# Speed Dating

*Cherise Woo, Shu Han Rachel Chang*

*November 14, 2017*

### Introduction

We will analyze the Speed Dating Experiment dataset from kaggle.com, This dataset was compiled by Columbia Business School professors Ray Fisman and Sheena Iyengar for their paper Gender Differences in Mate Selection: Evidence From a Speed Dating Experiment.

Data was gathered from participants in experimental speed dating events from 2002-2004. During the events, the attendees would have a four minute "first date" with every other participant of the opposite sex. At the end of their four minutes, participants were asked if they would like to see their date again. They were also asked to rate their date on six attributes: Attractiveness, Sincerity, Intelligence, Fun, Ambition, and Shared Interests.

The dataset also includes questionnaire data gathered from participants at different points in the process. These fields include: demographics, dating habits, self-perception across key attributes, beliefs on what others find valuable in a mate, and lifestyle information. See the Speed Dating Data Key document below for details. It is from https://www.kaggle.com/annavictoria/speed-dating-experiment. It was accessed on November 15, 2017 by clicking on the Download button. We used libraries dplyr, tidyr, tidyverse, scales, readr, ggplot2, ggthemes, stargazer in the analysis.

### Data Acquisition and Selection

After downloading the csv file, I imported the dataset into R and kept only the columns needed for analysis. The file includes 8378 rows and 195 variables.

### Libraries

```
library(dplyr)
```

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

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

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

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

```
library(tidyr)
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: readr
## Loading tidyverse: purrr
```

```
## Conflicts with tidy packages ----------------------------------------------
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
## The following object is masked from 'package:purrr':
##
##     discard
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(readr)
library(ggplot2)
library(ggthemes)
```

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

```
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

```
Speed_Dating <- read_csv("C:/Users/linds/OneDrive/Fall 2017/STAT-612 R/Project/Speed Dating Data.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   iid = col_integer(),
##   id = col_integer(),
##   gender = col_integer(),
##   idg = col_integer(),
##   condtn = col_integer(),
##   wave = col_integer(),
##   round = col_integer(),
##   position = col_integer(),
##   order = col_integer(),
##   partner = col_integer(),
##   pid = col_integer(),
##   match = col_integer(),
##   int_corr = col_double(),
##   samerace = col_integer(),
##   age_o = col_integer(),
##   race_o = col_integer(),
##   pf_o_att = col_double(),
##   pf_o_sin = col_double(),
##   pf_o_int = col_double(),
##   pf_o_fun = col_double()
##   # ... with 99 more columns
```

```
## )
## See spec(...) for full column specifications.
```

```
head(Speed_Dating)
```

```
## # A tibble: 6 x 195
##     iid    id gender   idg condtn  wave round position positin1 order
##   <int> <int>  <int> <int>  <int> <int> <int>    <int>    <chr> <int>
## 1     1     1      0     1      1     1    10        7     <NA>     4
## 2     1     1      0     1      1     1    10        7     <NA>     3
## 3     1     1      0     1      1     1    10        7     <NA>    10
## 4     1     1      0     1      1     1    10        7     <NA>     5
## 5     1     1      0     1      1     1    10        7     <NA>     7
## 6     1     1      0     1      1     1    10        7     <NA>     6
## # ... with 185 more variables: partner <int>, pid <int>, match <int>,
## #   int_corr <dbl>, samerace <int>, age_o <int>, race_o <int>,
## #   pf_o_att <dbl>, pf_o_sin <dbl>, pf_o_int <dbl>, pf_o_fun <dbl>,
## #   pf_o_amb <dbl>, pf_o_sha <dbl>, dec_o <int>, attr_o <dbl>,
## #   sinc_o <dbl>, intel_o <dbl>, fun_o <dbl>, amb_o <dbl>, shar_o <dbl>,
## #   like_o <dbl>, prob_o <dbl>, met_o <int>, age <int>, field <chr>,
## #   field_cd <dbl>, undergra <chr>, mn_sat <chr>, tuition <chr>,
## #   race <int>, imprace <int>, imprelig <int>, from <chr>, zipcode <dbl>,
## #   income <dbl>, goal <int>, date <int>, go_out <int>, career <chr>,
## #   career_c <dbl>, sports <int>, tvsports <int>, exercise <int>,
## #   dining <int>, museums <int>, art <int>, hiking <int>, gaming <int>,
## #   clubbing <int>, reading <int>, tv <int>, theater <int>, movies <int>,
## #   concerts <int>, music <int>, shopping <int>, yoga <int>,
## #   exphappy <int>, expnum <int>, attr1_1 <dbl>, sinc1_1 <dbl>,
## #   intel1_1 <dbl>, fun1_1 <dbl>, amb1_1 <dbl>, shar1_1 <dbl>,
## #   attr4_1 <chr>, sinc4_1 <chr>, intel4_1 <chr>, fun4_1 <chr>,
## #   amb4_1 <chr>, shar4_1 <chr>, attr2_1 <dbl>, sinc2_1 <dbl>,
## #   intel2_1 <dbl>, fun2_1 <dbl>, amb2_1 <dbl>, shar2_1 <dbl>,
## #   attr3_1 <int>, sinc3_1 <int>, fun3_1 <int>, intel3_1 <int>,
## #   amb3_1 <int>, attr5_1 <chr>, sinc5_1 <chr>, intel5_1 <chr>,
## #   fun5_1 <chr>, amb5_1 <chr>, dec <int>, attr <dbl>, sinc <dbl>,
## #   intel <dbl>, fun <dbl>, amb <dbl>, shar <dbl>, like <dbl>, prob <dbl>,
## #   met <int>, match_es <dbl>, attr1_s <chr>, sinc1_s <chr>, ...
```

### Dataset Variables (Original)

```
names(Speed_Dating)
```

```
##   [1] "iid"      "id"       "gender"   "idg"      "condtn"   "wave"
##   [7] "round"    "position" "positin1" "order"    "partner"  "pid"
##  [13] "match"    "int_corr" "samerace" "age_o"    "race_o"   "pf_o_att"
##  [19] "pf_o_sin" "pf_o_int" "pf_o_fun" "pf_o_amb" "pf_o_sha" "dec_o"
##  [25] "attr_o"   "sinc_o"   "intel_o"  "fun_o"    "amb_o"    "shar_o"
##  [31] "like_o"   "prob_o"   "met_o"    "age"      "field"    "field_cd"
##  [37] "undergra" "mn_sat"   "tuition"  "race"     "imprace"  "imprelig"
##  [43] "from"     "zipcode"  "income"   "goal"     "date"     "go_out"
##  [49] "career"   "career_c" "sports"   "tvsports" "exercise" "dining"
##  [55] "museums"  "art"      "hiking"   "gaming"   "clubbing" "reading"
##  [61] "tv"       "theater"  "movies"   "concerts" "music"    "shopping"
```

