R Notebook

EDA on Participants

```
# If errors occur check which functions are masked
library(readr)

## Warning: package 'readr' was built under R version 4.3.3

library(ggplot2)
library(corrplot)

## Warning: package 'corrplot' was built under R version 4.3.3

## corrplot 0.92 loaded

library(foreign)
library(leaps)
library(caret)

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: lattice

library(ROSE)

## Warning: package 'ROSE' was built under R version 4.3.3

## Loaded ROSE 0.0-4

library(tidyr)

## Warning: package 'tidyr' was built under R version 4.3.3

library(MASS)
library(rpart)
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 4.3.3

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.3

## randomForest 4.7-1.1

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

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

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

```r
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.3.3
```

```
## Loaded gbm 2.1.9
```

```
## This version of gbm is no longer under development. Consider transitioning
to gbm3, https://github.com/gbm-developers/gbm3
```

```r
library(dplyr)
```

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

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

```
## The following object is masked from 'package:MASS':
##
##     select
```

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

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

```r
attendees_data <- read_csv("csv_result-speeddating_unique.csv", na = "?")
```

```
## Rows: 551 Columns: 38
```

```
## ── Column specification
────────────────────────────────────────────────────────
```

```
## Delimiter: ","
## chr  (3): gender, race, field
## dbl (35): id, age, importance_same_race, importance_same_religion,
attractiv...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

Convert columns

```r
str(attendees_data)
```

```
## spc_tbl_ [551 × 38] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id                          : num [1:551] 3571 4845 849 879 4593 ...
##  $ gender                      : chr [1:551] "female" "female" "female"
"female" ...
##  $ age                         : num [1:551] 28 34 29 24 28 32 28 23 27
25 ...
##  $ race                        : chr [1:551] "Latino/Hispanic American"
"Asian/Pacific Islander/Asian-American" "Other" "Asian/Pacific
Islander/Asian-American" ...
##  $ importance_same_race        : num [1:551] 1 2 3 1 1 5 0 8 1 1 ...
##  $ importance_same_religion    : num [1:551] 1 1 10 3 10 6 1 10 4 10 ...
##  $ field                       : chr [1:551] "Political Science"
"Anthropology" "psychology" "Law" ...
##  $ attractive_important        : num [1:551] 0 2 5 5 5 5 5 6.67 6.67 7
...
##  $ sincere_important           : num [1:551] 25 60 20 15 20 ...
##  $ intelligence_important      : num [1:551] 25 15 25 45 25 ...
##  $ funny_important             : num [1:551] 25 8 15 25 25 ...
##  $ ambition_important          : num [1:551] 25 5 15 0 15 ...
##  $ shared_interests_important  : num [1:551] 0 10 20 10 10 ...
##  $ attractive                  : num [1:551] 9 5 7 6 6 4 9 7 6 6 ...
##  $ sincere                     : num [1:551] 9 9 9 8 10 8 8 8 10 10 ...
##  $ intelligence                : num [1:551] 9 8 9 9 10 6 5 8 6 8 ...
##  $ funny                       : num [1:551] 9 7 9 8 8 10 9 8 8 9 ...
##  $ ambition                    : num [1:551] 10 7 9 7 8 7 3 8 7 5 ...
##  $ sports                      : num [1:551] 4 8 3 5 10 6 6 7 8 9 ...
##  $ tvsports                    : num [1:551] 3 7 1 3 1 3 2 4 2 6 ...
##  $ exercise                    : num [1:551] 3 7 9 8 8 3 5 4 2 7 ...
##  $ dining                      : num [1:551] 8 6 9 6 7 10 8 8 8 7 ...
##  $ museums                     : num [1:551] 7 6 5 9 7 10 5 9 8 8 ...
##  $ art                         : num [1:551] 6 6 5 8 8 10 6 6 8 3 ...
##  $ hiking                      : num [1:551] 7 9 3 5 2 8 5 7 8 10 ...
##  $ gaming                      : num [1:551] 4 3 1 1 1 2 2 8 2 4 ...
##  $ clubbing                    : num [1:551] 9 6 5 2 4 3 9 6 8 1 ...
##  $ reading                     : num [1:551] 9 7 5 10 8 5 9 8 7 10 ...
##  $ tv                          : num [1:551] 2 2 2 1 1 4 8 5 1 3 ...
##  $ theater                     : num [1:551] 9 8 10 8 8 8 9 9 8 8 ...
##  $ movies                      : num [1:551] 9 8 7 9 9 9 9 5 9 5 ...
##  $ concerts                    : num [1:551] 9 8 6 4 7 9 6 7 7 2 ...
##  $ music                       : num [1:551] 9 8 9 5 7 9 6 8 7 6 ...
##  $ shopping                    : num [1:551] 9 3 5 6 8 9 4 8 2 6 ...
##  $ yoga                        : num [1:551] 2 1 5 2 9 7 7 8 2 2 ...
##  $ expected_happy_with_sd_people: num [1:551] 10 9 4 6 6 5 5 8 5 6 ...
##  $ expected_num_interested_in_me: num [1:551] NA NA 10 2 NA NA NA NA NA NA
...
##  $ expected_num_matches        : num [1:551] 2 2 2 NA 1 2 3 1 1 5 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   gender = col_character(),
```

```
##   ..   age = col_double(),
##   ..   race = col_character(),
##   ..   importance_same_race = col_double(),
##   ..   importance_same_religion = col_double(),
##   ..   field = col_character(),
##   ..   attractive_important = col_double(),
##   ..   sincere_important = col_double(),
##   ..   intelligence_important = col_double(),
##   ..   funny_important = col_double(),
##   ..   ambition_important = col_double(),
##   ..   shared_interests_important = col_double(),
##   ..   attractive = col_double(),
##   ..   sincere = col_double(),
##   ..   intelligence = col_double(),
##   ..   funny = col_double(),
##   ..   ambition = col_double(),
##   ..   sports = col_double(),
##   ..   tvsports = col_double(),
##   ..   exercise = col_double(),
##   ..   dining = col_double(),
##   ..   museums = col_double(),
##   ..   art = col_double(),
##   ..   hiking = col_double(),
##   ..   gaming = col_double(),
##   ..   clubbing = col_double(),
##   ..   reading = col_double(),
##   ..   tv = col_double(),
##   ..   theater = col_double(),
##   ..   movies = col_double(),
##   ..   concerts = col_double(),
##   ..   music = col_double(),
##   ..   shopping = col_double(),
##   ..   yoga = col_double(),
##   ..   expected_happy_with_sd_people = col_double(),
##   ..   expected_num_interested_in_me = col_double(),
##   ..   expected_num_matches = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>

attendees_data <- attendees_data %>%
    mutate(across(-all_of(c("gender", "field", "race")), as.numeric))

attendees_data <- attendees_data %>%
    mutate(across(all_of(c("gender", "field", "race")), as.factor))

str(attendees_data)

## tibble [551 × 38] (S3: tbl_df/tbl/data.frame)
##  $ id                            : num [1:551] 3571 4845 849 879 4593 ...
##  $ gender                        : Factor w/ 2 levels "female","male": 1 1
```

```
1 1 1 1 1 1 2 1 ...
##  $ age                        : num [1:551] 28 34 29 24 28 32 28 23 27
25 ...
##  $ race                       : Factor w/ 5 levels "Asian/Pacific
Islander/Asian-American",..: 4 1 5 1 3 3 3 3 3 5 ...
##  $ importance_same_race       : num [1:551] 1 2 3 1 1 5 0 8 1 1 ...
##  $ importance_same_religion   : num [1:551] 1 1 10 3 10 6 1 10 4 10 ...
##  $ field                      : Factor w/ 259 levels "Acting","African-
American Studies/History",..: 220 6 221 151 247 100 238 160 98 158 ...
##  $ attractive_important       : num [1:551] 0 2 5 5 5 5 5 6.67 6.67 7
...
##  $ sincere_important          : num [1:551] 25 60 20 15 20 ...
##  $ intelligence_important     : num [1:551] 25 15 25 45 25 ...
##  $ funny_important            : num [1:551] 25 8 15 25 25 ...
##  $ ambition_important         : num [1:551] 25 5 15 0 15 ...
##  $ shared_interests_important : num [1:551] 0 10 20 10 10 ...
##  $ attractive                 : num [1:551] 9 5 7 6 6 4 9 7 6 6 ...
##  $ sincere                    : num [1:551] 9 9 9 8 10 8 8 8 10 10 ...
##  $ intelligence               : num [1:551] 9 8 9 9 10 6 5 8 6 8 ...
##  $ funny                      : num [1:551] 9 7 9 8 8 10 9 8 8 9 ...
##  $ ambition                   : num [1:551] 10 7 9 7 8 7 3 8 7 5 ...
##  $ sports                     : num [1:551] 4 8 3 5 10 6 6 7 8 9 ...
##  $ tvsports                   : num [1:551] 3 7 1 3 1 3 2 4 2 6 ...
##  $ exercise                   : num [1:551] 3 7 9 8 8 3 5 4 2 7 ...
##  $ dining                     : num [1:551] 8 6 9 6 7 10 8 8 8 7 ...
##  $ museums                    : num [1:551] 7 6 5 9 7 10 5 9 8 8 ...
##  $ art                        : num [1:551] 6 6 5 8 8 10 6 6 8 3 ...
##  $ hiking                     : num [1:551] 7 9 3 5 2 8 5 7 8 10 ...
##  $ gaming                     : num [1:551] 4 3 1 1 1 2 2 8 2 4 ...
##  $ clubbing                   : num [1:551] 9 6 5 2 4 3 9 6 8 1 ...
##  $ reading                    : num [1:551] 9 7 5 10 8 5 9 8 7 10 ...
##  $ tv                         : num [1:551] 2 2 2 1 1 4 8 5 1 3 ...
##  $ theater                    : num [1:551] 9 8 10 8 8 8 9 9 8 8 ...
##  $ movies                     : num [1:551] 9 8 7 9 9 9 9 5 9 5 ...
##  $ concerts                   : num [1:551] 9 8 6 4 7 9 6 7 7 2 ...
##  $ music                      : num [1:551] 9 8 9 5 7 9 6 8 7 6 ...
##  $ shopping                   : num [1:551] 9 3 5 6 8 9 4 8 2 6 ...
##  $ yoga                       : num [1:551] 2 1 5 2 9 7 7 8 2 2 ...
##  $ expected_happy_with_sd_people: num [1:551] 10 9 4 6 6 5 5 8 5 6 ...
##  $ expected_num_interested_in_me: num [1:551] NA NA 10 2 NA NA NA NA NA NA
...
##  $ expected_num_matches       : num [1:551] 2 2 2 NA 1 2 3 1 1 5 ...
```

Summary statistics for numerical and categorical variables

```
summary(select_if(attendees_data, is.numeric))
```

```
##        id             age        importance_same_race
importance_same_religion
##  Min.   :   1    Min.   :18.00   Min.   : 0.000       Min.   : 1.000
##  1st Qu.:1884    1st Qu.:24.00   1st Qu.: 1.000       1st Qu.: 1.000
```

```
##  Median :4096   Median :26.00   Median : 3.000      Median : 3.000
##  Mean   :4078   Mean   :26.36   Mean   : 3.733      Mean   : 3.583
##  3rd Qu.:6402   3rd Qu.:28.00   3rd Qu.: 6.000      3rd Qu.: 6.000
##  Max.   :8357   Max.   :55.00   Max.   :10.000      Max.   :10.000
##                 NA's   :8       NA's   :7           NA's   :7
##  attractive_important sincere_important intelligence_important
funny_important
##  Min.   :  0.00       Min.   : 0.00     Min.   : 0.00          Min.   :
0.00
##  1st Qu.: 15.00       1st Qu.:14.93     1st Qu.:17.29          1st
Qu.:15.00
##  Median : 20.00       Median :18.00     Median :20.00          Median
:18.00
##  Mean   : 22.69       Mean   :17.29     Mean   :20.17          Mean
:17.45
##  3rd Qu.: 25.00       3rd Qu.:20.00     3rd Qu.:23.02          3rd
Qu.:20.00
##  Max.   :100.00       Max.   :60.00     Max.   :50.00          Max.
:50.00
##  NA's   :7            NA's   :7         NA's   :7              NA's   :8
##  ambition_important shared_interests_important   attractive
##  Min.   : 0.00      Min.   : 0.00              Min.   : 2.000
##  1st Qu.: 5.00      1st Qu.: 8.33              1st Qu.: 6.000
##  Median :10.00      Median :11.00              Median : 7.000
##  Mean   :10.81      Mean   :11.83              Mean   : 7.092
##  3rd Qu.:15.00      3rd Qu.:16.00              3rd Qu.: 8.000
##  Max.   :53.00      Max.   :30.00              Max.   :10.000
##  NA's   :9          NA's   :10                 NA's   :9
##     sincere        intelligence       funny           ambition
##  Min.   : 2.000   Min.   : 2.000   Min.   : 3.000   Min.   : 2.000
##  1st Qu.: 8.000   1st Qu.: 7.000   1st Qu.: 8.000   1st Qu.: 7.000
##  Median : 8.000   Median : 8.000   Median : 8.000   Median : 8.000
##  Mean   : 8.286   Mean   : 7.701   Mean   : 8.386   Mean   : 7.577
##  3rd Qu.: 9.000   3rd Qu.: 9.000   3rd Qu.: 9.000   3rd Qu.: 9.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##  NA's   :9        NA's   :9        NA's   :9        NA's   :9
##      sports          tvsports        exercise          dining
##  Min.   : 1.000   Min.   : 1.00   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 4.000   1st Qu.: 2.00   1st Qu.: 5.000   1st Qu.: 7.000
##  Median : 7.000   Median : 4.00   Median : 7.000   Median : 8.000
##  Mean   : 6.395   Mean   : 4.55   Mean   : 6.287   Mean   : 7.776
##  3rd Qu.: 8.250   3rd Qu.: 7.00   3rd Qu.: 8.000   3rd Qu.: 9.000
##  Max.   :10.000   Max.   :10.00   Max.   :10.000   Max.   :10.000
##  NA's   :7        NA's   :7       NA's   :7        NA's   :7
##     museums            art            hiking           gaming
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 6.000   1st Qu.: 5.000   1st Qu.: 4.000   1st Qu.: 1.00
##  Median : 7.000   Median : 7.000   Median : 6.000   Median : 3.00
##  Mean   : 6.972   Mean   : 6.689   Mean   : 5.757   Mean   : 3.84
##  3rd Qu.: 8.250   3rd Qu.: 8.000   3rd Qu.: 8.000   3rd Qu.: 6.00
```

```
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :14.00
##  NA's   :7        NA's   :7        NA's   :7        NA's   :7
##     clubbing         reading             tv            theater
##  Min.   : 0.000   Min.   : 1.000   Min.   : 1.000   Min.   : 0.000
##  1st Qu.: 4.000   1st Qu.: 7.000   1st Qu.: 3.000   1st Qu.: 5.000
##  Median : 6.000   Median : 8.000   Median : 6.000   Median : 7.000
##  Mean   : 5.752   Mean   : 7.647   Mean   : 5.325   Mean   : 6.761
##  3rd Qu.: 8.000   3rd Qu.: 9.000   3rd Qu.: 7.000   3rd Qu.: 9.000
##  Max.   :10.000   Max.   :13.000   Max.   :10.000   Max.   :10.000
##  NA's   :7        NA's   :7        NA's   :7        NA's   :7
##      movies          concerts          music           shopping
##  Min.   : 0.000   Min.   : 0.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 7.000   1st Qu.: 5.750   1st Qu.: 7.000   1st Qu.: 4.000
##  Median : 8.000   Median : 7.000   Median : 8.000   Median : 6.000
##  Mean   : 7.899   Mean   : 6.844   Mean   : 7.875   Mean   : 5.605
##  3rd Qu.: 9.000   3rd Qu.: 8.000   3rd Qu.: 9.000   3rd Qu.: 8.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
##  NA's   :7        NA's   :7        NA's   :7        NA's   :7
##       yoga        expected_happy_with_sd_people
expected_num_interested_in_me
##  Min.   : 0.000   Min.   : 1.000                Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 5.000                1st Qu.: 2.000
##  Median : 4.000   Median : 6.000                Median : 4.000
##  Mean   : 4.415   Mean   : 5.519                Mean   : 5.889
##  3rd Qu.: 7.000   3rd Qu.: 7.000                3rd Qu.: 9.000
##  Max.   :10.000   Max.   :10.000                Max.   :20.000
##  NA's   :7        NA's   :8                     NA's   :425
##  expected_num_matches
##  Min.   : 0.000
##  1st Qu.: 1.750
##  Median : 2.500
##  Mean   : 3.027
##  3rd Qu.: 4.000
##  Max.   :18.000
##  NA's   :72
```

```
summary(select_if(attendees_data, is.factor))
```

```
##     gender                                         race
##  female:274   Asian/Pacific Islander/Asian-American:136
##  male  :277   Black/African American               : 26
##               European/Caucasian-American          :304
##               Latino/Hispanic American             : 42
##               Other                                : 37
##               NA's                                 :  6
##
##                      field
##  Business                : 35
##  MBA                     : 35
##  Law                     : 33
```

```
##   Social Work          : 24
##   International Affairs: 15
##   (Other)              :403
##   NA's                 :  6
```

Check NA's

```
sum(is.na(attendees_data))

## [1] 751

dim(attendees_data)

## [1] 551  38
```

Define the mapping function

```
map_field <- function(field) {
    field <- tolower(field)  # Convert to lowercase for uniformity
    if (field %in% c("biology", "biochemistry", "biomedical engineering",
"biomedical informatics", "biochemistry & molecular biophysics",
                     "biochemistry/genetics", "cell biology", "chemistry",
"environmental science", "geology", "genetics",
                     "molecular biology", "neurobiology", "neuroscience",
"neurosciences/stem cells", "microbiology",
                     "statistics", "math", "math of finance", "mathematics",
"mathematics; phd", "climate dynamics", "climate-earth and environ. science",
                     "computational biochemsistry", "conservation biology",
"epidemiology", "nutritiron", "nutrition", "nutrition/genetics", "applied
maths/econs",
                     "mathematical finance", "computer science", "marine
geophysics", "physics", "physics [astrophysics]", "climate change", "ma
biotechnology",
                     "ecology", "earth and environmental science", "ma
science education")) {
        return("Science")
    } else if (field %in% c("mechanical engineering", "electrical
engineering", "civil engineering", "computer engineering",
                            "chemical engineering", "industrial engineering",
"industrial engineering/operations research",
                            "operations research", "operations research
[seas]", "environmental engineering", "biomedical engineering",
                            "engineering", "financial engineering",
"electrical engg.", "industrial engineering/operations research",
                            "masters of industrial engineering")) {
        return("Engineering")
    } else if (field %in% c("history", "philosophy", "literature",
"comparative literature", "english", "religious studies",
                            "classics", "french", "modern chinese
literature", "german literature", "english and comp lit",
                            "philosophy [ph.d.]", "philosophy and physics",
"history [gsas - phd]", "african-american studies/history", "religion",
```

```
                            "history of religion", "american studies",
"american studies [masters]", "japanese literature", "Art history",
"international politics",
                            "museum anthropology")) {
        return("Humanities")
    } else if (field %in% c("psychology", "sociology", "political science",
"economics", "anthropology", "education", "social work",
                            "social studies education", "international
relations", "international affairs", "international business",
                            "public policy", "cognitive studies in
education", "education administration", "education leadership - public school
administration",
                            "education policy", "elementary education",
"higher ed. - m.a.", "masters of social work&education", "organizational
psychology",
                            "sociomedical sciences- school of public health",
"social work", "speech language pathology", "speech languahe pathology",
                            "speech pathology", "speech pathology", "school
psychology", "instructional tech & media", "instructional media and
technology",
                            "education- literacy specialist", "bilingual
education", "finance/economics", "law and social work", "sociology and
education",
                            "clinical psychology", "international educational
development", "social work/sipa", "neuroscience and education", "sociology",
                            "neurosciences/stem cells", "education", "law",
"educational psychology")) {
        return("Social Sciences")
    } else if (field %in% c("business", "finance", "marketing", "management",
"accounting", "business administration",
                            "operations research", "business [mba]",
"business- mba", "mba", "mba finance", "mba / master of international affairs
[sipa]",
                            "mba - private equity / real estate", "business &
international affairs", "business school", "business; media", "financial
math",
                            "general management/finance", "business;
marketing", "international finance and business", "international finance;
economic policy",
                            "business/ finance/ real estate", "business;
international affairs", "consulting", "fundraising management",
                            "finance&economics", "finance", "finanace",
"financial engineering", "business administration", "operations research",
                            "financial math", "money")) {
        return("Business")
    } else if (field %in% c("medicine", "nursing", "public health",
"pharmacy", "dentistry", "veterinary medicine", "epidemiology",
                            "health policy", "medical informatics",
"biomedical informatics", "clinical psychology", "counseling psychology",
                            "medical informatics", "biomedical informatics",
```

```
"neuroscience", "public administration", "health policy", "tc [health ed]",
"biomedicine")) {
        return("Health")
    } else if (field %in% c("art", "music", "theater", "film", "design", "art
education", "arts administration", "theatre management & producing",
                            "mfa creative writing", "mfa -film", "mfa
writing", "mfa acting program", "creative writing", "creative writing -
nonfiction",
                            "writing: literary nonfiction", "creative writing
[nonfiction]", "nonfiction writing", "mfa poetry", "acting", "arts
administration",
                            "journalism")) {
        return("Arts")
    } else {
        return("Other")
    }
}
```

