# Analysis on Speed Dating Data Set

**Ren Ayangco**

**Rynz Daval**

**Gian Tan**

**Loading data (Using the built-in import function in RStudio)**

```
library(readr)
speed_data_data <- read_csv("C:/Users/Acer/Desktop/speed_data_data.csv")
df = data.frame(speed_data_data)
```

## I. Introduction

Dataset Link: https://www.kaggle.com/datasets/mexwell/speed-dating

We will be using the Speed Dating dataset from kaggle. In the dataset, we can see the fields which are
gender, age, income, goal, career, dec, attr, sinc, intel, fun, amb, shar, like, prob, met. According to the source of the dataset the meaning of the fields are as follows:

1. The first five columns are demographic - we may want to use them to look at subgroups later.
2. The next seven columns are important. dec is the raters decision on whether this individual was a match and then follows scores out of ten on six characteristics: attractiveness, sincerity, intelligence, fun, ambitiousness and shared interests.
3. The like column is an overall rating. The prob column is a rating on whether the rater believed that interest would be reciprocated and the final column is a binary on whether the two had met prior to the speed date, with the lower value indicating that they had met before.

**Further Explanation: Listed below are the meaning for the data in each field.**
**Gender:** 0 = Female, 1 = Male
**Goal:**
What is your primary goal in participating in this event?
Seemed like a fun night out=1
To meet new people=2
To get a date=3
Looking for a serious relationship=4
To say I did it=5
Other=6

**Dec (decision on whether this individual was a match):** 1 = Match, 0 = Not a match

**attr, sinc, int, fun, amb, shar:** attractiveness, sincerity, intelligence, fun, ambitiousness and shared interests rating scores out of ten

**Like:** Overall rating
**Prob:** Rating on whether the rater believed that interest would be reciprocated.

**Met:** Did the two had met prior to the speed date. 2 = Did not meet, 1 = Met before

## II. Cleaning the Data

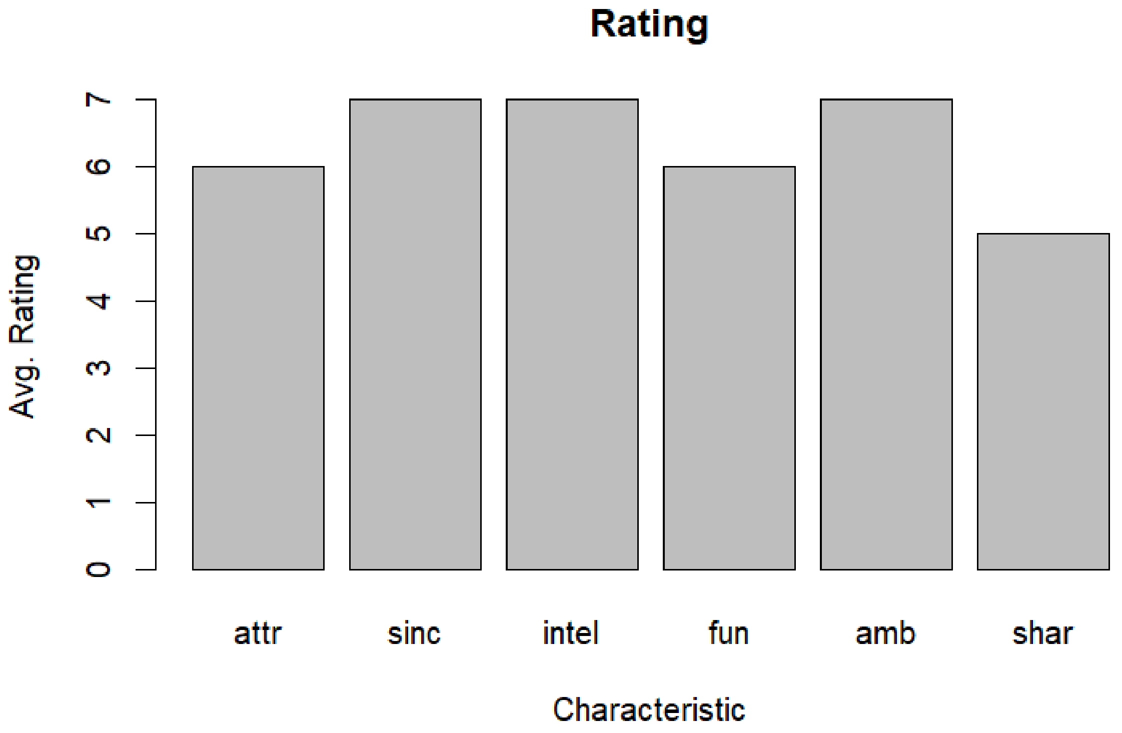
Before any form of analytics or processing will be done to the dataset the dataset must first be cleaned. In the dataset it has been observed that there are certain areas that had no data and is marked with (NA). To deal with this, the rows containing (NA) will be erased when needed.

```
clean_data <- na.omit(speed_data_data)
```

