Speed Dating R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Importing and Tidying the Data Set

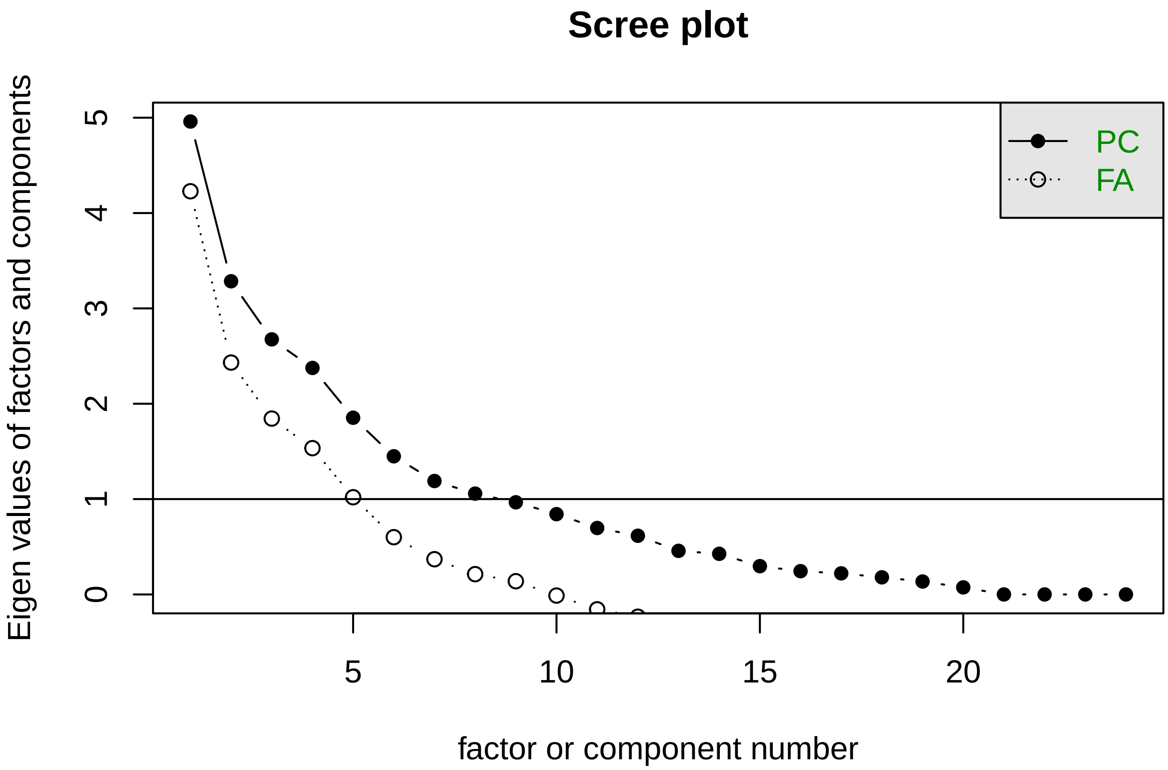
We imported libraries typically needed in this course and importing the Speed Dating Data Set. We decided to filter only Waves 6-9 because participants in those waves used a 1-10 scale and not a 1-100 scale like the other waves. After this we then divided up the observations into groups we hoped would later help answer our research questions. After our initial cleaning and exploration of the data set, we chose 3 attributes to study that we thought might yield some latent variables and fit into a cfa model. Those were attraction, fun, and sharing. Within these attributes we were looking at ratings of what people prioritized, and what they thought that the opposite sex prioritized. We thought those who prioritized a certain attribute would think that the other sex would also prioritize that attribute. After using a scree plot and determining that a three factor model was indeed best, we ran an efa and investigated the factor loadings. They did not come out cleanly as we had hoped, with the different attribute prioritizations lining up together. Due to the mix of positive and negative factor loadings as well as it not lining up with our predicted model, we decided to scrap this 3 attribute model and look at other attribute statistics to find some latent variables.

Research Questions:

What attributes (as rated by your partner) are most predictive of matching? Do these differ between genders? Do ratings of important attributes in a match fall into broader categories of prioritization? Do hobby ratings fall into broader categories of interest areas?

```
library(tidyverse)
library(Rcpp)
library(psych)
library(yarrr)
library(lavaan)
library(GPArotation)

speed <- read_csv("Speed Dating Data.csv") %>%
  filter(wave %in% c("6", "7", "8", "9"))


#get overall values to see how people value importance in the 6 factors in others and how they think ot
```

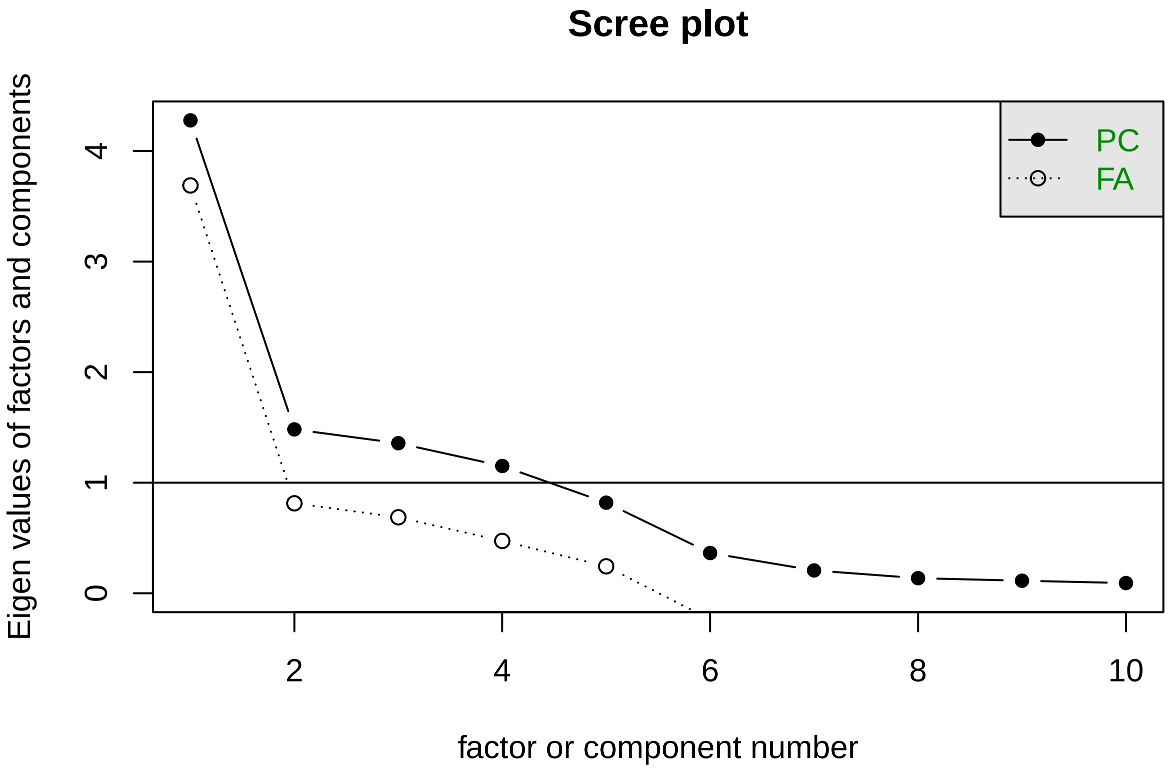
```r
values <- speed %>%
  select(iid, gender, pid, match, int_corr,attr1_1, sinc1_1, intel1_1, shar1_1, amb1_1, fun1_1, attr2_1
         attr1_2, sinc1_2, intel1_2, shar1_2, amb1_2, fun1_2, attr2_2, sinc2_2, intel2_2, shar2_2, amb2
  filter(!is.na(attr2_2))

#getting just the factors and getting rid of NAs in order to do statistical analysis (see below)

values_factor <- values %>%
  select(attr1_1, sinc1_1, intel1_1, shar1_1, amb1_1, fun1_1, attr2_1, sinc2_1, intel2_1, shar2_1, amb2
         attr1_2, sinc1_2, intel1_2, shar1_2, amb1_2, fun1_2, attr2_2, sinc2_2, intel2_2, shar2_2, amb2
  filter(!is.na(attr2_2))


#observe how many latent variables are possible for this data set through an EFA

scree(values_factor)
```



