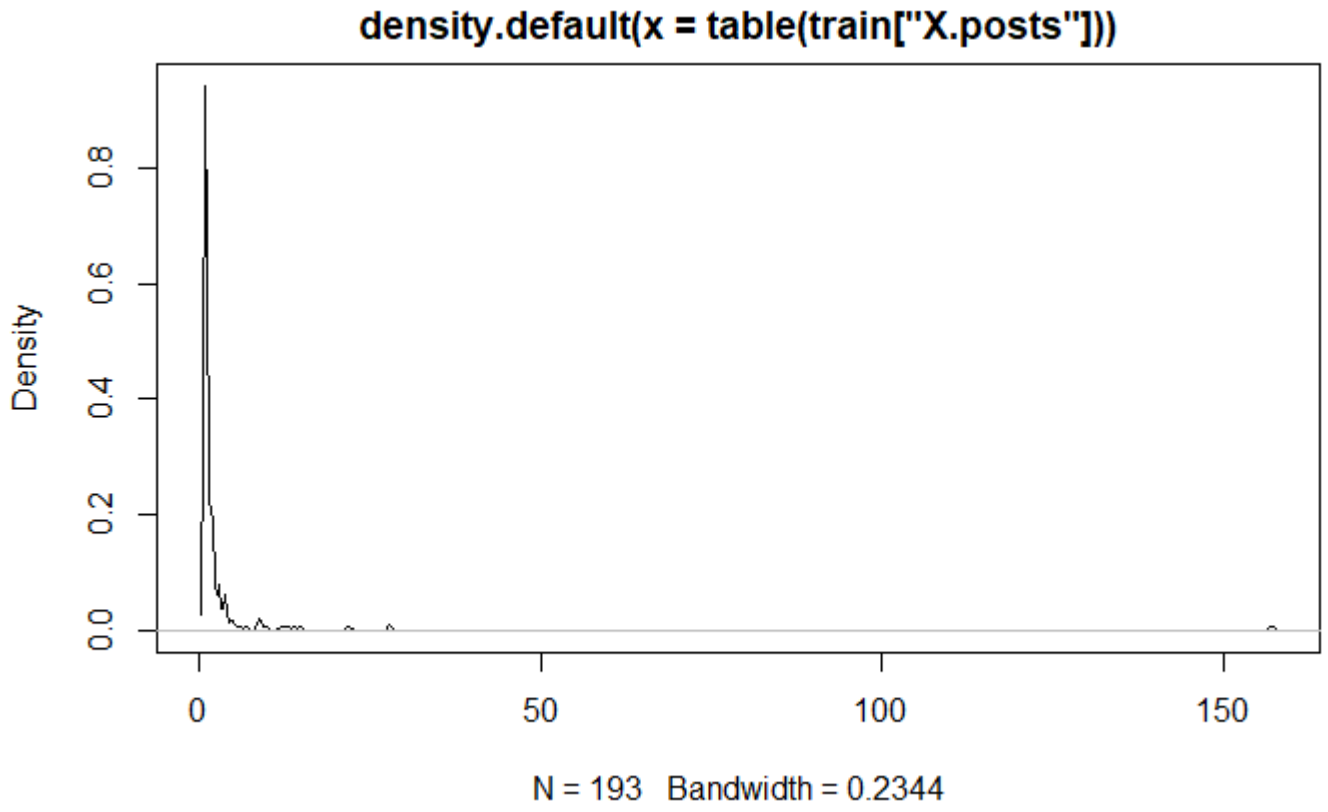
Hide

```
barplot(table(train['name..username']), main  = "Name and Username are same")
```



