Mental Health in Tech

Inlcude all the libraries here:

```
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(adabag)
library(leaps)
library(data.table)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(ggplot2)
library(adabag)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```
## corrplot 0.84 loaded
```

```
library(caret)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(tree)
library(car)
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
library(rpart)
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.2
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```
##
## Attaching package: 'glmnet'
```

```
## The following object is masked from 'package:pROC':
##
##     auc
```

```r
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(corrplot)
```

Exploratory Data Analysis:

Let us first read and understand the data:

```
datahealth <- read.csv("survey.csv", header=T)
```

```
summary(datahealth)
```

```
##                Timestamp         Age                  Gender
##  2014-08-27 12:31:41:   2   Min.   :-1.726e+03   Male   :615
##  2014-08-27 12:37:50:   2   1st Qu.: 2.700e+01   male   :206
##  2014-08-27 12:43:28:   2   Median : 3.100e+01   Female :121
##  2014-08-27 12:44:51:   2   Mean   : 7.943e+07   M      :116
##  2014-08-27 12:54:11:   2   3rd Qu.: 3.600e+01   female : 62
##  2014-08-27 14:22:43:   2   Max.   : 1.000e+11   F      : 38
##  (Other)            :1247                        (Other):101
##            Country        state     self_employed family_history treatment
##  United States :751   CA     :138   No  :1095     No :767        No :622
##  United Kingdom:185   WA     : 70   Yes : 146     Yes:492        Yes:637
##  Canada        : 72   NY     : 57   NA's:  18
##  Germany       : 45   TN     : 45
##  Ireland       : 27   TX     : 44
##  Netherlands   : 27   (Other):390
##  (Other)       :152   NA's   :515
##    work_interfere         no_employees remote_work tech_company
##  Never    :213    1-5           :162   No :883     No : 228
##  Often    :144    100-500       :176   Yes:376     Yes:1031
##  Rarely   :173    26-100        :289
##  Sometimes:465    500-1000      : 60
##  NA's     :264    6-25          :290
##                   More than 1000:282
##
##       benefits     care_options   wellness_program      seek_help
##  Don't know:408   No      :501   Don't know:188     Don't know:363
##  No        :374   Not sure:314   No        :842     No        :646
##  Yes       :477   Yes     :444   Yes       :229     Yes       :250
##
##
##
##
##      anonymity                    leave     mental_health_consequence
##  Don't know:819   Don't know         :563   Maybe:477
##  No        : 65   Somewhat difficult:126   No   :490
##  Yes       :375   Somewhat easy      :266   Yes  :292
##                   Very difficult     : 98
```

```
##                    Very easy         :206
##
##
##  phys_health_consequence        coworkers          supervisor
##  Maybe:273               No           :260   No           :393
##  No   :925               Some of them:774   Some of them:350
##  Yes  : 61               Yes          :225   Yes          :516
##
##
##
##
##  mental_health_interview phys_health_interview  mental_vs_physical
##  Maybe: 207              Maybe:557              Don't know:576
##  No   :1008              No   :500              No        :340
##  Yes  :  44              Yes  :202              Yes       :343
##
##
##
##
##  obs_consequence
##  No :1075
##  Yes: 184
##
##
##
##
##
##
comments
##  * Small family business - YMMV.
:   5
##
:   1
##  -
:   1
##  (yes but the situation was unusual and involved a change in leadership at a very
high level in the organization as well as an extended leave of absence)
:   1
##  A close family member of mine struggles with mental health so I try not to stigma
tize it. My employers/coworkers also seem compassionate toward any kind of health or
family needs.:   1
##  (Other)
: 155
##  NA's
:1095
```

```
data1 <- datahealth
dim(datahealth)
```

```
## [1] 1259   27
```

```
names(data1)
```

```
##  [1] "Timestamp"                 "Age"
##  [3] "Gender"                    "Country"
##  [5] "state"                     "self_employed"
##  [7] "family_history"            "treatment"
##  [9] "work_interfere"            "no_employees"
## [11] "remote_work"               "tech_company"
## [13] "benefits"                  "care_options"
## [15] "wellness_program"          "seek_help"
## [17] "anonymity"                 "leave"
## [19] "mental_health_consequence" "phys_health_consequence"
## [21] "coworkers"                 "supervisor"
## [23] "mental_health_interview"   "phys_health_interview"
## [25] "mental_vs_physical"        "obs_consequence"
## [27] "comments"
```

```
#CHANGE!!!!
GenMale <- c("cis male", "Cis Male", "Cis Man", "m", "M", "Mail", "maile", "Make", "M
al", "male", "Male", "Male ", "Male (CIS)", "Malr", "Man", "msle")
GenFemale <- c("cis-female/femme", "Cis Female", "f", "F", "femail", "Femake", "femal
e", "Female", "Female ", "Female (cis)", "Female (trans)", "Trans-female", "Trans wom
an", "woman", "Woman")

# Assigning the entries according to "categories"
data1$newgender <-
  ifelse((data1$Gender %in% GenMale), "Male", # Assigning "Male" to those who entered
a string contained in GenMale
  ifelse((data1$Gender %in% GenFemale), "Female", "Non-M/F")) %>% # Assigning "Female
" to those who entered a string contained in GenFemale
  as.factor()

# Observing cleaned table
table(data1$newgender)
```

```
##
##  Female    Male Non-M/F
##     251     990      18
```

```
#Clean the age column to eliminate spurious values like negatives and ages above 120
data1 = data1[(data1$Age > 15) & (data1$Age < 120),]
dim(data1)
```

```
## [1] 1251   28
```

```
data1 = subset(data1, select=-c(Gender, Timestamp, comments))
data1 <- data1 %>% rename(Gender = newgender )
names(data1)
```

```
##  [1] "Age"                       "Country"
##  [3] "state"                     "self_employed"
##  [5] "family_history"            "treatment"
##  [7] "work_interfere"            "no_employees"
##  [9] "remote_work"               "tech_company"
## [11] "benefits"                  "care_options"
## [13] "wellness_program"          "seek_help"
## [15] "anonymity"                 "leave"
## [17] "mental_health_consequence" "phys_health_consequence"
## [19] "coworkers"                 "supervisor"
## [21] "mental_health_interview"   "phys_health_interview"
## [23] "mental_vs_physical"        "obs_consequence"
## [25] "Gender"
```

```
#na.omit(data1)
dim(data1)
```

```
## [1] 1251   25
```

```
sapply(data1, class)
```

```
##                       Age                  Country
##                 "numeric"                 "factor"
##                     state            self_employed
##                  "factor"                 "factor"
##            family_history                treatment
##                  "factor"                 "factor"
##            work_interfere             no_employees
##                  "factor"                 "factor"
##               remote_work             tech_company
##                  "factor"                 "factor"
##                  benefits              care_options
##                  "factor"                 "factor"
##          wellness_program                seek_help
##                  "factor"                 "factor"
##                 anonymity                    leave
##                  "factor"                 "factor"
## mental_health_consequence   phys_health_consequence
##                  "factor"                 "factor"
##                 coworkers               supervisor
##                  "factor"                 "factor"
##   mental_health_interview     phys_health_interview
##                  "factor"                 "factor"
##        mental_vs_physical           obs_consequence
##                  "factor"                 "factor"
##                    Gender
##                  "factor"
```

```
names(data1)
```