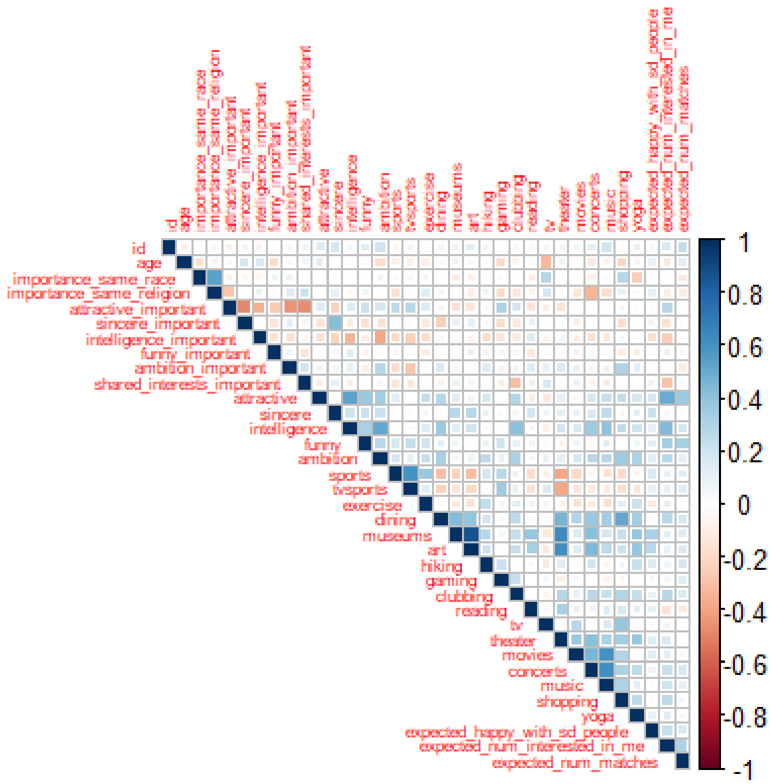
Categorize fields

```
attendees_data$field_category <- sapply(attendees_data$field, map_field)
attendees_data$field_category <- as.factor(attendees_data$field_category)
```

Graphs

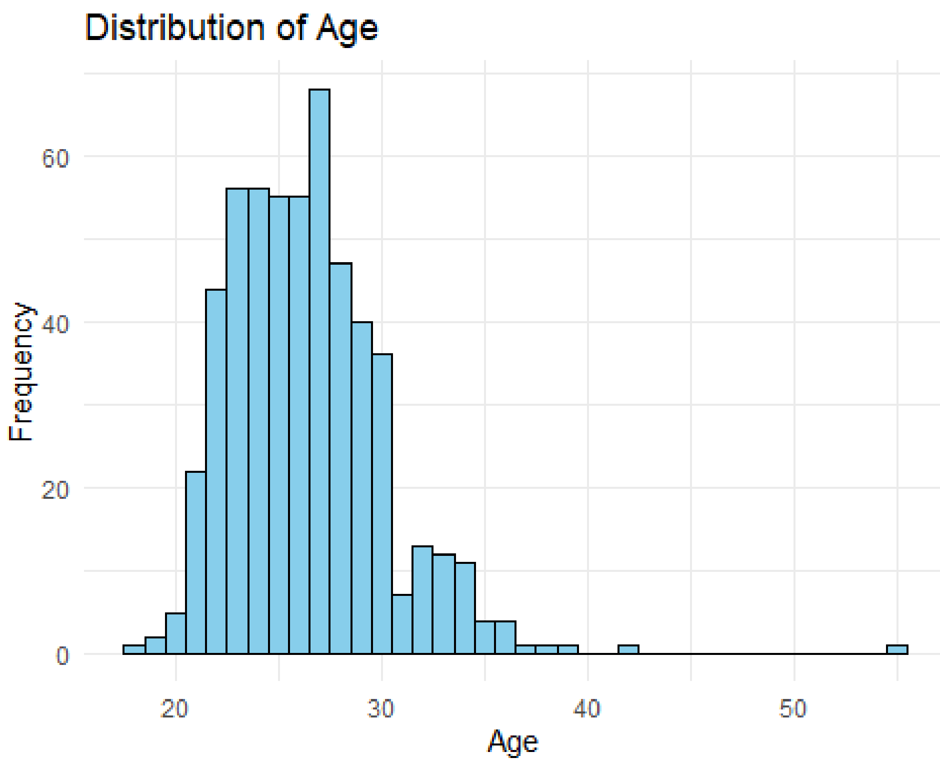
Correlation Matrix

```
cor_matrix <- cor(select_if(attendees_data, is.numeric), use =
"complete.obs")

corrplot(cor_matrix, method = "square", type = "upper", tl.cex = 0.5)
```



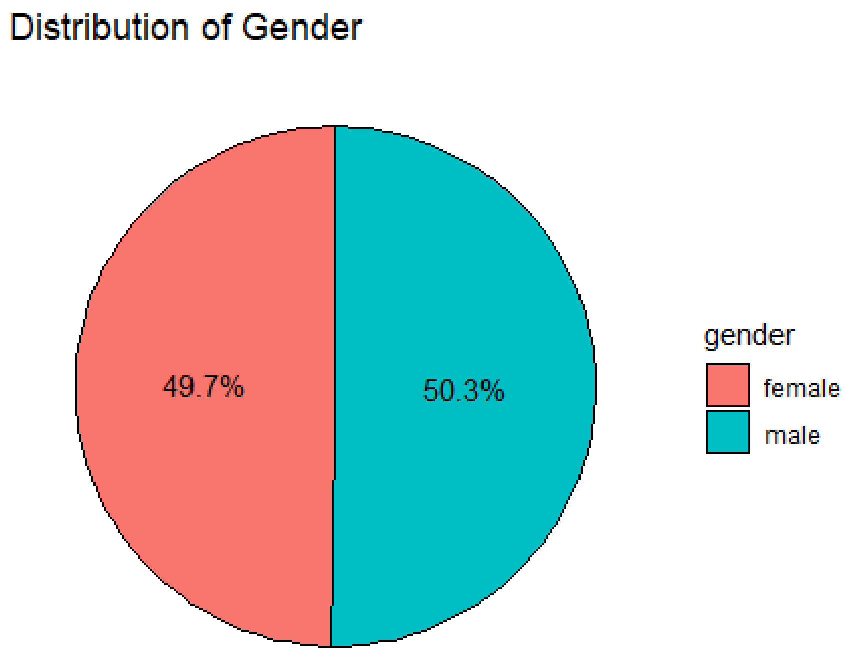
Age

```r
# Distribution of age
ggplot(attendees_data, aes(x = age)) +
    geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
    labs(title = "Distribution of Age", x = "Age", y = "Frequency") +
    theme_minimal()

## Warning: Removed 8 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



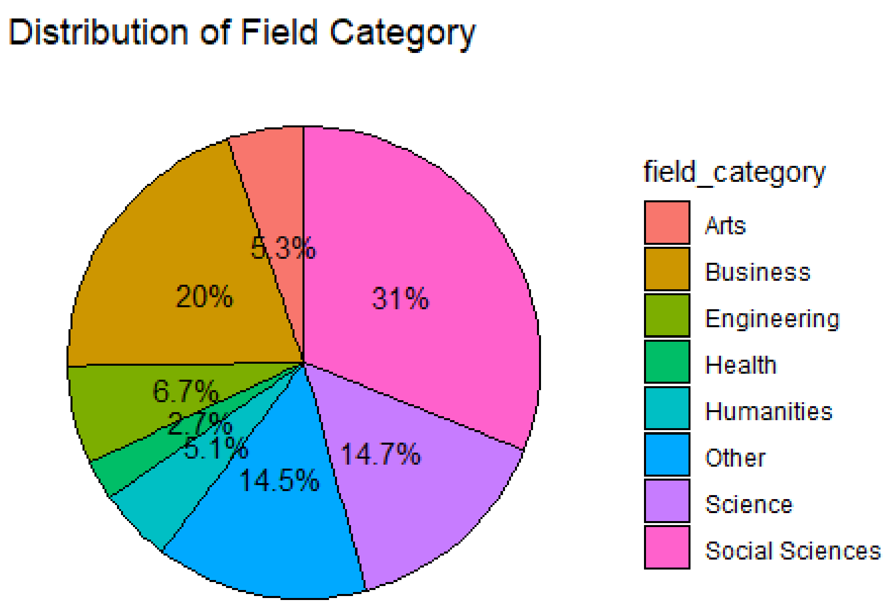
Gender

```
gender_distribution <- attendees_data %>%
    group_by(gender) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)

ggplot(gender_distribution, aes(x = "", y = count, fill = gender)) +
    geom_bar(stat = "identity", width = 1, color = "black") +
    coord_polar("y") +
    geom_text(aes(label = paste0(round(percentage, 1), "%")),
              position = position_stack(vjust = 0.5), color = "black") +
    labs(title = "Distribution of Gender", x = "", y = "") +
    theme_minimal() +
    theme(axis.text.x = element_blank(),  # Remove x-axis text
          axis.ticks = element_blank(),   # Remove x-axis ticks
          panel.grid = element_blank())   # Remove background grid
```



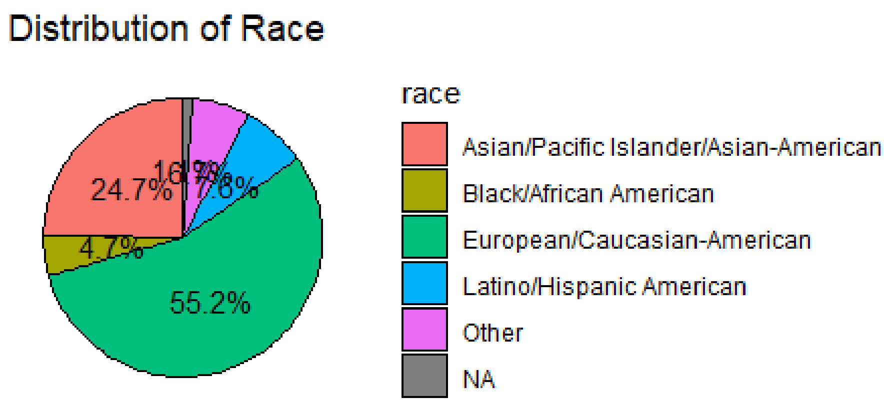
Field Category

```
field_category_distribution <- attendees_data %>%
    group_by(field_category) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)

ggplot(field_category_distribution, aes(x = "", y = count, fill =
field_category)) +
    geom_bar(stat = "identity", width = 1, color = "black") +
    coord_polar("y") +
    geom_text(aes(label = paste0(round(percentage, 1), "%")),
              position = position_stack(vjust = 0.5), color = "black") +
    labs(title = "Distribution of Field Category", x = "", y = "") +
    theme_minimal() +
    theme(axis.text.x = element_blank(),  # Remove x-axis text
          axis.ticks = element_blank(),   # Remove x-axis ticks
          panel.grid = element_blank())   # Remove background grid
```



Race

```
race_distribution <- attendees_data %>%
    group_by(race) %>%
    summarise(count = n()) %>%
    mutate(percentage = count / sum(count) * 100)

ggplot(race_distribution, aes(x = "", y = count, fill = race)) +
    geom_bar(stat = "identity", width = 1, color = "black") +
    coord_polar("y") +
    geom_text(aes(label = paste0(round(percentage, 1), "%")),
              position = position_stack(vjust = 0.5), color = "black") +
    labs(title = "Distribution of Race", x = "", y = "") +
    theme_minimal() +
    theme(axis.text.x = element_blank(),  # Remove x-axis text
          axis.ticks = element_blank(),   # Remove x-axis ticks
          panel.grid = element_blank())   # Remove background grid
```

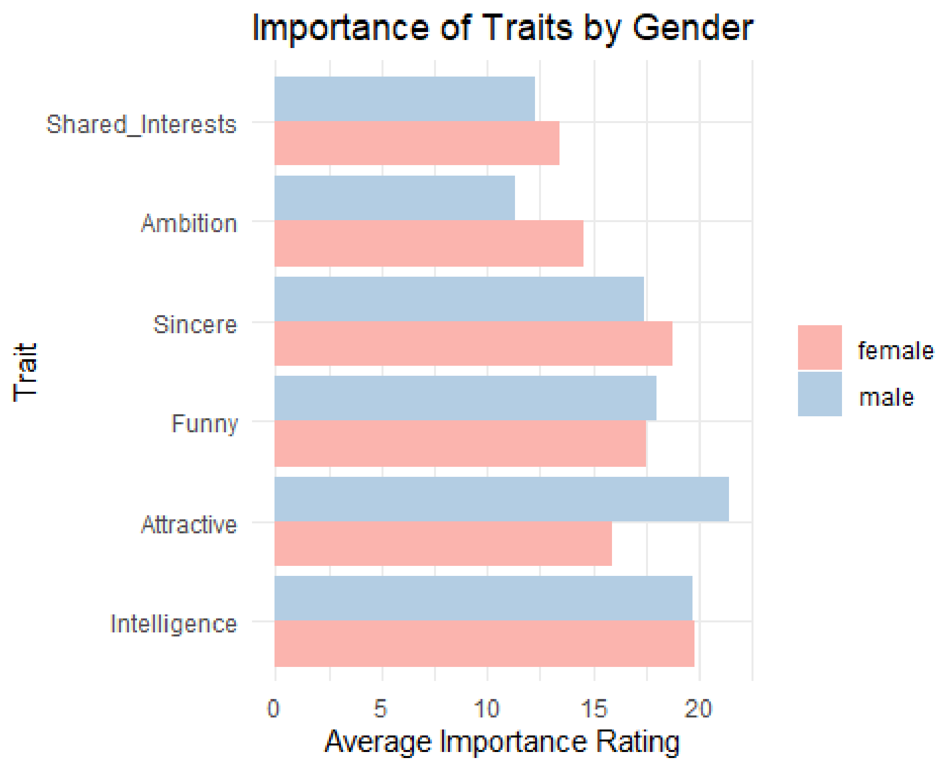


Importance of Traits by Gender

```
importance_vars <- c("attractive_important", "sincere_important",
"intelligence_important", "funny_important", "ambition_important",
"shared_interests_important")

# Define the function to remove outliers
remove_outliers <- function(df, cols) {
  for (col in cols) {
    Q1 <- quantile(df[[col]], 0.25, na.rm = TRUE)
    Q3 <- quantile(df[[col]], 0.75, na.rm = TRUE)
    IQR <- Q3 - Q1
    lower_bound <- Q1 - 1.5 * IQR
    upper_bound <- Q3 + 1.5 * IQR
    df <- df %>%
      filter(df[[col]] >= lower_bound & df[[col]] <= upper_bound)
  }
  return(df)
}

dim(attendees_data)

## [1] 551  39

attendees_data_no_outliers <- remove_outliers(attendees_data,
importance_vars)
dim(attendees_data_no_outliers)
```

```
## [1] 355  39
```

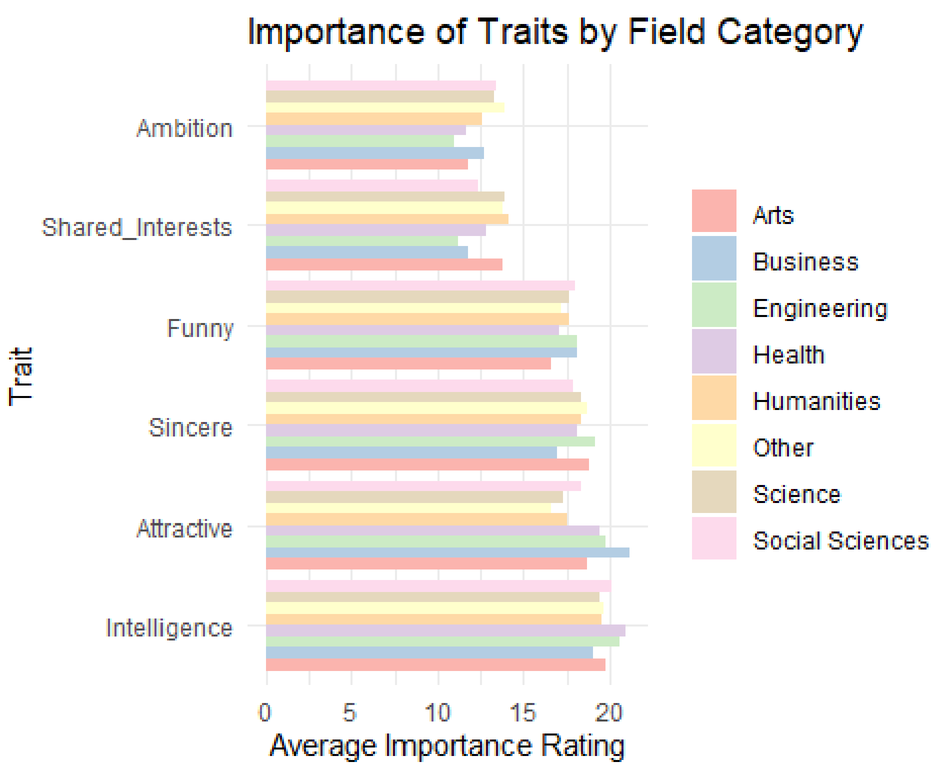
```r
# Calculate mean importance by gender for each attribute
importance_by_gender <- attendees_data_no_outliers %>%
  group_by(gender) %>%
  summarise(
    Attractive = mean(attractive_important),
    Sincere = mean(sincere_important),
    Intelligence = mean(intelligence_important),
    Funny = mean(funny_important),
    Ambition = mean(ambition_important),
    Shared_Interests = mean(shared_interests_important),
    .groups = 'drop'
  )

# Reshape data for plotting
importance_long <- importance_by_gender %>%
  pivot_longer(cols = -gender, names_to = "Trait", values_to = "Importance")

# Add ranking information within each gender
importance_long <- importance_long %>%
  group_by(gender) %>%
  mutate(Rank = rank(-Importance))  # Rank in descending order of importance