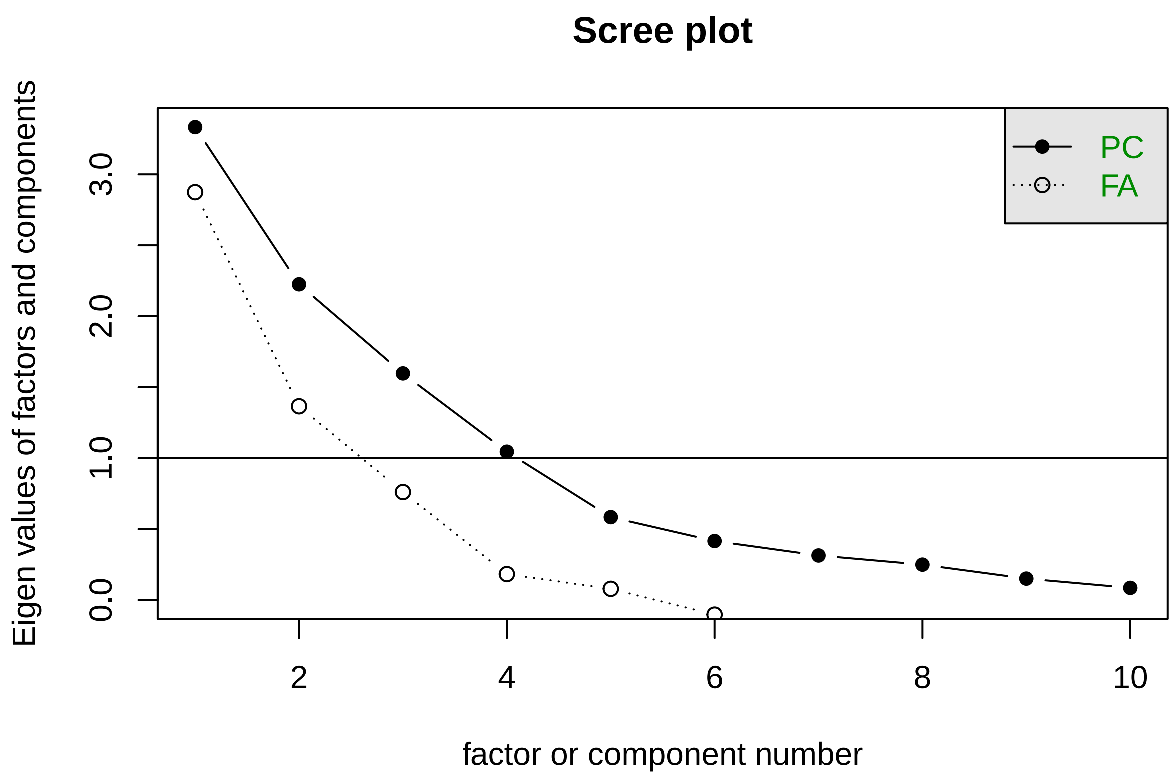

```r
values_oblique <- fa(values_factor,
                     nfactors = 6,
                     fm = "minres",
                     rotate = "oblimin")

values_oblique
```

```r
#get values for just how the participant feels about themself and if they got a match

self <- speed %>%
  select(iid, attr3_1, attr3_2, fun3_1, fun3_2, intel3_1, intel3_2, amb3_1, amb3_2, sinc3_1, sinc3_2) %
  filter(!is.na(attr3_2))
```

```r
self_factor <- self %>%
  select(-iid)

scree(self_factor)
```




```r
#values for how participant thinks about others when looking for a partner

others <- speed %>%
  select(iid, attr1_1, attr1_2, fun1_1, fun1_2, intel1_1, intel1_2, amb1_1, amb1_2, sinc1_1, sinc1_2) %>%
    filter(!is.na(attr1_2))

others_factor <- others %>%
  select(-iid)

scree(others_factor)
```




```
others_clean <- others %>%
  pivot_longer(2:11, names_to = "attr",
               values_to = "score") %>%
  group_by(iid, attr) %>%
  summarize()
```

Self Ratings vs. Match Visualizations

Along with our factor analyses, we were also interested in what attributes were most important in getting a match. We took people's self ratings of their own attributes and compared the groups that both did and did not match. We recognize that self-ratings are not entirely accurate measures, however the participants had a limited number of points to give themselves on all the attributes, which limits the self-valuation bias inaccuracies. We created pirateplots and found probability of superiority statistics for the attributes of Attractiveness, Fun, Intelligence, Ambition, and Sincerity. The largest differences were found in Fun and Attractiveness, which had probabilities of superiority in the matching group of 54% and 53% respectively. These differences were significant, while the differences in the other groups were even smaller with confidence interval overlap. Although 54% and 53% are small probabilities of superiority, the large sample size that was drawn from makes them reliable nonetheless.