## III. Analysis of Data

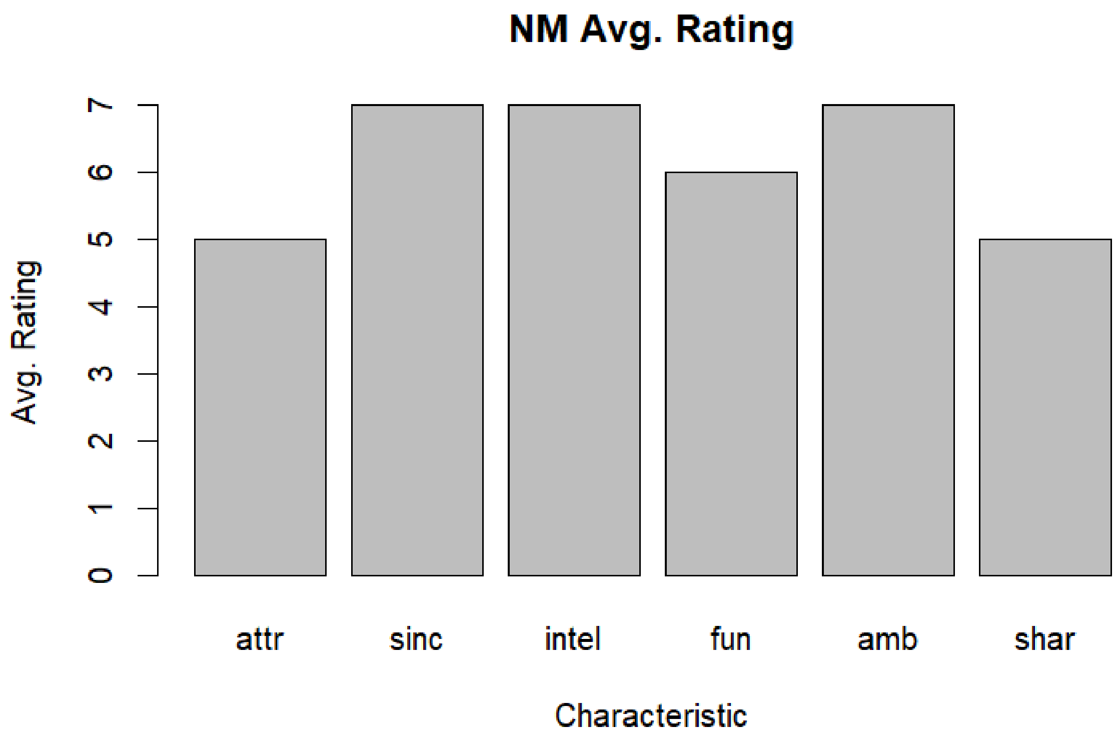
Finding the average for each rating category

The overall rating averages for each characteristic are as follows: 6, 7, 7, 6, 7, 5