# Plotting
ggplot(importance_long, aes(x = reorder(Trait, Rank), y = Importance, fill =
gender)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  coord_flip() +  # Flip coordinates for horizontal bars
  labs(title = "Importance of Traits by Gender",
       x = "Trait",
       y = "Average Importance Rating") +
  theme_minimal() +
  theme(legend.title = element_blank()) +
  scale_fill_brewer(palette = "Pastel1")
```



Importance of Traits by Field Category

```
importance_vars <- c("attractive_important", "sincere_important",
"intelligence_important", "funny_important", "ambition_important",
"shared_interests_important")

attendees_data_no_outliers <- remove_outliers(attendees_data,
importance_vars)

# Calculate mean importance by gender for each attribute
importance_by_field_category <- attendees_data_no_outliers %>%
  group_by(field_category) %>%
  summarise(
    Attractive = mean(attractive_important),
    Sincere = mean(sincere_important),
    Intelligence = mean(intelligence_important),
    Funny = mean(funny_important),
    Ambition = mean(ambition_important),
    Shared_Interests = mean(shared_interests_important),
    .groups = 'drop'
  )

# Reshape data for plotting
importance_long <- importance_by_field_category %>%
  pivot_longer(cols = -field_category, names_to = "Trait", values_to =
"Importance")
```

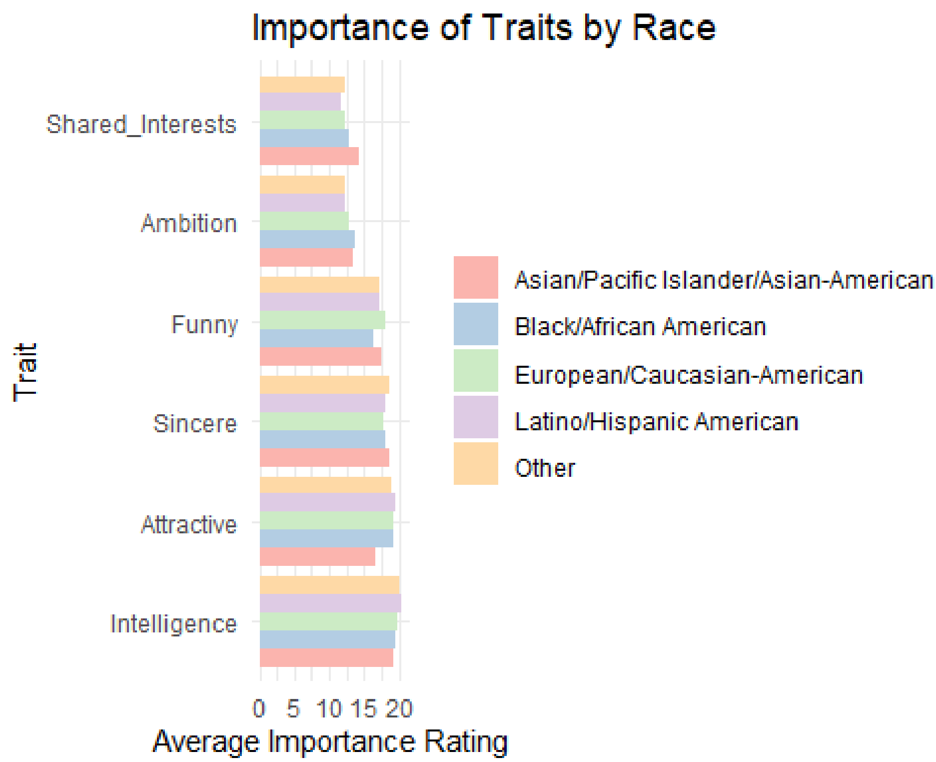
```r
# Add ranking information within each gender
importance_long <- importance_long %>%
  group_by(field_category) %>%
  mutate(Rank = rank(-Importance))  # Rank in descending order of importance

# Plotting
ggplot(importance_long, aes(x = reorder(Trait, Rank), y = Importance, fill =
field_category)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  coord_flip() +  # Flip coordinates for horizontal bars
  labs(title = "Importance of Traits by Field Category",
       x = "Trait",
       y = "Average Importance Rating") +
  theme_minimal() +
  theme(legend.title = element_blank()) +
  scale_fill_brewer(palette = "Pastel1")
```



Importance of Traits by Race

```r
importance_vars <- c("attractive_important", "sincere_important",
"intelligence_important", "funny_important", "ambition_important",
"shared_interests_important")

attendees_data_no_outliers <- remove_outliers(attendees_data,
importance_vars)

# Calculate mean importance by gender for each attribute
```

```r
importance_by_race <- attendees_data_no_outliers %>%
  group_by(race) %>%
  summarise(
    Attractive = mean(attractive_important),
    Sincere = mean(sincere_important),
    Intelligence = mean(intelligence_important),
    Funny = mean(funny_important),
    Ambition = mean(ambition_important),
    Shared_Interests = mean(shared_interests_important),
    .groups = 'drop'
  )

# Reshape data for plotting
importance_long <- importance_by_race %>%
  pivot_longer(cols = -race, names_to = "Trait", values_to = "Importance")

# Add ranking information within each gender
importance_long <- importance_long %>%
  group_by(race) %>%
  mutate(Rank = rank(-Importance))  # Rank in descending order of importance

# Plotting
ggplot(importance_long, aes(x = reorder(Trait, Rank), y = Importance, fill =
race)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.9)) +
  coord_flip() +  # Flip coordinates for horizontal bars
  labs(title = "Importance of Traits by Race",
       x = "Trait",
       y = "Average Importance Rating") +
  theme_minimal() +
  theme(legend.title = element_blank()) +
  scale_fill_brewer(palette = "Pastel1")
```

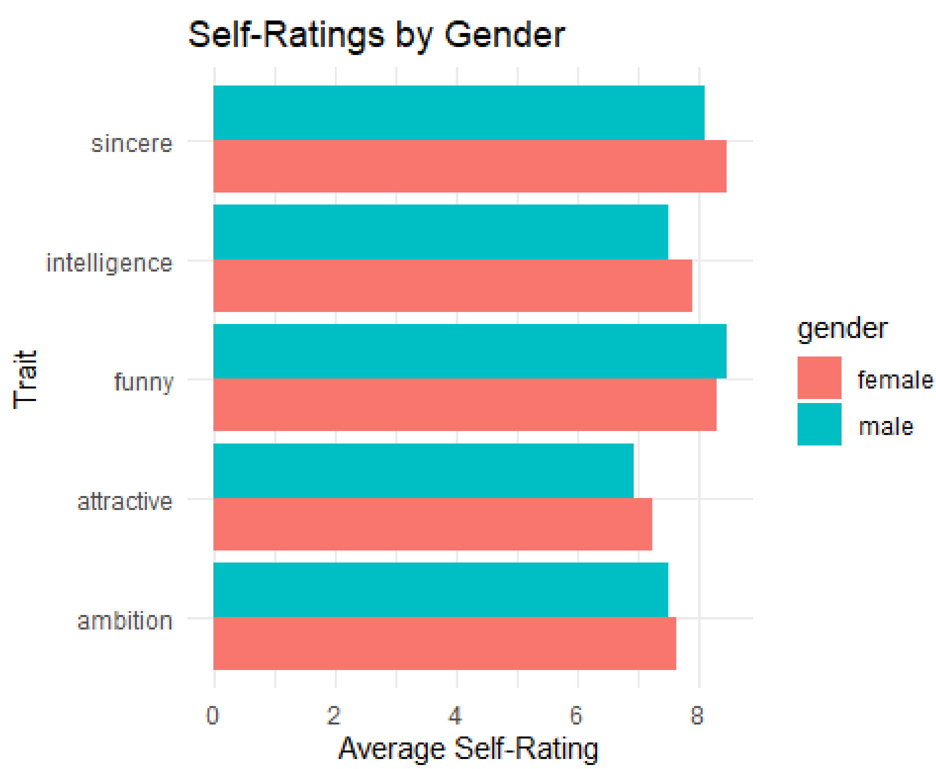


Self-ratings

```
self_ratings_vars <- c("attractive", "sincere", "intelligence", "funny",
"ambition")

dim(attendees_data)

## [1] 551  39

attendees_data_no_outliers <- remove_outliers(attendees_data,
self_ratings_vars)
dim(attendees_data_no_outliers)

## [1] 455  39

# Aggregate self-ratings by gender
self_ratings_gender <- attendees_data %>%
  group_by(gender) %>%
  summarise(across(self_ratings_vars, mean, na.rm = TRUE)) %>%
  pivot_longer(cols = -gender, names_to = "Trait", values_to =
"Average_Rating")

## Warning: There were 2 warnings in `summarise()`.
## The first warning was:
## ℹ In argument: `across(self_ratings_vars, mean, na.rm = TRUE)`.
## ℹ In group 1: `gender = female`.
## Caused by warning:
## ! Using an external vector in selections was deprecated in tidyselect
```

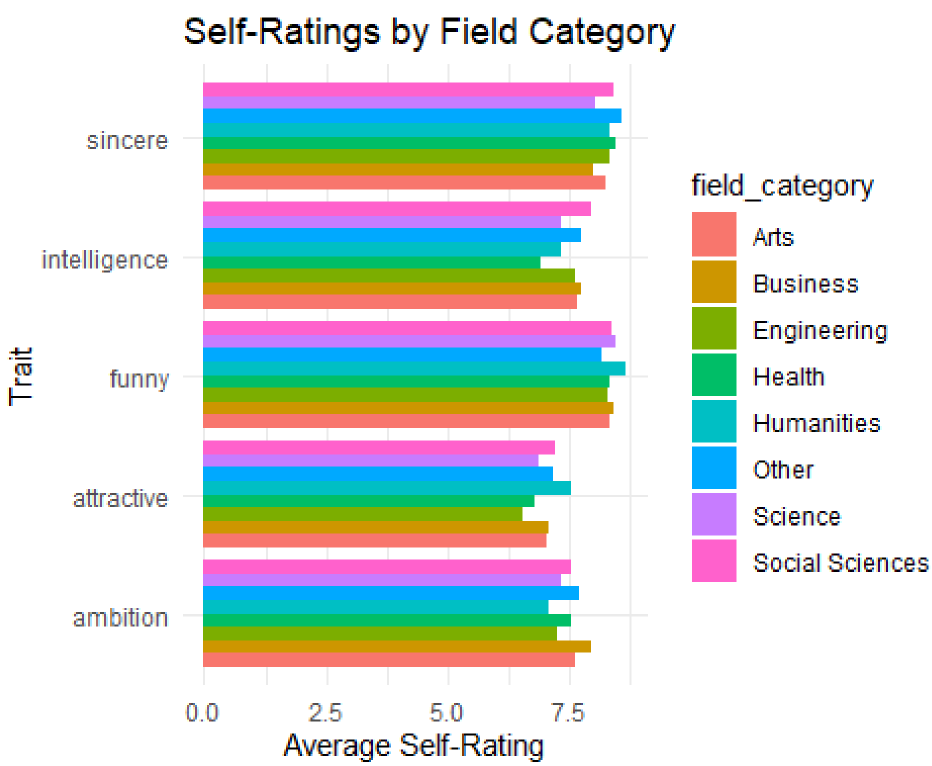
```
1.1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(self_ratings_vars)
##
##   # Now:
##   data %>% select(all_of(self_ratings_vars))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
# Plot self-ratings by gender
ggplot(self_ratings_gender, aes(x = Trait, y = Average_Rating, fill =
gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Self-Ratings by Gender", x = "Trait", y = "Average Self-
Rating") +
  theme_minimal() +
  coord_flip()
```

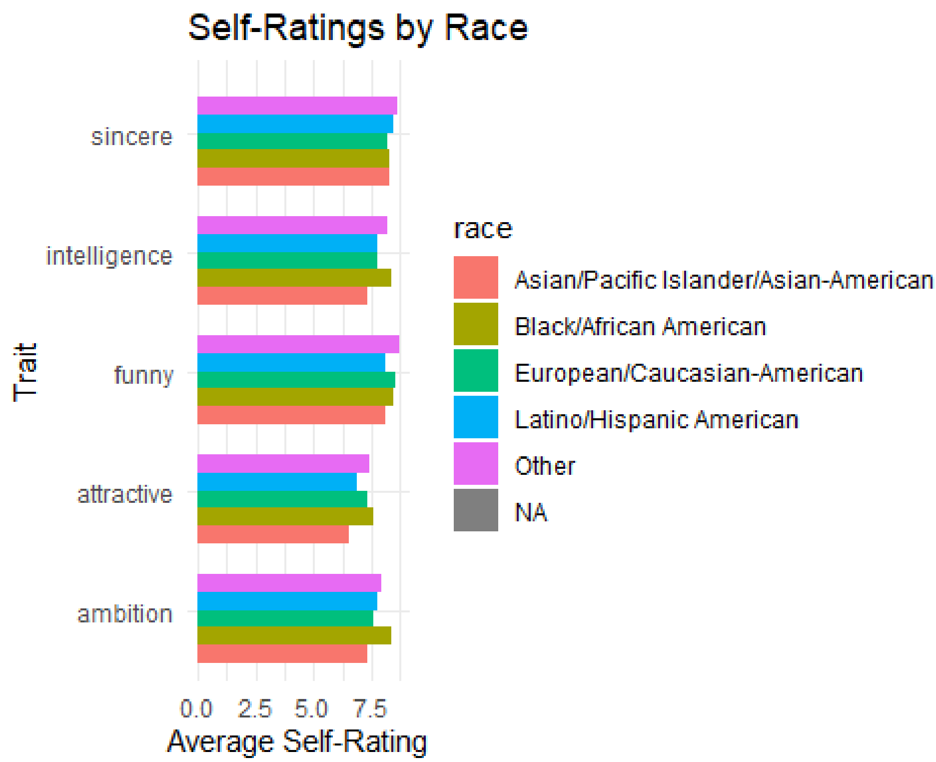



```
# Aggregate self-ratings by field category
self_ratings_field <- attendees_data %>%
  group_by(field_category) %>%
  summarise(across(self_ratings_vars, mean, na.rm = TRUE)) %>%
  pivot_longer(cols = -field_category, names_to = "Trait", values_to =
"Average_Rating")
```

```
# Plot self-ratings by field category
ggplot(self_ratings_field, aes(x = Trait, y = Average_Rating, fill =
field_category)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Self-Ratings by Field Category", x = "Trait", y = "Average
Self-Rating") +
  theme_minimal() +
  coord_flip()
```

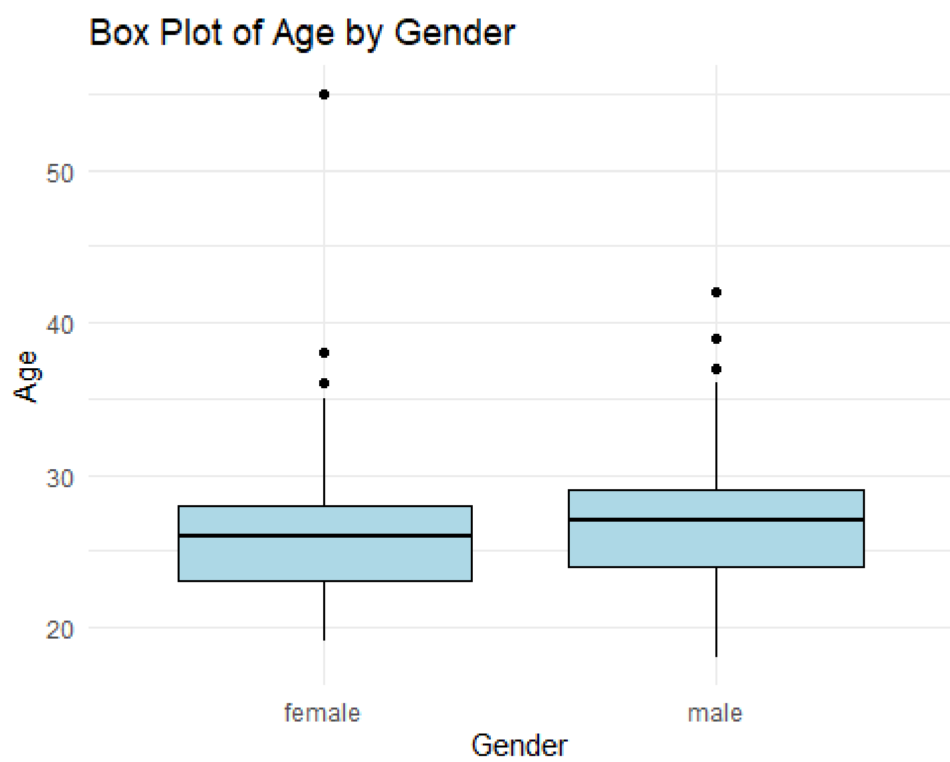



```
# Aggregate self-ratings by race
self_ratings_race <- attendees_data %>%
  group_by(race) %>%
  summarise(across(self_ratings_vars, mean, na.rm = TRUE)) %>%
  pivot_longer(cols = -race, names_to = "Trait", values_to =
"Average_Rating")

# Plot self-ratings by race
ggplot(self_ratings_race, aes(x = Trait, y = Average_Rating, fill = race)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Self-Ratings by Race", x = "Trait", y = "Average Self-
Rating") +
  theme_minimal() +
  coord_flip()

## Warning: Removed 5 rows containing missing values or values outside the
scale range
## (`geom_bar()`).
```



Box plots

```
ggplot(attendees_data, aes(x = gender, y = age)) +
    geom_boxplot(fill = "lightblue", color = "black") +
    labs(title = "Box Plot of Age by Gender", x = "Gender", y = "Age") +
    theme_minimal()

## Warning: Removed 8 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Data Cleaning

```
set.seed(12345)
speeddating_data <- read_csv("csv_result-speeddating.csv",
                             na = "?")

## Rows: 8378 Columns: 124
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr (59): gender, d_d_age, race, race_o, d_importance_same_race,
d_importanc...
## dbl (65): id, has_null, wave, age, age_o, d_age, samerace,
importance_same_r...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

Check NA's

```
sum(is.na(speeddating_data))

## [1] 18372

dim(speeddating_data)

## [1] 8378  124
```

Remove Columns

```
remove_vars <- c("wave","has_null", "expected_num_interested_in_me",
"decision", "decision_o", "met")
speeddating_data_new <- speeddating_data %>%
  dplyr::select(-starts_with("d_"), -all_of(remove_vars))
```

Add field categories and remove field

```
speeddating_data_new$field_category <- sapply(speeddating_data_new$field,
map_field)
speeddating_data_new$field_category <-
as.factor(speeddating_data_new$field_category)
speeddating_data_new <- select(speeddating_data_new, -"field")
str(speeddating_data_new)