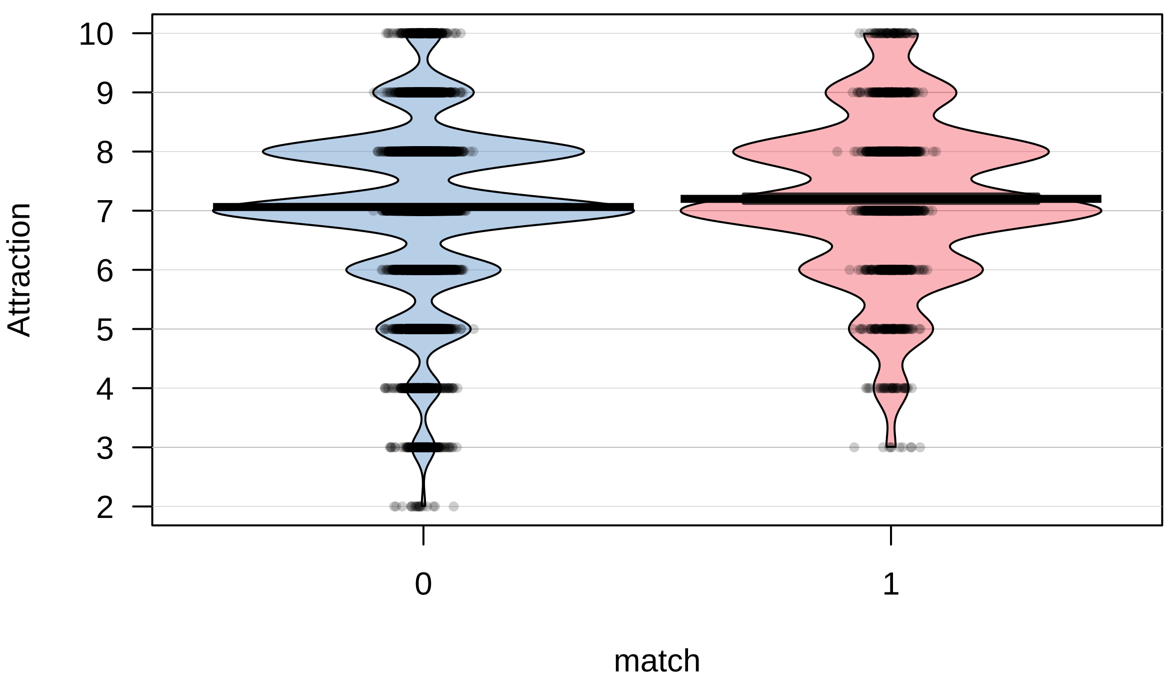
Gender Differences

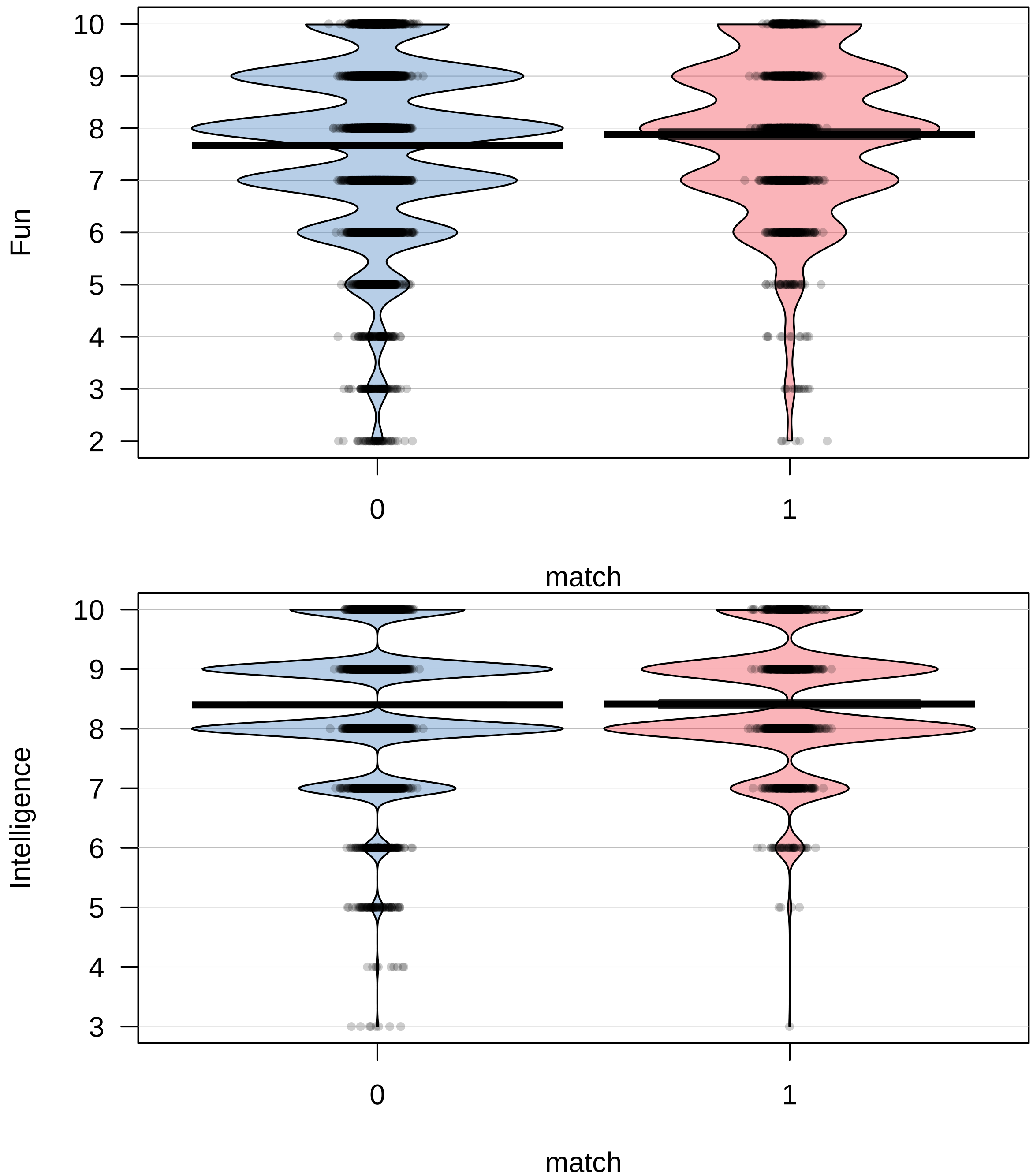
We also wanted to see if rating effects on matching were mediated by gender in any way. We found that the difference between fun and matching for varied by gender. There was no significant difference in matching between women's self ratings of fun, however an even greater effect was found among men than was found in the full group. Ambition had a 50% probability of superiority before being broken down by gender, which indicated that ambition had no effect on matching. However, when broken down by gender, it became clear that rating yourself higher in ambition as a woman lowers her chance of a match, while a man rating himself higher in ambition increases his chance of a match. These anaylses show that while generally it is beneficial for matching to consider oneself high in attractiveness and fun, that that doesn't tell the whole picture among different groups.

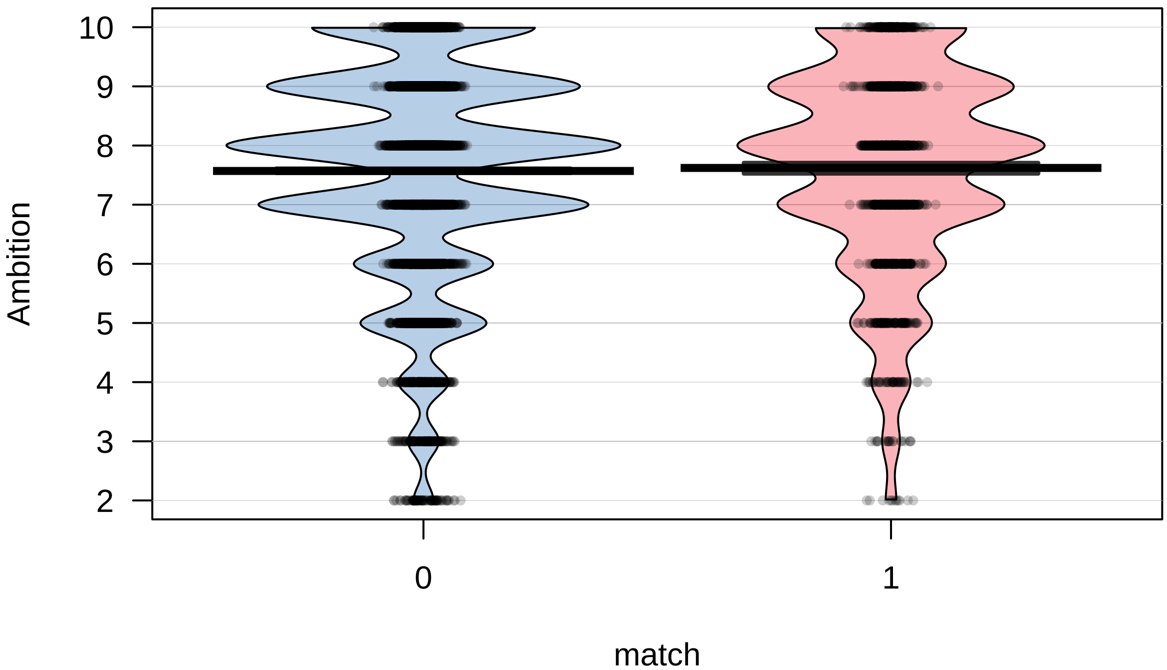
Correlations and GGplot

Considering the fact that attraction and fun were most associated with matching out of all the variables and that both were more associated with matching in men than in women, I thought that their self valuations might be related. It seemed possible that people who rate themselves as fun are more likely to also rate themselves as attractive. I ran a regression with fun ratings predicting attractiveness ratings. The multiple R-squared was .1918, meaning that 19% of the variation in attractiveness is explained by fun. Considering the size of the standardized beta nd multiple R-squared, I ran a correlation between all of the attributes to see if there were strong relationships other than between fun and attractiveness. Fun and Attractiveness had the highest positive correlation of .44, but all of the ratings were all at least slightly positively correlated with one another. The next highest was intelligence and attractiveness with a positive correlation of .37. I created a visualization of the relationships between these factors using a network_plot.