```
##  [67] "yoga"     "exphappy" "expnum"   "attr1_1"  "sinc1_1"  "intel1_1"
##  [73] "fun1_1"   "amb1_1"   "shar1_1"  "attr4_1"  "sinc4_1"  "intel4_1"
##  [79] "fun4_1"   "amb4_1"   "shar4_1"  "attr2_1"  "sinc2_1"  "intel2_1"
##  [85] "fun2_1"   "amb2_1"   "shar2_1"  "attr3_1"  "sinc3_1"  "fun3_1"
##  [91] "intel3_1" "amb3_1"   "attr5_1"  "sinc5_1"  "intel5_1" "fun5_1"
##  [97] "amb5_1"   "dec"      "attr"     "sinc"     "intel"    "fun"
## [103] "amb"      "shar"     "like"     "prob"     "met"      "match_es"
## [109] "attr1_s"  "sinc1_s"  "intel1_s" "fun1_s"   "amb1_s"   "shar1_s"
## [115] "attr3_s"  "sinc3_s"  "intel3_s" "fun3_s"   "amb3_s"   "satis_2"
## [121] "length"   "numdat_2" "attr7_2"  "sinc7_2"  "intel7_2" "fun7_2"
## [127] "amb7_2"   "shar7_2"  "attr1_2"  "sinc1_2"  "intel1_2" "fun1_2"
## [133] "amb1_2"   "shar1_2"  "attr4_2"  "sinc4_2"  "intel4_2" "fun4_2"
## [139] "amb4_2"   "shar4_2"  "attr2_2"  "sinc2_2"  "intel2_2" "fun2_2"
## [145] "amb2_2"   "shar2_2"  "attr3_2"  "sinc3_2"  "intel3_2" "fun3_2"
## [151] "amb3_2"   "attr5_2"  "sinc5_2"  "intel5_2" "fun5_2"   "amb5_2"
## [157] "you_call" "them_cal" "date_3"   "numdat_3" "num_in_3" "attr1_3"
## [163] "sinc1_3"  "intel1_3" "fun1_3"   "amb1_3"   "shar1_3"  "attr7_3"
## [169] "sinc7_3"  "intel7_3" "fun7_3"   "amb7_3"   "shar7_3"  "attr4_3"
## [175] "sinc4_3"  "intel4_3" "fun4_3"   "amb4_3"   "shar4_3"  "attr2_3"
## [181] "sinc2_3"  "intel2_3" "fun2_3"   "amb2_3"   "shar2_3"  "attr3_3"
## [187] "sinc3_3"  "intel3_3" "fun3_3"   "amb3_3"   "attr5_3"  "sinc5_3"
## [193] "intel5_3" "fun5_3"   "amb5_3"
```

### Unique ID-Surrogate Key

Variables iid and pid create a unique row.

```
Speed_Dating %>%
  count(iid,pid) %>%
  filter(n>1)
```

```
## # A tibble: 0 x 3
## # ... with 3 variables: iid <int>, pid <int>, n <int>
```

### Define a Subset of Variables to Use

- **new.id**: surrogate key
- **wave**: different dates of the speed dating
- **gender**: individual's gender (Female=0,Male=1)
- **from**: city the indiviudal is from
- **zipcode**: corresponding zipcode to the city
- **income**: Median household income based on zipcode using the Census Bureau website: (http://venus.census.gov/cdrom/lookup/CMD=LIST/DB=C90STF3B/LEV=ZIP); When there is no income it means that they are either from abroad or did not enter their zip code.
- **order**: the order the indiviual met their partner on that night
- **match**: whether the individual and partner both said yes=1 or no=0 to another date (mutual aggreement)
- **int_corr**: the correlation between participant's and partner's ratings of interests in Time 1
- **undergra**: school attended for undergraduate degree
- **field_cd**: field coded
- **field**: field of study (correspond to field_cd)
- **age**: individual's age
- **race**: individual's race

  - 1 = Black/African American
  - 2 = European/Caucasian-American
  - 3 = Latino/Hispanic American
  - 4 = Asian/Pacific Islander/Asian-American
  - 5 = Native American
  - 6 = Other
- **Imprace**: how important race is to you in a significant other on a scale of 1-10.
- **goal**: What is your primary goal in participating in this event?
  - 1 = Seemed like a fun night out
  - 2 = To meet new people
  - 3 = To get a date
  - 4 = Looking for a serious relationship
  - 5 = To say I did it
  - 6 = Other
- **date**: how frequently does the individual go on dates
  - 1 = Several times a week
  - 2 = Twice a week
  - 3 = Once a week
  - 4 = Twice a month
  - 5 = Once a month
  - 6 = Several times a year
  - 7 = Almost never
  - individual's stated preference at Time 1:
  - **attr1_1**: Attractive
  - **sinc1_1**: Sincere
  - **intel1_1**: Intelligent
  - **fun1_1**: Fun
  - **amb1_1**: Ambitious
  - **shar1_1**: Has shared interests/hobbies
  - partner's stated preference at Time 1 - adds up to 100:
  - **pf_o_att**: Attractive
  - **pf_o_sin**: Sincere
  - **pf_o_int**: Intelligent
  - **pf_o_fun**: Fun
  - **pf_o_amb**: Ambitious
  - **pf_o_sha**: Has shared interests/hobbies
  - Scorecard values: rated from 1-10, the partner's ratings
  - **dec_o**: decision of partner the night of event
  - existing names (**attr_o** - **shar_o**) explained above
  - New values:
    * **like_o**: How much do you like this person? (1=don't like at all, 10=like a lot)
    * **prob_o**: How probable do you think it is that this person will say 'yes' for you? (1=not probable, 10=extremely probable)
    * **met_o**: Have you met this person before? (1=yes, 2=no)
  - Scorecard Values: rated from 1-10, the individual's ratings (NOT partner)
  - **dec**: decision
  - existing names (**attr** - **met**) explained above
  - **imprace**: how important is it to you (on a scale of 1-10) that a person you date be of the same racial/ethnic background.
  - **samerace**: participant and the partner were the same race. 1= yes, 0=no
  - **race_o**: race of partner
  - **age_o**: age of partner

### Select Variables, Filter Out Waves 6:9