## tibble [8,378 × 62] (S3: tbl_df/tbl/data.frame)
##  $ id                            : num [1:8378] 1 2 3 4 5 6 7 8 9 10 ...
##  $ gender                        : chr [1:8378] "female" "female" "female"
"female" ...
##  $ age                           : num [1:8378] 21 21 21 21 21 21 21 21 21
21 ...
##  $ age_o                         : num [1:8378] 27 22 22 23 24 25 30 27 28
24 ...
##  $ race                          : chr [1:8378] "Asian/Pacific
Islander/Asian-American" "Asian/Pacific Islander/Asian-American"
"Asian/Pacific Islander/Asian-American" "Asian/Pacific Islander/Asian-
American" ...
##  $ race_o                        : chr [1:8378] "European/Caucasian-
American" "European/Caucasian-American" "Asian/Pacific Islander/Asian-
American" "European/Caucasian-American" ...
##  $ samerace                      : num [1:8378] 0 0 1 0 0 0 0 0 0 0 ...
##  $ importance_same_race          : num [1:8378] 2 2 2 2 2 2 2 2 2 2 ...
##  $ importance_same_religion      : num [1:8378] 4 4 4 4 4 4 4 4 4 4 ...
##  $ pref_o_attractive             : num [1:8378] 35 60 19 30 30 ...
##  $ pref_o_sincere                : num [1:8378] 20 0 18 5 10 ...
##  $ pref_o_intelligence           : num [1:8378] 20 0 19 15 20 ...
##  $ pref_o_funny                  : num [1:8378] 20 40 18 40 10 ...
##  $ pref_o_ambitious              : num [1:8378] 0 0 14 5 10 ...
##  $ pref_o_shared_interests       : num [1:8378] 5 0 12 5 20 ...
##  $ attractive_o                  : num [1:8378] 6 7 10 7 8 7 3 6 7 6 ...
##  $ sinsere_o                     : num [1:8378] 8 8 10 8 7 7 6 7 7 6 ...
##  $ intelligence_o                : num [1:8378] 8 10 10 9 9 8 7 5 8 6 ...
##  $ funny_o                       : num [1:8378] 8 7 10 8 6 8 5 6 8 6 ...
##  $ ambitous_o                    : num [1:8378] 8 7 10 9 9 7 8 8 8 6 ...
##  $ shared_interests_o            : num [1:8378] 6 5 10 8 7 7 7 6 9 6 ...
##  $ attractive_important          : num [1:8378] 15 15 15 15 15 15 15 15 15
15 ...
##  $ sincere_important             : num [1:8378] 20 20 20 20 20 20 20 20 20
20 ...
##  $ intellicence_important        : num [1:8378] 20 20 20 20 20 20 20 20 20
20 ...
```

```
##  $ funny_important            : num [1:8378] 15 15 15 15 15 15 15 15 15
15 ...
##  $ ambtition_important        : num [1:8378] 15 15 15 15 15 15 15 15 15
15 ...
##  $ shared_interests_important   : num [1:8378] 15 15 15 15 15 15 15 15 15
15 ...
##  $ attractive                 : num [1:8378] 6 6 6 6 6 6 6 6 6 6 ...
##  $ sincere                    : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ intelligence               : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ funny                      : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ ambition                   : num [1:8378] 7 7 7 7 7 7 7 7 7 7 ...
##  $ attractive_partner         : num [1:8378] 6 7 5 7 5 4 7 4 7 5 ...
##  $ sincere_partner            : num [1:8378] 9 8 8 6 6 9 6 9 6 6 ...
##  $ intelligence_partner       : num [1:8378] 7 7 9 8 7 7 7 7 8 6 ...
##  $ funny_partner              : num [1:8378] 7 8 8 7 7 4 4 6 9 8 ...
##  $ ambition_partner           : num [1:8378] 6 5 5 6 6 6 6 5 8 10 ...
##  $ shared_interests_partner   : num [1:8378] 5 6 7 8 6 4 7 6 8 8 ...
##  $ sports                     : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ tvsports                   : num [1:8378] 2 2 2 2 2 2 2 2 2 2 ...
##  $ exercise                   : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ dining                     : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ museums                    : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ art                        : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ hiking                     : num [1:8378] 5 5 5 5 5 5 5 5 5 5 ...
##  $ gaming                     : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ clubbing                   : num [1:8378] 5 5 5 5 5 5 5 5 5 5 ...
##  $ reading                    : num [1:8378] 6 6 6 6 6 6 6 6 6 6 ...
##  $ tv                         : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ theater                    : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ movies                     : num [1:8378] 10 10 10 10 10 10 10 10 10
10 ...
##  $ concerts                   : num [1:8378] 10 10 10 10 10 10 10 10 10
10 ...
##  $ music                      : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ shopping                   : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ yoga                       : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ interests_correlate        : num [1:8378] 0.14 0.54 0.16 0.61 0.21
0.25 0.34 0.5 0.28 -0.36 ...
##  $ expected_happy_with_sd_people: num [1:8378] 3 3 3 3 3 3 3 3 3 3 ...
##  $ expected_num_matches       : num [1:8378] 4 4 4 4 4 4 4 4 4 4 ...
##  $ like                       : num [1:8378] 7 7 7 7 6 6 6 6 7 6 ...
##  $ guess_prob_liked           : num [1:8378] 6 5 NA 6 6 5 5 7 7 6 ...
##  $ match                      : num [1:8378] 0 0 1 1 1 0 0 0 1 0 ...
##  $ field_category             : Factor w/ 8 levels "Arts","Business",..:
8 8 8 8 8 8 8 8 8 8 ...
##   ..- attr(*, "names")= chr [1:8378] "Law" "Law" "Law" "Law" ...
```

Check NA's

```r
sum(is.na(speeddating_data_new))
```

```
## [1] 11356
```

```r
dim(speeddating_data_new)
```

```
## [1] 8378   62
```

Convert character to factor

```r
speeddating_data_new[] <- lapply(speeddating_data_new, function(x) {
    if (is.character(x)) as.factor(x) else x
})
str(speeddating_data_new)
```

```
## tibble [8,378 × 62] (S3: tbl_df/tbl/data.frame)
##  $ id                            : num [1:8378] 1 2 3 4 5 6 7 8 9 10 ...
##  $ gender                        : Factor w/ 2 levels "female","male": 1 1
1 1 1 1 1 1 1 1 ...
##  $ age                           : num [1:8378] 21 21 21 21 21 21 21 21 21
21 ...
##  $ age_o                         : num [1:8378] 27 22 22 23 24 25 30 27 28
24 ...
##  $ race                          : Factor w/ 5 levels "Asian/Pacific
Islander/Asian-American",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ race_o                        : Factor w/ 5 levels "Asian/Pacific
Islander/Asian-American",..: 3 3 1 3 4 3 3 3 3 3 ...
##  $ samerace                      : num [1:8378] 0 0 1 0 0 0 0 0 0 0 ...
##  $ importance_same_race          : num [1:8378] 2 2 2 2 2 2 2 2 2 2 ...
##  $ importance_same_religion      : num [1:8378] 4 4 4 4 4 4 4 4 4 4 ...
##  $ pref_o_attractive             : num [1:8378] 35 60 19 30 30 ...
##  $ pref_o_sincere                : num [1:8378] 20 0 18 5 10 ...
##  $ pref_o_intelligence           : num [1:8378] 20 0 19 15 20 ...
##  $ pref_o_funny                  : num [1:8378] 20 40 18 40 10 ...
##  $ pref_o_ambitious              : num [1:8378] 0 0 14 5 10 ...
##  $ pref_o_shared_interests       : num [1:8378] 5 0 12 5 20 ...
##  $ attractive_o                  : num [1:8378] 6 7 10 7 8 7 3 6 7 6 ...
##  $ sinsere_o                     : num [1:8378] 8 8 10 8 7 7 6 7 7 6 ...
##  $ intelligence_o                : num [1:8378] 8 10 10 9 9 8 7 5 8 6 ...
##  $ funny_o                       : num [1:8378] 8 7 10 8 6 8 5 6 8 6 ...
##  $ ambitous_o                    : num [1:8378] 8 7 10 9 9 7 8 8 8 6 ...
##  $ shared_interests_o            : num [1:8378] 6 5 10 8 7 7 7 6 9 6 ...
##  $ attractive_important          : num [1:8378] 15 15 15 15 15 15 15 15 15
15 ...
##  $ sincere_important             : num [1:8378] 20 20 20 20 20 20 20 20 20
20 ...
##  $ intellicence_important        : num [1:8378] 20 20 20 20 20 20 20 20 20
20 ...
##  $ funny_important               : num [1:8378] 15 15 15 15 15 15 15 15 15
15 ...
##  $ ambtition_important           : num [1:8378] 15 15 15 15 15 15 15 15 15
15 ...
##  $ shared_interests_important    : num [1:8378] 15 15 15 15 15 15 15 15 15
```

```
15 ...
##  $ attractive                    : num [1:8378] 6 6 6 6 6 6 6 6 6 6 ...
##  $ sincere                       : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ intelligence                  : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ funny                         : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ ambition                      : num [1:8378] 7 7 7 7 7 7 7 7 7 7 ...
##  $ attractive_partner            : num [1:8378] 6 7 5 7 5 4 7 4 7 5 ...
##  $ sincere_partner               : num [1:8378] 9 8 8 6 6 9 6 9 6 6 ...
##  $ intelligence_partner          : num [1:8378] 7 7 9 8 7 7 7 7 8 6 ...
##  $ funny_partner                 : num [1:8378] 7 8 8 7 7 4 4 6 9 8 ...
##  $ ambition_partner              : num [1:8378] 6 5 5 6 6 6 6 5 8 10 ...
##  $ shared_interests_partner      : num [1:8378] 5 6 7 8 6 4 7 6 8 8 ...
##  $ sports                        : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ tvsports                      : num [1:8378] 2 2 2 2 2 2 2 2 2 2 ...
##  $ exercise                      : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ dining                        : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ museums                       : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ art                           : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ hiking                        : num [1:8378] 5 5 5 5 5 5 5 5 5 5 ...
##  $ gaming                        : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ clubbing                      : num [1:8378] 5 5 5 5 5 5 5 5 5 5 ...
##  $ reading                       : num [1:8378] 6 6 6 6 6 6 6 6 6 6 ...
##  $ tv                            : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ theater                       : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ movies                        : num [1:8378] 10 10 10 10 10 10 10 10 10
10 ...
##  $ concerts                      : num [1:8378] 10 10 10 10 10 10 10 10 10
10 ...
##  $ music                         : num [1:8378] 9 9 9 9 9 9 9 9 9 9 ...
##  $ shopping                      : num [1:8378] 8 8 8 8 8 8 8 8 8 8 ...
##  $ yoga                          : num [1:8378] 1 1 1 1 1 1 1 1 1 1 ...
##  $ interests_correlate           : num [1:8378] 0.14 0.54 0.16 0.61 0.21
0.25 0.34 0.5 0.28 -0.36 ...
##  $ expected_happy_with_sd_people: num [1:8378] 3 3 3 3 3 3 3 3 3 3 ...
##  $ expected_num_matches          : num [1:8378] 4 4 4 4 4 4 4 4 4 4 ...
##  $ like                          : num [1:8378] 7 7 7 7 6 6 6 6 7 6 ...
##  $ guess_prob_liked              : num [1:8378] 6 5 NA 6 6 5 5 7 7 6 ...
##  $ match                         : num [1:8378] 0 0 1 1 1 0 0 0 1 0 ...
##  $ field_category                : Factor w/ 8 levels "Arts","Business",..:
8 8 8 8 8 8 8 8 8 8 ...
##   ..- attr(*, "names")= chr [1:8378] "Law" "Law" "Law" "Law" ...
```

Data Split

```r
s <- createDataPartition(y = speeddating_data_new$match,p = 0.7,list = FALSE)
train <- speeddating_data_new[s,] # 70% training
test <- speeddating_data_new[-s,] # 30% testing
```

Check subsets

```r
dim(speeddating_data_new)

## [1] 8378   62

dim(train)

## [1] 5865   62

dim(test)

## [1] 2513   62

summary(train$match)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.1598  0.0000  1.0000

summary(test$match)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.0000  0.0000  0.1763  0.0000  1.0000

# Proportions in the original data
prop_original <- prop.table(table(speeddating_data_new$match))
print("Original Data Proportions")

## [1] "Original Data Proportions"

print(prop_original)

## 
##         0         1 
## 0.8352829 0.1647171

# Proportions in the training data
prop_train <- prop.table(table(train$match))
print("Training Data Proportions")

## [1] "Training Data Proportions"

print(prop_train)

## 
##         0         1 
## 0.8402387 0.1597613

# Proportions in the testing data
prop_test <- prop.table(table(test$match))
print("Testing Data Proportions")

## [1] "Testing Data Proportions"

print(prop_test)
```

```
##
##         0         1
## 0.8237167 0.1762833
```

Check NA's

```r
colSums(is.na(train))[colSums(is.na(train))>100]
```

```
##            attractive_o                sinsere_o          intelligence_o
##                     131                      186                     204
##                 funny_o               ambitous_o      shared_interests_o
##                     234                      509                     746
##      attractive_partner          sincere_partner    intelligence_partner
##                     132                      182                     199
##           funny_partner         ambition_partner shared_interests_partner
##                     237                      502                     737
##     interests_correlate     expected_num_matches                    like
##                     118                      807                     161
##         guess_prob_liked
##                     210
```

```r
colSums(is.na(test))[colSums(is.na(test))>100]
```

```
##               sinsere_o          intelligence_o                 funny_o
##                     101                      102                     126
##              ambitous_o      shared_interests_o           funny_partner
##                     213                      330                     113
##        ambition_partner shared_interests_partner    expected_num_matches
##                     210                      330                     366
```

Median Imputation

```r
# Median Imputation
train[] <- lapply(train, function(x) {
    # Calculate the number of missing values in the column
    num_missing <- sum(is.na(x))

    # Check if the column has more than 100 missing values
    if(num_missing > 100) {
        if(is.numeric(x)) {
            # Replace NA with median for numeric columns
            x[is.na(x)] <- median(x, na.rm = TRUE)
        } else if(is.factor(x)) {
            # Calculate mode for factor columns
            mode <- names(sort(table(x), decreasing = TRUE))[1]
            x[is.na(x)] <- mode
        }
    }
    return(x)
})

test[] <- lapply(test, function(x) {
```

```r
    # Calculate the number of missing values in the column
    num_missing <- sum(is.na(x))

    # Check if the column has more than 100 missing values
    if(num_missing > 100) {
        if(is.numeric(x)) {
            # Replace NA with median for numeric columns
            x[is.na(x)] <- median(x, na.rm = TRUE)
        } else if(is.factor(x)) {
            # Calculate mode for factor columns
            mode <- names(sort(table(x), decreasing = TRUE))[1]
            x[is.na(x)] <- mode
        }
    }
    return(x)
})
```

Check NA's

```r
colSums(is.na(train))[colSums(is.na(train))!=0]

##                          age                         age_o
##                           70                            80
##                         race                        race_o
##                           46                            56
##          importance_same_race      importance_same_religion
##                           60                            60
##             pref_o_attractive               pref_o_sincere
##                           66                            66
##           pref_o_intelligence                 pref_o_funny
##                           66                            71
##              pref_o_ambitious      pref_o_shared_interests
##                           79                            95
##          attractive_important            sincere_important
##                           60                            60
##        intellicence_important              funny_important
##                           60                            66
##           ambtition_important   shared_interests_important
##                           74                            90
##                    attractive                      sincere
##                           77                            77
##                  intelligence                        funny
##                           77                            77
##                      ambition                       sports
##                           77                            60
##                      tvsports                     exercise
##                           60                            60
##                        dining                      museums
##                           60                            60
##                           art                       hiking
##                           60                            60
```

```
##                          gaming                         clubbing
##                              60                               60
##                         reading                               tv
##                              60                               60
##                         theater                           movies
##                              60                               60
##                        concerts                            music
##                              60                               60
##                        shopping                             yoga
##                              60                               60
## expected_happy_with_sd_people
##                              76
```

```
colSums(is.na(test))[colSums(is.na(test))!=0]
```

```
##                             age                            age_o
##                              25                               24
##                            race                           race_o
##                              17                               17
##            importance_same_race       importance_same_religion
##                              19                               19
##               pref_o_attractive                  pref_o_sincere
##                              23                               23
##             pref_o_intelligence                    pref_o_funny
##                              23                               27
##                pref_o_ambitious        pref_o_shared_interests
##                              28                               34
##                    attractive_o             attractive_important
##                              81                               19
##               sincere_important           intellicence_important
##                              19                               19
##                 funny_important              ambtition_important
##                              23                               25
##     shared_interests_important                       attractive
##                              31                               28
##                         sincere                     intelligence
##                              28                               28
##                           funny                         ambition
##                              28                               28
##              attractive_partner                  sincere_partner
##                              70                               95
##            intelligence_partner                           sports
##                              97                               19
##                        tvsports                         exercise
##                              19                               19
##                          dining                          museums
##                              19                               19
##                             art                           hiking
##                              19                               19
##                          gaming                         clubbing
```

```
##                           19                          19
##                      reading                          tv
##                           19                          19
##                      theater                      movies
##                           19                          19
##                     concerts                       music
##                           19                          19
##                     shopping                        yoga
##                           19                          19
##          interests_correlate expected_happy_with_sd_people
##                           40                          25
##                         like             guess_prob_liked
##                           79                          99
```

```
sum(!complete.cases(train))
```

```
## [1] 232
```

```
dim(train)
```

```
## [1] 5865   62
```

```
sum(!complete.cases(test))
```

```
## [1] 246
```

```
dim(test)
```

```
## [1] 2513   62
```

Omit rows

```
train=na.omit(train)
dim(train)
```

```
## [1] 5633   62
```

```
test=na.omit(test)
dim(test)
```

```
## [1] 2267   62
```

Model Selection

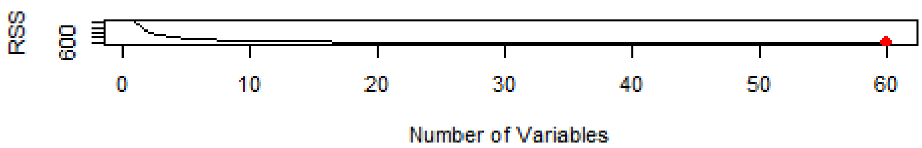
Backwards

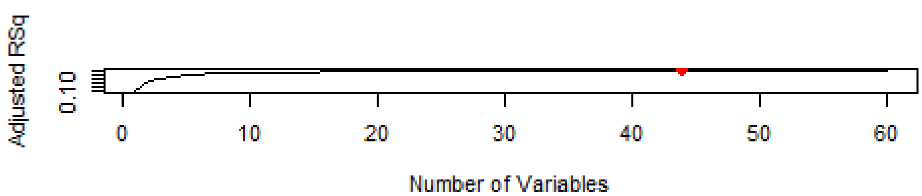
```
regfit.bwd <- regsubsets(match ~ ., data = train, nvmax = 60, method =
"backward")
reg.summary <- summary(regfit.bwd)
```

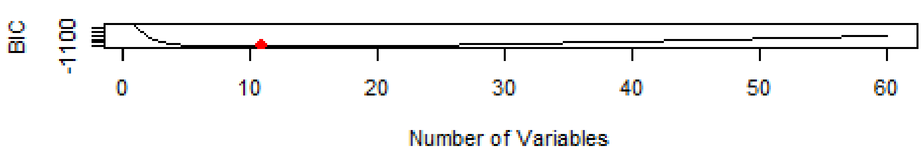
Graphs

```
par(mfrow = c(3, 1))
plot(reg.summary$rss, xlab = "Number of Variables",
```

```
     ylab = "RSS", type = "l")
n = which.min(reg.summary$rss)
points(n, reg.summary$rss[n], col = "red", cex = 2,
       pch = 20)

plot(reg.summary$adjr2, xlab = "Number of Variables",
     ylab = "Adjusted RSq", type = "l")
n = which.max(reg.summary$adjr2)
points(n, reg.summary$adjr2[n], col = "red", cex = 2,
       pch = 20)

plot(reg.summary$bic, xlab = "Number of Variables",
     ylab = "BIC", type = "l")
n = which.min(reg.summary$bic)
points(n, reg.summary$bic[n], col = "red", cex = 2,
       pch = 20)
```







Best variables

```
# Determine which models to check - here we use the model with the highest
adjusted R^2
best.model.size <- which.min(reg.summary$bic)
# Get coefficients of the best model
coefficients <- coef(regfit.bwd, id = best.model.size)
# Print the coefficients to see which are significant
print(coefficients)
```

```
##          (Intercept)                age_o  pref_o_intelligence
##         -0.580180808         -0.003851993          0.002235853
##         pref_o_funny         attractive_o              funny_o
##          0.002469989          0.028425277          0.017644961
##    shared_interests_o         intelligence   attractive_partner
##          0.017472929         -0.010095009          0.024395302
## expected_num_matches                 like     guess_prob_liked
##          0.010486012          0.027856538          0.016611360
```