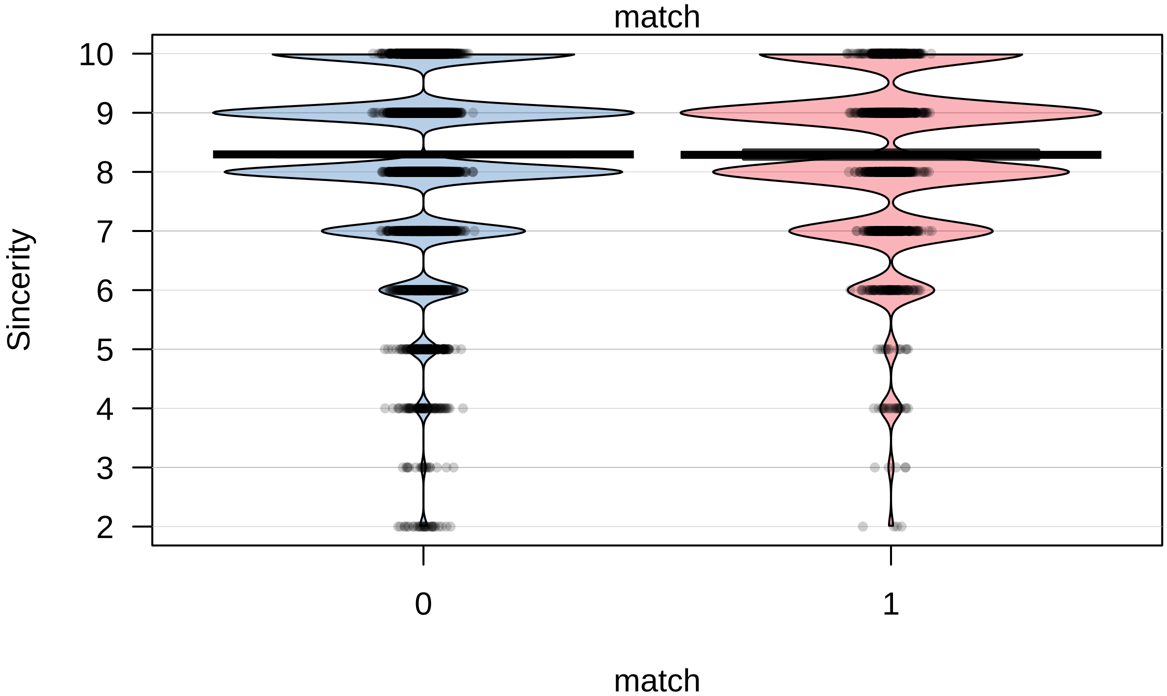
```
## Rows: 8378 Columns: 195
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): field, undergra, from, career
## dbl (187): iid, id, gender, idg, condtn, wave, round, position, positin1, or...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```



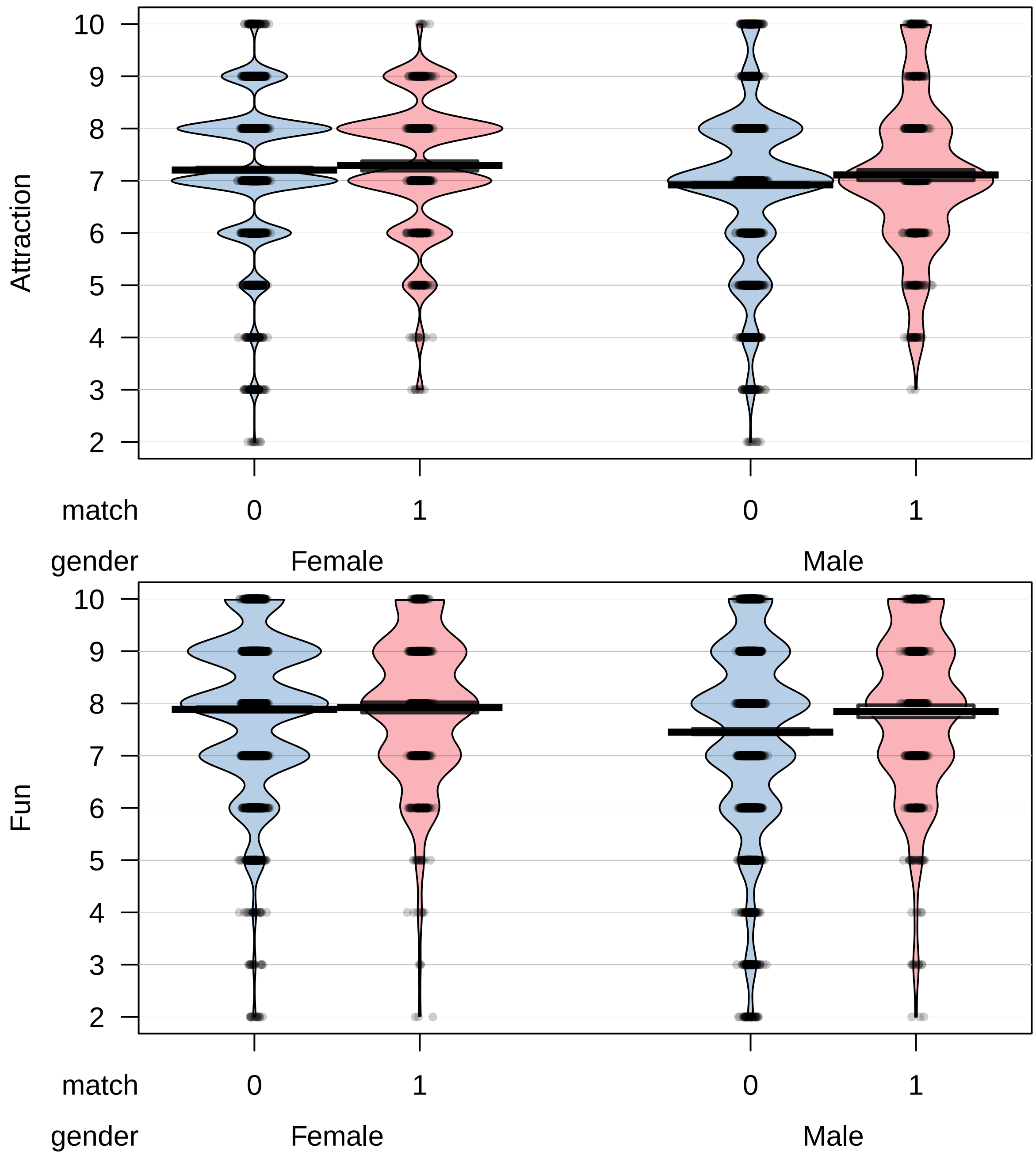

Fun
10
9
8
7
6
5
4
3
2
0
1
match
Intelligence
10
9
8
7
6
5
4
3
0
1
match

