Speed Dating Project

Aria Wang

March 12, 2020

```r
setwd("~/Desktop/MSiA 420/speed-dating-project")
data <- read.csv("Speed Dating Data.csv")
```

```r
# check NA
na_rate <- rep(1, dim(data)[2])
# drop the variables that end with _3
for (i in 1:156){
  na_rate[i] <- sum(is.na(data[,i]))/nrow(data)
}
na_columns <- colnames(data)[na_rate > 0.5]
```

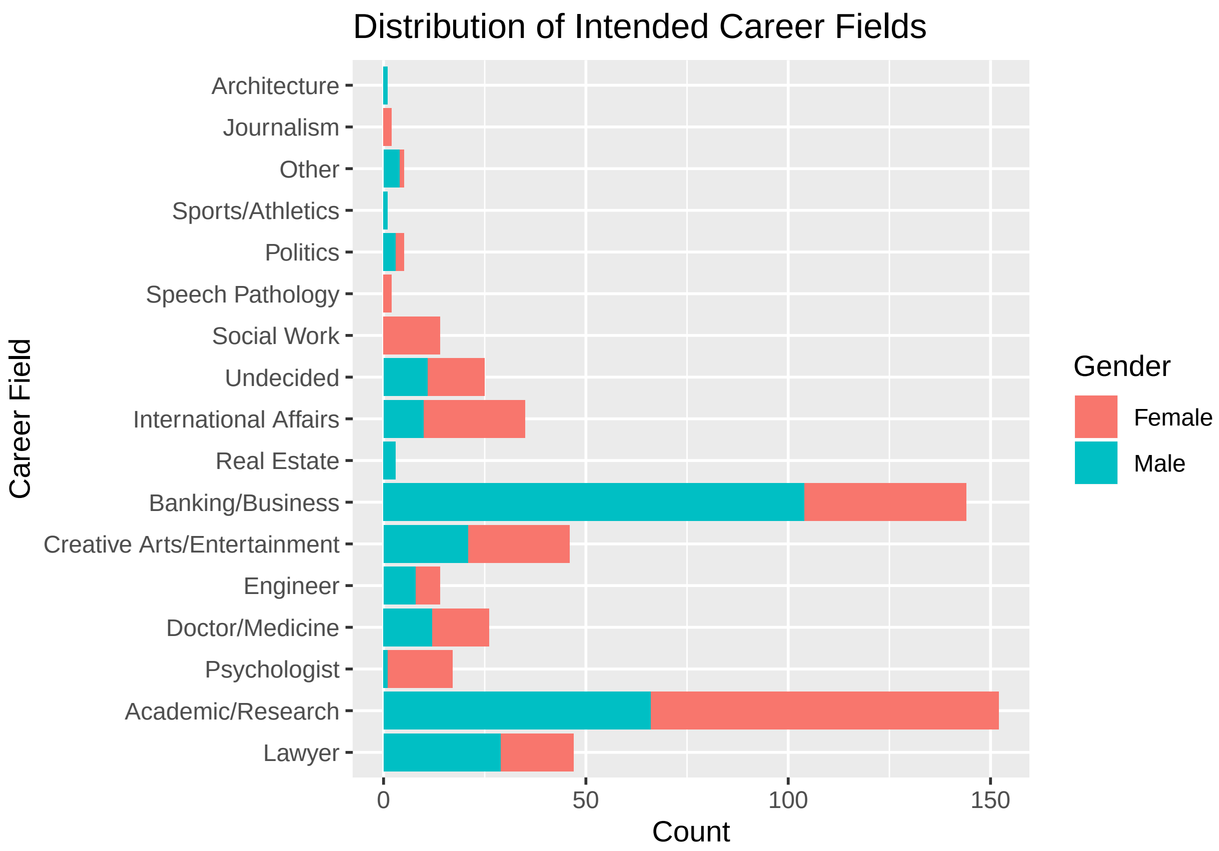
```r
library(dplyr)
data <- data %>%
  select(-na_columns)
```

```r
data$gender <- as.factor(data$gender)
data$career_c <- as.factor(data$career_c)
data$samerace <- as.factor(data$samerace)
data$race <- as.factor(data$race)
data$dec <- as.factor(data$dec)
data$date <- as.factor(data$date)
```