Hide

```
plot(density(table(train["X.posts"])))
```

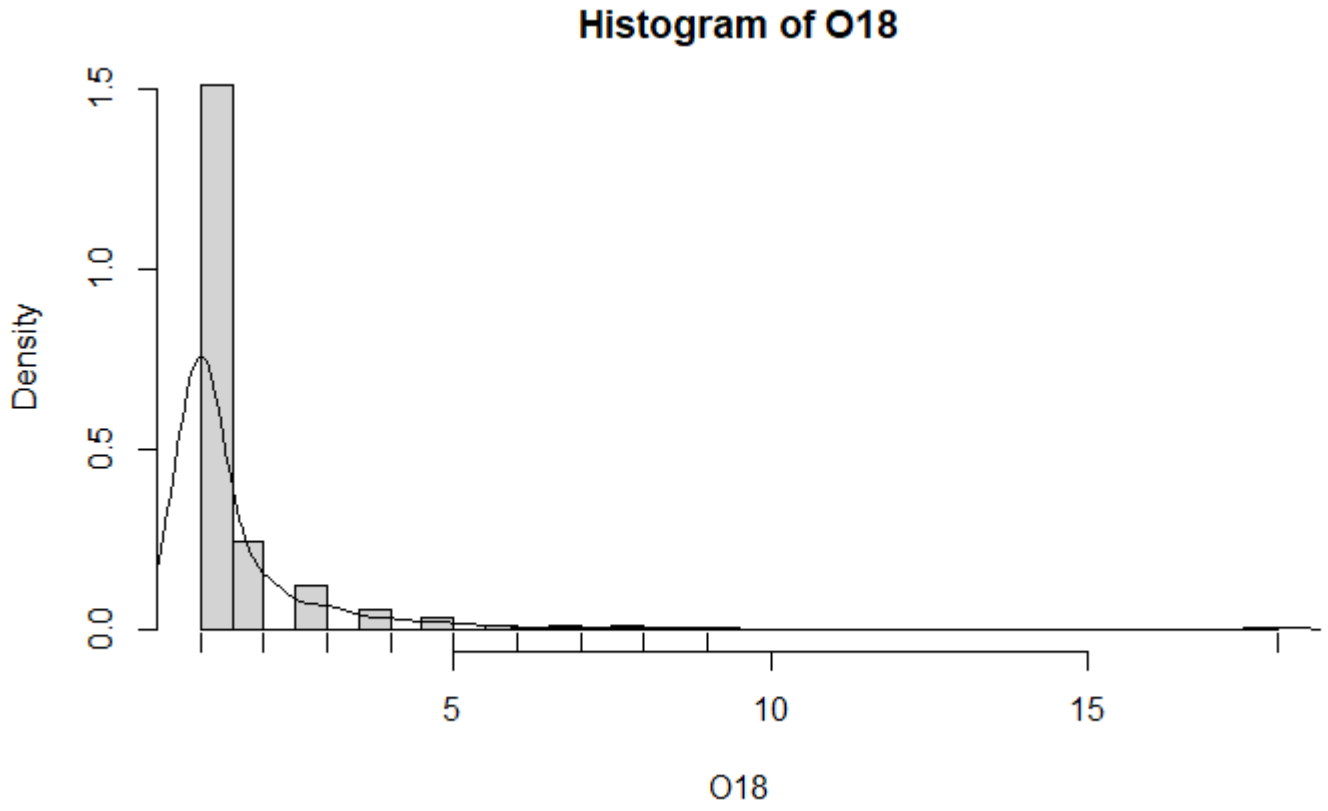


Hide

```
O18 = table(train["X.followers"])

O18.density <- density(O18)
hist(O18, breaks=40, probability=TRUE)
lines(O18.density)
```