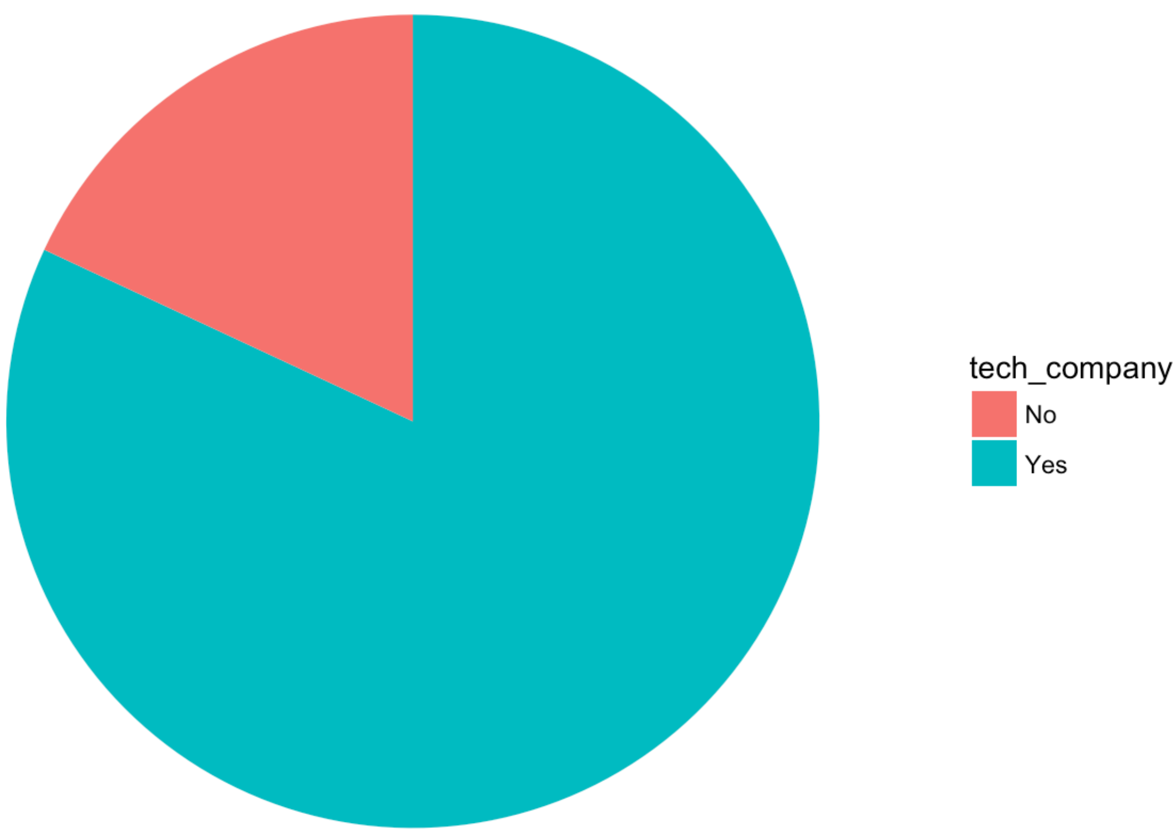
```
##  [1] "Age"                       "Country"
##  [3] "state"                     "self_employed"
##  [5] "family_history"            "treatment"
##  [7] "work_interfere"            "no_employees"
##  [9] "remote_work"               "tech_company"
## [11] "benefits"                  "care_options"
## [13] "wellness_program"          "seek_help"
## [15] "anonymity"                 "leave"
## [17] "mental_health_consequence" "phys_health_consequence"
## [19] "coworkers"                 "supervisor"
## [21] "mental_health_interview"   "phys_health_interview"
## [23] "mental_vs_physical"        "obs_consequence"
## [25] "Gender"
```

```
data1$work_interfere <- as.character(data1$work_interfere)
data1$work_interfere[is.na(data1$work_interfere)] <- "Never"
data1$work_interfere <- as.factor(data1$work_interfere)
summary(data1$work_interfere)
```

```
##     Never     Often    Rarely Sometimes
##       474       140       173       464
```

Let us see the distribution of data with respect to tech and non-tech companies:

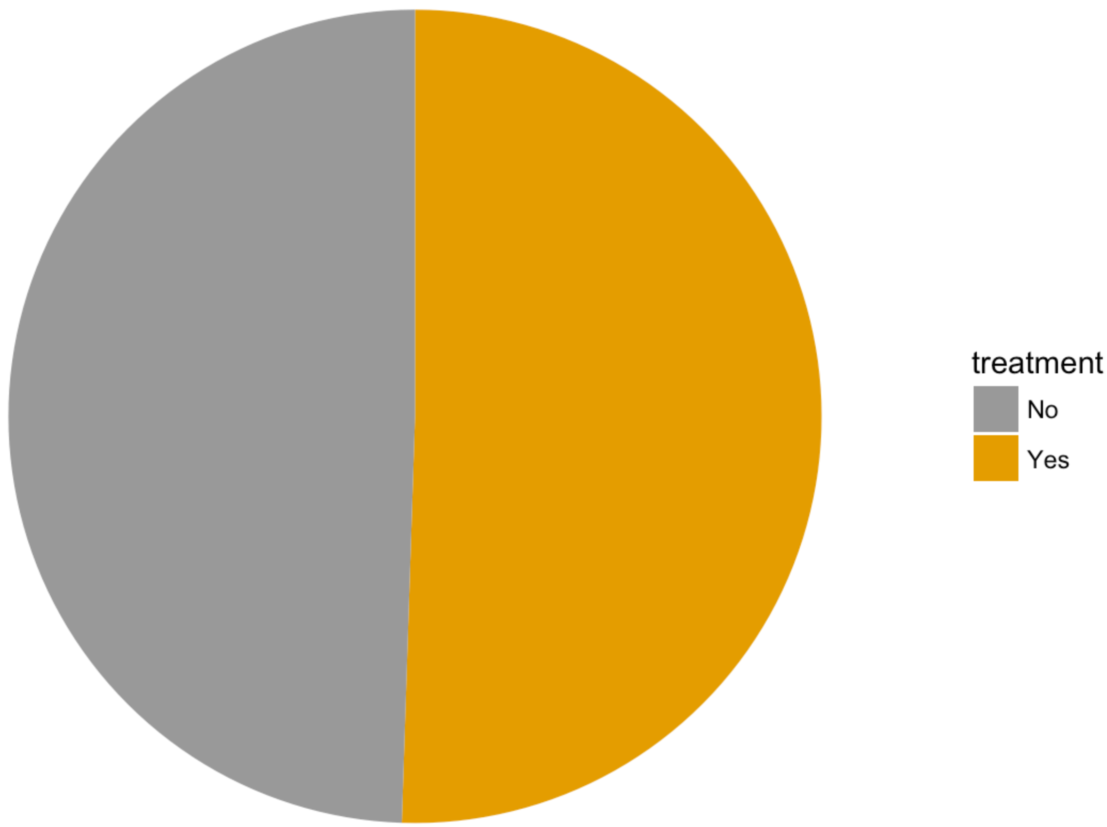
```
bar <- ggplot(data=data1, aes(x = sum(tech_company =="Yes"), fill = tech_company)) +
geom_bar(width = 0.2) +coord_fixed(ratio = 0.2)
pie <- bar + coord_polar("y", start=0) +theme_void()
pie
```



Clearly, our data is skewed in favor of the tech companies.