```
SpeedDatingNarrow <- Speed_Dating %>%
  mutate(new.id = paste0(iid,pid)) %>%
  select(new.id,wave,gender,from,zipcode,income,order,match,int_corr,undergra,field_cd,field,age,goal,da
  filter(!wave %in% c(6:9))

head(SpeedDatingNarrow)
```

```
## # A tibble: 6 x 52
##   new.id  wave gender    from zipcode income order match int_corr undergra
##    <chr> <int>  <int>   <chr>   <dbl>  <dbl> <int> <int>    <dbl>    <chr>
## 1    111     1      0 Chicago   60521  69487     4     0     0.14     <NA>
## 2    112     1      0 Chicago   60521  69487     3     0     0.54     <NA>
## 3    113     1      0 Chicago   60521  69487    10     1     0.16     <NA>
## 4    114     1      0 Chicago   60521  69487     5     1     0.61     <NA>
## 5    115     1      0 Chicago   60521  69487     7     1     0.21     <NA>
## 6    116     1      0 Chicago   60521  69487     6     0     0.25     <NA>
## # ... with 42 more variables: field_cd <dbl>, field <chr>, age <int>,
## #   goal <int>, date <int>, attr1_1 <dbl>, sinc1_1 <dbl>, intel1_1 <dbl>,
## #   fun1_1 <dbl>, amb1_1 <dbl>, shar1_1 <dbl>, pf_o_att <dbl>,
## #   pf_o_sin <dbl>, pf_o_int <dbl>, pf_o_fun <dbl>, pf_o_amb <dbl>,
## #   pf_o_sha <dbl>, dec_o <int>, attr_o <dbl>, sinc_o <dbl>,
## #   intel_o <dbl>, fun_o <dbl>, amb_o <dbl>, shar_o <dbl>, like_o <dbl>,
## #   prob_o <dbl>, met_o <int>, dec <int>, attr <dbl>, sinc <dbl>,
## #   intel <dbl>, fun <dbl>, amb <dbl>, shar <dbl>, like <dbl>, prob <dbl>,
## #   met <int>, race <int>, race_o <int>, imprace <int>, samerace <int>,
## #   age_o <int>
```

We selected variables only in Time 1, which is before and during the event. These variables include the stated preference of the participant and the partner, and the scorecard values of the participant and the partner.

### Summary of selected variables

```
SpeedDatingNarrow %>%
  select(age, income, imprace, attr1_1, sinc1_1, intel1_1, fun1_1, amb1_1, shar1_1) %>%
  summary()
```

```
##       age            income          imprace          attr1_1
##  Min.   :18.00   Min.   :  8607   Min.   : 0.000   Min.   :  0.00
##  1st Qu.:24.00   1st Qu.: 31148   1st Qu.: 1.000   1st Qu.: 15.00
##  Median :26.00   Median : 42390   Median : 3.000   Median : 20.00
##  Mean   :26.28   Mean   : 44275   Mean   : 3.652   Mean   : 23.98
##  3rd Qu.:28.00   3rd Qu.: 53940   3rd Qu.: 6.000   3rd Qu.: 30.00
##  Max.   :55.00   Max.   :109031   Max.   :10.000   Max.   :100.00
##  NA's   :90      NA's   :3473     NA's   :74       NA's   :74
##     sinc1_1        intel1_1         fun1_1          amb1_1
##  Min.   : 0.0   Min.   : 0.00   Min.   : 0.00   Min.   : 0.000
##  1st Qu.:10.0   1st Qu.:18.00   1st Qu.:14.00   1st Qu.: 5.000
##  Median :20.0   Median :20.00   Median :18.00   Median :10.000
##  Mean   :17.3   Mean   :20.56   Mean   :17.35   Mean   : 9.744
##  3rd Qu.:20.0   3rd Qu.:25.00   3rd Qu.:20.00   3rd Qu.:15.000
##  Max.   :60.0   Max.   :50.00   Max.   :50.00   Max.   :53.000
```

```
##  NA's   :74     NA's   :74      NA's   :84      NA's   :94
##     shar1_1
##  Min.   : 0.00
##  1st Qu.: 5.00
##  Median :10.00
##  Mean   :11.25
##  3rd Qu.:15.00
##  Max.   :30.00
##  NA's   :116
```

As we can tell from the summary above, the median age of participants is 26, and didn't care a lot about the race of their partners. On the other hand, attractiveness of partner is the most important attribute with the highest median among all attributes.

### Counts for Waves

```
SpeedDatingNarrow %>%
  count(wave)
```

```
## # A tibble: 17 x 2
##     wave     n
##    <int> <int>
##  1     1   200
##  2     2   608
##  3     3   200
##  4     4   648
##  5     5   190
##  6    10   162
##  7    11   882
##  8    12   392
##  9    13   180
## 10    14   720
## 11    15   684
## 12    16    96
## 13    17   280
## 14    18    72
## 15    19   450
## 16    20    84
## 17    21   968
```

### Create Weighted Variables for Individual

```
weighted <- SpeedDatingNarrow %>%
  filter(!wave %in% c('6','7','8','9')) %>%
  mutate(per_attr1_1 = attr1_1/100,
         per_sinc1_1 = sinc1_1/100,
         per_intel1_1 = intel1_1/100,
         per_fun1_1 = fun1_1/100,
         per_amb1_1 = amb1_1/100,
         per_shar1_1 = shar1_1/100) %>%
  mutate(WeightedScore = ((per_attr1_1*attr) +(per_sinc1_1*sinc)+(per_intel1_1*intel)+(per_fun1_1*fun)+
  select(new.id,gender,from,zipcode,income,match,dec,WeightedScore,wave,order,everything())
```

```
weighted
```

```
## # A tibble: 6,816 x 59
##    new.id gender    from zipcode income match   dec WeightedScore  wave
##     <chr>  <int>   <chr>   <dbl>  <dbl> <int> <int>         <dbl> <int>
##  1    111      0 Chicago   60521  69487     0     1         0.680     1
##  2    112      0 Chicago   60521  69487     0     1         0.690     1
##  3    113      0 Chicago   60521  69487     1     1         0.715     1
##  4    114      0 Chicago   60521  69487     1     1         0.700     1
##  5    115      0 Chicago   60521  69487     1     1         0.620     1
##  6    116      0 Chicago   60521  69487     0     0         0.590     1
##  7    117      0 Chicago   60521  69487     0     1         0.620     1
##  8    118      0 Chicago   60521  69487     0     0         0.635     1
##  9    119      0 Chicago   60521  69487     1     1         0.760     1
## 10    120      0 Chicago   60521  69487     0     1         0.705     1
## # ... with 6,806 more rows, and 50 more variables: order <int>,
## #   int_corr <dbl>, undergra <chr>, field_cd <dbl>, field <chr>,
## #   age <int>, goal <int>, date <int>, attr1_1 <dbl>, sinc1_1 <dbl>,
## #   intel1_1 <dbl>, fun1_1 <dbl>, amb1_1 <dbl>, shar1_1 <dbl>,
## #   pf_o_att <dbl>, pf_o_sin <dbl>, pf_o_int <dbl>, pf_o_fun <dbl>,
## #   pf_o_amb <dbl>, pf_o_sha <dbl>, dec_o <int>, attr_o <dbl>,
## #   sinc_o <dbl>, intel_o <dbl>, fun_o <dbl>, amb_o <dbl>, shar_o <dbl>,
## #   like_o <dbl>, prob_o <dbl>, met_o <int>, attr <dbl>, sinc <dbl>,
## #   intel <dbl>, fun <dbl>, amb <dbl>, shar <dbl>, like <dbl>, prob <dbl>,
## #   met <int>, race <int>, race_o <int>, imprace <int>, samerace <int>,
## #   age_o <int>, per_attr1_1 <dbl>, per_sinc1_1 <dbl>, per_intel1_1 <dbl>,
## #   per_fun1_1 <dbl>, per_amb1_1 <dbl>, per_shar1_1 <dbl>
```

### Create Weighted Variables for Partner