Hide

```
rug(O18)
```

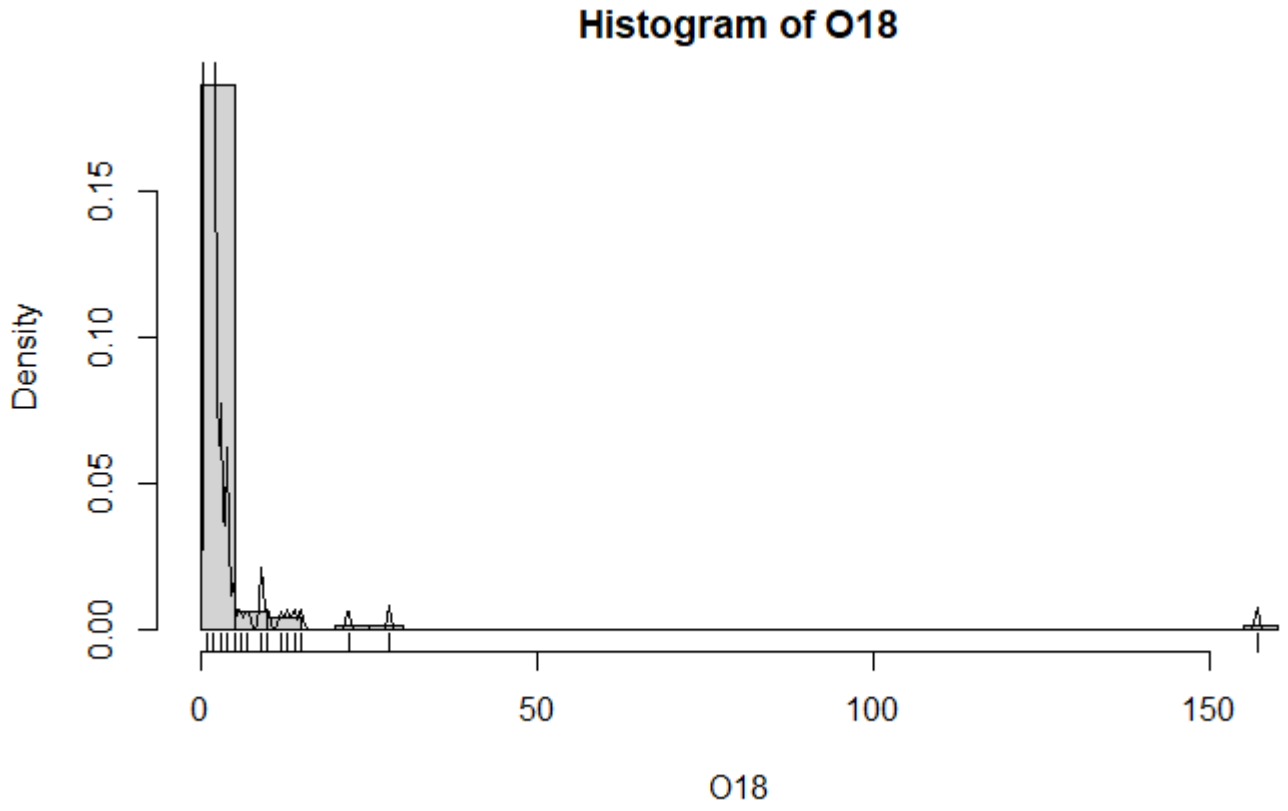


Hide

```
O18 = table(train["X.posts"])

O18.density <- density(O18)
hist(O18, breaks=40, probability=TRUE)
lines(O18.density)
```