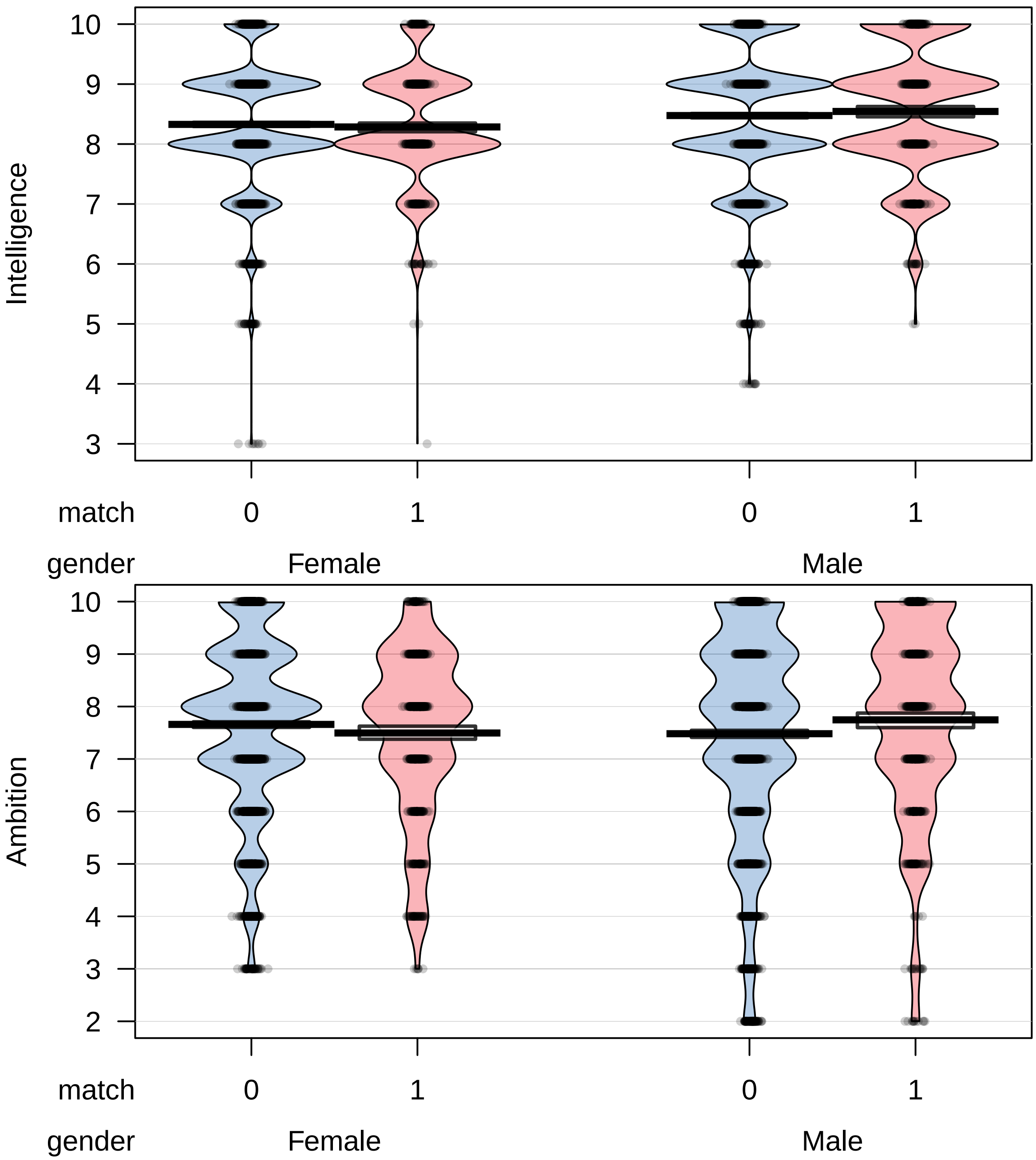


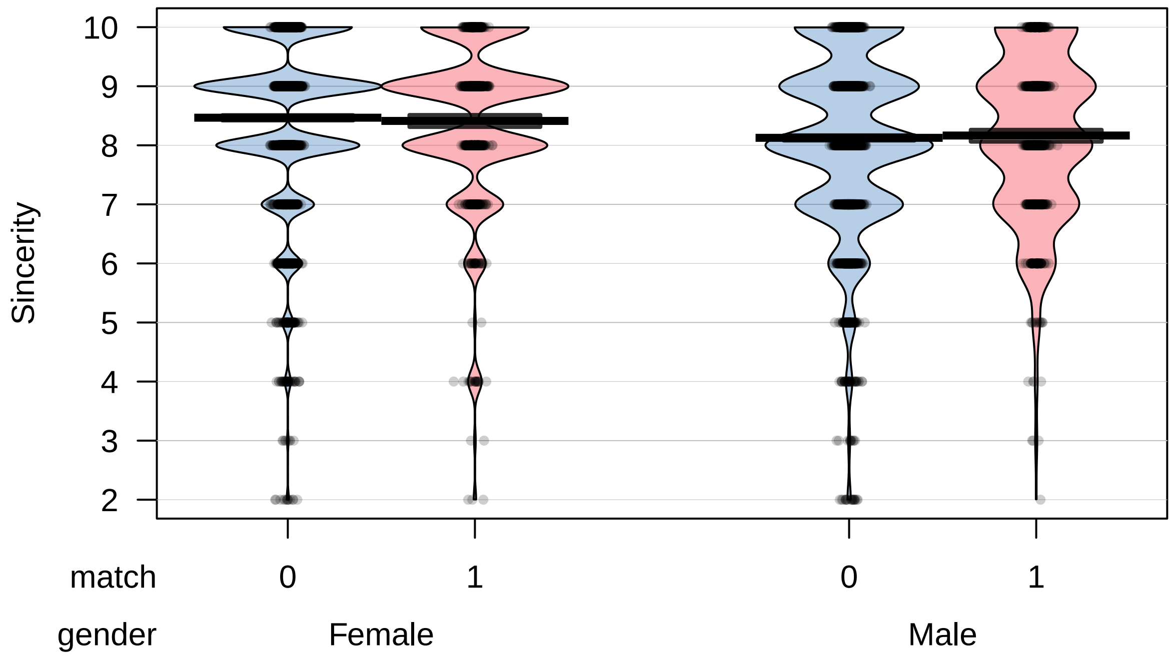
Ambition
10
9
8
7
6
5
4
3
2
0
1
match



Sincerity
10
9
8
7
6
5
4
3
2
0
1
match


Attraction
10
9
8
7
6
5
4
3
2
match 0 1 0 1
gender Female Male
Fun
10
9
8
7
6
5
4
3
2
match 0 1 0 1
gender Female Male


Intelligence
10
9
8
7
6
5
4
3
match
0
1
0
1
gender
Female
Male
Ambition
10
9
8
7
6
5
4
3
2
match
0
1
0
1
gender
Female
Male



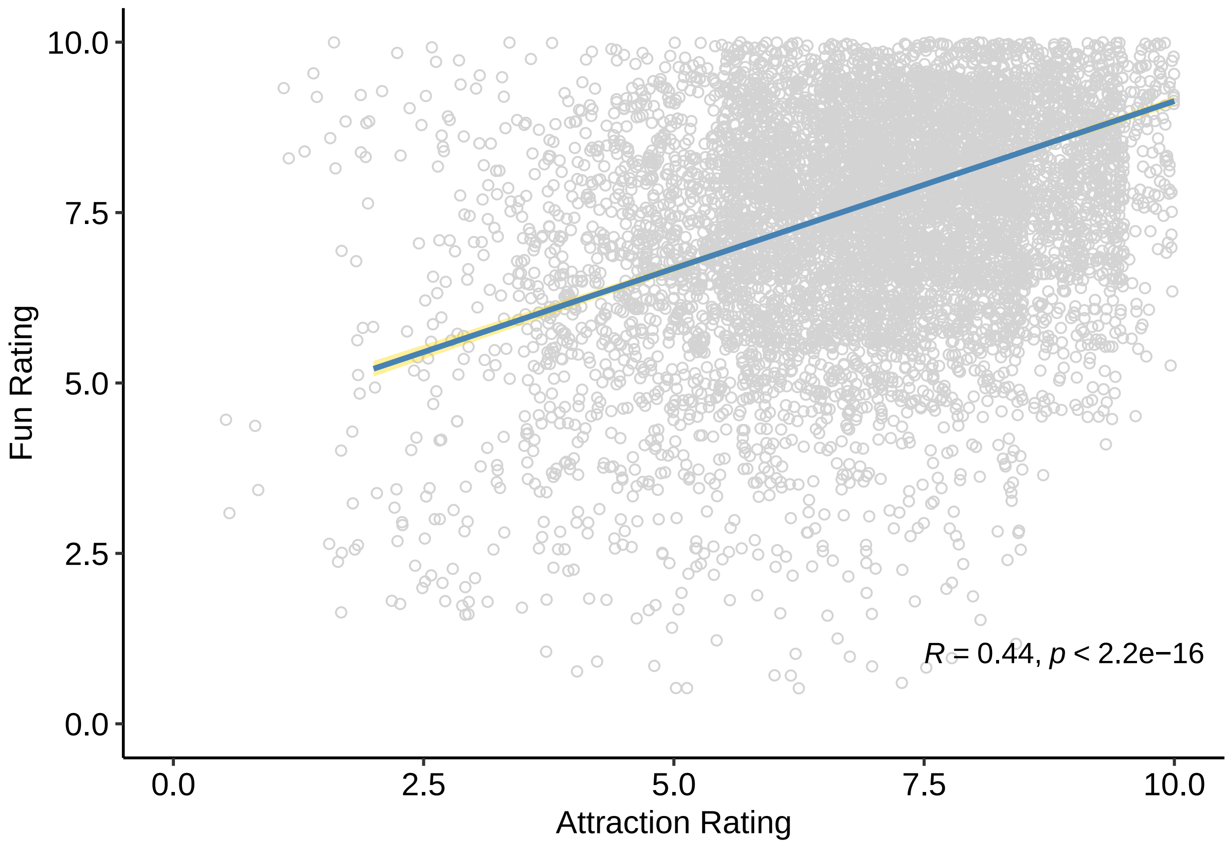