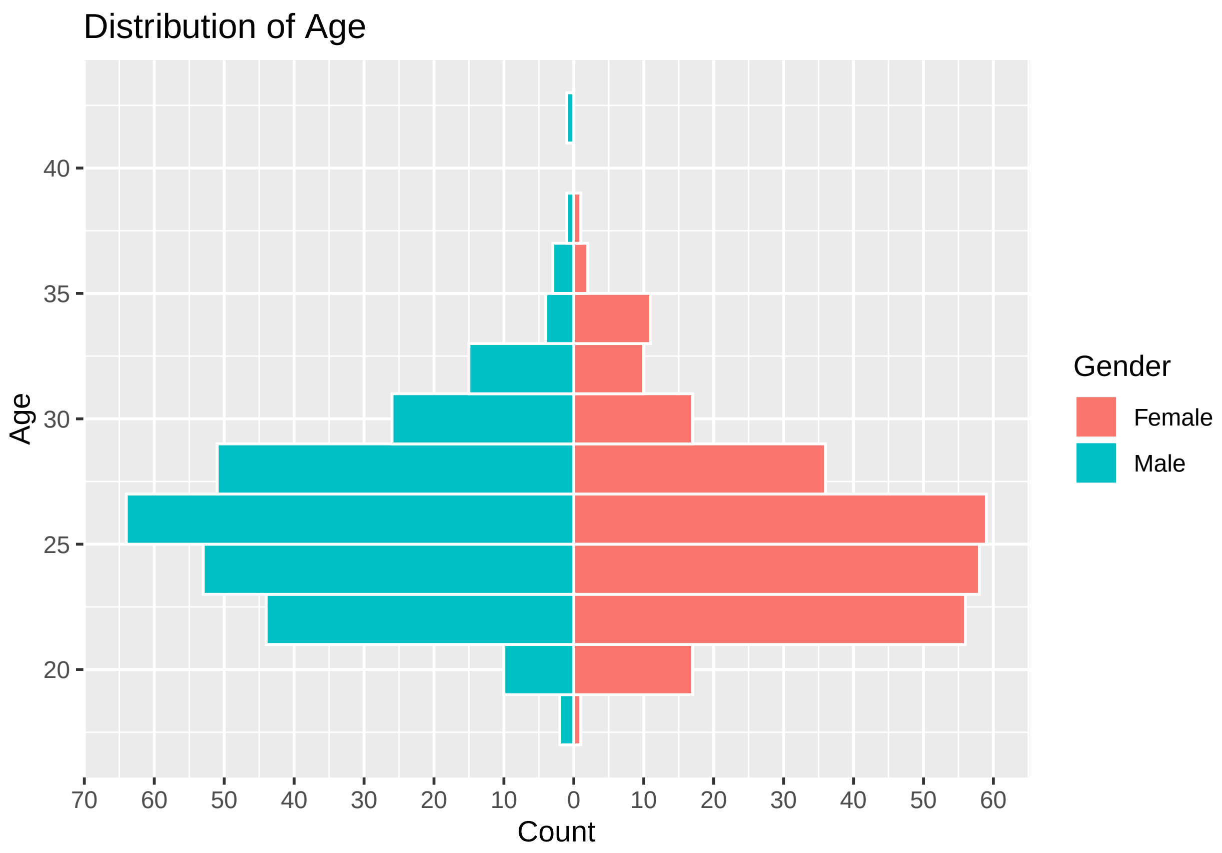
```r
# scale the ratings
data <- data %>%
  mutate(pf_sum_o = pf_o_att + pf_o_sin + pf_o_int + pf_o_fun + pf_o_amb + pf_o_sha,
         sum_o = attr_o + sinc_o + intel_o + fun_o + amb_o + shar_o,
         sum1_1 = attr1_1 + sinc1_1 + intel1_1 + fun1_1 + amb1_1 + shar1_1,
         sum4_1 = attr4_1 + sinc4_1 + intel4_1 + fun4_1 + amb4_1 + shar4_1,
         sum2_1 = attr2_1 + sinc2_1 + intel2_1 + fun2_1 + amb2_1 + shar2_1,
         sum3_1 = attr3_1 + sinc3_1 + intel3_1 + fun3_1 + amb3_1,
         sum5_1 = attr5_1 + sinc5_1 + intel5_1 + fun5_1 + amb5_1,
         sum1_2 = attr1_2 + sinc1_2 + intel1_2 + fun1_2 + amb1_2 + shar1_2,
         sum4_2 = attr4_2 + sinc4_2 + intel4_2 + fun4_2 + amb4_2 + shar4_2,
         sum2_2 = attr2_2 + sinc2_2 + intel2_2 + fun2_2 + amb2_2 + shar2_2,
         sum3_2 = attr3_2 + sinc3_2 + intel3_2 + fun3_2 + amb3_2,
         sum5_2 = attr5_2 + sinc5_2 + intel5_2 + fun5_2 + amb5_2) %>%
  mutate_at(c("pf_o_att", "pf_o_sin", "pf_o_int", "pf_o_fun", "pf_o_amb", "pf_o_sha"),
            funs(./pf_sum_o*100)) %>%
  mutate_at(c("attr_o", "sinc_o", "intel_o", "fun_o", "amb_o", "shar_o"),
            funs(./sum_o*100)) %>%
  mutate_at(c("attr1_1", "sinc1_1", "intel1_1", "fun1_1", "amb1_1", "shar1_1"),
            funs(./sum1_1*100)) %>%
  mutate_at(c("attr4_1", "sinc4_1", "intel4_1", "fun4_1", "amb4_1", "shar4_1"),
            funs(./sum4_1*100)) %>%
  mutate_at(c("attr2_1", "sinc2_1", "intel2_1", "fun2_1", "amb2_1", "shar2_1"),
            funs(./sum2_1*100)) %>%
  mutate_at(c("attr3_1", "sinc3_1", "fun3_1", "intel3_1", "amb3_1"),
            funs(./sum3_1*100)) %>%
  mutate_at(c("attr5_1", "sinc5_1", "fun5_1", "intel5_1", "amb5_1"),
            funs(./sum5_1*100)) %>%
  mutate_at(c("attr1_2", "sinc1_2", "intel1_2", "fun1_2", "amb1_2", "shar1_2"),
            funs(./sum1_2*100)) %>%
  mutate_at(c("attr4_2", "sinc4_2", "intel4_2", "fun4_2", "amb4_2", "shar4_2"),
            funs(./sum4_2*100)) %>%
  mutate_at(c("attr2_2", "sinc2_2", "intel2_2", "fun2_2", "amb2_2", "shar2_2"),
            funs(./sum2_2*100)) %>%
  mutate_at(c("attr3_2", "sinc3_2", "intel3_2", "fun3_2", "amb3_2"),
            funs(./sum3_2*100)) %>%
    mutate_at(c("attr5_2", "sinc5_2", "intel5_2", "fun5_2", "amb5_2"),
            funs(./sum5_2*100)) %>%
  select(-c("pf_sum_o", "sum_o", "sum1_1", "sum4_1", "sum2_1", "sum3_1",
            "sum5_1", "sum1_2", "sum4_2", "sum2_2", "sum3_2", "sum5_2"))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

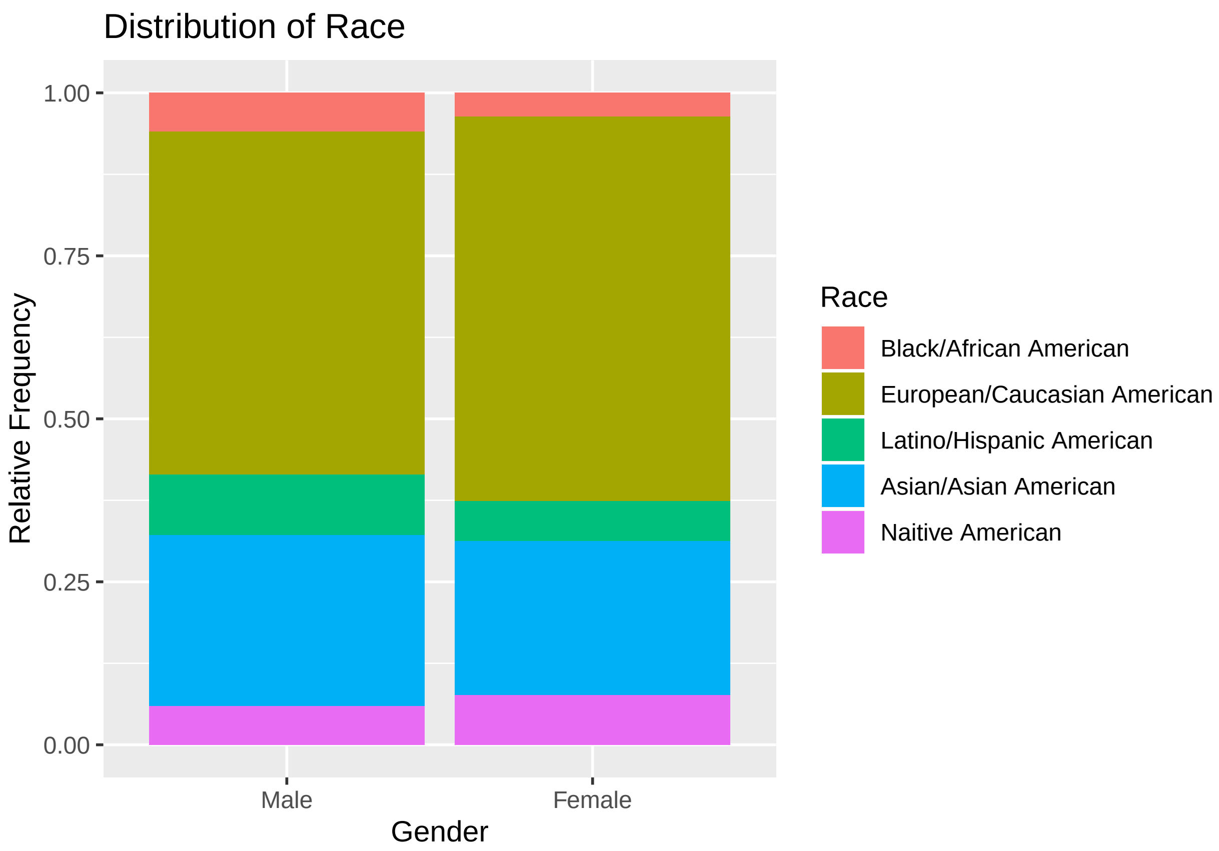
```
library(ggplot2)
# career distribution plot
career_label <- c("Lawyer", "Academic/Research", "Psychologist",
                  "Doctor/Medicine", "Engineer", "Creative Arts/Entertainment",
                  "Banking/Business", "Real Estate", "International Affairs",
                  "Undecided", "Social Work", "Speech Pathology", "Politics",
                  "Sports/Athletics", "Other", "Journalism", "Architecture")