```
weighted <- weighted %>%
    mutate(per_pf_o_att = pf_o_att/100,
         per_pf_o_sin=pf_o_sin/100,
         per_pf_o_int = pf_o_int/100,
         per_pf_o_fun = pf_o_fun/100,
         per_pf_o_amb=pf_o_amb/100,
         per_pf_o_sha=pf_o_sha/100) %>%
  mutate(PartnerWeightedScore = ((per_pf_o_att*attr_o)+(per_pf_o_sin*sinc_o)+(per_pf_o_int*intel_o)+(pe
  select(new.id,gender,from,zipcode,income,match,dec,WeightedScore,dec_o,PartnerWeightedScore,wave,orde

weighted
```

```
## # A tibble: 6,816 x 66
##    new.id gender    from zipcode income match   dec WeightedScore dec_o
##     <chr>  <int>   <chr>   <dbl>  <dbl> <int> <int>         <dbl> <int>
##  1    111      0 Chicago   60521  69487     0     1         0.680     0
##  2    112      0 Chicago   60521  69487     0     1         0.690     0
##  3    113      0 Chicago   60521  69487     1     1         0.715     1
##  4    114      0 Chicago   60521  69487     1     1         0.700     1
##  5    115      0 Chicago   60521  69487     1     1         0.620     1
##  6    116      0 Chicago   60521  69487     0     0         0.590     1
##  7    117      0 Chicago   60521  69487     0     1         0.620     0
```

```
##  8    118      0 Chicago   60521  69487     0     0         0.635     0
##  9    119      0 Chicago   60521  69487     1     1         0.760     1
## 10    120      0 Chicago   60521  69487     0     1         0.705     0
## # ... with 6,806 more rows, and 57 more variables:
## #   PartnerWeightedScore <dbl>, wave <int>, order <int>, int_corr <dbl>,
## #   undergra <chr>, field_cd <dbl>, field <chr>, age <int>, goal <int>,
## #   date <int>, attr1_1 <dbl>, sinc1_1 <dbl>, intel1_1 <dbl>,
## #   fun1_1 <dbl>, amb1_1 <dbl>, shar1_1 <dbl>, pf_o_att <dbl>,
## #   pf_o_sin <dbl>, pf_o_int <dbl>, pf_o_fun <dbl>, pf_o_amb <dbl>,
## #   pf_o_sha <dbl>, attr_o <dbl>, sinc_o <dbl>, intel_o <dbl>,
## #   fun_o <dbl>, amb_o <dbl>, shar_o <dbl>, like_o <dbl>, prob_o <dbl>,
## #   met_o <int>, attr <dbl>, sinc <dbl>, intel <dbl>, fun <dbl>,
## #   amb <dbl>, shar <dbl>, like <dbl>, prob <dbl>, met <int>, race <int>,
## #   race_o <int>, imprace <int>, samerace <int>, age_o <int>,
## #   per_attr1_1 <dbl>, per_sinc1_1 <dbl>, per_intel1_1 <dbl>,
## #   per_fun1_1 <dbl>, per_amb1_1 <dbl>, per_shar1_1 <dbl>,
## #   per_pf_o_att <dbl>, per_pf_o_sin <dbl>, per_pf_o_int <dbl>,
## #   per_pf_o_fun <dbl>, per_pf_o_amb <dbl>, per_pf_o_sha <dbl>
```

To calculate the impacts of the attributes of the participant's decision, we created a weighted score for the participant and the partner by weighting the stated preferences and the scorecard values.

### Individual Race

Then, we transformed the coded race to text for the ease of future analysis.

```
weighted$race[weighted$race==1] <- "Black/African American"
weighted$race[weighted$race==2] <- "European/Caucasian-American"
weighted$race[weighted$race==3] <- "Latino/Hispanic American"
weighted$race[weighted$race==4] <- "Asian/Pacific Islander/Asian-American"
weighted$race[weighted$race==5] <- "Native American"
weighted$race[weighted$race==6] <- "Other"
```

### Partner Race

```
weighted$race_o[weighted$race_o==1] <- "Black/African American"
weighted$race_o[weighted$race_o==2] <- "European/Caucasian-American"
weighted$race_o[weighted$race_o==3] <- "Latino/Hispanic American"
weighted$race_o[weighted$race_o==4] <- "Asian/Pacific Islander/Asian-American"
weighted$race_o[weighted$race_o==5] <- "Native American"
weighted$race_o[weighted$race_o==6] <- "Other"
```

### Counts for Races

```
race <- weighted %>%
  count(race) %>%
  na.omit(race)
race
```

```
## # A tibble: 5 x 2
##                                    race     n
##                                   <chr> <int>
## 1 Asian/Pacific Islander/Asian-American  1649
```

```
## 2                Black/African American   308
## 3           European/Caucasian-American  3786
## 4              Latino/Hispanic American   569
## 5                                 Other   446
```

Most of the participants in the experiment are European/Caucasion-American, and followed by Asian/Pacific Islander/Asian-American.

### Avg. Importance of Race vs. Avg. Weighted Score

```
weighted %>%
  filter(gender==0) %>%
  group_by(race,dec)%>%
  summarise(
    avg_imprace = mean(imprace,na.rm=TRUE),
    avg_weighted = mean(WeightedScore,na.rm = TRUE),
    avg_partner_weighted = mean(PartnerWeightedScore,na.rm=TRUE)
  )
```

```
## # A tibble: 12 x 5
## # Groups:   race [?]
##                                    race   dec avg_imprace avg_weighted
##                                   <chr> <int>       <dbl>        <dbl>
##  1 Asian/Pacific Islander/Asian-American     0    3.617587    0.5930846
##  2 Asian/Pacific Islander/Asian-American     1    3.378517    0.6970364
##  3                Black/African American     0    4.310811    0.5899206
##  4                Black/African American     1    3.535211    0.7460317
##  5           European/Caucasian-American     0    4.381116    0.6263791
##  6           European/Caucasian-American     1    4.461929    0.7651189
##  7              Latino/Hispanic American     0    2.461187    0.5322828
##  8              Latino/Hispanic American     1    2.544776    0.6863800
##  9                                 Other     0    3.851562    0.5869550
## 10                                 Other     1    3.609195    0.7404930
## 11                                  <NA>     0         NaN          NaN
## 12                                  <NA>     1         NaN          NaN
## # ... with 1 more variables: avg_partner_weighted <dbl>
```

To compare the importance of race and the weighted score across different race, we emplyed dplyr to compute the differences. The chart above shows that European/Caucasian American rated their partners higher than other races, and Latino/Hispanic American and European/Caucasian American received scores slightly higher than other races.

### Are people more likely to get a match if they're the same race?

We also wanted to see if participants are more likely to get a match if they're both the same race, the result shows that out of all matches, 646 participants who got a match are not the same race.