```r
best.vars <- names(which(reg.summary$which[best.model.size, ]))
formula <- as.formula(paste("match ~", paste(best.vars, collapse = " + ")))
print(formula)
```

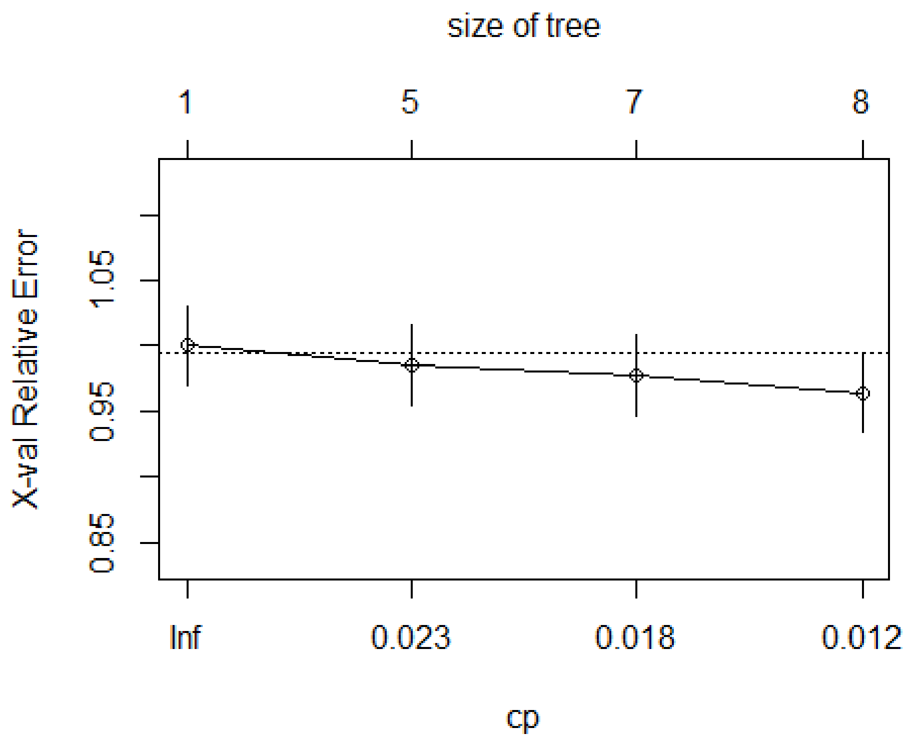
```
## match ~ (Intercept) + age_o + pref_o_intelligence + pref_o_funny +
##     attractive_o + funny_o + shared_interests_o + intelligence +
##     attractive_partner + expected_num_matches + like + guess_prob_liked
```

Decision Trees

```r
set.seed(12345)
tree.speeddating=rpart(match ~ ., data=train, method="class")
printcp(tree.speeddating)
```

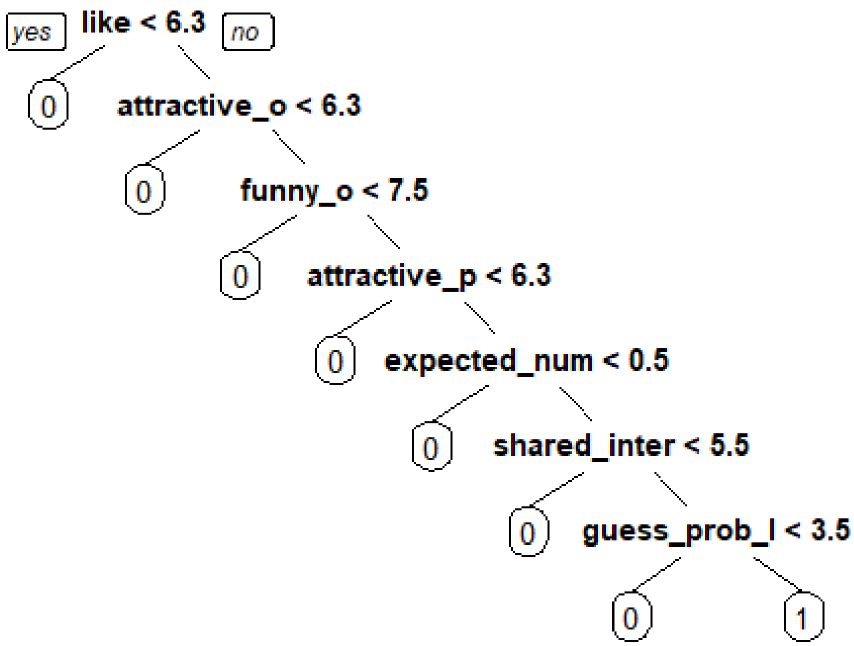
```
##
## Classification tree:
## rpart(formula = match ~ ., data = train, method = "class")
##
## Variables actually used in tree construction:
## [1] attractive_o         attractive_partner   expected_num_matches
## [4] funny_o              guess_prob_liked     like
## [7] shared_interests_o
##
## Root node error: 896/5633 = 0.15906
##
## n= 5633
##
##          CP nsplit rel error  xerror     xstd
## 1 0.023158      0   1.00000 1.00000 0.030636
## 2 0.022321      4   0.90737 0.98549 0.030454
## 3 0.014509      6   0.86272 0.97768 0.030356
## 4 0.010000      7   0.84821 0.96429 0.030185
```

```r
plotcp(tree.speeddating)
```

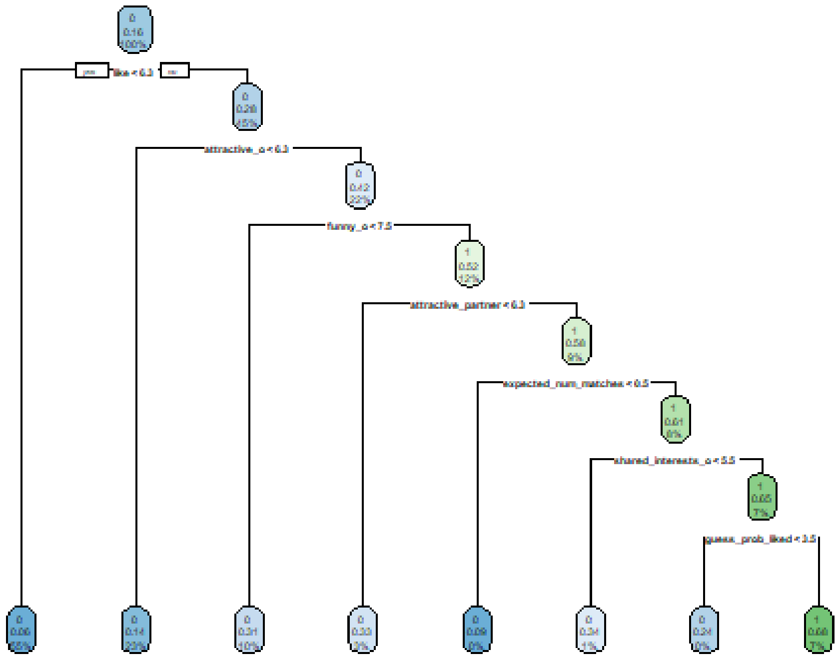



```
names(tree.speeddating)

##  [1] "frame"              "where"              "call"
##  [4] "terms"              "cptable"            "method"
##  [7] "parms"              "control"            "functions"
## [10] "numresp"            "splits"             "variable.importance"
## [13] "y"                  "ordered"

tree.speeddating$cptable

##           CP nsplit rel error    xerror       xstd
## 1 0.02315848      0 1.0000000 1.0000000 0.03063570
## 2 0.02232143      4 0.9073661 0.9854911 0.03045435
## 3 0.01450893      6 0.8627232 0.9776786 0.03035573
## 4 0.01000000      7 0.8482143 0.9642857 0.03018510

min_cp_index=which.min(tree.speeddating$cptable[,"CP"])
min_cp_index

## 4
## 4

cp=tree.speeddating$cptable[min_cp_index,"CP"]
cp

## [1] 0.01
```

```
prune.speeddating = prune(tree.speeddating,cp=cp)
prp(prune.speeddating)
```




```
rpart.plot(prune.speeddating)
```

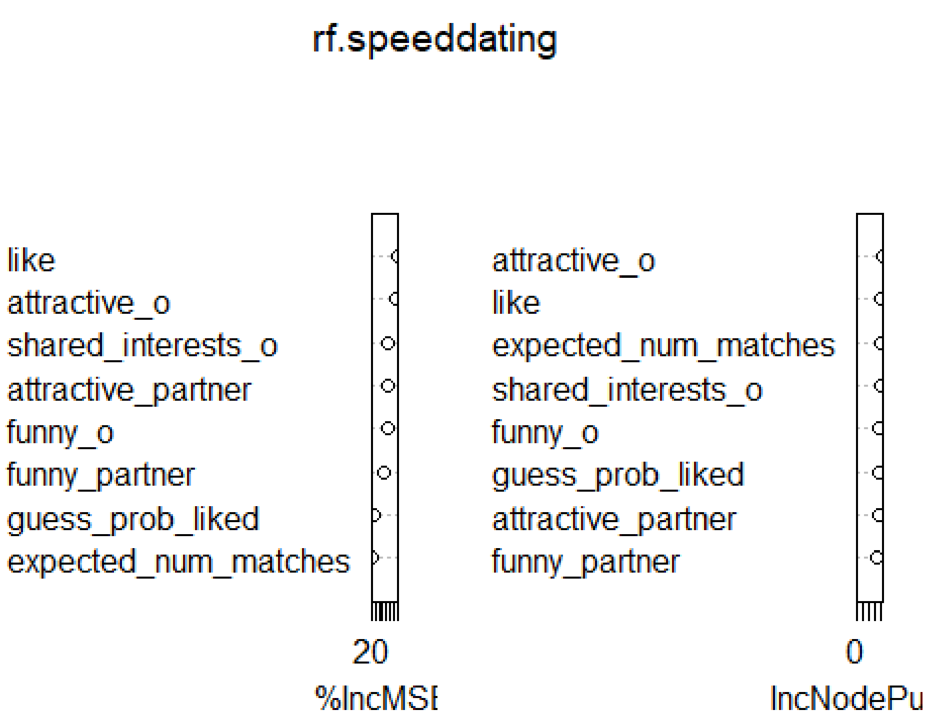



```
tree.pred=predict(prune.speeddating,test,type="class")
table(predicted=tree.pred,actual=test$match)

##          actual
## predicted    0    1
##         0 1808  321
##         1   42   96

mean(tree.pred==test$match)

## [1] 0.8398765

mean(tree.pred!=test$match)

## [1] 0.1601235
```

Random Forest

```
##Random forest with best variables
rf.speeddating=randomForest(match~attractive_o+funny_o+shared_interests_o+att
ractive_partner+funny_partner+expected_num_matches+like+guess_prob_liked,
data=train ,mtry=3,importance=TRUE)

## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values.  Are you sure you want to do regression?
```

```r
yhat.rf = predict(rf.speeddating,newdata=test, type="class")
#table(yhat.rf,test$match)
mean(yhat.rf != test$match, na.rm = TRUE)

## [1] 1

mean(yhat.rf == test$match, na.rm = TRUE) #Accuracy with train 0.8563025

## [1] 0

#yhat.rf
importance(rf.speeddating)

##                      %IncMSE  IncNodePurity
## attractive_o         52.29385      87.65910
## funny_o              42.20061      76.43797
## shared_interests_o   43.13634      78.74716
## attractive_partner   42.23233      73.07161
## funny_partner        35.84503      65.25120
## expected_num_matches 20.92230      81.40768
## like                 55.53831      85.01055
## guess_prob_liked     23.60100      73.69470

varImpPlot(rf.speeddating)
```



Logistic Regression

```
logistic_model <- glm(match ~ .-id, data = train, family = binomial())
summary(logistic_model)

##
## Call:
## glm(formula = match ~ . - id, family = binomial(), data = train)
##
## Coefficients:
##                                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -1.045e+01  3.133e+00  -3.336 0.000849
***
## gendermale                         -2.041e-01  1.345e-01  -1.517 0.129333
## age                                -2.283e-02  1.442e-02  -1.583 0.113444
## age_o                              -2.961e-02  1.306e-02  -2.267 0.023373 *
## raceBlack/African American          1.952e-01  2.201e-01   0.887 0.375174
## raceEuropean/Caucasian-American    -3.714e-01  1.377e-01  -2.698 0.006984
**
## raceLatino/Hispanic American       -1.150e-01  1.945e-01  -0.591 0.554219
## raceOther                           8.712e-02  2.080e-01   0.419 0.675405
## race_oBlack/African American       -4.094e-02  2.055e-01  -0.199 0.842115
## race_oEuropean/Caucasian-American -1.371e-01  1.228e-01  -1.117 0.264032
## race_oLatino/Hispanic American      1.235e-01  1.731e-01   0.713 0.475573
## race_oOther                         2.288e-02  2.029e-01   0.113 0.910202
## samerace                            8.595e-02  1.142e-01   0.753 0.451651
## importance_same_race               -1.017e-02  1.982e-02  -0.513 0.607849
## importance_same_religion            7.746e-03  1.898e-02   0.408 0.683112
## pref_o_attractive                  -2.965e-03  2.192e-02  -0.135 0.892424
## pref_o_sincere                      7.366e-04  2.236e-02   0.033 0.973720
## pref_o_intelligence                 2.042e-02  2.268e-02   0.901 0.367828
## pref_o_funny                        1.591e-02  2.261e-02   0.704 0.481573
## pref_o_ambitious                    1.310e-03  2.198e-02   0.060 0.952484
## pref_o_shared_interests            -3.909e-03  2.258e-02  -0.173 0.862574
## attractive_o                        3.296e-01  3.234e-02  10.195  < 2e-16
***
## sinsere_o                          -5.038e-02  3.930e-02  -1.282 0.199841
## intelligence_o                      1.140e-01  4.726e-02   2.412 0.015877 *
## funny_o                             1.804e-01  3.585e-02   5.031 4.87e-07
***
## ambitous_o                         -1.097e-01  3.678e-02  -2.981 0.002870
**
## shared_interests_o                  1.955e-01  2.987e-02   6.543 6.01e-11
***
## attractive_important                1.740e-02  2.088e-02   0.833 0.404677
## sincere_important                   1.562e-02  2.170e-02   0.720 0.471668
## intellicence_important              3.719e-02  2.180e-02   1.706 0.088048 .
## funny_important                     3.364e-02  2.182e-02   1.542 0.123155
## ambtition_important                 7.293e-03  2.063e-02   0.353 0.723719
## shared_interests_important          1.239e-02  2.185e-02   0.567 0.570730
## attractive                         -5.657e-02  4.611e-02  -1.227 0.219834
```

```
## sincere                          -1.729e-02  3.765e-02  -0.459 0.646194
## intelligence                     -5.636e-02  3.905e-02  -1.444 0.148879
## funny                            -3.287e-02  5.144e-02  -0.639 0.522916
## ambition                          1.465e-02  3.215e-02   0.456 0.648636
## attractive_partner                2.210e-01  3.372e-02   6.554 5.61e-11
***
## sincere_partner                  -9.776e-02  3.998e-02  -2.445 0.014468 *
## intelligence_partner              7.916e-02  4.804e-02   1.648 0.099401 .
## funny_partner                     1.040e-01  3.811e-02   2.730 0.006335
**
## ambition_partner                 -1.087e-01  3.691e-02  -2.945 0.003230
**
## shared_interests_partner          3.454e-02  3.219e-02   1.073 0.283270
## sports                           -2.222e-02  2.348e-02  -0.947 0.343887
## tvsports                         -3.387e-02  2.105e-02  -1.609 0.107588
## exercise                         -8.159e-03  2.140e-02  -0.381 0.702989
## dining                            4.026e-03  3.203e-02   0.126 0.899957
## museums                          -4.466e-02  4.828e-02  -0.925 0.354960
## art                               9.799e-02  4.226e-02   2.319 0.020408 *
## hiking                           -2.115e-03  1.975e-02  -0.107 0.914714
## gaming                            7.794e-03  1.980e-02   0.394 0.693787
## clubbing                          2.644e-02  1.966e-02   1.345 0.178733
## reading                           1.147e-02  2.604e-02   0.441 0.659567
## tv                                5.666e-02  2.460e-02   2.304 0.021240 *
## theater                          -1.977e-02  2.793e-02  -0.708 0.479099
## movies                           -6.127e-02  3.478e-02  -1.762 0.078117 .
## concerts                          5.238e-02  3.162e-02   1.657 0.097589 .
## music                            -3.688e-02  3.517e-02  -1.049 0.294390
## shopping                         -8.343e-02  2.356e-02  -3.542 0.000397
***
## yoga                              6.699e-03  1.804e-02   0.371 0.710340
## interests_correlate               3.108e-01  1.607e-01   1.935 0.053036 .
## expected_happy_with_sd_people    -1.297e-02  2.854e-02  -0.454 0.649505
## expected_num_matches              9.141e-02  2.036e-02   4.490 7.12e-06
***
## like                              3.375e-01  4.696e-02   7.187 6.62e-13
***
## guess_prob_liked                  1.790e-01  2.718e-02   6.585 4.56e-11
***
## field_categoryBusiness           -6.670e-02  2.206e-01  -0.302 0.762363
## field_categoryEngineering        -4.488e-01  2.754e-01  -1.630 0.103160
## field_categoryHealth              1.308e-02  3.017e-01   0.043 0.965424
## field_categoryHumanities         -3.221e-01  2.709e-01  -1.189 0.234492
## field_categoryOther              -4.574e-01  2.231e-01  -2.050 0.040326 *
## field_categoryScience           -1.419e-01  2.241e-01  -0.633 0.526648
## field_categorySocial Sciences    -3.710e-01  2.005e-01  -1.850 0.064309 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 4935.8  on 5632  degrees of freedom
## Residual deviance: 3545.9  on 5560  degrees of freedom
## AIC: 3691.9
##
## Number of Fisher Scoring iterations: 6
```

Predict

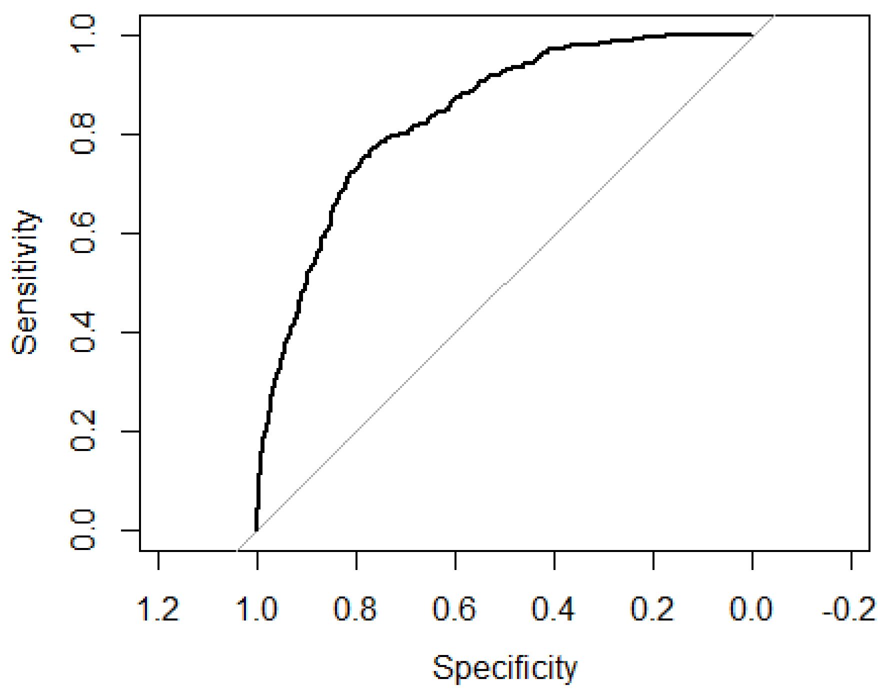
```
# Predict on test data
predicted_probabilities <- predict(logistic_model, newdata = test, type =
"response")
predicted_classes <- ifelse(predicted_probabilities > 0.5, 1, 0)
#predicted_classes
```

Confusion Matrix and Accuracy