Let us se the distribution of data with respect to the number individuals seeking treatment for mental illnesses:

```
bar <- ggplot(data=data1, aes(x = sum(treatment =="Yes"), fill = treatment)) + geom_b
ar(width = 0.2) +coord_fixed(ratio = 0.2)
pie <- bar + coord_polar("y", start=0) + theme_void() + scale_fill_manual(values=c("#
999999", "#E69F00"))
pie
```

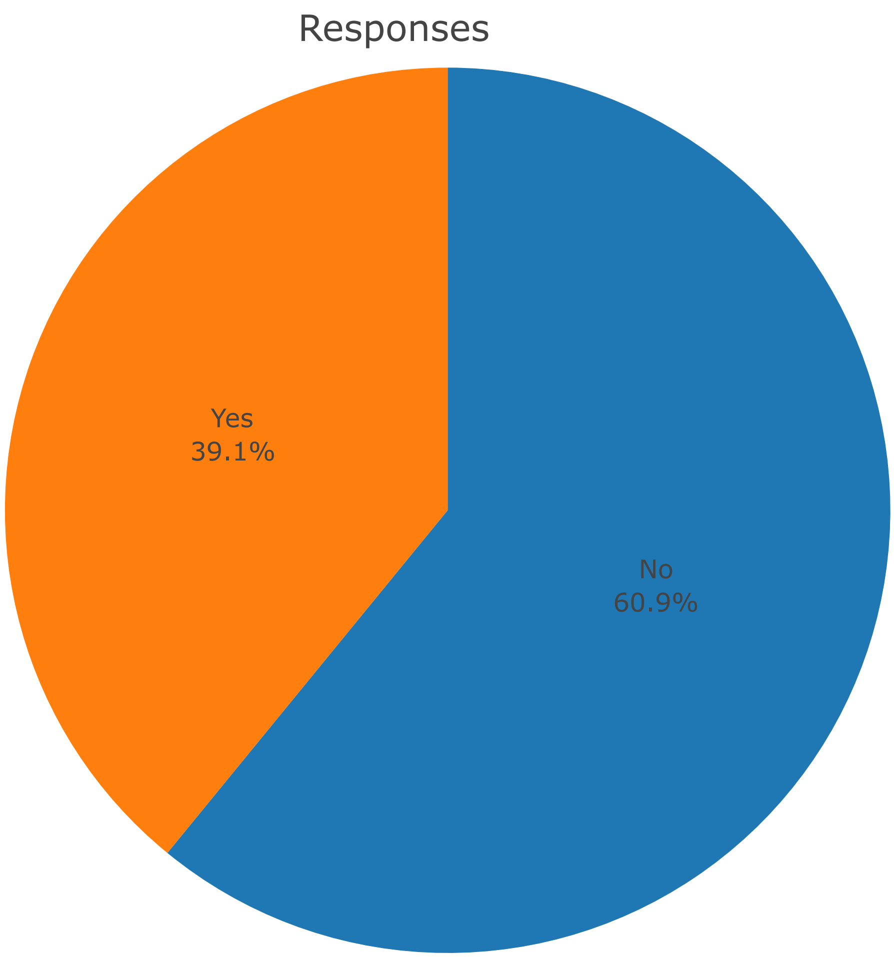


We have close to even distribution of data with respect to individuals seeking treatment.

What is the percentage of folks with a Family history of mental illnesses?

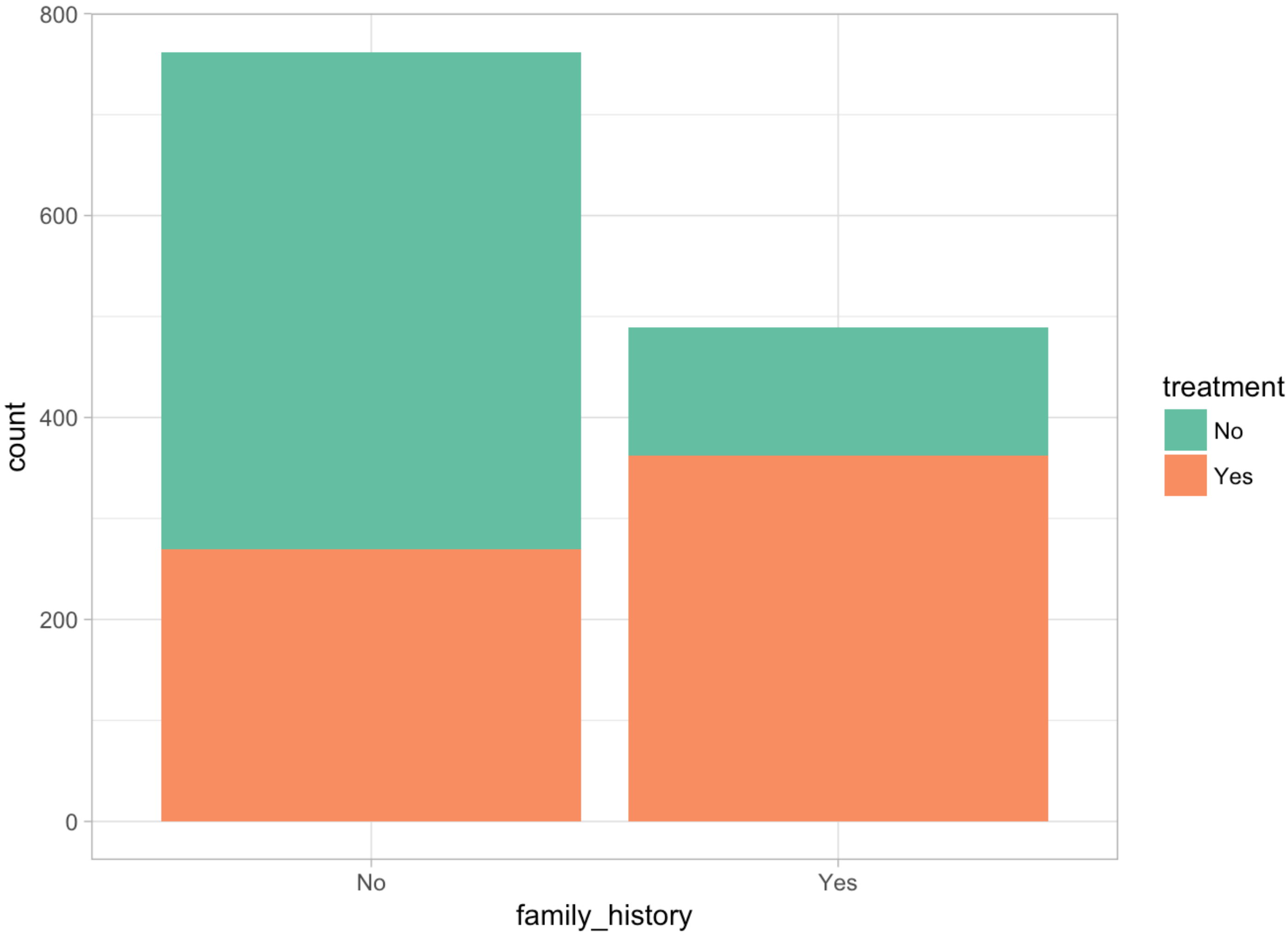
```
colors1 <- c("No" = "#fffff", "Yes" = "qqqqq", "Maybe" = "#11111", "Not sure" = "#111
11", "Don't know" = "#11111")

data1 %>%
  count(family_history) %>%
  plot_ly(
    labels = ~family_history,
    values = ~n,
    type = "pie",
    textposition = 'inside',
    textinfo = 'label+percent',
    hoverinfo = 'text', # Setting text on hover (see text variable on next line)
    text = ~paste(n, "Respondents"), # Setting text on hover
    marker = list(colors = colors1)) %>% # Setting up colors for clarity
  layout(title = "Responses")
```