```
##
## Attaching package: 'effectsize'
##
## The following object is masked from 'package:psych':
##
##     phi

## Pr(superiority) |       95% CI
## ------------------------------
## 0.47            | [0.46, 0.49]

## Pr(superiority) |       95% CI
## ------------------------------
## 0.46            | [0.44, 0.48]

## Pr(superiority) |       95% CI
## ------------------------------
## 0.50            | [0.48, 0.51]

## Pr(superiority) |       95% CI
## ------------------------------
## 0.49            | [0.48, 0.51]

## Pr(superiority) |       95% CI
## ------------------------------
## 0.50            | [0.49, 0.52]

##
## Attaching package: 'table1'
##
## The following objects are masked from 'package:base':
##
##     units, units<-

##
## Call:
```

```
## lm(formula = Attraction ~ Fun, data = all2)
##
## Coefficients:
## (Intercept)          Fun
##      4.0744       0.3907
```

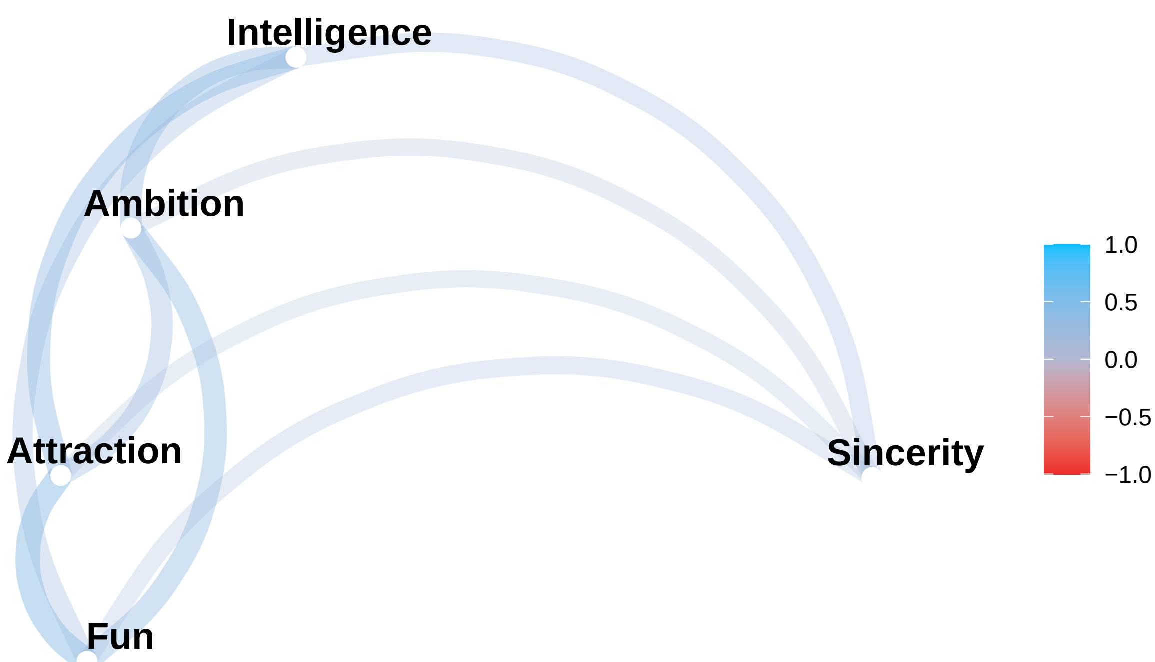
```
##
## Call:
## lm(formula = Attraction ~ Fun, data = all2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5909 -0.5909  0.1905  0.7998  3.5812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.07442    0.06934   58.76   <2e-16 ***
## Fun          0.39072    0.00882   44.30   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.255 on 8271 degrees of freedom
##   (105 observations deleted due to missingness)
## Multiple R-squared:  0.1918, Adjusted R-squared:  0.1917
## F-statistic:  1962 on 1 and 8271 DF,  p-value: < 2.2e-16
```

```
##
##
## Regression results using Attraction as the criterion
##
##
##    Predictor      b     b_95%_CI beta  beta_95%_CI sr2 sr2_95%_CI     r
##  (Intercept) 4.07** [3.94, 4.21]
##          Fun 0.39** [0.37, 0.41] 0.44 [0.42, 0.46] .19 [.18, .21] .44**
##
##
##
##             Fit
##
##
##      R2 = .192**
##  95% CI[.18,.21]
##
##
## Note. A significant b-weight indicates the beta-weight and semi-partial correlation are also signific
## b represents unstandardized regression weights. beta indicates the standardized regression weights.
## sr2 represents the semi-partial correlation squared. r represents the zero-order correlation.
## Square brackets are used to enclose the lower and upper limits of a confidence interval.
## * indicates p < .05. ** indicates p < .01.
##
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
##
## Correlation method: 'pearson'
## Missing treated using: 'pairwise.complete.obs'
```



First attempt at a CFA model

Our first CFA model based off our initial presentation idea was to see if we could condense the attribute groupings to make a less "busy" model. However, once we ran this model we soon learned that this would not be statistically feasible because the values were no where close to what is needed for a parsimonious model. Because of this we decided to go into another direction and look at the means of each attribute without putting them into a model to see how people on average value these traits in a relationship.