```
confusion_matrix <- table(Predicted = predicted_classes, Actual = test$match)
print(confusion_matrix)

##          Actual
## Predicted    0    1
##         0 1778  288
##         1   72  129

accuracy <- sum(predicted_classes == test$match) / nrow(test)
print(paste("Accuracy:", accuracy))

## [1] "Accuracy: 0.841199823555359"
```

ROC Curve and AUC

```
library(pROC)

## Warning: package 'pROC' was built under R version 4.3.3

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

##
## Attaching package: 'pROC'

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

roc_result <- roc(response = test$match, predictor = predicted_probabilities)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(roc_result)
```




```
auc(roc_result)

## Area under the curve: 0.8361
```

Logistic Regression With Best Variables

```
logistic_model <- glm(match ~ age_o + pref_o_intelligence + pref_o_funny +
    attractive_o + funny_o + shared_interests_o + intelligence +
    attractive_partner + expected_num_matches + like + guess_prob_liked, data
= train, family = binomial())
summary(logistic_model)

##
## Call:
## glm(formula = match ~ age_o + pref_o_intelligence + pref_o_funny +
##     attractive_o + funny_o + shared_interests_o + intelligence +
##     attractive_partner + expected_num_matches + like + guess_prob_liked,
##     family = binomial(), data = train)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -10.087924   0.546463 -18.460  < 2e-16 ***
## age_o                 -0.028650   0.012197  -2.349 0.018832 *
## pref_o_intelligence    0.023099   0.006382   3.619 0.000295 ***
## pref_o_funny           0.018847   0.007023   2.684 0.007279 **
## attractive_o           0.286976   0.029068   9.872  < 2e-16 ***
## funny_o                0.188965   0.032414   5.830 5.55e-09 ***
```

```
## shared_interests_o     0.176154   0.027577   6.388 1.68e-10 ***
## intelligence          -0.109703   0.028879  -3.799 0.000145 ***
## attractive_partner     0.212558   0.031122   6.830 8.50e-12 ***
## expected_num_matches   0.086838   0.017993   4.826 1.39e-06 ***
## like                   0.347176   0.038199   9.089  < 2e-16 ***
## guess_prob_liked       0.155357   0.024495   6.342 2.26e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4935.8  on 5632  degrees of freedom
## Residual deviance: 3674.4  on 5621  degrees of freedom
## AIC: 3698.4
##
## Number of Fisher Scoring iterations: 6
```

Predict

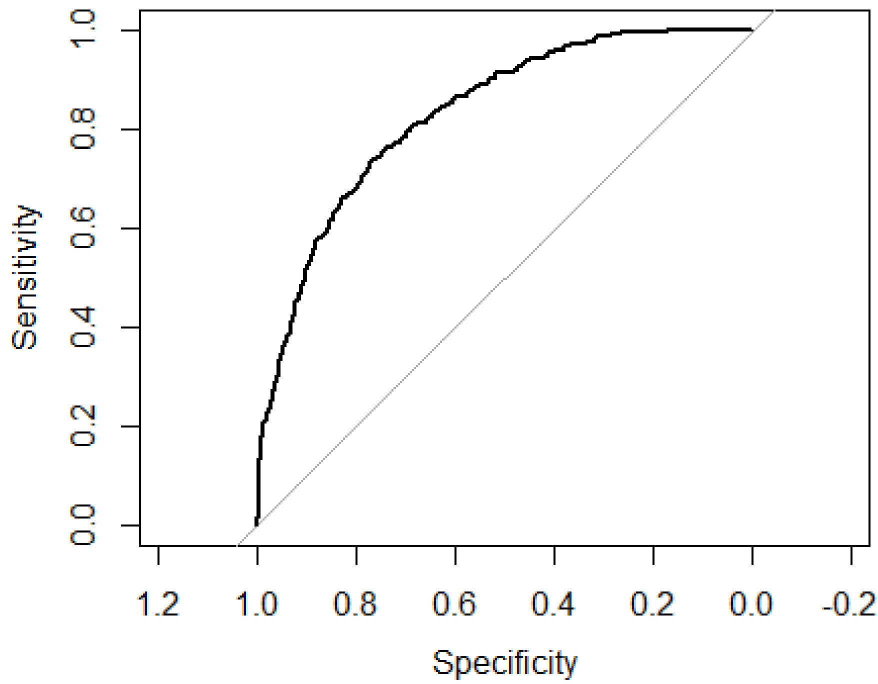
```
# Predict on test data
predicted_probabilities <- predict(logistic_model, newdata = test, type =
"response")
predicted_classes <- ifelse(predicted_probabilities > 0.5, 1, 0)
#predicted_classes
```

Confusion Matrix and Accuracy

```
confusion_matrix <- table(Predicted = predicted_classes, Actual = test$match)
print(confusion_matrix)

##          Actual
## Predicted    0    1
##         0 1792  311
##         1   58  106

accuracy <- sum(predicted_classes == test$match) / nrow(test)
print(paste("Accuracy:", accuracy))

## [1] "Accuracy: 0.837229819144244"
```

ROC Curve and AUC

```
roc_result <- roc(response = test$match, predictor = predicted_probabilities)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(roc_result)
```




```
auc(roc_result)

## Area under the curve: 0.8285
```

Gradient Boosting

```
# Gradient Boosting

# Train the GBM model
boost.dating = gbm(match ~ attractive_o + funny_o + shared_interests_o +
                     attractive_partner + funny_partner +
expected_num_matches + like +
                     guess_prob_liked, data = train, distribution =
"bernoulli", n.trees = 5000, interaction.depth = 1)

# Predict probabilities on the test data
yhat.boost = predict(boost.dating, newdata = test, n.trees = 5000, type =
"response")

# Function to calculate accuracy for different thresholds
calculate_accuracy <- function(threshold) {
  yhat.boost.class = ifelse(yhat.boost > threshold, 1, 0)
  accuracy = mean(yhat.boost.class == test$match)
  return(accuracy)
}
```

```r
# Evaluate accuracy for thresholds from 0 to 1
thresholds = seq(0, 1, by = 0.01)
accuracies = sapply(thresholds, calculate_accuracy)

# Find the best threshold
best_threshold = thresholds[which.max(accuracies)]
best_threshold

## [1] 0.48

# Predict classes using the best threshold
yhat.boost.class = ifelse(yhat.boost > best_threshold, 1, 0)

# Confusion matrix and misclassification rate
conf_matrix = table(yhat.boost.class, test$match)
print(conf_matrix)

##
## yhat.boost.class    0    1
##                0 1781  284
##                1   69  133

mean(yhat.boost.class!=test$match)

## [1] 0.1557124

mean(yhat.boost.class==test$match)

## [1] 0.8442876
```

Recommendation System

```r
# Load necessary libraries
library(dplyr)
library(Matrix)

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

library(recommenderlab)

## Warning: package 'recommenderlab' was built under R version 4.3.3

## Loading required package: arules

## Warning: package 'arules' was built under R version 4.3.3
```

```
##
## Attaching package: 'arules'

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

## The following objects are masked from 'package:base':
##
##     abbreviate, write

## Loading required package: proxy

## Warning: package 'proxy' was built under R version 4.3.3

##
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':
##
##     as.matrix

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

## Registered S3 methods overwritten by 'registry':
##   method               from
##   print.registry_field proxy
##   print.registry_entry proxy

##
## Attaching package: 'recommenderlab'

## The following objects are masked from 'package:caret':
##
##     MAE, RMSE

# Group to assign each person a unique ID
data_recommendation <- train %>%
  group_by(gender, age, race, importance_same_race, importance_same_religion,
           attractive_important, sincere_important, intellicence_important,
           funny_important, ambtition_important, shared_interests_important,
           attractive, sincere, intelligence, funny, ambition) %>%
  mutate(personID = cur_group_id()) %>%
  ungroup()

# Group to assign each partner a unique ID
```

```
data_recommendation <- data_recommendation %>%
  group_by(age_o, race_o, pref_o_attractive, pref_o_sincere,
pref_o_intelligence,
           pref_o_funny, pref_o_ambitious, pref_o_shared_interests) %>%
  mutate(partnerID = cur_group_id()) %>%
  ungroup()

# Add a progress counter
total_iterations <- nrow(data_recommendation) * nrow(data_recommendation)
progress_counter <- 0

for (i in 1:nrow(data_recommendation)) {
  for (j in 1:nrow(data_recommendation)) {
    if (data_recommendation$age[i] == data_recommendation$age_o[j] &&
        data_recommendation$race[i] == data_recommendation$race_o[j] &&
        data_recommendation$attractive_important[i] ==
data_recommendation$pref_o_attractive[j] &&
        data_recommendation$sincere_important[i] ==
data_recommendation$pref_o_sincere[j] &&
        data_recommendation$intellicence_important[i] ==
data_recommendation$pref_o_intelligence[j] &&
        data_recommendation$funny_important[i] ==
data_recommendation$pref_o_funny[j] &&
        data_recommendation$ambtition_important[i] ==
data_recommendation$pref_o_ambitious[j] &&
        data_recommendation$shared_interests_important[i] ==
data_recommendation$pref_o_shared_interests[j])
      {
        data_recommendation$partnerID[j] <- data_recommendation$personID[i]
      }

    # Update and print progress counter
    progress_counter <- progress_counter + 1
    if (progress_counter %% 100000 == 0) {  # Print progress every 100000
iterations
      print(paste("Progress:", progress_counter, "out of", total_iterations))
    }
  }
}

## [1] "Progress: 1e+05 out of 31730689"
## [1] "Progress: 2e+05 out of 31730689"
## [1] "Progress: 3e+05 out of 31730689"
## [1] "Progress: 4e+05 out of 31730689"
## [1] "Progress: 5e+05 out of 31730689"
## [1] "Progress: 6e+05 out of 31730689"
## [1] "Progress: 7e+05 out of 31730689"
## [1] "Progress: 8e+05 out of 31730689"
## [1] "Progress: 9e+05 out of 31730689"
## [1] "Progress: 1e+06 out of 31730689"
```

```
## [1] "Progress: 1100000 out of 31730689"
## [1] "Progress: 1200000 out of 31730689"
## [1] "Progress: 1300000 out of 31730689"
## [1] "Progress: 1400000 out of 31730689"
## [1] "Progress: 1500000 out of 31730689"
## [1] "Progress: 1600000 out of 31730689"
## [1] "Progress: 1700000 out of 31730689"
## [1] "Progress: 1800000 out of 31730689"
## [1] "Progress: 1900000 out of 31730689"
## [1] "Progress: 2e+06 out of 31730689"
## [1] "Progress: 2100000 out of 31730689"
## [1] "Progress: 2200000 out of 31730689"
## [1] "Progress: 2300000 out of 31730689"
## [1] "Progress: 2400000 out of 31730689"
## [1] "Progress: 2500000 out of 31730689"
## [1] "Progress: 2600000 out of 31730689"
## [1] "Progress: 2700000 out of 31730689"
## [1] "Progress: 2800000 out of 31730689"
## [1] "Progress: 2900000 out of 31730689"
## [1] "Progress: 3e+06 out of 31730689"
## [1] "Progress: 3100000 out of 31730689"
## [1] "Progress: 3200000 out of 31730689"
## [1] "Progress: 3300000 out of 31730689"
## [1] "Progress: 3400000 out of 31730689"
## [1] "Progress: 3500000 out of 31730689"
## [1] "Progress: 3600000 out of 31730689"
## [1] "Progress: 3700000 out of 31730689"
## [1] "Progress: 3800000 out of 31730689"
## [1] "Progress: 3900000 out of 31730689"
## [1] "Progress: 4e+06 out of 31730689"
## [1] "Progress: 4100000 out of 31730689"
## [1] "Progress: 4200000 out of 31730689"
## [1] "Progress: 4300000 out of 31730689"
## [1] "Progress: 4400000 out of 31730689"
## [1] "Progress: 4500000 out of 31730689"
## [1] "Progress: 4600000 out of 31730689"
## [1] "Progress: 4700000 out of 31730689"
## [1] "Progress: 4800000 out of 31730689"
## [1] "Progress: 4900000 out of 31730689"
## [1] "Progress: 5e+06 out of 31730689"
## [1] "Progress: 5100000 out of 31730689"
## [1] "Progress: 5200000 out of 31730689"
## [1] "Progress: 5300000 out of 31730689"
## [1] "Progress: 5400000 out of 31730689"
## [1] "Progress: 5500000 out of 31730689"
## [1] "Progress: 5600000 out of 31730689"
## [1] "Progress: 5700000 out of 31730689"
## [1] "Progress: 5800000 out of 31730689"
## [1] "Progress: 5900000 out of 31730689"
## [1] "Progress: 6e+06 out of 31730689"
```

```
## [1] "Progress: 6100000 out of 31730689"
## [1] "Progress: 6200000 out of 31730689"
## [1] "Progress: 6300000 out of 31730689"
## [1] "Progress: 6400000 out of 31730689"
## [1] "Progress: 6500000 out of 31730689"
## [1] "Progress: 6600000 out of 31730689"
## [1] "Progress: 6700000 out of 31730689"
## [1] "Progress: 6800000 out of 31730689"
## [1] "Progress: 6900000 out of 31730689"
## [1] "Progress: 7e+06 out of 31730689"
## [1] "Progress: 7100000 out of 31730689"
## [1] "Progress: 7200000 out of 31730689"
## [1] "Progress: 7300000 out of 31730689"
## [1] "Progress: 7400000 out of 31730689"
## [1] "Progress: 7500000 out of 31730689"
## [1] "Progress: 7600000 out of 31730689"
## [1] "Progress: 7700000 out of 31730689"
## [1] "Progress: 7800000 out of 31730689"
## [1] "Progress: 7900000 out of 31730689"
## [1] "Progress: 8e+06 out of 31730689"
## [1] "Progress: 8100000 out of 31730689"
## [1] "Progress: 8200000 out of 31730689"
## [1] "Progress: 8300000 out of 31730689"
## [1] "Progress: 8400000 out of 31730689"
## [1] "Progress: 8500000 out of 31730689"
## [1] "Progress: 8600000 out of 31730689"
## [1] "Progress: 8700000 out of 31730689"
## [1] "Progress: 8800000 out of 31730689"
## [1] "Progress: 8900000 out of 31730689"
## [1] "Progress: 9e+06 out of 31730689"
## [1] "Progress: 9100000 out of 31730689"
## [1] "Progress: 9200000 out of 31730689"
## [1] "Progress: 9300000 out of 31730689"
## [1] "Progress: 9400000 out of 31730689"
## [1] "Progress: 9500000 out of 31730689"
## [1] "Progress: 9600000 out of 31730689"
## [1] "Progress: 9700000 out of 31730689"
## [1] "Progress: 9800000 out of 31730689"
## [1] "Progress: 9900000 out of 31730689"
## [1] "Progress: 1e+07 out of 31730689"
## [1] "Progress: 10100000 out of 31730689"
## [1] "Progress: 10200000 out of 31730689"
## [1] "Progress: 10300000 out of 31730689"
## [1] "Progress: 10400000 out of 31730689"
## [1] "Progress: 10500000 out of 31730689"
## [1] "Progress: 10600000 out of 31730689"
## [1] "Progress: 10700000 out of 31730689"
## [1] "Progress: 10800000 out of 31730689"
## [1] "Progress: 10900000 out of 31730689"
## [1] "Progress: 1.1e+07 out of 31730689"
```

```
## [1] "Progress: 11100000 out of 31730689"
## [1] "Progress: 11200000 out of 31730689"
## [1] "Progress: 11300000 out of 31730689"
## [1] "Progress: 11400000 out of 31730689"
## [1] "Progress: 11500000 out of 31730689"
## [1] "Progress: 11600000 out of 31730689"
## [1] "Progress: 11700000 out of 31730689"
## [1] "Progress: 11800000 out of 31730689"
## [1] "Progress: 11900000 out of 31730689"
## [1] "Progress: 1.2e+07 out of 31730689"
## [1] "Progress: 12100000 out of 31730689"
## [1] "Progress: 12200000 out of 31730689"
## [1] "Progress: 12300000 out of 31730689"
## [1] "Progress: 12400000 out of 31730689"
## [1] "Progress: 12500000 out of 31730689"
## [1] "Progress: 12600000 out of 31730689"
## [1] "Progress: 12700000 out of 31730689"
## [1] "Progress: 12800000 out of 31730689"
## [1] "Progress: 12900000 out of 31730689"
## [1] "Progress: 1.3e+07 out of 31730689"
## [1] "Progress: 13100000 out of 31730689"
## [1] "Progress: 13200000 out of 31730689"
## [1] "Progress: 13300000 out of 31730689"
## [1] "Progress: 13400000 out of 31730689"
## [1] "Progress: 13500000 out of 31730689"
## [1] "Progress: 13600000 out of 31730689"
## [1] "Progress: 13700000 out of 31730689"
## [1] "Progress: 13800000 out of 31730689"
## [1] "Progress: 13900000 out of 31730689"
## [1] "Progress: 1.4e+07 out of 31730689"
## [1] "Progress: 14100000 out of 31730689"
## [1] "Progress: 14200000 out of 31730689"
## [1] "Progress: 14300000 out of 31730689"
## [1] "Progress: 14400000 out of 31730689"
## [1] "Progress: 14500000 out of 31730689"
## [1] "Progress: 14600000 out of 31730689"
## [1] "Progress: 14700000 out of 31730689"
## [1] "Progress: 14800000 out of 31730689"
## [1] "Progress: 14900000 out of 31730689"
## [1] "Progress: 1.5e+07 out of 31730689"
## [1] "Progress: 15100000 out of 31730689"
## [1] "Progress: 15200000 out of 31730689"
## [1] "Progress: 15300000 out of 31730689"
## [1] "Progress: 15400000 out of 31730689"
## [1] "Progress: 15500000 out of 31730689"
## [1] "Progress: 15600000 out of 31730689"
## [1] "Progress: 15700000 out of 31730689"
## [1] "Progress: 15800000 out of 31730689"
## [1] "Progress: 15900000 out of 31730689"
## [1] "Progress: 1.6e+07 out of 31730689"
```