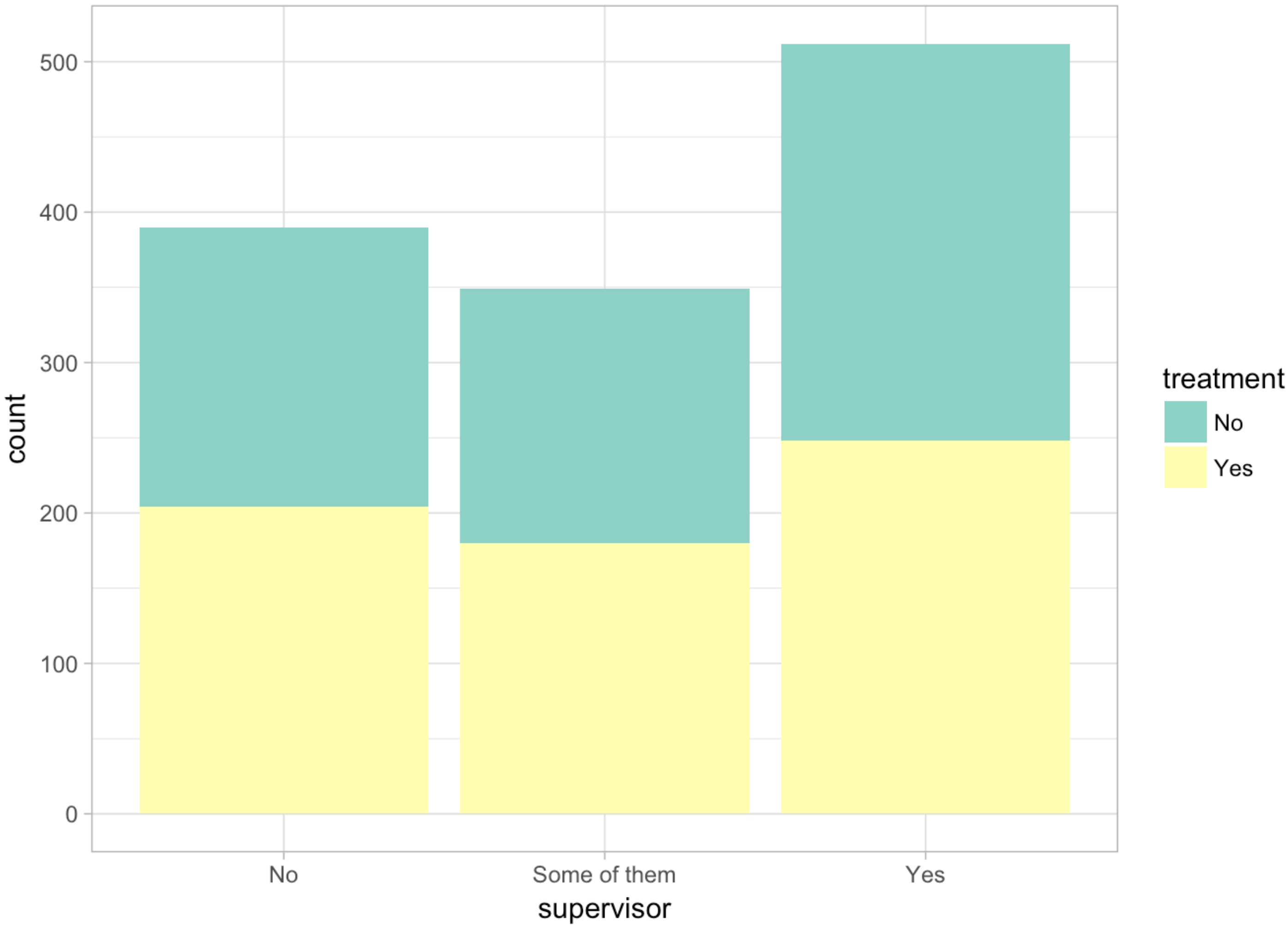
Do the ones with a family history of mental illness seek treatment?

```
ggplot(data=data1, aes(x=family_history, fill = treatment)) +geom_bar() +theme_light(
) +scale_fill_brewer(palette="Set2")
```



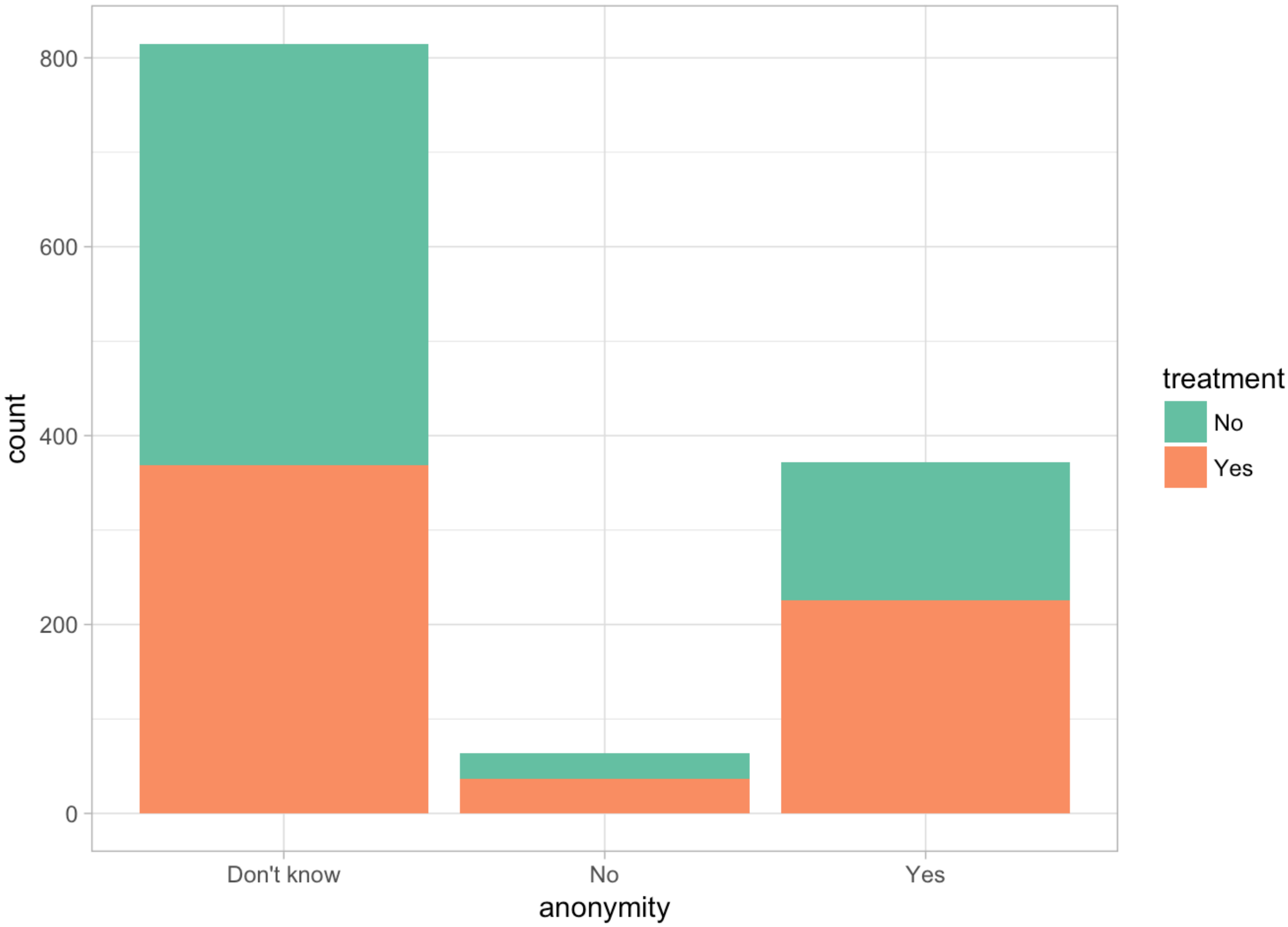
If the worker is willing to discuss the mental health issue with the supervisor, is he or she more probable to seek treatment?

```
ggplot(data=data1, aes(x=supervisor, fill = treatment)) +geom_bar() +theme_light() +s
cale_fill_brewer(palette="Set3")
```