Hide

```
rug(O18)
```

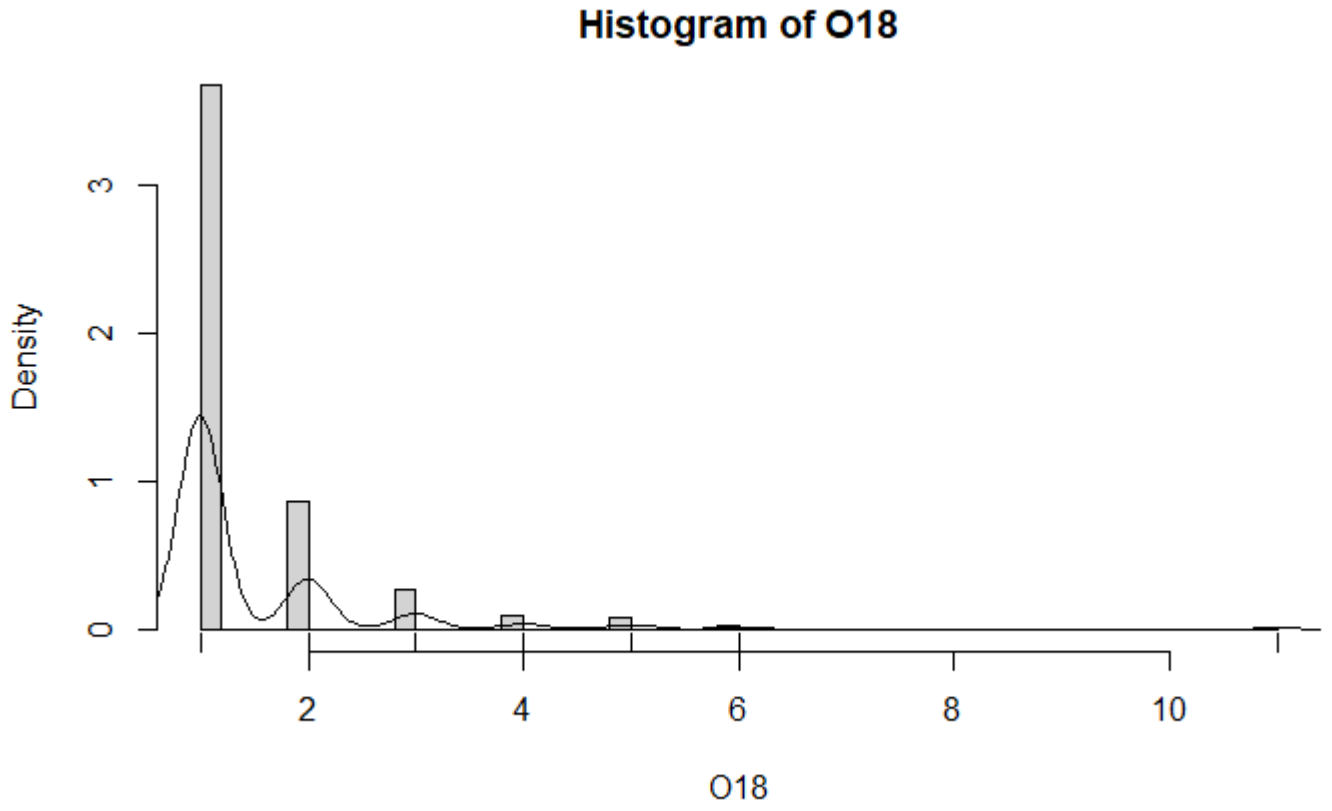


Hide

```
O18 = table(train["X.follows"])

O18.density <- density(O18)
hist(O18, breaks=40, probability=TRUE)
lines(O18.density)
```