```
#CFA for self and others data sets

library(tidyverse)
library(lavaan)

Traits.model <-
'
extraversion =~ fun1_1 + fun2_1 + attr1_1 + attr2_1
status =~ intel1_1 + intel2_1 + amb1_1 + amb2_1
similar =~ sinc1_1 + sinc2_1 + shar1_1 + shar2_1

'

#look at stat summary for this CFA model. From the results we came to the realization that this model w

fit <- cfa(Traits.model, data = speed)
summary(fit, fit.measures=TRUE)
lavInspect(fit, what = "std")
```

What each participant looks for (attributes wise)

After some debate and brainstorming, we decided on a 3 factor model that would combine attribute ratings that we predicted would correlate. Extraversion, we decided, was latent in the fun and attraction measures, while status comes from high intelligence and ambition valuation and similarity prioritization comes from sincerity and sharing valuation. Once we had determined that this was our cfa model that we would investigate, we encountered a data wrangling issue. There were many repeat scores for the same participant that went along with each person they went on a speed date with. Some participants had more duplicates than others and we wanted to prevent this from skewing our data. So, we pivoted the data wider to remove the excess repeats from each variable, then pivoted it back longer and joined the data frames together so that there was just 1 observation for every participant for every attribute valuation rating.

```
#get values for just self and if they got a match

tot <- read_csv("Speed Dating Data.csv") %>%
  select(iid, attr1_1, attr2_1, fun1_1, fun2_1, intel1_1, intel2_1, amb1_1, amb2_1, sinc1_1, sinc2_1)

self <- speed %>%
  select(iid, attr3_1, attr3_2, fun3_1, fun3_2, intel3_1, intel3_2, amb3_1, amb3_2, sinc3_1, sinc3_2) %
  filter(!is.na(attr3_2))


others <- speed %>%
  select(iid, partner, attr1_1, attr1_2, fun1_1, fun1_2, intel1_1, intel1_2, amb1_1, amb1_2, sinc1_1, s
  filter(!is.na(attr1_1))
```

```r
attr_wide <- others %>%
  pivot_wider(names_from = "iid", values_from = "attr1_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

attr <- attr_wide %>%
  pivot_longer(2:102, names_to = "iid", values_to = "attr1_1") %>%
  select(-partner)

fun_wide <- others %>%
  pivot_wider(names_from = "iid", values_from = "fun1_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

fun <- fun_wide %>%
  pivot_longer(2:102, names_to = "iid", values_to = "fun1_1") %>%
  select(-partner)

intel_wide <- others %>%
  pivot_wider(names_from = "iid", values_from = "intel1_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

intel <- intel_wide %>%
  pivot_longer(2:102, names_to = "iid", values_to = "intel1_1") %>%
  select(-partner)

amb_wide <- others %>%
  pivot_wider(names_from = "iid", values_from = "amb1_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

amb <- amb_wide %>%
  pivot_longer(2:102, names_to = "iid", values_to = "amb1_1") %>%
  select(-partner)

sinc_wide <- others %>%
  pivot_wider(names_from = "iid", values_from = "sinc1_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

sinc <- sinc_wide %>%
  pivot_longer(2:102, names_to = "iid", values_to = "sinc1_1") %>%
  select(-partner)

shar_wide <- others %>%
  pivot_wider(names_from = "iid", values_from = "shar1_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

shar <- shar_wide %>%
  pivot_longer(2:102, names_to = "iid", values_to = "shar1_1") %>%
  select(-partner)

self_eval <- full_join(fun, attr) %>%
  full_join(amb) %>%
  full_join(sinc) %>%
  full_join(shar) %>%
  full_join(intel)
```

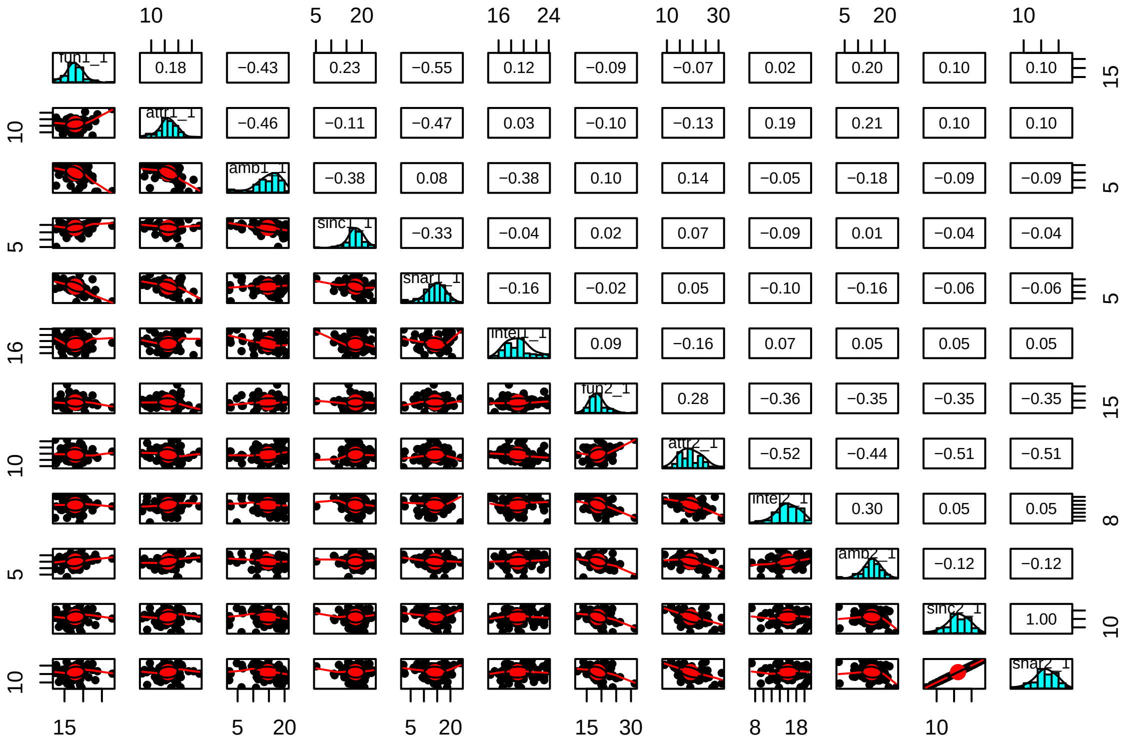
What you think opposite sex looks for (attributes wise)

In our cfa model, we wanted to include not just what individuals looked for in the opposite sex, but what they thought the opposite sex looked for. So, we repeated the pivoting data wrangling process for those measures as well.

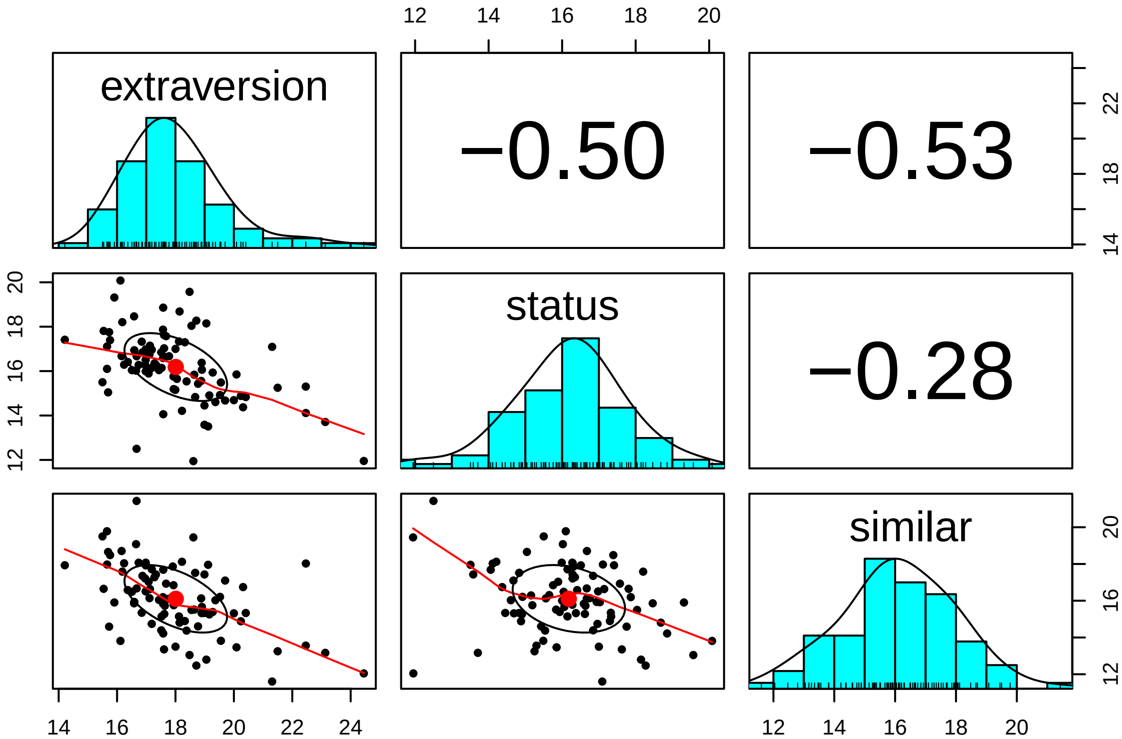
```
others_2 <- speed %>%
  select(iid, partner, attr1_1, attr2_1, attr1_2, fun2_1, fun1_2, intel2_1, intel1_2, amb2_1, amb1_2, s
  filter(!is.na(attr1_2))