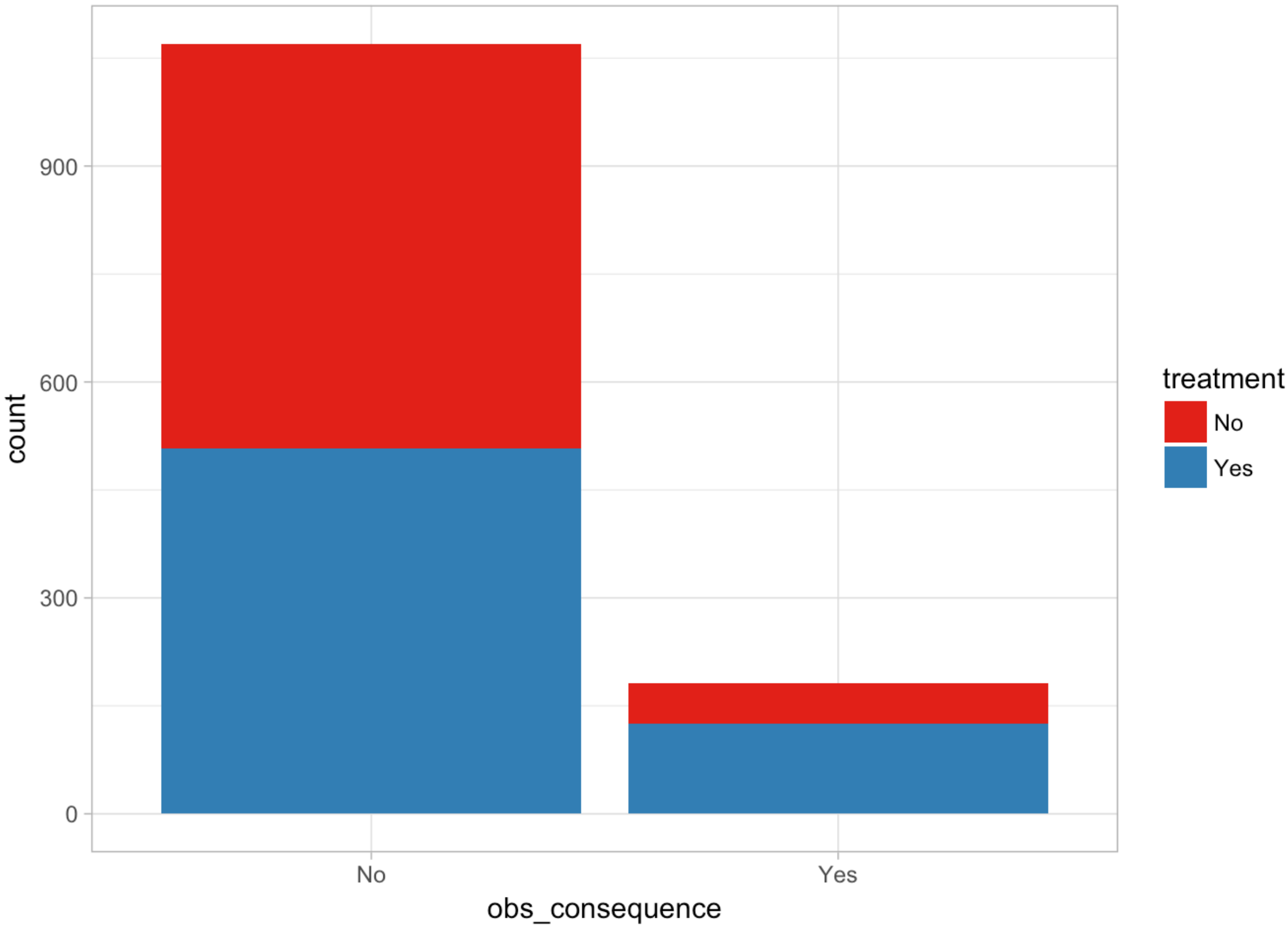
Does the anonymity of the worker affect the individual seeking treatment?

```
ggplot(data=data1, aes(x=anonymity, fill = treatment)) +geom_bar() +theme_light() +sc
ale_fill_brewer(palette="Set2")
```