data %>%
  filter(!is.na(career_c)) %>%
  select(iid, gender, career_c) %>%
  unique(by = iid) %>%
  ggplot() +
    geom_bar(aes(career_c, fill=gender)) +
    scale_x_discrete(label = career_label) + coord_flip() +
    labs(title = "Distribution of Intended Career Fields", x = "Career Field", y = "Count") +
    scale_fill_discrete("Gender", labels = c("Female", "Male"))
```



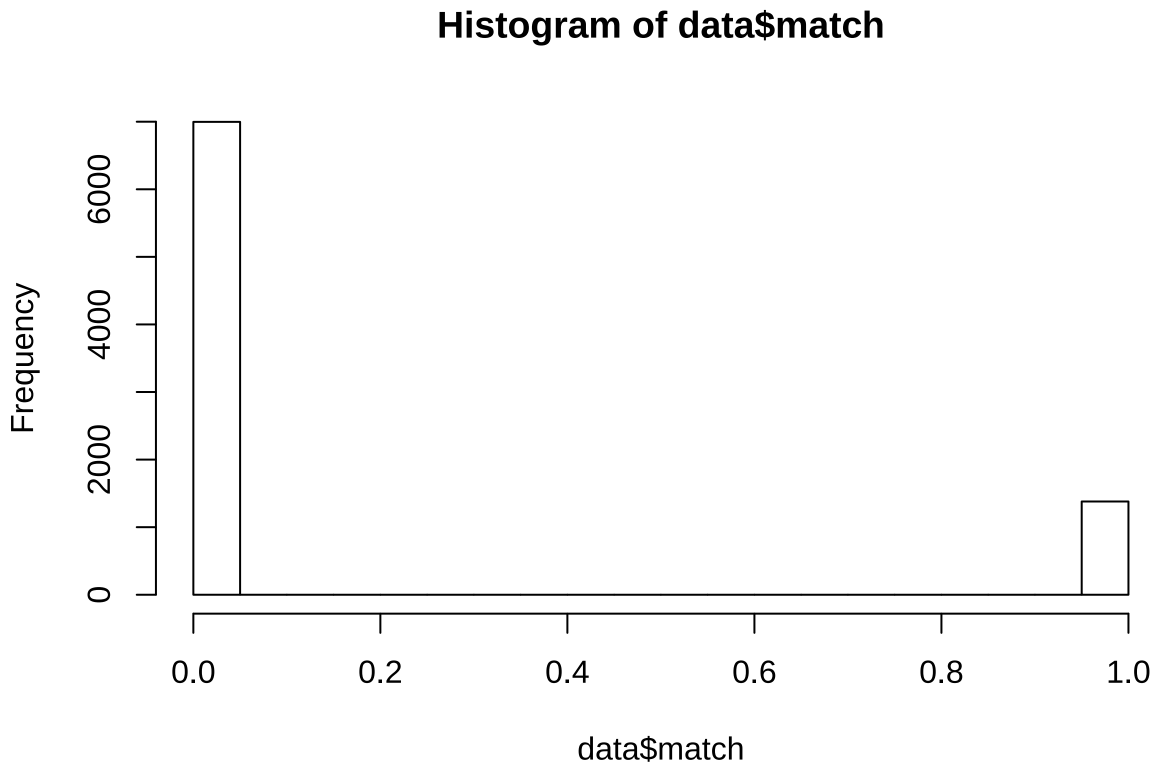
```
# age distribution plot
temp_age <- data %>%
  filter(!is.na(age)) %>%
  filter(age < max(age)) %>%
  select(iid, gender, age) %>%
  unique(by = iid)