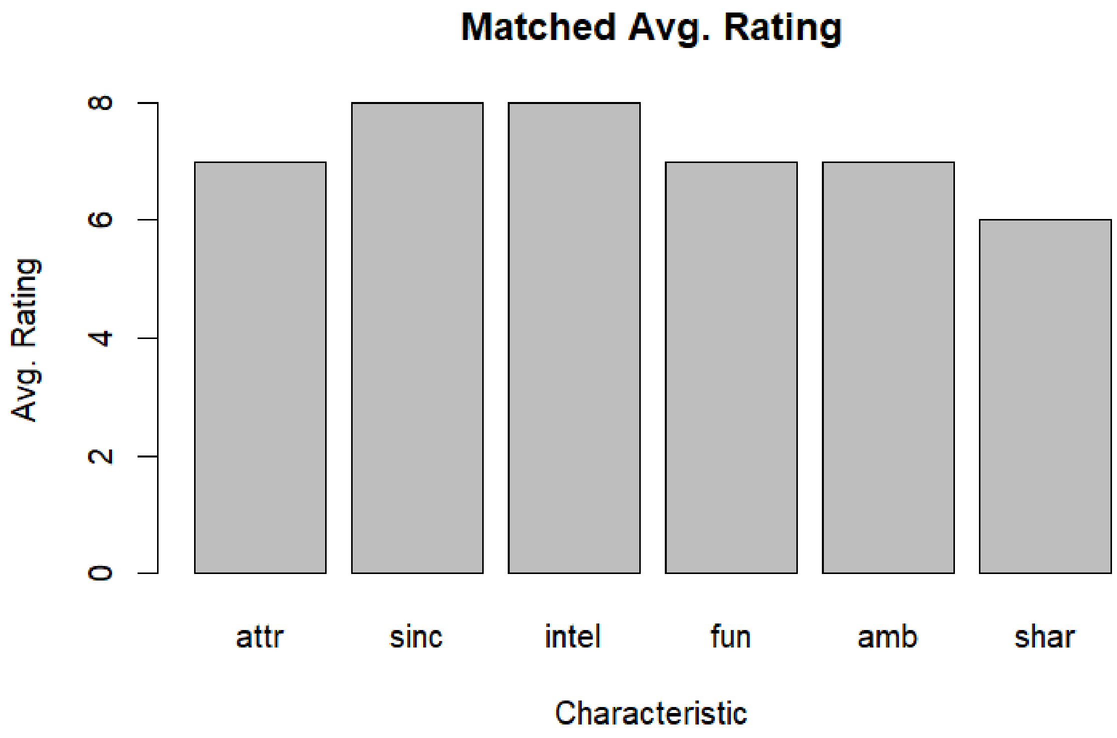
**The rating averages for not matched:**