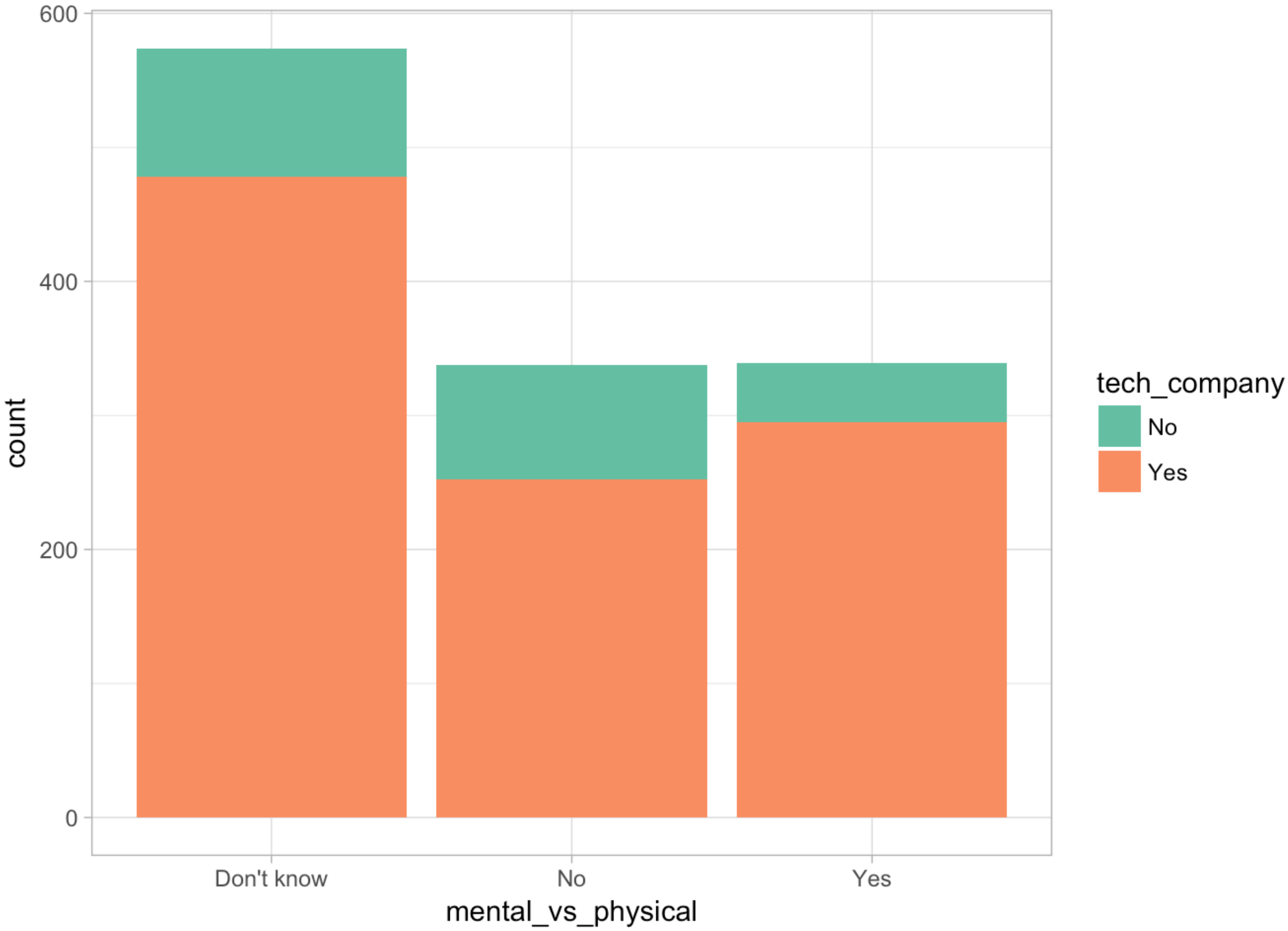
Do the consequences of seeking mental help affect the worker seeking treatment?

```
ggplot(data=data1, aes(x=obs_consequence, fill = treatment)) +geom_bar() +theme_light
() +scale_fill_brewer(palette="Set1")
```



How seriously are issues related to mental health taken in comparison to physical health, in tech and non-tech companies:

```
ggplot(data=data1, aes(x=mental_vs_physical,  fill = tech_company)) +geom_bar() +them
e_light() +scale_fill_brewer(palette="Set2")
```



Some functions that can be resued later.

```
# #Define some functions that can be resued later.
#
# getNumericColumns<-function(t){
#     tn = sapply(t,function(x){is.numeric(x)})
#     return(names(tn)[which(tn)])
# }
#
# getFactorColumns<-function(t){
#     tn = sapply(t,function(x){is.factor(x)})
#     return(names(tn)[which(tn)])
# }
```

Model building:

Out of the 1251 samples, we are reserving 1000 samples for training and 251 samples for testing.

```
set.seed(1)
n <- nrow(data1)

train.index <- sample(n,1000)
health.train <- data1[train.index,]
health.test <- data1[-train.index,]

x.train <- health.train[,-6]
y.train <- health.train$treatment

x.test <- health.test[,-6]
y.test <- health.test$treatment
```

```
#Creating a dataframe to save results of each method in order to plot a graph
success <- data.frame(methods=c("Logistic Regression","Single Tree", "Random Forest",
"Bagging","Neural Nets"), percentages=c(0,0,0,0,0))
```

Logistic regression:

```
fit0 <- glm(treatment~ ., data = health.train, family=binomial(logit))
Anova(fit0) #Perform Anova to get significant variables
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: treatment
##                           LR Chisq Df Pr(>Chisq)
## Age                          0.003  1  0.9568538
## Country                      0.560  3  0.9056227
## state                       49.069 42  0.2107834
## self_employed                2.427  1  0.1192484
## family_history              13.176  1  0.0002835 ***
## work_interfere             221.559  3  < 2.2e-16 ***
## no_employees                10.483  5  0.0626501 .
## remote_work                  0.050  1  0.8235812
## tech_company                 2.467  1  0.1162323
## benefits                     9.899  2  0.0070874 **
## care_options                10.121  2  0.0063423 **
## wellness_program             0.841  2  0.6567468
## seek_help                    7.612  2  0.0222392 *
## anonymity                   10.474  2  0.0053164 **
## leave                        1.530  4  0.8213834
## mental_health_consequence    4.036  2  0.1328910
## phys_health_consequence      1.303  2  0.5213697
## coworkers                    4.325  2  0.1150401
## supervisor                   2.316  2  0.3140699
## mental_health_interview      2.352  2  0.3084630
## phys_health_interview        0.554  2  0.7578853
## mental_vs_physical           2.286  2  0.3188246
## obs_consequence              0.034  1  0.8540161
## Gender                       4.035  2  0.1330195
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since state and self_employed have NA values but are not significant at the 0.05 level, we can remove these columns from our data.

```
data1 <- data1[, -c(3,4)]
health.train <- health.train[, -c(3,4)]
health.test <- health.test[, -c(3,4)]
x.train <- x.train[, -c(3,4)]
x.test <- x.test[, -c(3,4)]
```

Picking out only the significant variables, we get a better model with the variables - family_history, work_interfere, benefits, care_options, seek_help, anonymity.