ggplot(data = temp_age, aes(x = age,fill = gender)) + coord_flip() +
  geom_histogram(data = subset(temp_age, gender == "0"), binwidth = 2, color = "white") +
  geom_histogram(data = subset(temp_age, gender == "1"),
               aes(y = ..count.. * (-1)), binwidth = 2, color = "white") +
  scale_y_continuous(breaks = seq(-70, 70, 10), labels = abs(seq(-70, 70, 10)))+
  scale_x_continuous(breaks = seq(10, 45, 5), labels = seq(10, 45,5)) +
  labs(title = "Distribution of Age", x = "Age", y = "Count") +
  scale_fill_discrete("Gender", labels = c("Female", "Male"))
```



```
# race distribution plot
race_label <- c("Black/African American", "European/Caucasian American",
                "Latino/Hispanic American", "Asian/Asian American",
                "Naitive American", "Other")

data %>%
  filter(!is.na(race)) %>%
  select(iid, gender, race) %>%
  unique(by = iid) %>%
  ggplot() +
    geom_bar(aes(x = gender, fill = race), position = "fill") +
    labs(title = "Distribution of Race", x = "Gender", y = "Relative Frequency") +
    scale_fill_discrete("Race", labels = race_label) +
    scale_x_discrete(labels=c("0" = "Male", "1" = "Female"))
```

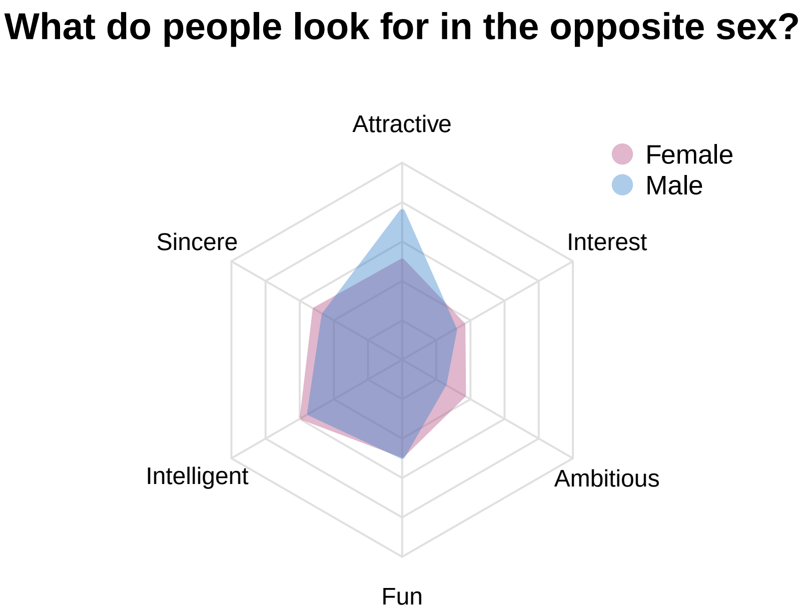


```
hist(data$match)
```



```
library(fmsb)
# what do you look for in the opposite sex
test1 <- data %>%
  filter(!is.na(attr1_1 + sinc1_1 + intel1_1 + fun1_1 + amb1_1 + shar1_1)) %>%
  select(iid, gender, attr1_1:shar1_1) %>%
  unique(by = idd) %>%
  group_by(gender) %>%
  summarise(Attractive = mean(attr1_1), Sincere = mean(sinc1_1),
            Intelligent = mean(intel1_1), Fun = mean(fun1_1),
            Ambitious = mean(amb1_1), Interest = mean(shar1_1))

test1forplot <- test1 %>%
  select(-gender)

maxmin <- data.frame(
 Attractive = c(36, 0),
 Sincere = c(36, 0),
 Intelligent = c(36, 0),
 Fun = c(36, 0),
 Ambitious = c(36, 0),
 Interest = c(36, 0))

test11 <- rbind(maxmin, test1forplot)