Hide

```
rug(O18)
```

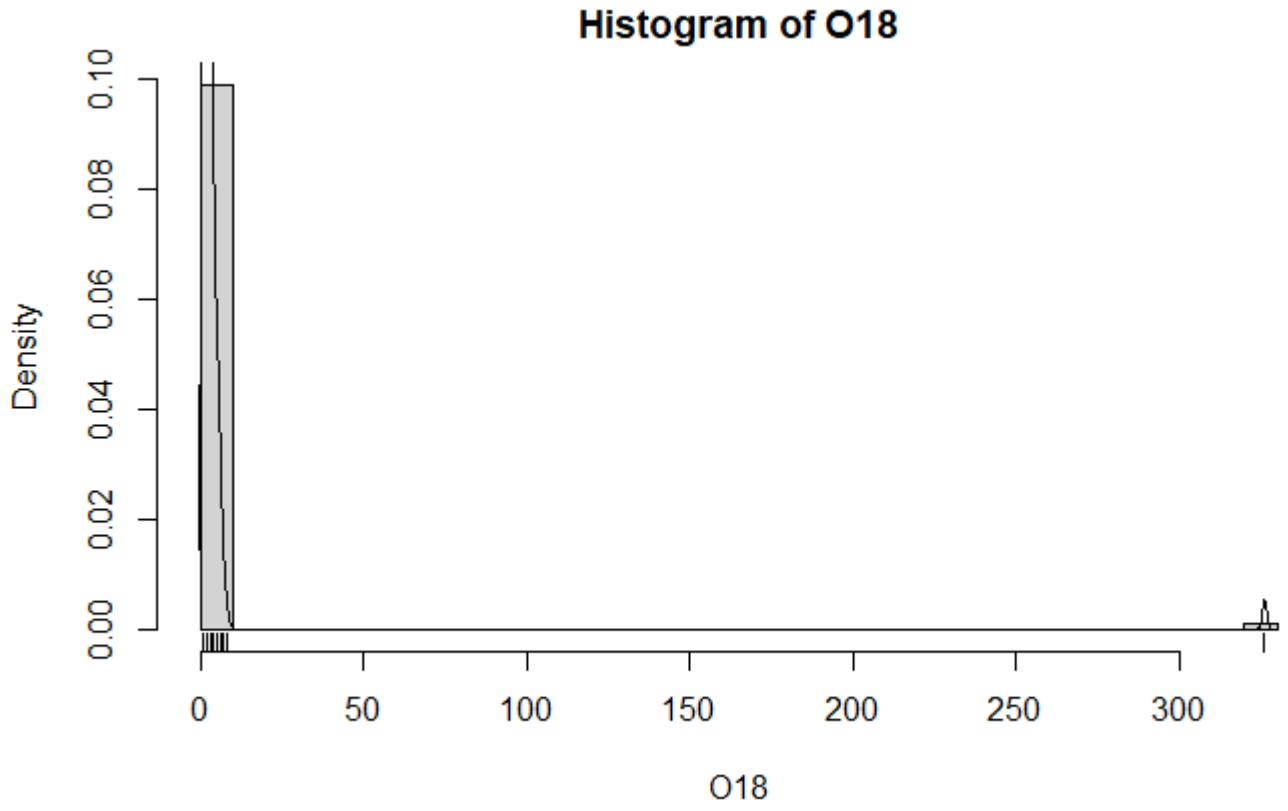


Hide

```
O18 = table(train["description.length"])

O18.density <- density(O18)
hist(O18, breaks=40, probability=TRUE)
lines(O18.density)
```