```
fit1 <- glm(treatment ~ family_history + work_interfere + benefits + care_options + s
eek_help + anonymity, data = health.train, family=binomial(logit))
Anova(fit1) #Anonymity is not significant. Remove it.
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: treatment
##                LR Chisq Df Pr(>Chisq)
## family_history    31.66  1  1.834e-08 ***
## work_interfere   385.89  3  < 2.2e-16 ***
## benefits          12.86  2   0.001609 **
## care_options      10.51  2   0.005212 **
## seek_help          7.02  2   0.029952 *
## anonymity          4.81  2   0.090433 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit2 <- glm(treatment ~ family_history + work_interfere + benefits + care_options + s
eek_help , data = health.train, family=binomial(logit))
Anova(fit2) #seek_help is not significant. Remove it.
```

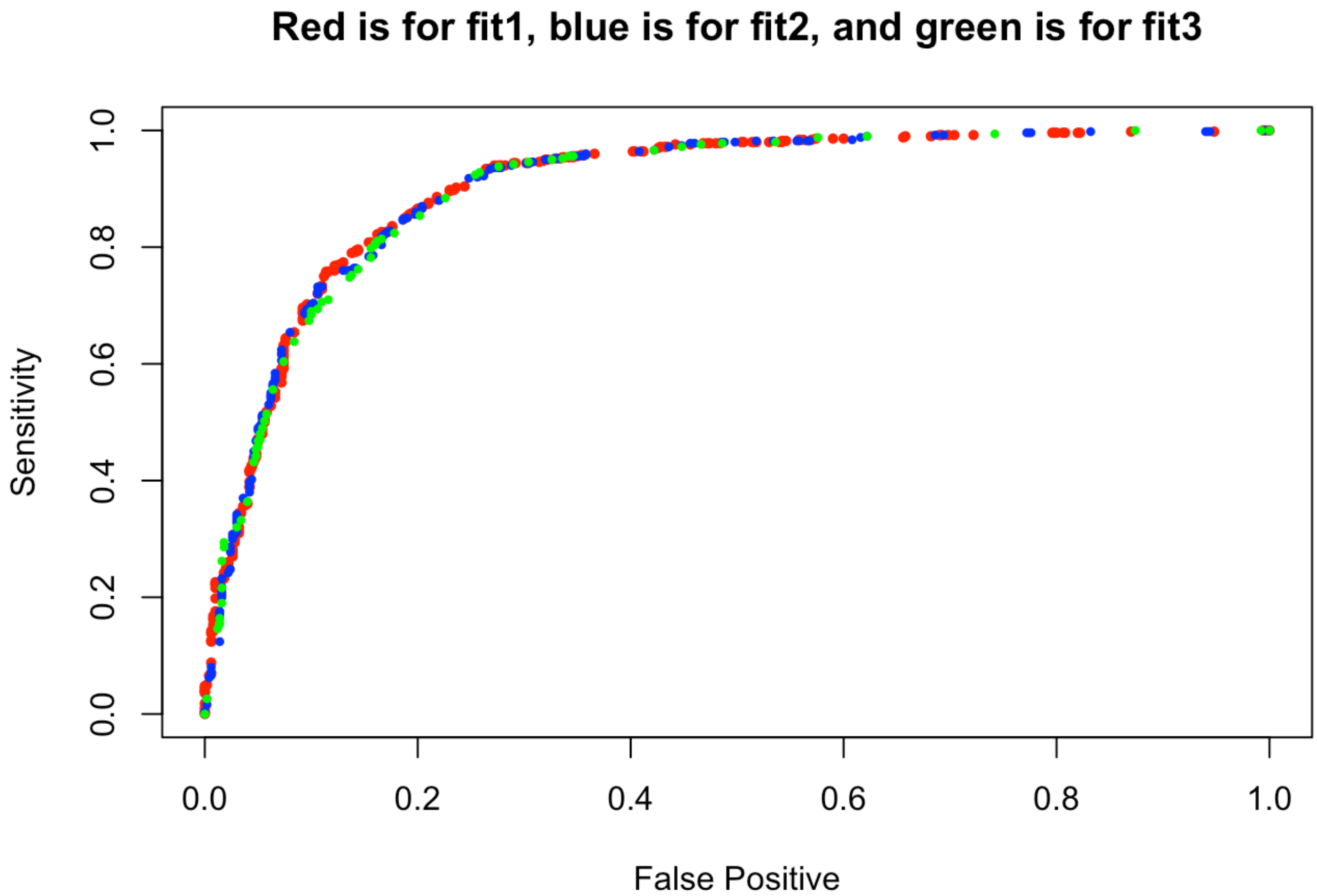
```
## Analysis of Deviance Table (Type II tests)
##
## Response: treatment
##                LR Chisq Df Pr(>Chisq)
## family_history    32.48  1  1.202e-08 ***
## work_interfere   383.88  3  < 2.2e-16 ***
## benefits          14.83  2  0.0006009 ***
## care_options      14.38  2  0.0007522 ***
## seek_help          5.85  2  0.0535892 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit3 <- glm(treatment ~ family_history + work_interfere + benefits + care_options  ,
data = health.train, family=binomial(logit))
Anova(fit3) #All variables significant at 0.05 level
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: treatment
##                LR Chisq Df Pr(>Chisq)
## family_history    32.90  1  9.708e-09 ***
## work_interfere   378.80  3  < 2.2e-16 ***
## benefits          19.24  2  6.644e-05 ***
## care_options      13.02  2   0.001488 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fit1.roc <- roc(health.train$treatment, fit1$fitted, plot=F)
fit2.roc <- roc(health.train$treatment, fit2$fitted, plot=F)
fit3.roc <- roc(health.train$treatment, fit3$fitted, plot=F)
#Not much difference between the 3 fits.
plot(1-fit1.roc$specificities, fit1.roc$sensitivities, col="red", pch=16, cex=.7,
     xlab="False Positive",
     ylab="Sensitivity")
points(1-fit2.roc$specificities, fit2.roc$sensitivities, col="blue", pch=16, cex=.6)
points(1-fit3.roc$specificities, fit3.roc$sensitivities, col="green", pch=16, cex=.6)

title("Red is for fit1, blue is for fit2, and green is for fit3")
```




```
# roccurve <- roc(health.test$treatment ~ predict(fit3, health.test))
# plot(roccurve)

fit.pred <- rep("No", 1000)
fit.pred[fit3$fitted > 2/3]="Yes"
MCE = (sum((fit.pred[health.train$treatment == "Yes"] != "Yes"))
         + sum((fit.pred[health.train$treatment == "No"] != "No")))/length(health.tra
in$treatment)
MCE #0.191
```

```
## [1] 0.191
```

```
success$percentages[success$methods == "Logistic Regression"] <- (100 - MCE*100)
```

Single tree:

```
set.seed(1)
fit.single <- randomForest(treatment~., health.train, mtry=2, ntree=1)
```

```
names(fit.single)
```

```
##  [1] "call"            "type"            "predicted"
##  [4] "err.rate"        "confusion"       "votes"
##  [7] "oob.times"       "classes"         "importance"
## [10] "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"
## [16] "y"               "test"            "inbag"
## [19] "terms"
```

```
fit.single$mtry
```

```
## [1] 2
```

```
fit.single$votes[1:20, ]   #  prob of 0 and 1 using oob's
```

```
##       No Yes
## 334    0   1
## 469  NaN NaN
## 720    0   1
## 1142   1   0
## 253  NaN NaN
## 1127   0   1
## 1185   1   0
## 828    1   0
## 787  NaN NaN
## 77   NaN NaN
## 257    0   1
## 220  NaN NaN
## 857    0   1
## 479  NaN NaN
## 958    1   0
## 619    0   1
## 892  NaN NaN
## 1233 NaN NaN
## 472    0   1
## 963    0   1
```

```
fit.single$predicted[1:20] #  lables using oob's and majority vote. Notice those with NA because they are not in any OOB's
```

```
##  334  469  720 1142  253 1127 1185  828  787   77  257  220  857  479  958
##  Yes <NA>  Yes   No <NA>  Yes   No   No <NA> <NA>  Yes <NA>  Yes <NA>   No
##  619  892 1233  472  963
##  Yes <NA> <NA>  Yes  Yes
## Levels: No Yes
```

```
fit.single$err.rate[1,]["OOB"]  #  mis-classification errors of oob's/0/1
```

```
##       OOB
## 0.3905817
```

```
predict(fit.single, health.test)[1:20]  # prediction by using the RF based on all the training data.
```