test11male <- test11[c(1,2,4),]
test11female <- test11[c(1,2,3),]
```

```r
radarchart(test11,
           pty = 32,
           axistype = 0,
           pcol = c(rgb(0.7, 0.3, 0.5, 0.4), rgb(0.2, 0.5, 0.8, 0.4)),
           pfcol = c(rgb(0.7, 0.3, 0.5, 0.4), rgb(0.2, 0.5, 0.8, 0.4)),
           plty = 1,
           plwd = 3,
           cglty = 1,
           cglcol = "gray88",
           centerzero = TRUE,
           seg = 5,
           vlcex = 0.75,
           palcex = 0.75,
           title = "What do people look for in the opposite sex?")
legend(x = 1, y = 1.2, legend = c("Female", "Male"),
       bty = "n", pch = 20 , col = c(rgb(0.7, 0.3, 0.5, 0.4), rgb(0.2, 0.5, 0.8, 0.4)),
       text.col = "black", cex = 0.8, pt.cex = 2)
```

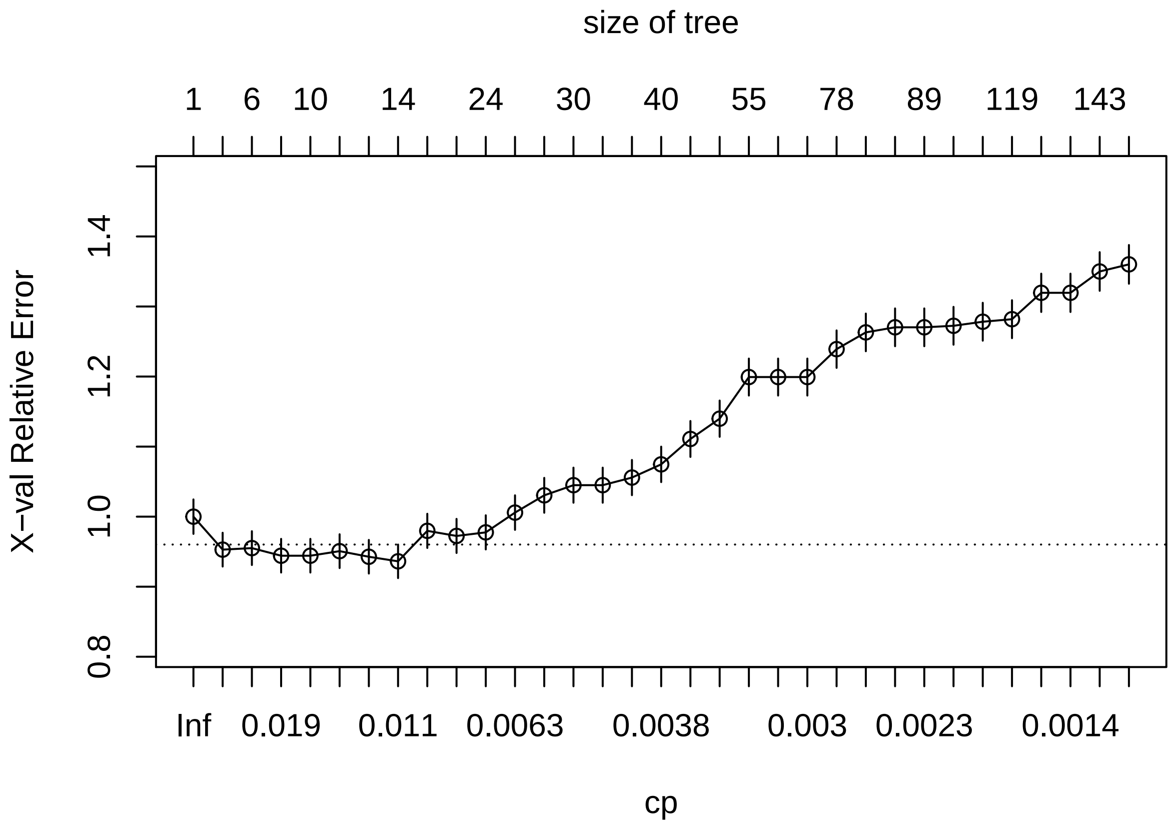


Decision Tree

```r
# drop 3 or s
data <- data[, !grepl(".*_[3s]$",colnames(data))]
```

```r
# drop columns with >50% null values
t = data.frame(colSums(is.na(data))/nrow(data))
colnames(t)=c("nullrate")
t <- t %>% subset(nullrate<0.5)
data <- data[,rownames(t)]

# other
data <- data[, colnames(data)!='match_es']
data <- data[, colnames(data)!='pid']
data <- data[, colnames(data)!='dec_o']
data <- data[, colnames(data)!='dec']
data <- data[, colnames(data)!='like_o']
data <- data[, colnames(data)!='like']
data <- data[, colnames(data)!='partner']
# data <- data[, colnames(data)!='attr_o']
# data <- data[, colnames(data)!='attr']


data$income <- as.numeric(data$income)
data$tuition <- as.numeric(data$tuition)
data <- data[, !(colnames(data) %in% c('zipcode','from','career','field','undergra','mn_sat','attr5_2'
                                       ,'attr5_1'))]
```

```r
# fit inital model by cv
library(rpart)
set.seed(420)
control <- rpart.control(minbucket = 5, cp = 0.001, maxsurrogate = 0, usesurrogate = 0, xval = 10)
date.tr <- rpart(match ~.,data, method = "class", control = control)
plotcp(date.tr) #plot of CV r^2 vs. size
```