The rating averages are as follows: 5 7 7 6 7 5

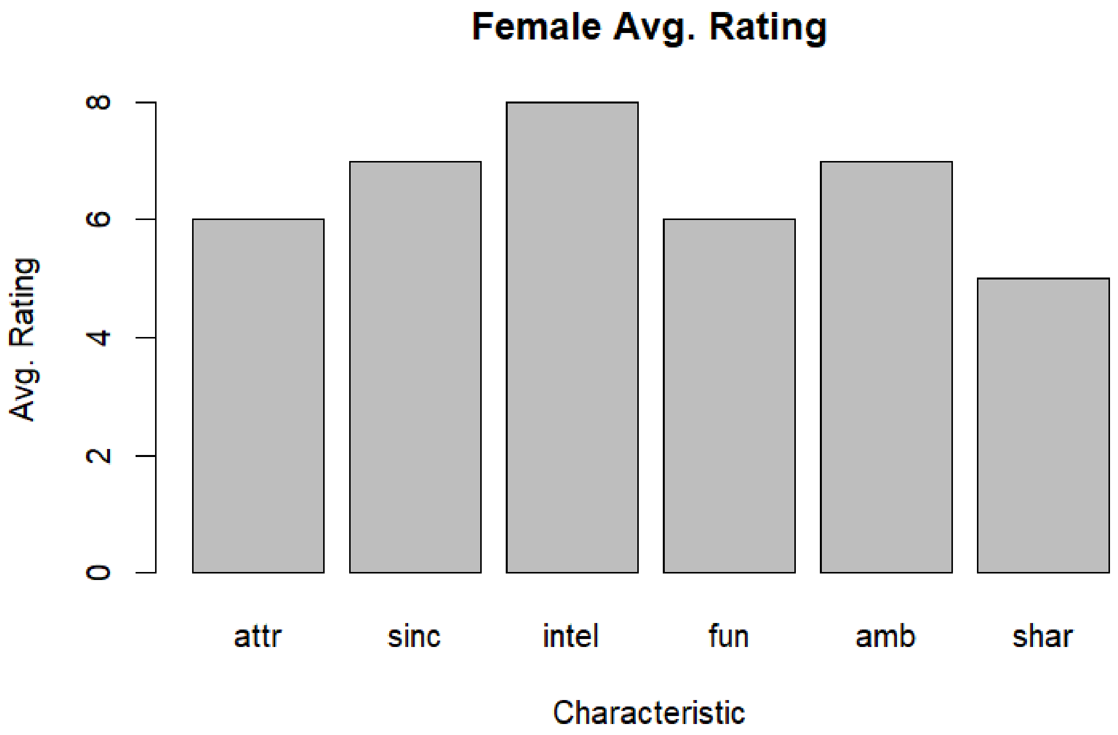


**The rating averages for matched:**

The rating averages are as follows: 7 8 8 7 7 6

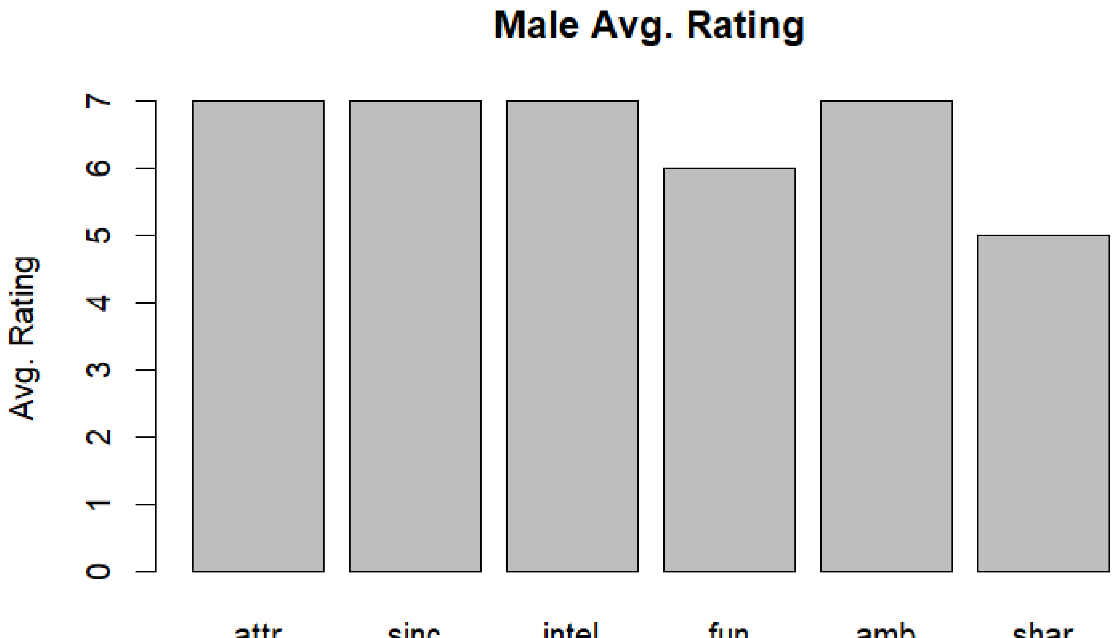


**The rating average for females:**
The rating averages are as follows:  6 7 8 6 7 5



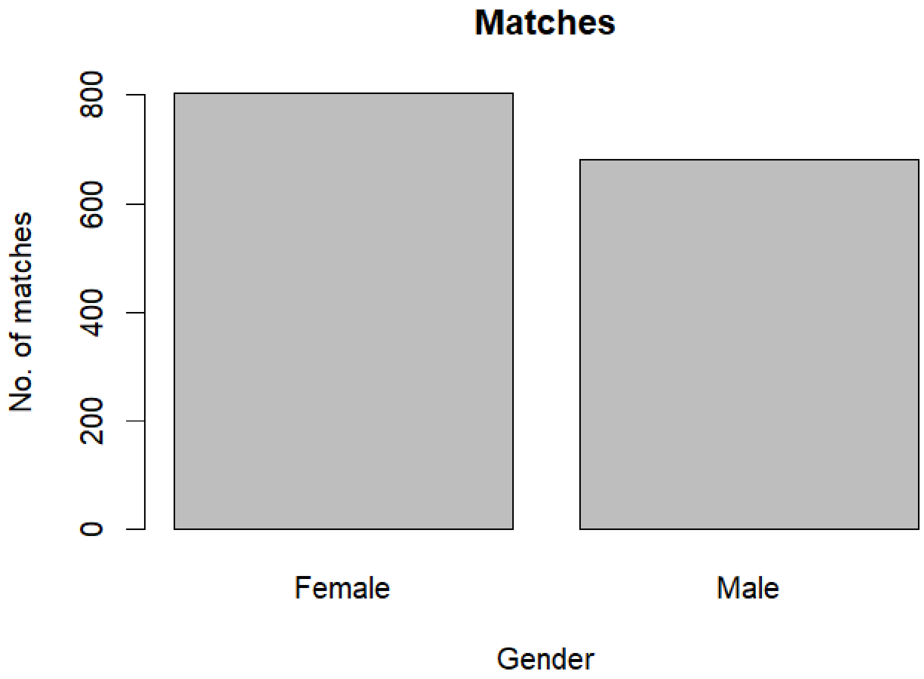
**The rating average for males:**

The rating average for males are as follows: 7 7 7 6 7 5



**Which gender gets more matches?**

The gender that gets more matches are Females with 804 getting matches in comparison to 682 Males getting Matches.



**Which characteristic in which did matter most in getting a match?**

```
|            |   Estimate| Std. Error|     z value| Pr(>|z|)          | odds_ratio|
|:-----------|----------:|----------:|-----------:|------------------:|----------:|
|(Intercept) | -7.5310787|  0.3339955| -22.5484422|          0.0000000|  0.0005362|
|attr        |  0.4985030|  0.0336310|  14.8227158|          0.0000000|  1.6462550|
|sinc        | -0.0977123|  0.0391876|  -2.4934506|          0.0126508|  0.9069098|
|intel       |  0.0026323|  0.0460121|   0.0572091|          0.9543786|  1.0026358|
|fun         |  0.1509904|  0.0354351|   4.2610401|          0.0000203|  1.1629855|
|amb         | -0.1993916|  0.0356382|  -5.5948847|          0.0000000|  0.8192290|
|shar        |  0.1444750|  0.0300887|   4.8016333|          0.0000016|  1.1554328|
|like        |  0.5061097|  0.0454431|  11.1372238|          0.0000000|  1.6588254|
|prob        |  0.1489714|  0.0244328|   6.0972019|          0.0000000|  1.1606398|
```

We want to determine the strength and direction of the relationship between match decision and other factors. We would therefore be applying a binomial logistical regression to the question of determining which characteristic matter most as we would be dealing with binary data which is match decision (our dependent variable) and find its relationship with ratings based on their characteristic.

All the data above have statistical significance due to most of their values being below 0.05. When we look at the Estimate column we can see that attr, fun, and like which represent attractiveness, fun, and likeness(overall rating) would be the biggest factor in deciding whether a person gets a match while characteristics like sincerity and ambitiousness would lower your chances of getting a match.

### Does meeting the person have an effect on decision?

Pearson's Chi-squared test with Yates' continuity correction

| X-squared | df | p-value |
|---|---|---|
| 15.318 | 1 | 9.083e-05 |

When processing for this analysis the met data column has been observed to contain values that are not binary in nature. To remedy this additional data processing is performed to the met column, this process involves turning numbers which are greater than 1 to simply 1. After that we now have 1’s and 0’s.

Applying a chi square test between the met column and the match decision column we return with the result above. We have a very low p-value ($p < 0.05$) which would indicate that there is indeed a correlation or relationship between meeting the person on the decision of whether a match will be made or not.