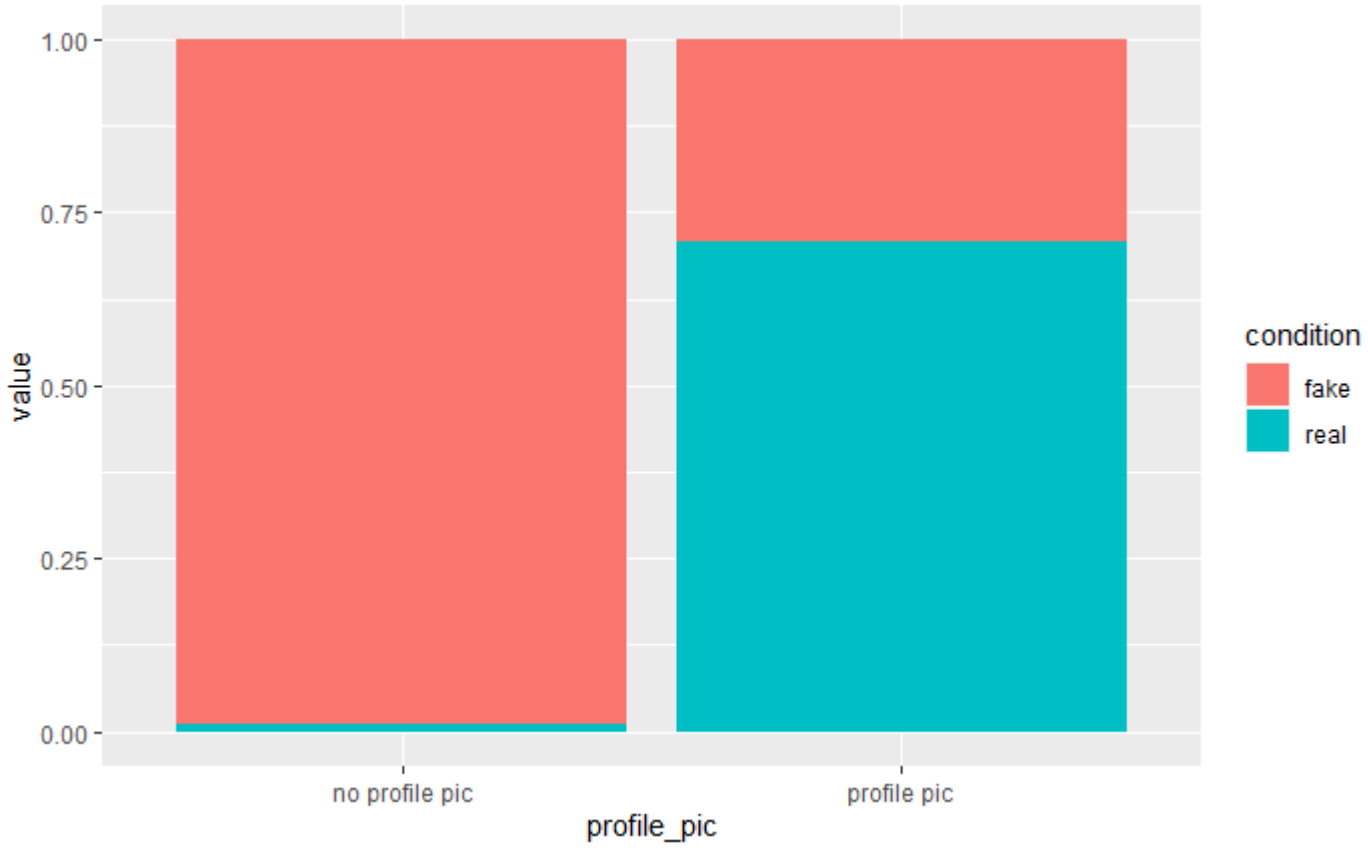
Hide

```
rug(O18)
```



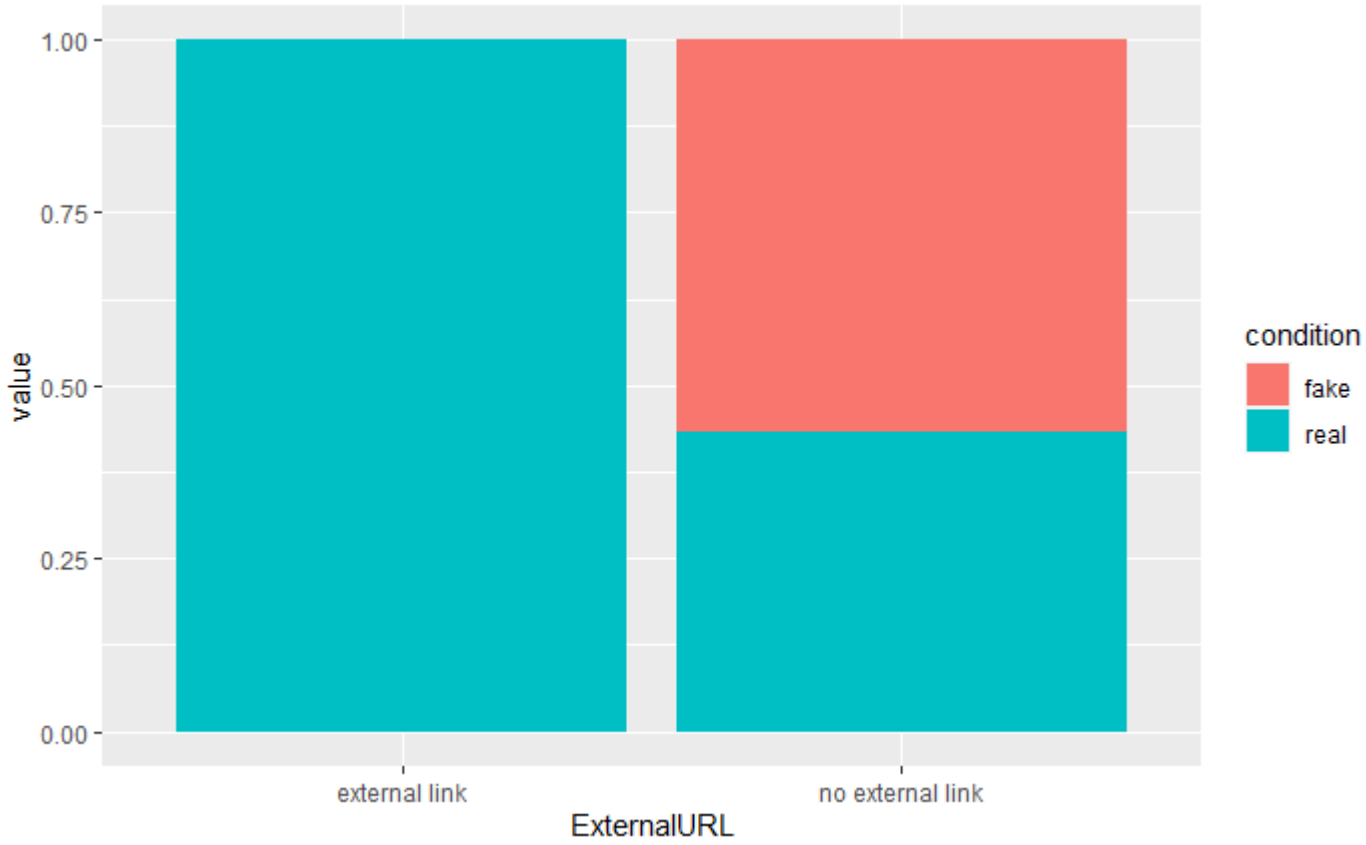
Hide

```
#specie <- c(rep("sorgho" , 3) , rep("poacee" , 3) , rep("banana" , 3) , rep("triticum" , 3) )
#condition <- rep(c("normal" , "stress" , "Nitrogen") , 4)
#value <- abs(rnorm(12 , 0 , 15))
#data <- data.frame(table(train["profile.pic"]), table(train["fake"]))

# Stacked
#ggplot(data, aes(fill=condition, y=value, x=specie)) + geom_bar(position="stack", stat="identit
y")

profile_pic <- c(rep("profile pic" , 2) , rep("no profile pic" , 2) )
condition <- rep(c("fake" , "real") , 2)
value <- c(nrow(train[train$profile.pic != "0"& train$fake != "0",]), nrow(train[train$profile.p
ic != "0"& train$fake != "1",]), nrow(train[train$profile.pic != "1"& train$fake != "0",]), nrow
(train[train$profile.pic != "1"& train$fake != "1",]))
data <- data.frame(profile_pic,condition,value)

# Stacked
ggplot(data, aes(fill=condition, y=value, x=profile_pic)) +
  geom_bar(position="fill", stat="identity")
```