```
date.tr1 <- prune(date.tr, cp=0.011)
date.tr1$variable.importance
```

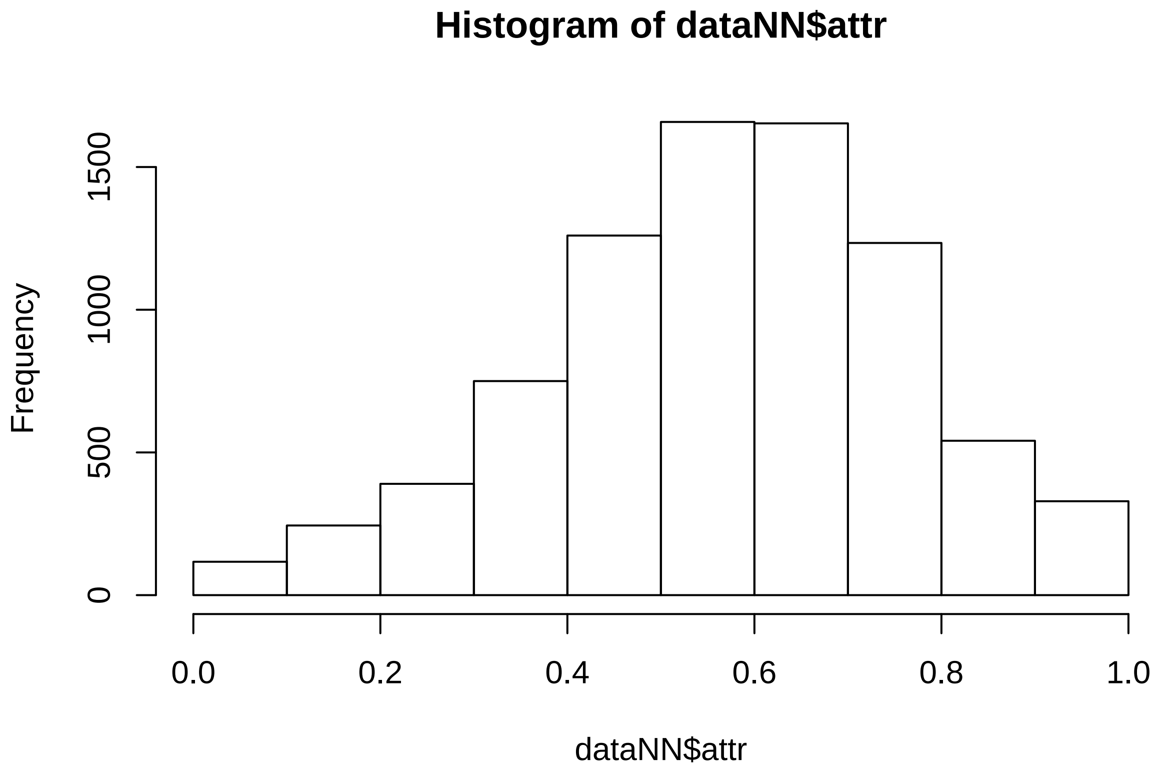
```
##        fun     prob_o       prob       attr       shar      fun_o     attr_o
## 144.494863 106.295276  29.474435  27.760189  21.157164  16.199978  15.171390
##    intel_o   pf_o_int
##  13.931836   8.071229
```