```
## [1] "Progress: 16100000 out of 31730689"
## [1] "Progress: 16200000 out of 31730689"
## [1] "Progress: 16300000 out of 31730689"
## [1] "Progress: 16400000 out of 31730689"
## [1] "Progress: 16500000 out of 31730689"
## [1] "Progress: 16600000 out of 31730689"
## [1] "Progress: 16700000 out of 31730689"
## [1] "Progress: 16800000 out of 31730689"
## [1] "Progress: 16900000 out of 31730689"
## [1] "Progress: 1.7e+07 out of 31730689"
## [1] "Progress: 17100000 out of 31730689"
## [1] "Progress: 17200000 out of 31730689"
## [1] "Progress: 17300000 out of 31730689"
## [1] "Progress: 17400000 out of 31730689"
## [1] "Progress: 17500000 out of 31730689"
## [1] "Progress: 17600000 out of 31730689"
## [1] "Progress: 17700000 out of 31730689"
## [1] "Progress: 17800000 out of 31730689"
## [1] "Progress: 17900000 out of 31730689"
## [1] "Progress: 1.8e+07 out of 31730689"
## [1] "Progress: 18100000 out of 31730689"
## [1] "Progress: 18200000 out of 31730689"
## [1] "Progress: 18300000 out of 31730689"
## [1] "Progress: 18400000 out of 31730689"
## [1] "Progress: 18500000 out of 31730689"
## [1] "Progress: 18600000 out of 31730689"
## [1] "Progress: 18700000 out of 31730689"
## [1] "Progress: 18800000 out of 31730689"
## [1] "Progress: 18900000 out of 31730689"
## [1] "Progress: 1.9e+07 out of 31730689"
## [1] "Progress: 19100000 out of 31730689"
## [1] "Progress: 19200000 out of 31730689"
## [1] "Progress: 19300000 out of 31730689"
## [1] "Progress: 19400000 out of 31730689"
## [1] "Progress: 19500000 out of 31730689"
## [1] "Progress: 19600000 out of 31730689"
## [1] "Progress: 19700000 out of 31730689"
## [1] "Progress: 19800000 out of 31730689"
## [1] "Progress: 19900000 out of 31730689"
## [1] "Progress: 2e+07 out of 31730689"
## [1] "Progress: 20100000 out of 31730689"
## [1] "Progress: 20200000 out of 31730689"
## [1] "Progress: 20300000 out of 31730689"
## [1] "Progress: 20400000 out of 31730689"
## [1] "Progress: 20500000 out of 31730689"
## [1] "Progress: 20600000 out of 31730689"
## [1] "Progress: 20700000 out of 31730689"
## [1] "Progress: 20800000 out of 31730689"
## [1] "Progress: 20900000 out of 31730689"
## [1] "Progress: 2.1e+07 out of 31730689"
```

```
## [1] "Progress: 21100000 out of 31730689"
## [1] "Progress: 21200000 out of 31730689"
## [1] "Progress: 21300000 out of 31730689"
## [1] "Progress: 21400000 out of 31730689"
## [1] "Progress: 21500000 out of 31730689"
## [1] "Progress: 21600000 out of 31730689"
## [1] "Progress: 21700000 out of 31730689"
## [1] "Progress: 21800000 out of 31730689"
## [1] "Progress: 21900000 out of 31730689"
## [1] "Progress: 2.2e+07 out of 31730689"
## [1] "Progress: 22100000 out of 31730689"
## [1] "Progress: 22200000 out of 31730689"
## [1] "Progress: 22300000 out of 31730689"
## [1] "Progress: 22400000 out of 31730689"
## [1] "Progress: 22500000 out of 31730689"
## [1] "Progress: 22600000 out of 31730689"
## [1] "Progress: 22700000 out of 31730689"
## [1] "Progress: 22800000 out of 31730689"
## [1] "Progress: 22900000 out of 31730689"
## [1] "Progress: 2.3e+07 out of 31730689"
## [1] "Progress: 23100000 out of 31730689"
## [1] "Progress: 23200000 out of 31730689"
## [1] "Progress: 23300000 out of 31730689"
## [1] "Progress: 23400000 out of 31730689"
## [1] "Progress: 23500000 out of 31730689"
## [1] "Progress: 23600000 out of 31730689"
## [1] "Progress: 23700000 out of 31730689"
## [1] "Progress: 23800000 out of 31730689"
## [1] "Progress: 23900000 out of 31730689"
## [1] "Progress: 2.4e+07 out of 31730689"
## [1] "Progress: 24100000 out of 31730689"
## [1] "Progress: 24200000 out of 31730689"
## [1] "Progress: 24300000 out of 31730689"
## [1] "Progress: 24400000 out of 31730689"
## [1] "Progress: 24500000 out of 31730689"
## [1] "Progress: 24600000 out of 31730689"
## [1] "Progress: 24700000 out of 31730689"
## [1] "Progress: 24800000 out of 31730689"
## [1] "Progress: 24900000 out of 31730689"
## [1] "Progress: 2.5e+07 out of 31730689"
## [1] "Progress: 25100000 out of 31730689"
## [1] "Progress: 25200000 out of 31730689"
## [1] "Progress: 25300000 out of 31730689"
## [1] "Progress: 25400000 out of 31730689"
## [1] "Progress: 25500000 out of 31730689"
## [1] "Progress: 25600000 out of 31730689"
## [1] "Progress: 25700000 out of 31730689"
## [1] "Progress: 25800000 out of 31730689"
## [1] "Progress: 25900000 out of 31730689"
## [1] "Progress: 2.6e+07 out of 31730689"
```

```
## [1] "Progress: 26100000 out of 31730689"
## [1] "Progress: 26200000 out of 31730689"
## [1] "Progress: 26300000 out of 31730689"
## [1] "Progress: 26400000 out of 31730689"
## [1] "Progress: 26500000 out of 31730689"
## [1] "Progress: 26600000 out of 31730689"
## [1] "Progress: 26700000 out of 31730689"
## [1] "Progress: 26800000 out of 31730689"
## [1] "Progress: 26900000 out of 31730689"
## [1] "Progress: 2.7e+07 out of 31730689"
## [1] "Progress: 27100000 out of 31730689"
## [1] "Progress: 27200000 out of 31730689"
## [1] "Progress: 27300000 out of 31730689"
## [1] "Progress: 27400000 out of 31730689"
## [1] "Progress: 27500000 out of 31730689"
## [1] "Progress: 27600000 out of 31730689"
## [1] "Progress: 27700000 out of 31730689"
## [1] "Progress: 27800000 out of 31730689"
## [1] "Progress: 27900000 out of 31730689"
## [1] "Progress: 2.8e+07 out of 31730689"
## [1] "Progress: 28100000 out of 31730689"
## [1] "Progress: 28200000 out of 31730689"
## [1] "Progress: 28300000 out of 31730689"
## [1] "Progress: 28400000 out of 31730689"
## [1] "Progress: 28500000 out of 31730689"
## [1] "Progress: 28600000 out of 31730689"
## [1] "Progress: 28700000 out of 31730689"
## [1] "Progress: 28800000 out of 31730689"
## [1] "Progress: 28900000 out of 31730689"
## [1] "Progress: 2.9e+07 out of 31730689"
## [1] "Progress: 29100000 out of 31730689"
## [1] "Progress: 29200000 out of 31730689"
## [1] "Progress: 29300000 out of 31730689"
## [1] "Progress: 29400000 out of 31730689"
## [1] "Progress: 29500000 out of 31730689"
## [1] "Progress: 29600000 out of 31730689"
## [1] "Progress: 29700000 out of 31730689"
## [1] "Progress: 29800000 out of 31730689"
## [1] "Progress: 29900000 out of 31730689"
## [1] "Progress: 3e+07 out of 31730689"
## [1] "Progress: 30100000 out of 31730689"
## [1] "Progress: 30200000 out of 31730689"
## [1] "Progress: 30300000 out of 31730689"
## [1] "Progress: 30400000 out of 31730689"
## [1] "Progress: 30500000 out of 31730689"
## [1] "Progress: 30600000 out of 31730689"
## [1] "Progress: 30700000 out of 31730689"
## [1] "Progress: 30800000 out of 31730689"
## [1] "Progress: 30900000 out of 31730689"
## [1] "Progress: 3.1e+07 out of 31730689"
```

```
## [1] "Progress: 31100000 out of 31730689"
## [1] "Progress: 31200000 out of 31730689"
## [1] "Progress: 31300000 out of 31730689"
## [1] "Progress: 31400000 out of 31730689"
## [1] "Progress: 31500000 out of 31730689"
## [1] "Progress: 31600000 out of 31730689"
## [1] "Progress: 31700000 out of 31730689"

# Check number of distinct participants identified
n_distinct(data_recommendation$personID)

## [1] 538

# List of interests to group by similarity
attributes <- c("sports", "tvsports", "exercise", "dining", "museums", "art",
                "hiking", "gaming", "clubbing", "reading", "tv", "theater",
                "movies", "concerts", "music", "shopping", "yoga")

# List of values and traits to group by similarity
attributes <- c("importance_same_race", "importance_same_religion",
"attractive_important",
                "sincere_important", "intellicence_important",
"funny_important",
                "ambtition_important", "shared_interests_important",
"intelligence",
                "attractive", "funny", "ambition", "sincere")

data_recommendation[, attributes] <- scale(data_recommendation[, attributes])
# Normalize attributes

# Calculate similarity matrix for content-based filtering
similarity_matrix <- as.matrix(dist(data_recommendation[, attributes], method
= "cosine"))
similarity_score <- 1 / (1 + similarity_matrix)  # Convert distances to
similarity scores

# Assume user_ids from the content-based filtering
user_ids <- sort(unique(c(data_recommendation$personID,
data_recommendation$partnerID)))

# Create a sparse matrix for all user_ids for collaborative filtering
rating_matrix <- sparseMatrix(i = match(data_recommendation$personID,
user_ids),
                              j = match(data_recommendation$partnerID,
user_ids),
                              x = as.numeric(data_recommendation$match),
                              dims = c(length(user_ids), length(user_ids)))

# Convert to a realRatingMatrix
rating_matrix <- as(rating_matrix, "realRatingMatrix")
```

```r
# Rebuild the model and predict
cf_model <- Recommender(data = rating_matrix, method = "UBCF")
cf_recommendations <- predict(cf_model, rating_matrix)
cf_scores <- as(cf_recommendations, "matrix")
# Normalize
cf_scores_normalized <- cf_scores / max(cf_scores, na.rm = TRUE)
# Ensure similarity scores cover the same users
similarity_score <- similarity_score[user_ids, user_ids]
# Combined Score
combined_scores <- (cf_scores_normalized + similarity_score) / 2

# If only want based on similarities only
# combined_scores <- similarity_score

gender_vector <- data_recommendation$gender[match(user_ids,
data_recommendation$personID)]
# Ensure opposite genders are only recommended
for (i in 1:length(user_ids)) {
  for (j in 1:length(user_ids)) {
    if (gender_vector[i] == gender_vector[j]) {
      # Set score to zero if genders match
      combined_scores[i, j] <- 0
    }
  }
}

top_recommendations <- apply(combined_scores, 1, function(x) order(x,
decreasing = TRUE)[1:5])
#View(top_recommendations)

id <- 516
View(data_recommendation[data_recommendation$personID == id,])

# Example: Get top 5 recommendations for each user
top_n <- 5
top_recommendations <- apply(combined_scores, 1, function(x) {
  ordered_indices <- order(x, decreasing = TRUE)
  recommendations <- rep(0, length(x)) # initialize to 0
  recommendations[ordered_indices[1:top_n]] <- 1 # set top recommendations to
1
  return(recommendations)
})

top_recommendations_matrix <- matrix(unlist(top_recommendations), nrow =
nrow(combined_scores), byrow = TRUE)
#top_recommendations_matrix
```

PCA on Participants

```
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.3.3

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

library(ggbiplot)

## Warning: package 'ggbiplot' was built under R version 4.3.3

library(shiny)

## Warning: package 'shiny' was built under R version 4.3.3

attendees_data_pca <- attendees_data
```

Median Imputation

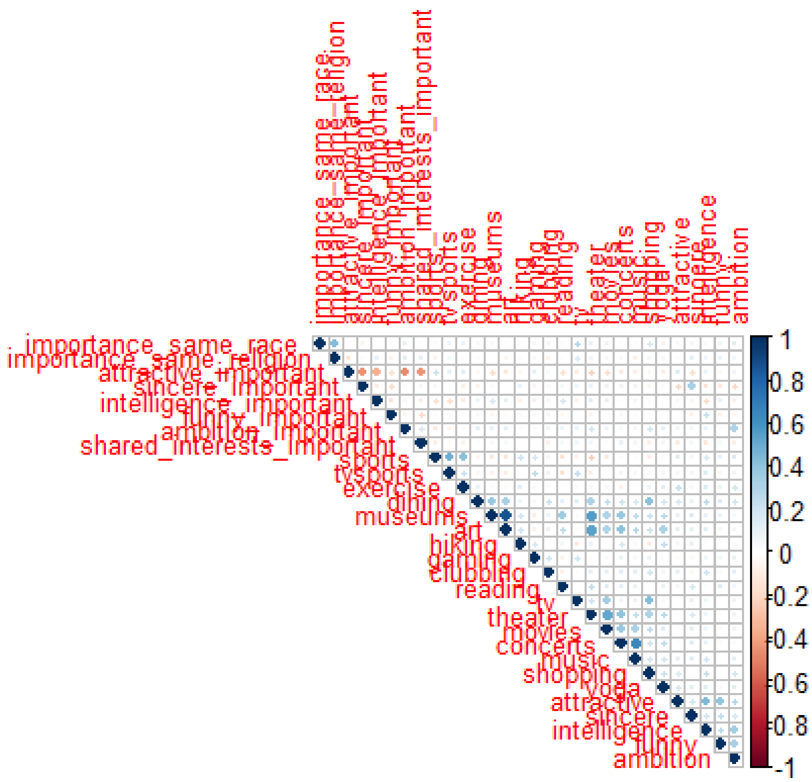
```
sum(is.na(attendees_data_pca))

## [1] 751

attendees_data_pca[] <- lapply(attendees_data_pca, function(x) {
    if (is.numeric(x)) {
        x[is.na(x)] <- median(x, na.rm = TRUE)
    } else if (is.factor(x)) {
        mode <- names(sort(table(x), decreasing = TRUE))[1]
        x[is.na(x)] <- mode
    }
    return(x)
})
sum(is.na(attendees_data_pca))

## [1] 0
```

Correlation Matrix

```
relevant_vars <- c("importance_same_race", "importance_same_religion",
"attractive_important",
                   "sincere_important", "intelligence_important",
"funny_important",
                   "ambition_important", "shared_interests_important",
"sports", "tvsports", "exercise", "dining", "museums", "art",
                   "hiking", "gaming", "clubbing", "reading", "tv",
"theater",
                   "movies", "concerts", "music", "shopping", "yoga",
"attractive", "sincere", "intelligence", "funny", "ambition")

cor_matrix <- cor(select(attendees_data_pca, relevant_vars), use =
"complete.obs")
```

```
## Warning: Using an external vector in selections was deprecated in
tidyselect 1.1.0.
## ℹ Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(relevant_vars)
##
##   # Now:
##   data %>% select(all_of(relevant_vars))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

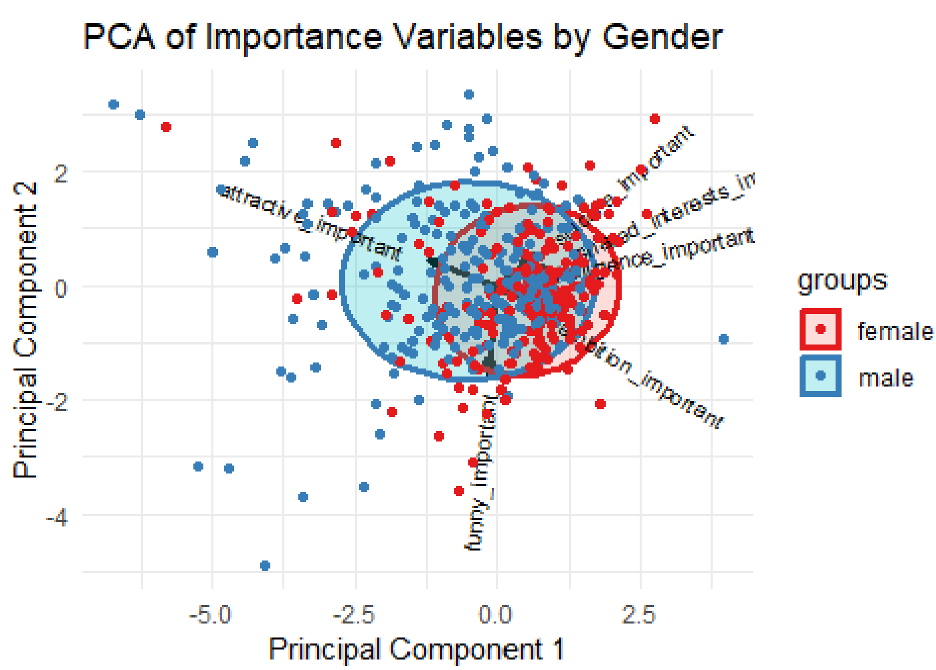
```r
corrplot(cor_matrix, method = "circle", type = "upper", tl.cex = 0.8)
```



Importance Variables

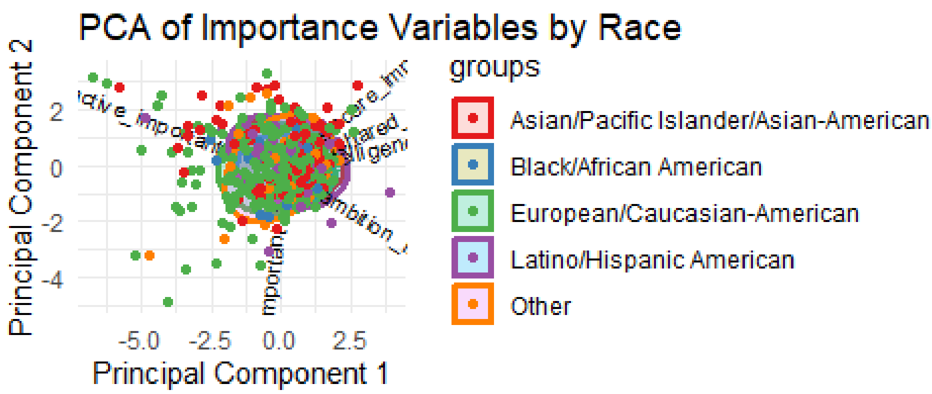
```r
# Select importance variables
importance_vars <- c("attractive_important", "sincere_important",
"intelligence_important",
                     "funny_important", "ambition_important",
"shared_interests_important")

data_for_pca <- attendees_data_pca %>% select(all_of(importance_vars))

# Standardize the data
```

```
data_scaled <- scale(data_for_pca)

# Perform PCA
pca_result <- prcomp(data_scaled, center = TRUE, scale. = TRUE)

# Summary of PCA
summary(pca_result)

## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6
## Standard deviation     1.3815 1.0713 1.0605 0.9954 0.8956 0.16218
## Proportion of Variance 0.3181 0.1913 0.1875 0.1651 0.1337 0.00438
## Cumulative Proportion  0.3181 0.5094 0.6968 0.8619 0.9956 1.00000

# Create PCA biplot with gender labels
pca_scores <- data.frame(pca_result$x)
pca_scores$gender <- attendees_data_pca$gender

ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$gender, scale = 0) +
    geom_point(aes(color = pca_scores$gender)) +
    labs(title = "PCA of Importance Variables by Gender", x = "Principal
Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```

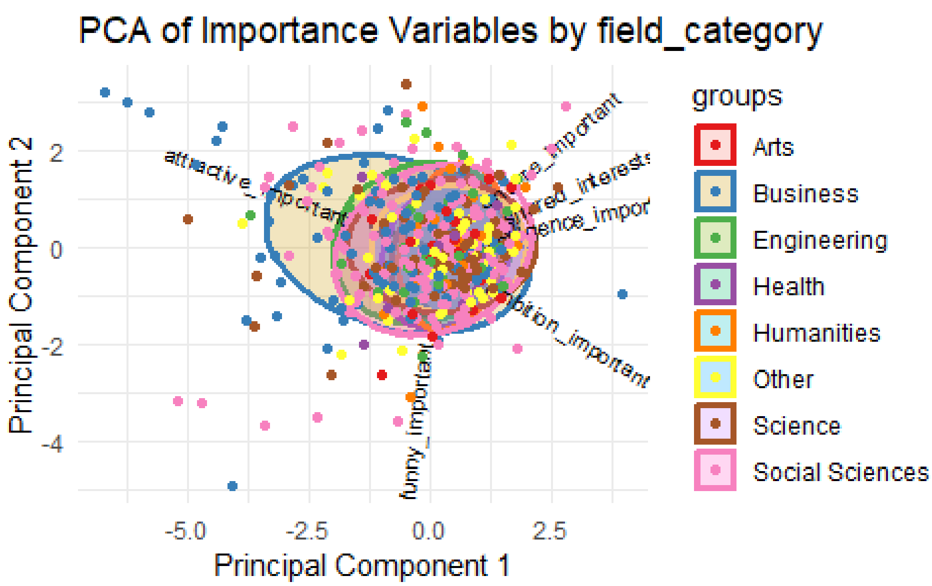