```
SpeedDatingNarrow %>%
  filter(match == 1) %>%
  count(samerace)
```

```
## # A tibble: 2 x 2
##   samerace     n
##      <int> <int>
```

```
## 1        0   646
## 2        1   478
```

We also wanted to see if participants are more likely to say yes to a second date if their partner is the same race. The result shows that the partners of the 1691 participants who agreed to go on a second date are not the in the same race as the participant, which is 500 more than the partners who are in the same race as the participants.

```
SpeedDatingNarrow %>%
  filter(dec == 1) %>%
  count(samerace)
```

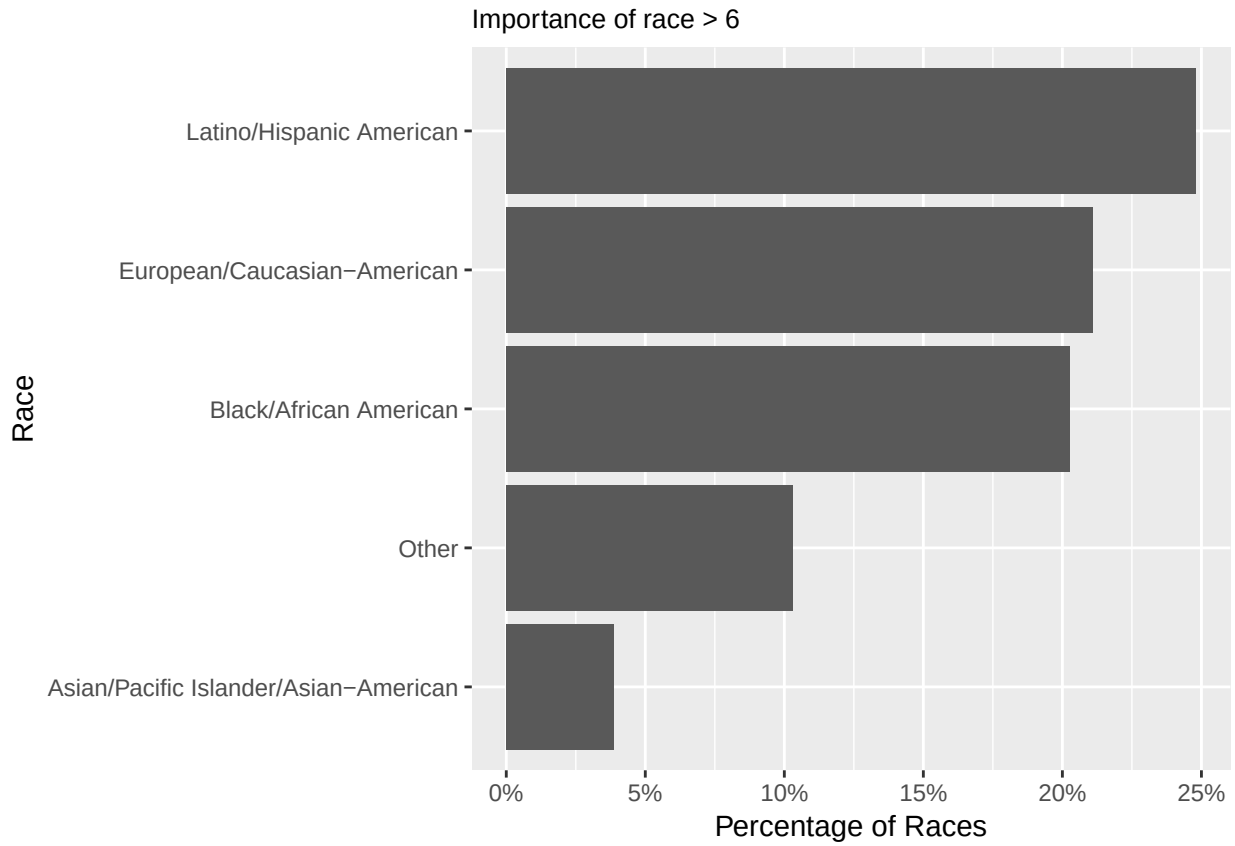
```
## # A tibble: 2 x 2
##   samerace     n
##      <int> <int>
## 1        0  1691
## 2        1  1152
```

### Which race cares more about their partner's race?

### imprace > 6

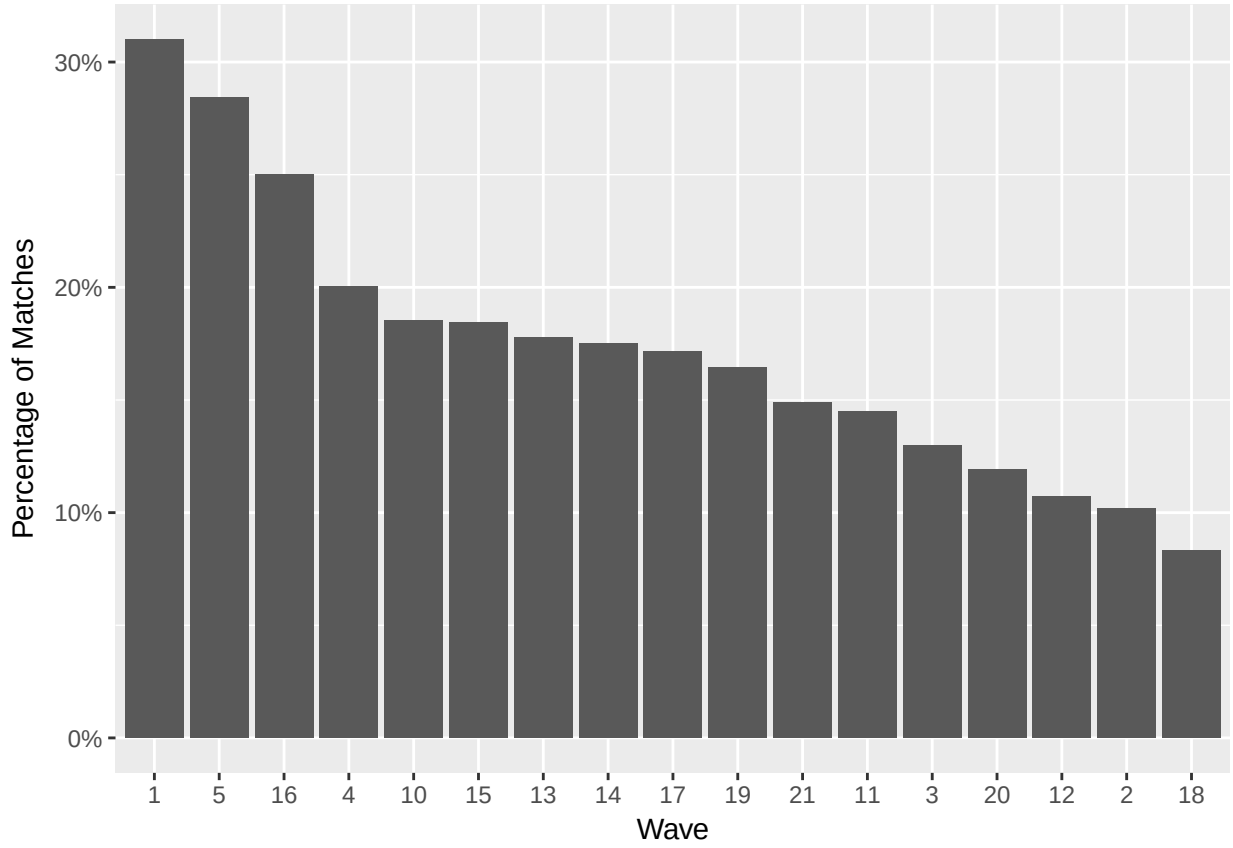
We also created a bar graph to see which race cares more about their partner's race. I divided the number of participants with imprace larger than 6, and divided by the total number of participants of each race. The result shows that Latino/Hispanic American is the race with the highest percentage, followed by European/Caucasian American, and Asian/Pacific Islander/Asian American has the lowest percentage.