## IV. Full R Code

```
library(dplyr)
library(tidyverse)
library(readr)
library(corrplot)
library(stats)

speed_data_data <- read_csv("C:/Users/Acer/Desktop/speed_data_data.csv")

clean_data <- na.omit(speed_data_data)

view(clean_data)

#chisquare test
df_valid <- clean_data
for (x in 1:length(df_valid$met)){
  if( df_valid$met[x] >= 1 ){
    df_valid$met[x] <- 1
  }
}
print(df_valid$met)

result <- chisq.test(df_valid$dec, df_valid$met)
print(result)


#binomial logistical regression test

print(clean_data$met)

model <- glm(dec ~ attr + sinc + intel + fun + amb + shar + like + prob + met,
            data = clean_data,, family = binomial(link = "logit"))
summary(model)

coef(model)

ctable <- coef(summary(model))
odds_ratio <- exp(coef(summary(model))[ , c("Estimate")])
(coef_summary <-  cbind(ctable, as.data.frame(odds_ratio, nrow = nrow(ctable), ncol = 1)))
%>%
  knitr::kable()


#avg
avgVector <- c(mean(clean_data$attr),
              mean(clean_data$sinc),
              mean(clean_data$intel),
              mean(clean_data$fun),
              mean(clean_data$amb),
              mean(clean_data$shar)   )

avgVectorDat <- round(avgVector)
charNames <- c("attr","sinc","intel","fun","amb","shar")
barplot(avgVectorDat,names.arg=charNames,xlab="Characteristic",ylab="Avg.
Rating",main="Rating")


#avg by no matches

no_matches <- clean_data[clean_data$dec == 0,]
print(no_matches)
avgVector <- c(mean(no_matches$attr),
              mean(no_matches$sinc),
              mean(no_matches$intel),
              mean(no_matches$fun),
              mean(no_matches$amb),
              mean(no_matches$shar)   )

avgVectorDat <- round(avgVector)
print(avgVectorDat)
charNames <- c("attr","sinc","intel","fun","amb","shar")
barplot(avgVectorDat,names.arg=charNames,xlab="Characteristic",ylab="Avg.
Rating",main="NM Avg. Rating")

#avg by matches
has_matches <- clean_data[clean_data$dec == 1,]
print(has_matches)
avgVector <- c(mean(has_matches$attr),
            mean(has_matches$sinc),
            mean(has_matches$intel),
            mean(has_matches$fun),
            mean(has_matches$amb),
            mean(has_matches$shar)   )

avgVectorDat <- round(avgVector)
print(avgVectorDat)
charNames <- c("attr","sinc","intel","fun","amb","shar")
barplot(avgVectorDat,names.arg=charNames,xlab="Characteristic",ylab="Avg.
Rating",main="Matched Avg. Rating")

#avg by gender (female)
is_female <- clean_data[clean_data$gender == 0,]
print(is_female)
avgVector <- c(mean(is_female$attr),
            mean(is_female$sinc),
            mean(is_female$intel),
            mean(is_female$fun),
            mean(is_female$amb),
            mean(is_female$shar)   )

avgVectorDat <- round(avgVector)
print(avgVectorDat)
charNames <- c("attr","sinc","intel","fun","amb","shar")
barplot(avgVectorDat,names.arg=charNames,xlab="Characteristic",ylab="Avg.
Rating",main="Female Avg. Rating")

#avg by gender(male)
is_male <- clean_data[clean_data$gender == 1,]
print(is_male)
avgVector <- c(mean(is_male$attr),
            mean(is_male$sinc),
            mean(is_male$intel),
            mean(is_male$fun),
            mean(is_male$amb),
            mean(is_male$shar)   )

avgVectorDat <- round(avgVector)
print(avgVectorDat)
charNames <- c("attr","sinc","intel","fun","amb","shar")
barplot(avgVectorDat,names.arg=charNames,xlab="Characteristic",ylab="Avg.
Rating",main="Male Avg. Rating")


#Checking which gender matched more
match <- with(clean_data,tapply(gender,dec, FUN=sum))
```

```
print(match)
genderNames <- c("Female","Male")
barplot(match,names.arg=genderNames,xlab="Gender",ylab="No. of
matches",main="Matches")

#Checking which career received more matches
job <- clean_data %>% group_by(career) %>%
  summarise(Matches = sum(dec))

#Getting the index of the max of the matches
indexOfJob <- which.max(job$Matches)


#Output for which career received more matches
print(job[indexOfJob, ])


#Count of people that actually matched
sum(clean_data$dec)
```