```
# Create PCA biplot with race labels
pca_scores <- data.frame(pca_result$x)
```

```
pca_scores$race <- attendees_data_pca$race

ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$race, scale = 0) +
    geom_point(aes(color = pca_scores$race)) +
    labs(title = "PCA of Importance Variables by Race", x = "Principal
Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```



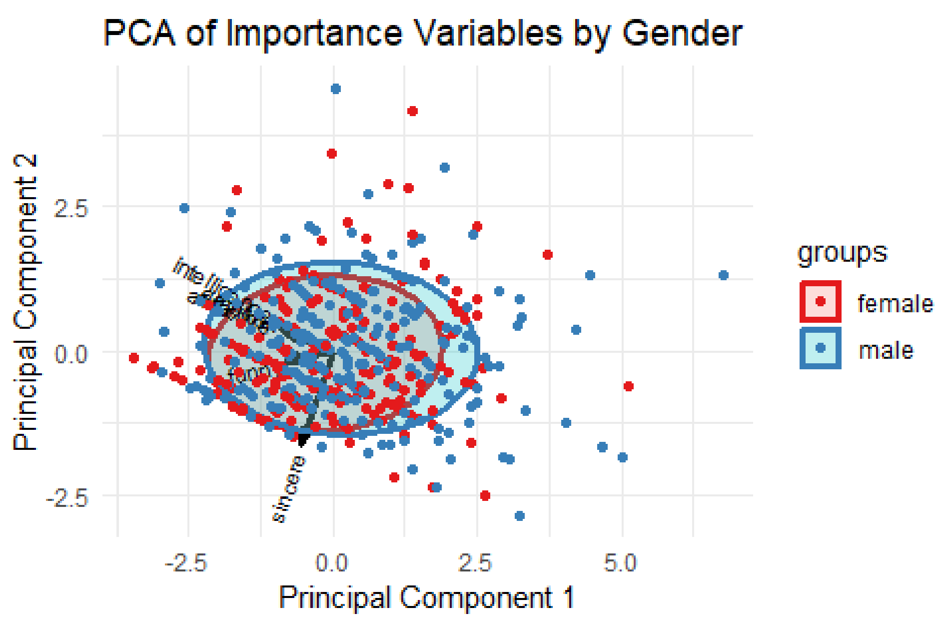

```
# Create PCA biplot with field category labels
pca_scores <- data.frame(pca_result$x)
pca_scores$field_category <- attendees_data_pca$field_category

ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$field_category,
scale = 0) +
    geom_point(aes(color = pca_scores$field_category)) +
    labs(title = "PCA of Importance Variables by field_category", x =
"Principal Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```



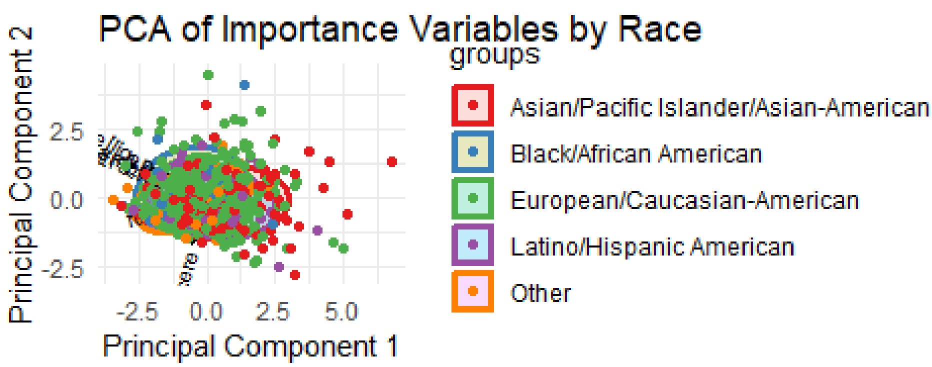
Self-Ratings Variables

```
# Select importance variables
self_ratings_vars <- c("attractive", "sincere", "intelligence", "funny",
"ambition")

data_for_pca <- attendees_data_pca %>% select(all_of(self_ratings_vars))

# Standardize the data
data_scaled <- scale(data_for_pca)

# Perform PCA
pca_result <- prcomp(data_scaled, center = TRUE, scale. = TRUE)

# Summary of PCA
summary(pca_result)

## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5
## Standard deviation     1.4679 0.9426 0.8580 0.8501 0.70558
## Proportion of Variance 0.4309 0.1777 0.1472 0.1446 0.09957
## Cumulative Proportion  0.4309 0.6086 0.7559 0.9004 1.00000

# Create PCA biplot with gender labels
pca_scores <- data.frame(pca_result$x)
pca_scores$gender <- attendees_data_pca$gender
```

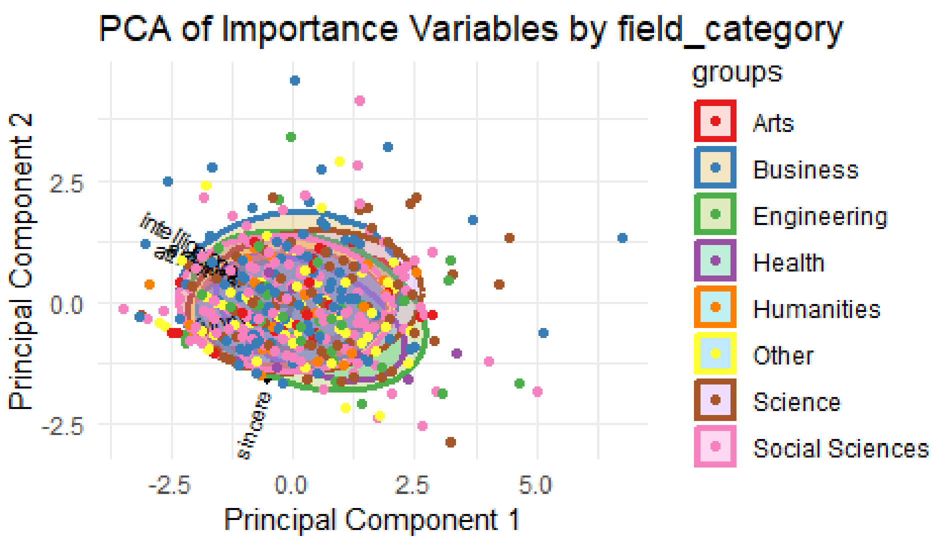
```r
ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$gender, scale = 0) +
    geom_point(aes(color = pca_scores$gender)) +
    labs(title = "PCA of Importance Variables by Gender", x = "Principal
Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```




```r
# Create PCA biplot with race labels
pca_scores <- data.frame(pca_result$x)
pca_scores$race <- attendees_data_pca$race

ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$race, scale = 0) +
    geom_point(aes(color = pca_scores$race)) +
    labs(title = "PCA of Importance Variables by Race", x = "Principal
Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```




```r
# Create PCA biplot with field category labels
pca_scores <- data.frame(pca_result$x)
pca_scores$field_category <- attendees_data_pca$field_category

ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$field_category,
scale = 0) +
    geom_point(aes(color = pca_scores$field_category)) +
    labs(title = "PCA of Importance Variables by field_category", x =
"Principal Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```

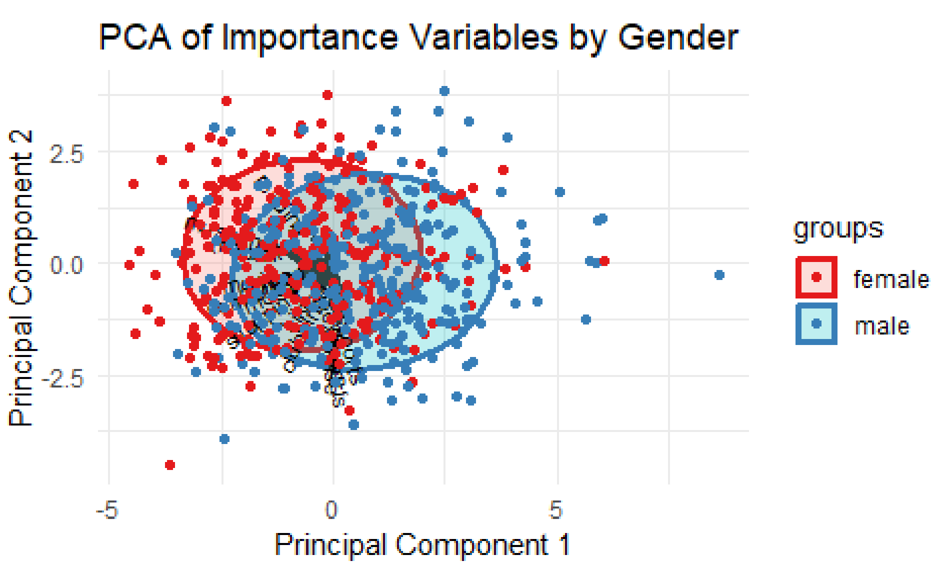


Hobbies Variables

```
# Select importance variables
hobbies_vars <- c("sports", "tvsports", "exercise", "dining", "museums",
"art", "hiking", "gaming", "clubbing", "reading", "tv", "theater", "movies",
"concerts", "music", "shopping", "yoga")

data_for_pca <- attendees_data_pca %>% select(all_of(hobbies_vars))

# Standardize the data
data_scaled <- scale(data_for_pca)

# Perform PCA
pca_result <- prcomp(data_scaled, center = TRUE, scale. = TRUE)

# Summary of PCA
summary(pca_result)

## Importance of components:
##                           PC1    PC2     PC3    PC4     PC5     PC6
PC7
## Standard deviation     1.9869 1.4418 1.29715 1.1224 1.05184 1.00668
0.98143
## Proportion of Variance 0.2322 0.1223 0.09898 0.0741 0.06508 0.05961
0.05666
## Cumulative Proportion  0.2322 0.3545 0.45348 0.5276 0.59266 0.65227
0.70893
```

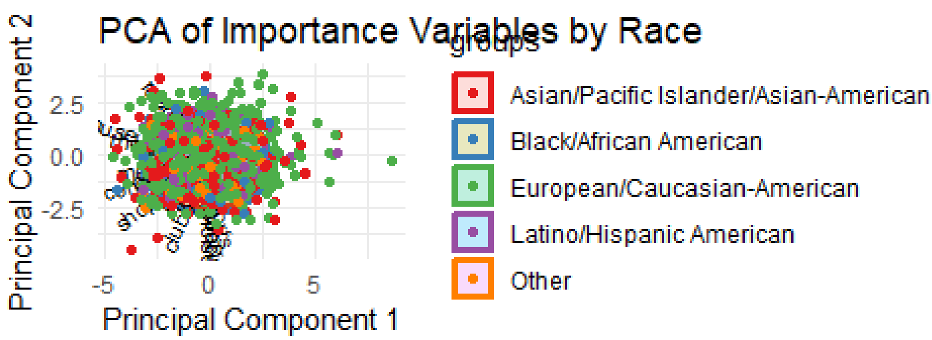
```
##                              PC8     PC9    PC10    PC11    PC12    PC13
PC14
## Standard deviation     0.89567 0.86461 0.84209 0.75458 0.71240 0.66362
0.65297
## Proportion of Variance 0.04719 0.04397 0.04171 0.03349 0.02985 0.02591
0.02508
## Cumulative Proportion  0.75612 0.80010 0.84181 0.87530 0.90516 0.93106
0.95614
##                           PC15    PC16    PC17
## Standard deviation     0.56807 0.54683 0.35196
## Proportion of Variance 0.01898 0.01759 0.00729
## Cumulative Proportion  0.97512 0.99271 1.00000
```

```r
# Create PCA biplot with gender labels
pca_scores <- data.frame(pca_result$x)
pca_scores$gender <- attendees_data_pca$gender

ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$gender, scale = 0) +
    geom_point(aes(color = pca_scores$gender)) +
    labs(title = "PCA of Importance Variables by Gender", x = "Principal
Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```

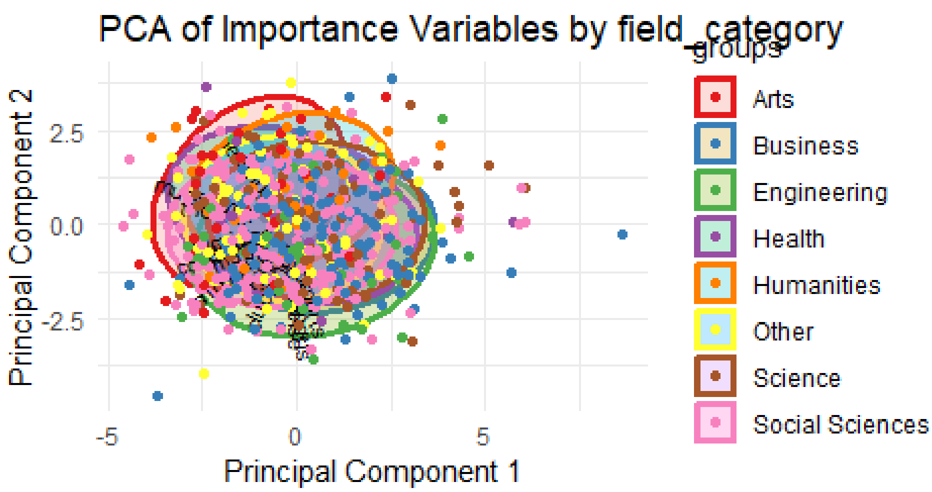



```r
# Create PCA biplot with race labels
pca_scores <- data.frame(pca_result$x)
pca_scores$race <- attendees_data_pca$race
```

```r
ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$race, scale = 0) +
    geom_point(aes(color = pca_scores$race)) +
    labs(title = "PCA of Importance Variables by Race", x = "Principal
Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```




```r
# Create PCA biplot with field category labels
pca_scores <- data.frame(pca_result$x)
pca_scores$field_category <- attendees_data_pca$field_category

ggbiplot(pca_result, ellipse = TRUE, groups = pca_scores$field_category,
scale = 0) +
    geom_point(aes(color = pca_scores$field_category)) +
    labs(title = "PCA of Importance Variables by field_category", x =
"Principal Component 1", y = "Principal Component 2") +
    theme_minimal() +
    scale_color_brewer(palette = "Set1")
```



App

```
# Define the function to remove outliers
remove_outliers <- function(df, cols) {
  for (col in cols) {
    Q1 <- quantile(df[[col]], 0.25, na.rm = TRUE)
    Q3 <- quantile(df[[col]], 0.75, na.rm = TRUE)
    IQR <- Q3 - Q1
    lower_bound <- Q1 - 1.5 * IQR
    upper_bound <- Q3 + 1.5 * IQR
    df <- df %>%
      filter(df[[col]] >= lower_bound & df[[col]] <= upper_bound)
  }
  return(df)
}

# Define UI for the application
ui <- fluidPage(
  titlePanel("Interactive PCA Biplot with Multiple Filters"),
  sidebarLayout(
    sidebarPanel(
      # Variable group selection
      checkboxGroupInput("importance_vars", "Select Importance Variables:",
                         choices = c("importance_same_race",
"importance_same_religion", "attractive_important",
                                     "sincere_important",
"intelligence_important", "funny_important",
```

```r
                                     "ambition_important",
"shared_interests_important")),

      checkboxGroupInput("hobbies_vars", "Select Hobbies Variables:",
                         choices = c("sports", "tvsports", "exercise",
"dining", "museums", "art",
                                     "hiking", "gaming", "clubbing",
"reading", "tv", "theater",
                                     "movies", "concerts", "music",
"shopping", "yoga")),

      checkboxGroupInput("self_ratings_vars", "Select Self Ratings
Variables:",
                         choices = c("attractive", "sincere", "intelligence",
"funny", "ambition")),

      # Grouping variable selection
      selectInput("group_by", "Group By:",
                  choices = list("Race" = "race",
                                 "Field Category" = "field_category",
                                 "Gender" = "gender")),

      # Dynamic checkboxes for filtering based on the grouping variable
      uiOutput("dynamic_filters"),

      # Dynamic checkboxes for toggling ellipses based on the selected group
      uiOutput("ellipse_toggles"),

      sliderInput("width", "Plot Width:",
                  min = 400, max = 1000, value = 600),
      sliderInput("height", "Plot Height:",
                  min = 400, max = 1000, value = 600),

      # Checkbox to remove outliers
      checkboxInput("remove_outliers", "Remove Outliers", value = FALSE),

      # Download button
      downloadButton("downloadPlot", "Download Plot")
    ),
    mainPanel(
      plotOutput("scaledPlot", width = "100%", height = "auto"),
      textOutput("resolution")
    )
  )
)

# Define server logic for the application
server <- function(input, output, session) {
```

```r
  # Update dynamic checkboxes based on the selected grouping variable
  output$dynamic_filters <- renderUI({
    choices <- unique(attendees_data_pca[[input$group_by]])
    checkboxGroupInput("filters", paste("Select", input$group_by, ":"),
                       choices = choices, selected = choices)
  })

  # Update dynamic checkboxes for toggling ellipses
  output$ellipse_toggles <- renderUI({
    choices <- unique(attendees_data_pca[[input$group_by]])
    checkboxGroupInput("toggle_ellipses", "Toggle Ellipses:",
                       choices = choices, selected = choices)
  })

  plot <- reactive({
    # Combine selected variables from different groups
    selected_vars <- c(input$importance_vars, input$hobbies_vars,
input$self_ratings_vars)

    # Filter data based on selected filters
    filtered_data <- attendees_data_pca %>%
      filter(get(input$group_by) %in% input$filters)

    # Remove outliers if checkbox is selected
    if (input$remove_outliers) {
      filtered_data <- remove_outliers(filtered_data, selected_vars)
    }

    # Ensure at least two levels in the factor for ellipses
    if (length(unique(filtered_data[[input$group_by]])) < 2) {
      return(ggplot() + labs(title = "Select at least two categories for the
selected grouping variable"))
    }

    # Select variables for PCA
    data_for_pca <- filtered_data %>% select(all_of(selected_vars))

    # Standardize the data
    data_scaled <- scale(data_for_pca)

    # Perform PCA on the selected variables
    pca_result <- prcomp(data_scaled, center = TRUE, scale. = TRUE)

    # Create PCA biplot with optional ellipses
    pca_scores <- data.frame(pca_result$x)
    pca_scores$group <- filtered_data[[input$group_by]]

    plot <- ggbiplot(pca_result, groups = pca_scores$group, scale = 0) +
      geom_point(aes(color = pca_scores$group)) +
```

```r
      labs(title = "PCA Biplot",
           x = "Principal Component 1", y = "Principal Component 2") +
      theme_minimal() +
      scale_color_brewer(palette = "Set1")

    for (group in input$toggle_ellipses) {
      plot <- plot + stat_ellipse(data = pca_scores[pca_scores$group ==
group, ], aes(x = PC1, y = PC2, color = group), level = 0.95)
    }

    plot
  })

  output$pcaPlot <- renderPlot({
    plot()
  }, height = function() {
    input$scale
  }, width = function() {
    input$scale
  })

  output$downloadPlot <- downloadHandler(
    filename = function() {
      paste("PCA_Biplot", Sys.Date(), ".png", sep = "")
   },
    content = function(file) {
      ggsave(file, plot = plot() + theme_minimal(), device = "png", bg =
"white")
    }
  )


  output$resolution <- renderText({
    paste("Current resolution:", input$scale, "x", input$scale)
  })
}

# Run the application
#shinyApp(ui = ui, server = server)
```