```
countrace1 <- weighted %>%
  filter( imprace > 6) %>%
  count(race)

imprace1 <- countrace1$n/race$n
imprace1 <- data.frame(imprace1)

racedata <- data.frame(race = c('Black/African American', 'European/Caucasian-American', 'Latino/Hispani

imprace2 <- imprace1 %>%
  cbind(racedata) %>%
  ggplot() +
    geom_bar(mapping = aes(x =reorder(race,imprace1), y = imprace1), stat = "identity") +
    labs(
      subtitle="Importance of race > 6",
      x = "Race",
      y = "Percentage of Races"
    ) +
    coord_flip()+
  scale_y_continuous(labels = scales::percent)

imprace2
```




**Which wave has more matches?**

```
countwave <- SpeedDatingNarrow %>%
  count(wave)

countwave1 <- SpeedDatingNarrow %>%
  filter(match == 1) %>%
  count(wave)

wavepercentage <- countwave1$n/countwave$n
wavepercentage <- as.data.frame(wavepercentage)

wavedata <- data.frame(wave = c(1:5,10:21))
wavedata$wave <- as.character(wavedata$wave)

wavepercentage %>%
  cbind(wavedata) %>%
  arrange(desc(wavepercentage)) %>%
  ggplot() +
    geom_bar(mapping = aes(x =reorder(wave,-wavepercentage), y = wavepercentage), stat = "identity") +
    labs(x="Wave",
         y="Percentage of Matches")+
  scale_y_continuous(labels = scales::percent)
```




The result above shows wave 1 has the most matches with over 30% match. The wave with the most participants, wave 14, has around 20% match. We believe the restrictions in different waves have caused the difference.

**Are men more likely to say yes to a second date?**

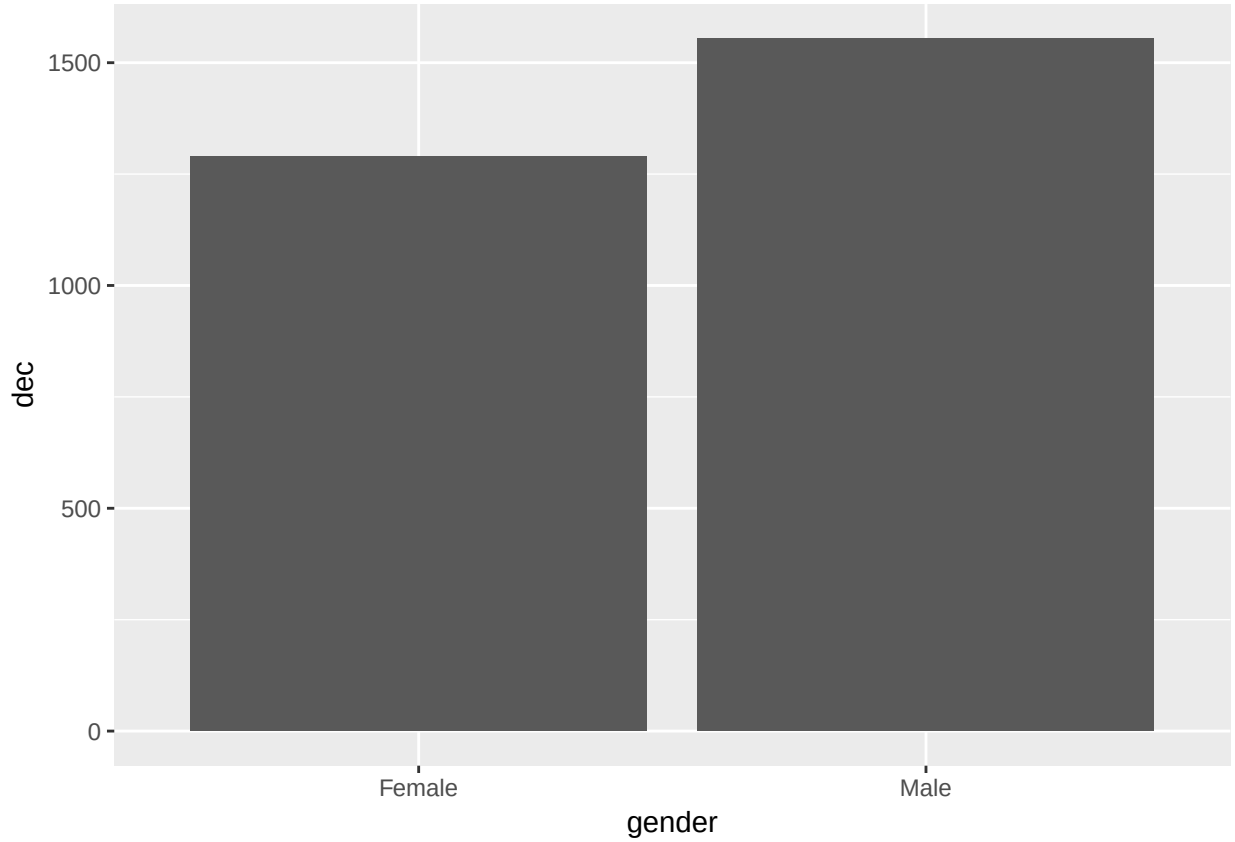
We transformed the gender variable to character variable for ease of analysis. Before computing the result of the question, we computed the number of each genders in the experiment. The population of male participants is slightly larger than that of female participants.