Hide

```
ExternalURL <- c(rep("external link" , 2) , rep("no external link" , 2) )
condition <- rep(c("fake" , "real") , 2)
value <- c(nrow(train[train$external.URL != "0"& train$fake != "0",]), nrow(train[train$externa
l.URL != "0"& train$fake != "1",]), nrow(train[train$external.URL != "1"& train$fake != "0",]),
 nrow(train[train$external.URL != "1"& train$fake != "1",]))
data <- data.frame(ExternalURL,condition,value)

# Stacked
ggplot(data, aes(fill=condition, y=value, x=ExternalURL)) +
  geom_bar(position="fill", stat="identity")
```



Hide

```
private <- c(rep("private" , 2) , rep("public" , 2) )
condition <- rep(c("fake" , "real") , 2)
value <- c(nrow(train[train$private != "0"& train$fake != "0",]), nrow(train[train$private !=
"0"& train$fake != "1",]), nrow(train[train$private != "1"& train$fake != "0",]), nrow(train[tra
in$private != "1"& train$fake != "1",]))
data <- data.frame(private,condition,value)

# Stacked
ggplot(data, aes(fill=condition, y=value, x=private)) +
  geom_bar(position="fill", stat="identity")
```

Hide

```
nameisusername <- c(rep("name==username" , 2) , rep("name!=username" , 2) )
condition <- rep(c("fake" , "real") , 2)
value <- c(nrow(train[train$name..username != "0"& train$fake != "0",]), nrow(train[train$name..
username != "0"& train$fake != "1",]), nrow(train[train$name..username != "1"& train$fake != "0"
,]), nrow(train[train$name..username != "1"& train$fake != "1",]))
data <- data.frame(nameisusername,condition,value)

# Stacked
ggplot(data, aes(fill=condition, y=value, x=nameisusername)) +
  geom_bar(position="fill", stat="identity")
```

Hide

```
NA
NA
NA
NA
```

Hide

```
# Save the file.
train$fake = as.factor(train$fake)
intrain <- createDataPartition(y = train$fake, p= 0.7, list = FALSE)
training <- train[intrain,]
testing <- train[-intrain,]
dim(training); dim(testing);
```

```
[1] 404  12
[1] 172  12
```