```
# data preprocessing
selected = colnames(data) %in%
  c("fun", "prob_o", "prob", "attr", "shar", "fun_o", "attr_o", "intel_o", "pf_o_int", "match")
dataNN = data[,selected]
dataNN[,1] = as.factor(dataNN[,1])
dataNN[, -1] <- sapply(dataNN[,-1], function(x) x/(max(x, na.rm=TRUE) - min(x, na.rm = TRUE)))
write.csv(dataNN, '~/Desktop/MSiA 420/speed-dating-project/data_clean.csv')
```

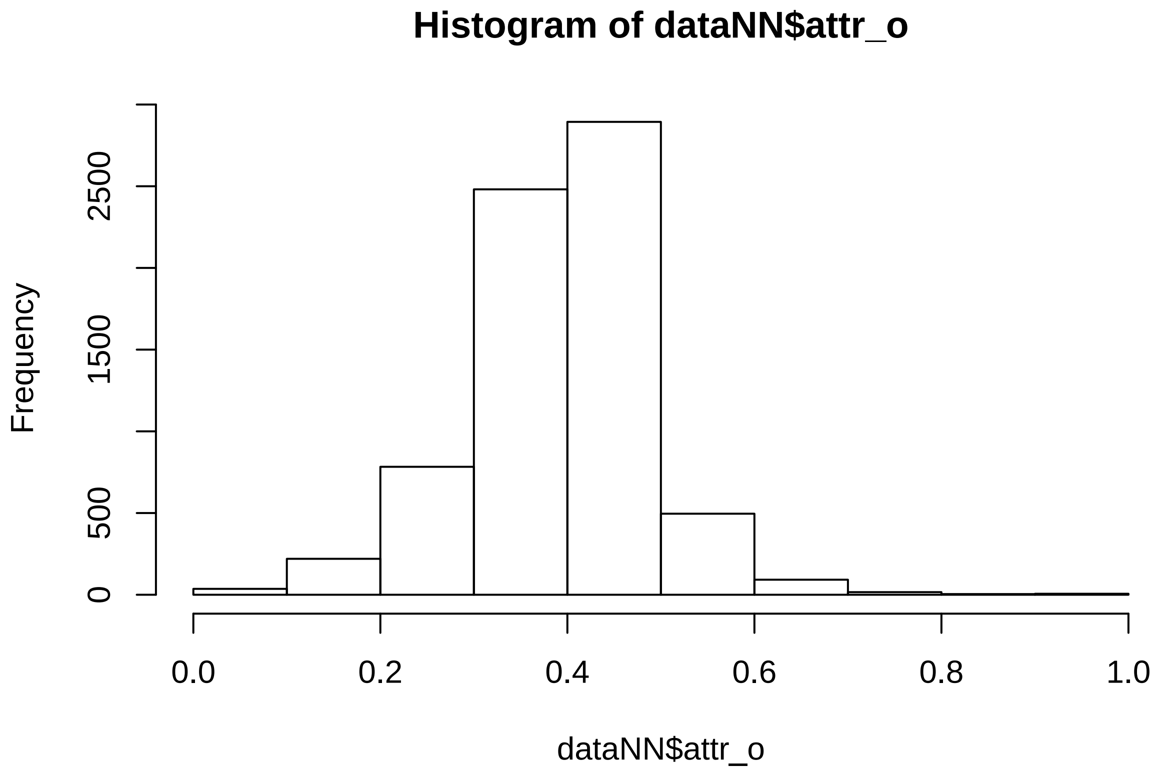
======================= end of data cleaning ========================

Logistic regression

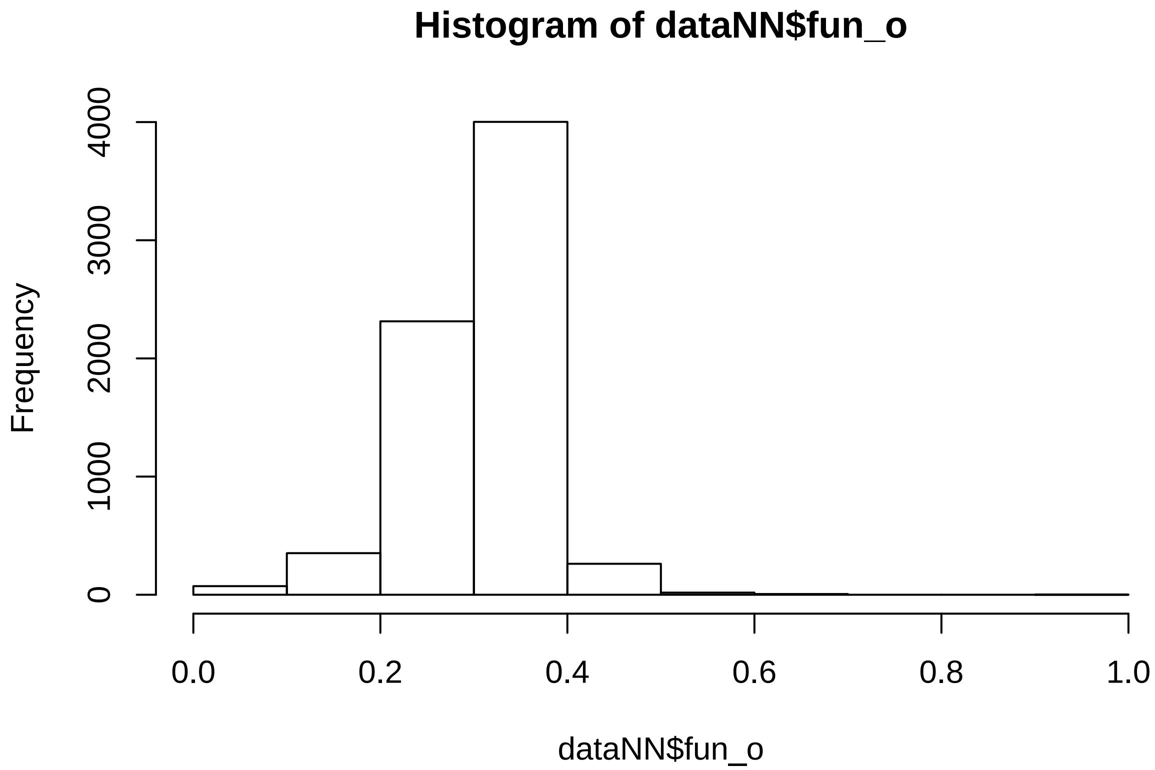
```
hist(dataNN$attr)
```



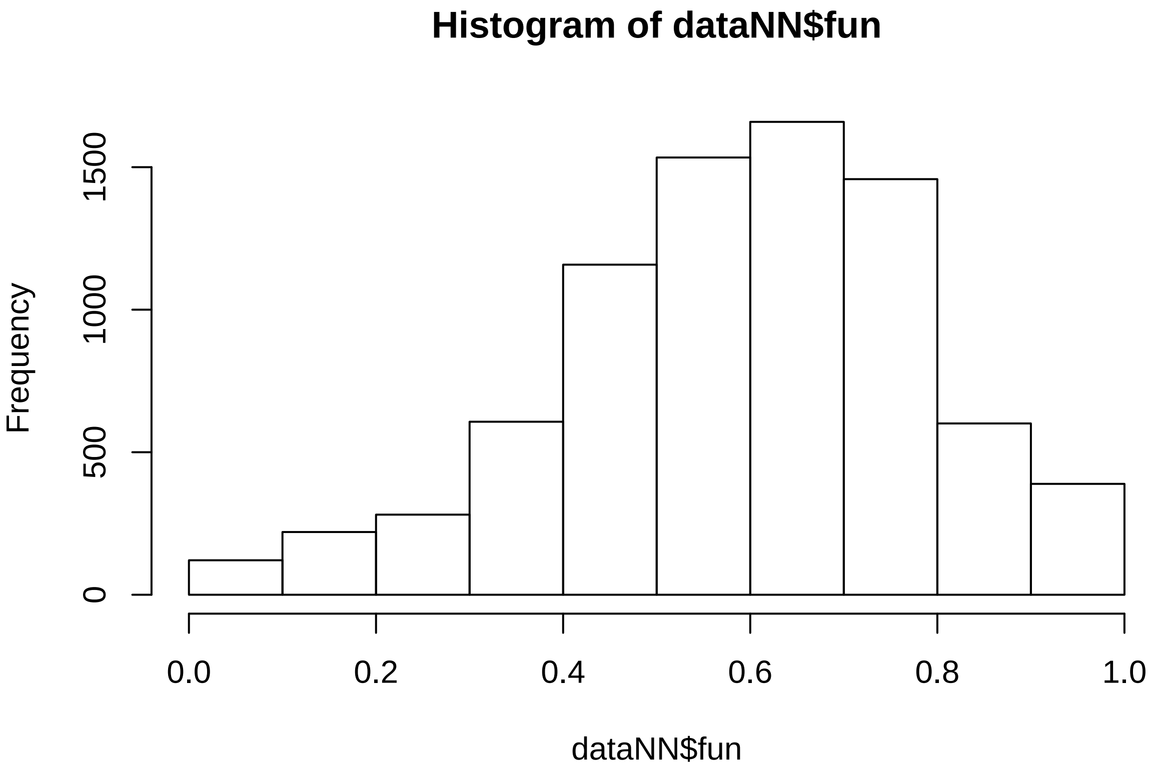
```r
hist(dataNN$attr_o)
```



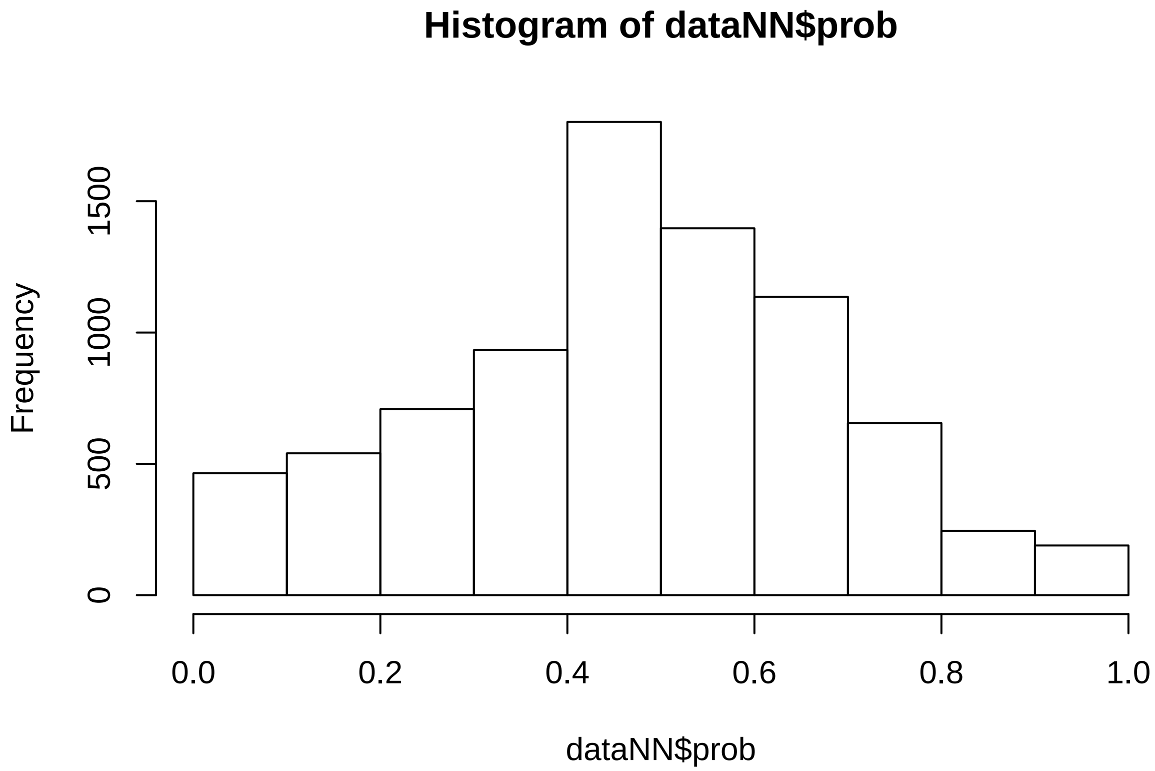
```
hist(dataNN$fun_o)
```