```
SpeedDatingNarrow$gender[SpeedDatingNarrow$gender==0] <- "Female"
SpeedDatingNarrow$gender[SpeedDatingNarrow$gender==1] <- "Male"

# number of males and females
SpeedDatingNarrow %>%
  count(gender)
```

```
## # A tibble: 2 x 2
##   gender     n
##    <chr> <int>
## 1 Female  3403
## 2   Male  3413
```

```
SpeedDatingNarrow$gender <- as.character(SpeedDatingNarrow$gender)

SpeedDatingNarrow %>%
  filter(dec == 1) %>%
```

```
  ggplot() +
  geom_bar(mapping = aes(x = gender, y = dec), stat = "identity")
```




The result shows that male participants are more likely to agree to go on a second date by a 200 difference.

### Regression Data Variable Selection

```
RegressionData <- weighted %>%
  select(gender,income,age,match,dec,WeightedScore,like,prob,met,age_o,dec_o,PartnerWeightedScore,like_

RegressionData
```

```
## # A tibble: 6,816 x 21
##    gender income   age match   dec WeightedScore  like  prob   met age_o
##     <int>  <dbl> <int> <int> <int>         <dbl> <dbl> <dbl> <int> <int>
##  1      0  69487    21     0     1         0.680     7     6     2    27
##  2      0  69487    21     0     1         0.690     7     5     1    22
##  3      0  69487    21     1     1         0.715     7    NA     1    22
##  4      0  69487    21     1     1         0.700     7     6     2    23
##  5      0  69487    21     1     1         0.620     6     6     2    24
##  6      0  69487    21     0     0         0.590     6     5     2    25
##  7      0  69487    21     0     1         0.620     6     5     2    30
##  8      0  69487    21     0     0         0.635     6     7    NA    27
##  9      0  69487    21     1     1         0.760     7     7     2    28
## 10      0  69487    21     0     1         0.705     6     6     2    24
## # ... with 6,806 more rows, and 11 more variables: dec_o <int>,
```

```
## #   PartnerWeightedScore <dbl>, like_o <dbl>, prob_o <dbl>, met_o <int>,
## #   order <int>, int_corr <dbl>, goal <int>, date <int>, imprace <int>,
## #   samerace <int>
```

### Summary of Regression

```
summary(RegressionData)
```

```
##      gender           income            age            match
##  Min.   :0.0000   Min.   :  8607   Min.   :18.00   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.: 31148   1st Qu.:24.00   1st Qu.:0.0000
##  Median :1.0000   Median : 42390   Median :26.00   Median :0.0000
##  Mean   :0.5007   Mean   : 44275   Mean   :26.28   Mean   :0.1649
##  3rd Qu.:1.0000   3rd Qu.: 53940   3rd Qu.:28.00   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :109031   Max.   :55.00   Max.   :1.0000
##                   NA's   :3473     NA's   :90
##       dec         WeightedScore         like             prob
##  Min.   :0.0000   Min.   :0.0000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:0.0000   1st Qu.:0.5750   1st Qu.: 5.000   1st Qu.: 4.000
##  Median :0.0000   Median :0.6700   Median : 6.000   Median : 5.000
##  Mean   :0.4171   Mean   :0.6611   Mean   : 6.124   Mean   : 5.151
##  3rd Qu.:1.0000   3rd Qu.:0.7550   3rd Qu.: 7.000   3rd Qu.: 7.000
##  Max.   :1.0000   Max.   :1.0910   Max.   :10.000   Max.   :10.000
##                   NA's   :1168     NA's   :212      NA's   :270
##       met             age_o           dec_o        PartnerWeightedScore
##  Min.   :0.0000   Min.   :18.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:24.00   1st Qu.:0.0000   1st Qu.:0.5750
##  Median :0.0000   Median :26.00   Median :0.0000   Median :0.6700
##  Mean   :0.8599   Mean   :26.29   Mean   :0.4167   Mean   :0.6612
##  3rd Qu.:2.0000   3rd Qu.:28.00   3rd Qu.:1.0000   3rd Qu.:0.7550
##  Max.   :7.0000   Max.   :55.00   Max.   :1.0000   Max.   :1.0910
##  NA's   :333      NA's   :99                       NA's   :1176
##      like_o           prob_o           met_o           order
##  Min.   : 0.000   Min.   : 0.000   Min.   :1.000   Min.   : 1.000
##  1st Qu.: 5.000   1st Qu.: 4.000   1st Qu.:2.000   1st Qu.: 4.000
##  Median : 6.000   Median : 5.000   Median :2.000   Median : 8.000
##  Mean   : 6.124   Mean   : 5.152   Mean   :1.965   Mean   : 8.919
##  3rd Qu.: 7.000   3rd Qu.: 7.000   3rd Qu.:2.000   3rd Qu.:13.000
##  Max.   :10.000   Max.   :10.000   Max.   :7.000   Max.   :22.000
##  NA's   :222      NA's   :279      NA's   :343
##     int_corr            goal            date          imprace
##  Min.   :-0.7300   Min.   :1.000   Min.   :1.000   Min.   : 0.000
##  1st Qu.:-0.0200   1st Qu.:1.000   1st Qu.:4.000   1st Qu.: 1.000
##  Median : 0.2100   Median :2.000   Median :5.000   Median : 3.000
##  Mean   : 0.1958   Mean   :2.134   Mean   :5.015   Mean   : 3.652
##  3rd Qu.: 0.4300   3rd Qu.:2.000   3rd Qu.:6.000   3rd Qu.: 6.000
##  Max.   : 0.9100   Max.   :6.000   Max.   :7.000   Max.   :10.000
##  NA's   :148       NA's   :74      NA's   :92      NA's   :74
##     samerace
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3903
```