Hide

```
anyNA(train)
```

```
[1] FALSE
```

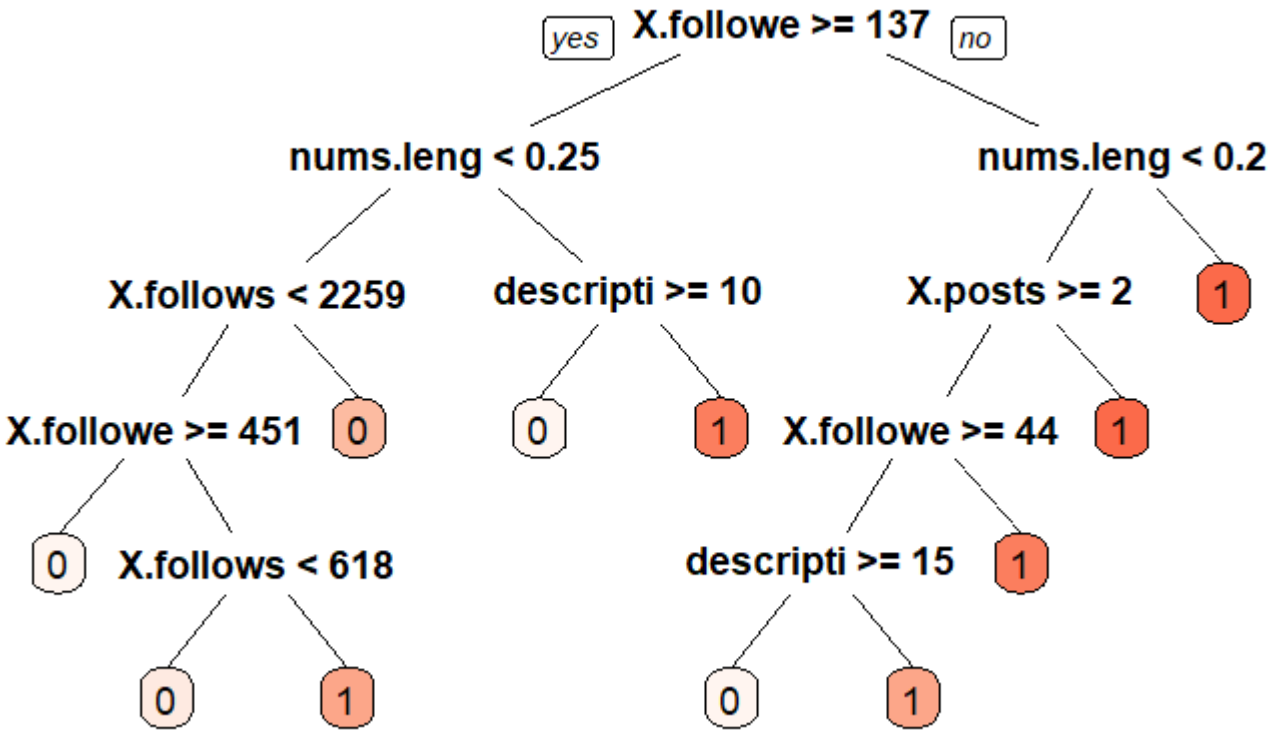
Hide

```
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(3333)
dtree_fit <- train(fake  ~., data = training, method = "rpart",
                   parms = list(split = "information"),
                   trControl=trctrl,
                   tuneLength = 10)
dtree_fit
```

```
CART

404 samples
 11 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 364, 364, 363, 364, 364, 364, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.00000000  0.8827391  0.7654534
  0.08360836  0.8680043  0.7359401
  0.16721672  0.8680043  0.7359401
  0.25082508  0.8680043  0.7359401
  0.33443344  0.8680043  0.7359401
  0.41804180  0.8680043  0.7359401
  0.50165017  0.8680043  0.7359401
  0.58525853  0.8680043  0.7359401
  0.66886689  0.8680043  0.7359401
  0.75247525  0.6497967  0.3044439

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.
```

Hide

```
prp(dtree_fit$finalModel, box.palette = "Reds", tweak = 1.2)
```



Hide

```
predict(dtree_fit, newdata = testing[1,])
```

```
[1] 0
Levels: 0 1
```

Hide

```
testing[1,]
```

profile.pic <int>	nums.length.username <dbl>	fullname.words <int>	nums.length.fullname <dbl>	name..usernar <ir
2	1	0	2	0

1 row | 1-6 of 12 columns

Hide

```
predict(dtree_fit, newdata = testing[1,])
```

```
[1] 0
Levels: 0 1
```

Hide

```
test_pred <- predict(dtree_fit, newdata = testing)
confusionMatrix(factor(test_pred, levels=0:1), factor(testing$fake, levels=0:1) )  #check accura
cy
```

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 69  6
         1 17 80

               Accuracy : 0.8663
                 95% CI : (0.8061, 0.9133)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.7326

 Mcnemar's Test P-Value : 0.03706

            Sensitivity : 0.8023
            Specificity : 0.9302
         Pos Pred Value : 0.9200
         Neg Pred Value : 0.8247
             Prevalence : 0.5000
         Detection Rate : 0.4012
   Detection Prevalence : 0.4360
      Balanced Accuracy : 0.8663

       'Positive' Class : 0
```