```r
hist(dataNN$fun)
```



```r
hist(dataNN$prob)
```



```r
logistic1 <- glm(match ~., data=dataNN, family = binomial(link="logit"))
summary(logistic1)
```

```
##
## Call:
## glm(formula = match ~ ., family = binomial(link = "logit"), data = dataNN)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7019  -0.6036  -0.3695  -0.1731   2.8670
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -9.2359     0.6344 -14.558  < 2e-16 ***
## pf_o_int      0.7928     0.2965   2.674 0.007504 **
## attr_o        3.5685     0.4862   7.339 2.15e-13 ***
## intel_o      -2.8524     0.8408  -3.392 0.000693 ***
## fun_o         3.8263     0.7526   5.084 3.69e-07 ***
## prob_o        2.7657     0.2049  13.500  < 2e-16 ***
## attr          2.2627     0.2674   8.463  < 2e-16 ***
## fun           1.5845     0.3002   5.278 1.31e-07 ***
## shar          1.3477     0.2525   5.336 9.48e-08 ***
## prob          1.4173     0.2150   6.593 4.30e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5614.7  on 5985  degrees of freedom
## Residual deviance: 4425.0  on 5976  degrees of freedom
##   (2392 observations deleted due to missingness)
## AIC: 4445
##
## Number of Fisher Scoring iterations: 5
```

```
library(car)
```

```
## Loading required package: carData
```

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

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

```
vif(logistic1)
```

```
## pf_o_int   attr_o  intel_o    fun_o   prob_o     attr      fun     shar
## 1.013389 1.187225 1.287538 1.168298 1.073482 1.395422 1.657829 1.571304
##     prob
## 1.209785
```