```
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

### Logistic Regression - Individual Selection

We wanted to predict the factors influencing an individual's decision to go on a second date. Since decision (dec) is a binomial variable, we used logistic regression to predict the outcome.

```
attach(RegressionData)
Regression.Formula <- dec~gender+income+age+WeightedScore+like+prob+met+age_o+PartnerWeightedScore+like_

prob.DecMatch<- glm(Regression.Formula,
                family = binomial(link="logit"),  data=RegressionData)
summary(prob.DecMatch)
```

```
##
## Call:
## glm(formula = Regression.Formula, family = binomial(link = "logit"),
##     data = RegressionData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8343  -0.7493  -0.2538   0.7582   2.3961
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -7.827e+00  9.142e-01  -8.561  < 2e-16 ***
## gender                3.772e-01  1.186e-01   3.181 0.001468 **
## income                6.286e-06  3.111e-06   2.020 0.043359 *
## age                  -6.700e-02  1.725e-02  -3.885 0.000102 ***
## WeightedScore         5.534e+00  6.680e-01   8.285  < 2e-16 ***
## like                  6.450e-01  5.509e-02  11.709  < 2e-16 ***
## prob                  9.006e-02  2.947e-02   3.057 0.002239 **
## met                  -9.454e-03  5.885e-02  -0.161 0.872365
## age_o                 1.280e-03  1.690e-02   0.076 0.939635
## PartnerWeightedScore -8.602e-01  5.898e-01  -1.459 0.144695
## like_o               -1.075e-01  4.931e-02  -2.180 0.029289 *
## prob_o                1.364e-01  3.075e-02   4.434 9.23e-06 ***
## met_o                 3.394e-01  2.273e-01   1.493 0.135425
## order                 6.687e-03  1.020e-02   0.655 0.512156
## int_corr             -1.805e-01  1.848e-01  -0.977 0.328669
## goal                  5.118e-02  3.862e-02   1.325 0.185091
## date                  3.233e-02  3.880e-02   0.833 0.404772
## imprace              -6.294e-02  1.903e-02  -3.307 0.000942 ***
## samerace              3.167e-01  1.143e-01   2.771 0.005589 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2979.4  on 2212  degrees of freedom
## Residual deviance: 2038.8  on 2194  degrees of freedom
##   (4603 observations deleted due to missingness)
```

```
## AIC: 2076.8
##
## Number of Fisher Scoring iterations: 5
```

The results above show that gender, age, Weighted Score, like, prob, prob_o, like_o, imprace, and samerace are significant variables for prediction. This means that how much the participant likes the partner and how much the participant thinks the partner like him back are important, and also the less a participant cares about race, the more likely he/she'll agree to go on a second date.

### Logistic Regression - Match

We're also interested about the factors influencing the likelihood of getting a match. We used logistic regression since match is a binomial variable.

```
Regression.Formula.match <- match~RegressionData$gender+RegressionData$income+RegressionData$age+Regres
```

```
prob.DecMatch2<- glm(Regression.Formula.match,
                family = binomial(link="logit"),  data=RegressionData)
summary(prob.DecMatch2)
```

```
##
## Call:
## glm(formula = Regression.Formula.match, family = binomial(link = "logit"),
##     data = RegressionData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6480  -0.5653  -0.3182  -0.1325   2.9686
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -9.831e+00  1.134e+00  -8.673  < 2e-16
## RegressionData$gender              2.205e-01  1.420e-01   1.553  0.12046
## RegressionData$income             -1.258e-06  3.736e-06  -0.337  0.73624
## RegressionData$age                -5.051e-02  2.150e-02  -2.349  0.01883
## RegressionData$WeightedScore       4.397e+00  7.897e-01   5.568 2.58e-08
## RegressionData$like                2.940e-01  6.447e-02   4.561 5.10e-06
## RegressionData$prob                1.059e-01  3.428e-02   3.088  0.00202
## RegressionData$met                -3.371e-02  7.228e-02  -0.466  0.64091
## RegressionData$age_o              -2.076e-02  2.050e-02  -1.012  0.31135
## RegressionData$PartnerWeightedScore  1.726e+00  7.679e-01   2.247  0.02462
## RegressionData$like_o              3.681e-01  6.824e-02   5.395 6.87e-08
## RegressionData$prob_o              1.071e-01  3.592e-02   2.982  0.00286
## RegressionData$met_o               1.403e-01  2.731e-01   0.514  0.60735
## RegressionData$order               1.392e-02  1.241e-02   1.121  0.26213
## RegressionData$int_corr           -1.348e-01  2.185e-01  -0.617  0.53715
## RegressionData$goal               -2.809e-02  4.512e-02  -0.623  0.53361
## RegressionData$date                3.382e-03  4.592e-02   0.074  0.94128
## RegressionData$imprace            -4.665e-02  2.265e-02  -2.060  0.03944
## RegressionData$samerace            1.542e-01  1.355e-01   1.138  0.25521
##
## (Intercept)                       ***
## RegressionData$gender
## RegressionData$income
## RegressionData$age                *
```

```
## RegressionData$WeightedScore        ***
## RegressionData$like                 ***
## RegressionData$prob                 **
## RegressionData$met
## RegressionData$age_o
## RegressionData$PartnerWeightedScore *
## RegressionData$like_o               ***
## RegressionData$prob_o               **
## RegressionData$met_o
## RegressionData$order
## RegressionData$int_corr
## RegressionData$goal
## RegressionData$date
## RegressionData$imprace              *
## RegressionData$samerace
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2097.4  on 2212  degrees of freedom
## Residual deviance: 1522.6  on 2194  degrees of freedom
##   (4603 observations deleted due to missingness)
## AIC: 1560.6
##
## Number of Fisher Scoring iterations: 6
```

The model shows hat age, weighted score, like, prob, like_o, prob_o, and imprace are significant variables in the prediction. Race has less effect in getting a match compared to agreeing to go on a second date, therefore, the most important factors that contribute to a match is how much the pair likes each other.

### System Information

```
sessionInfo()
```

```
## R version 3.4.1 (2017-06-30)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 16299)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2    stargazer_5.2   ggthemes_3.4.0  scales_0.5.0
##  [5] purrr_0.2.3     readr_1.1.1     tibble_1.3.4    ggplot2_2.2.1
```

```
##  [9] tidyverse_1.1.1 tidyr_0.7.1     dplyr_0.7.4
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.12     cellranger_1.1.0 compiler_3.4.1   plyr_1.8.4
##  [5] bindr_0.1        forcats_0.2.0    tools_3.4.1      digest_0.6.12
##  [9] lubridate_1.6.0  jsonlite_1.5     evaluate_0.10.1  nlme_3.1-131
## [13] gtable_0.2.0     lattice_0.20-35  pkgconfig_2.0.1  rlang_0.1.2
## [17] psych_1.7.8      yaml_2.1.14      parallel_3.4.1   haven_1.1.0
## [21] xml2_1.1.1       httr_1.3.1       stringr_1.2.0    knitr_1.17
## [25] hms_0.3          rprojroot_1.2    grid_3.4.1       glue_1.1.1
## [29] R6_2.2.2         readxl_1.0.0     foreign_0.8-69   rmarkdown_1.6
## [33] modelr_0.1.1     reshape2_1.4.2   magrittr_1.5     backports_1.1.0
## [37] htmltools_0.3.6  rvest_0.3.2      assertthat_0.2.0 mnormt_1.5-5
## [41] colorspace_1.3-2 labeling_0.3     stringi_1.1.5    lazyeval_0.2.0
## [45] munsell_0.4.3    broom_0.4.2
```