```
##   6  11  12  20  28  33  36  39  41  46  50  55  59  68  73  75  76  80
## Yes Yes  No  No  No  No Yes Yes Yes  No Yes  No  No Yes Yes  No  No  No
##  84  88
## Yes  No
## Levels: No Yes
```

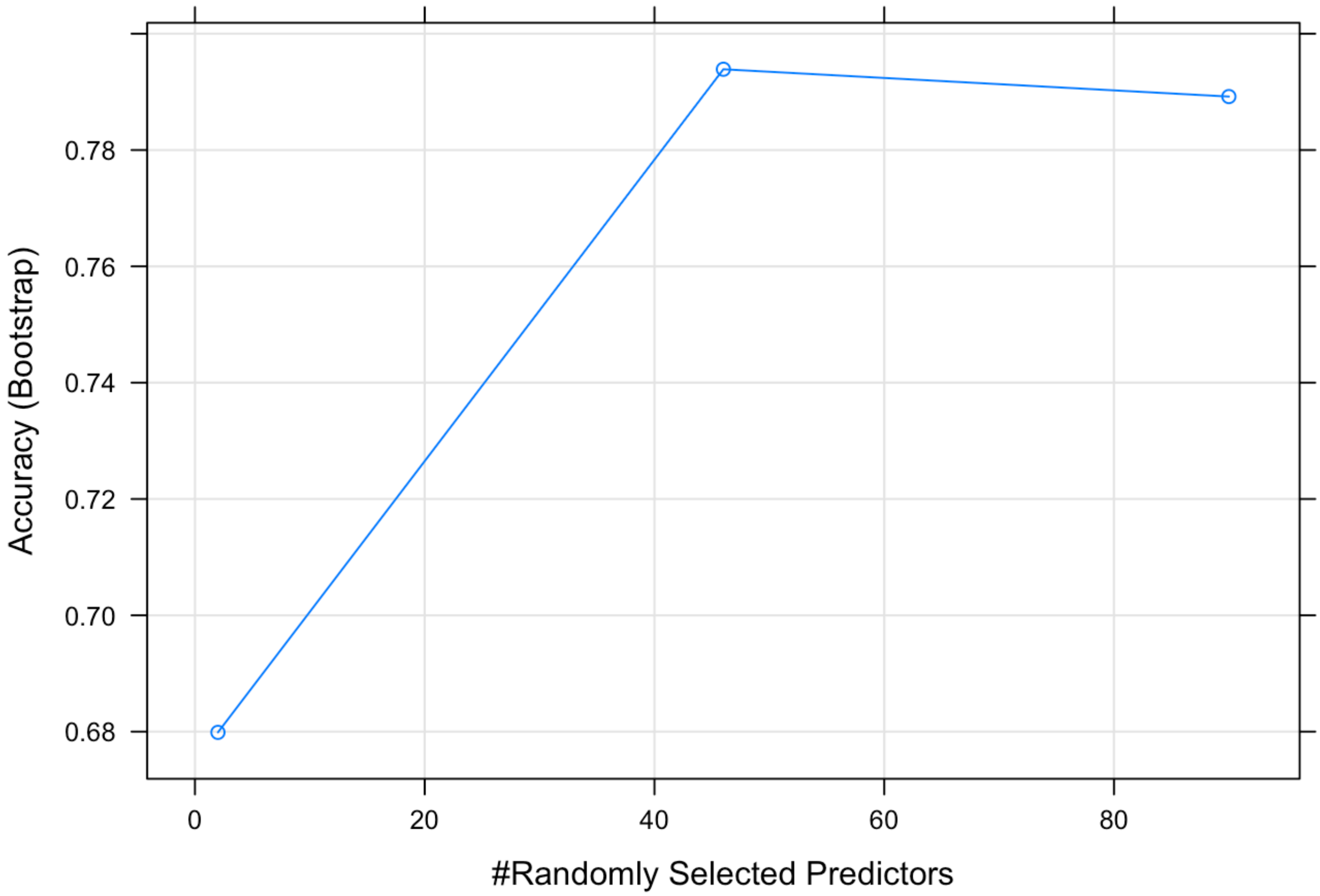
```
data.frame(fit.single$votes[1:20, ], fit.single$predicted[1:20], predict(fit.single,
health.test)[1:20] )
```

```
##       No Yes fit.single.predicted.1.20.
## 334    0   1                        Yes
## 469  NaN NaN                       <NA>
## 720    0   1                        Yes
## 1142   1   0                         No
## 253  NaN NaN                       <NA>
## 1127   0   1                        Yes
## 1185   1   0                         No
## 828    1   0                         No
## 787  NaN NaN                       <NA>
## 77   NaN NaN                       <NA>
## 257    0   1                        Yes
## 220  NaN NaN                       <NA>
## 857    0   1                        Yes
## 479  NaN NaN                       <NA>
## 958    1   0                         No
## 619    0   1                        Yes
## 892  NaN NaN                       <NA>
## 1233 NaN NaN                       <NA>
## 472    0   1                        Yes
## 963    0   1                        Yes
##      predict.fit.single..health.test..1.20.
## 334                                     Yes
## 469                                     Yes
## 720                                      No
## 1142                                     No
## 253                                      No
## 1127                                     No
## 1185                                    Yes
## 828                                     Yes
## 787                                     Yes
## 77                                       No
## 257                                     Yes
## 220                                      No
## 857                                      No
## 479                                     Yes
## 958                                     Yes
## 619                                      No
## 892                                      No
## 1233                                     No
## 472                                     Yes
## 963                                      No
```

```
success$percentages[success$methods == "Single Tree"] <- (100 - 100*fit.single$err.ra
te[1,]["OOB"])
```

Random forests:

```
health.rf <- train(treatment~., data=health.train, method="rf",metric="Accuracy", ntr
ee=20)
plot(health.rf)
```




```
predict.rf <- predict(health.rf,health.test)
#Accuracy
confusionMatrix(predict.rf, health.test$treatment)$overall[1]
```

```
##  Accuracy
## 0.7968127
```

```
success$percentages[success$methods == "Random Forest"] <- confusionMatrix(predict.rf
, health.test$treatment)$overall[1]*100
```

Neural nets:

```
# Let us first calculate the number of hidden layers/nodes and the decay parameters
# size: number of intermediate hidden nodes
# decay: parameter to avoid overfitting
parameter <- train( treatment ~ . , data=health.train, method="nnet", trace=F)
size <- parameter$bestTune$size
decay <- parameter$bestTune$decay

# Neural net model:
model.nn <- nnet(treatment ~ ., size=size, decay=decay, trace=F, data=health.train)
predict.nn <- predict(model.nn, health.test, type = "class")
sum(predict.nn==y.test)/length(predict.nn) #Accuracy
```

```
## [1] 0.812749
```

```
success$percentages[success$methods == "Neural Nets"] <- confusionMatrix(predict.nn,h
ealth.test$treatment)$overall[1]*100
```

Bagging:

```
bag.model <- bagging(treatment ~ .,  data=health.train)
predict.bag <- predict(bag.model, health.test, type="class")
confusionMatrix(predict.bag$class, health.test$treatment)$overall[1]
```

```
##  Accuracy
## 0.8366534
```

```
success$percentages[success$methods == "Bagging"] <- confusionMatrix(predict.bag$clas
s, health.test$treatment)$overall[1]*100
```

Lets plot our success rates for different methods:

```
success
```

```
##               methods percentages
## 1 Logistic Regression    80.90000
## 2         Single Tree    60.94183
## 3       Random Forest    79.68127
## 4             Bagging    83.66534
## 5         Neural Nets    81.27490
```

```
ggplot(success, aes(x=methods, y=percentages)) + geom_bar(stat="identity", fill=c("ye
llowgreen", "hotpink2", "dodgerblue3", "orange2","Red"), width = 0.2) + coord_flip()
+ theme(legend.position = "none") + geom_text(aes(label = format(round(percentages, 2
), nsmall = 2)), size = 3, hjust = 3, vjust = 3)
```