attr_wide_2 <- others_2 %>%
  pivot_wider(names_from = "iid", values_from = "attr2_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

attr_2 <- attr_wide_2 %>%
  pivot_longer(2:92, names_to = "iid", values_to = "attr2_1") %>%
  select(-partner)

fun_wide_2 <- others_2 %>%
  pivot_wider(names_from = "iid", values_from = "fun2_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

fun_2 <- fun_wide_2 %>%
  pivot_longer(2:92, names_to = "iid", values_to = "fun2_1") %>%
  select(-partner)

intel_wide_2 <- others_2 %>%
  pivot_wider(names_from = "iid", values_from = "intel2_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

intel_2 <- intel_wide_2 %>%
  pivot_longer(2:92, names_to = "iid", values_to = "intel2_1") %>%
  select(-partner)

amb_wide_2 <- others_2 %>%
  pivot_wider(names_from = "iid", values_from = "amb2_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

amb_2 <- amb_wide_2 %>%
  pivot_longer(2:92, names_to = "iid", values_to = "amb2_1") %>%
  select(-partner)

sinc_wide_2 <- others_2 %>%
  pivot_wider(names_from = "iid", values_from = "sinc2_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)

sinc_2 <- sinc_wide_2 %>%
  pivot_longer(2:92, names_to = "iid", values_to = "sinc2_1") %>%
  select(-partner)

shar_wide_2 <- others_2 %>%
  pivot_wider(names_from = "iid", values_from = "shar2_1", id_cols = "partner", values_fn = list) %>%
  filter(partner == 1)
```

```r
shar_2 <- sinc_wide_2 %>%
  pivot_longer(2:92, names_to = "iid", values_to = "shar2_1") %>%
  select(-partner)

oppsex_eval <- full_join(fun_2, attr_2) %>%
  full_join(intel_2) %>%
  full_join(amb_2) %>%
  full_join(sinc_2) %>%
  full_join(shar_2)
```

```r
#joining the two data sets in order for further statistical evaluation

no_wammies <- full_join(self_eval, oppsex_eval, by = c("iid"))

no_wammies2 <- no_wammies %>%
  filter(attr2_1 != "NULL") %>%
  mutate(attr2_1 = as.numeric(attr2_1)) %>%
  mutate(attr1_1 = as.numeric(attr1_1)) %>%
  mutate(fun2_1 = as.numeric(fun2_1)) %>%
  mutate(fun1_1 = as.numeric(fun1_1)) %>%
  mutate(amb2_1 = as.numeric(amb2_1)) %>%
  mutate(amb1_1 = as.numeric(amb1_1)) %>%
  mutate(sinc2_1 = as.numeric(sinc2_1)) %>%
  mutate(sinc1_1 = as.numeric(sinc1_1)) %>%
  mutate(intel2_1 = as.numeric(intel2_1)) %>%
  mutate(intel1_1 = as.numeric(intel1_1)) %>%
    mutate(shar2_1 = as.numeric(shar2_1)) %>%
  mutate(shar1_1 = as.numeric(shar1_1)) %>%
  select(-iid)


summary(no_wammies2)


pairs.panels(no_wammies2)
```



```
#create a new CFA model based on newly updated traits. Still was not very good/significant but we were 

library(tidyverse)
library(lavaan)

Traits.model <-
'
extraversion =~ fun1_1 + fun2_1 + attr1_1 + attr2_1
status =~ intel1_1 + intel2_1 + amb1_1 + amb2_1
similar =~ sinc1_1 + sinc2_1 + shar1_1 + shar2_1

'


#fit <- cfa(Traits.model, data = no_wammies2)
summary(fit, fit.measures=TRUE)
lavInspect(fit, what = "std")



#get a mean for each attribute based on participant and see the correlations between these values
means <- no_wammies2 %>%
  mutate(extraversion = (attr1_1 + attr2_1 + fun1_1 + fun2_1) / 4) %>%
  mutate(status = (intel1_1 + intel2_1 + amb1_1 + amb2_1) / 4) %>%
  mutate(similar = (shar1_1 + shar2_1 + sinc1_1 + sinc2_1) / 4) %>%
  select(extraversion, status, similar)

pairs.panels(means)
```



Multiple Regressions for CFA model

```
#multiple regression which further shows this model is not the right fit for this data set and what we

library(lm.beta)

multireg1 <- lm.beta(lm(extraversion ~ status + similar, data = means))
summary(multireg1)

multireg2 <- lm.beta(lm(status ~ extraversion + similar, data = means))
summary(multireg2)

multireg3 <- lm.beta(lm(similar ~ extraversion + status, data = means))
summary(multireg3)

#look for unique variance


extrastat_u <- summary(lm(extraversion ~ status + similar, data = means))$r.squared -
              summary(lm(extraversion ~ status, data = means))$r.squared

extrasim_u <- summary(lm(extraversion ~ status + similar, data = means))$r.squared -
              summary(lm(extraversion ~ similar, data = means))$r.squared




statusextra_u <- summary(lm(status ~ extraversion + similar, data = means))$r.squared -
             summary(lm(status ~ extraversion, data = means))$r.squared
```

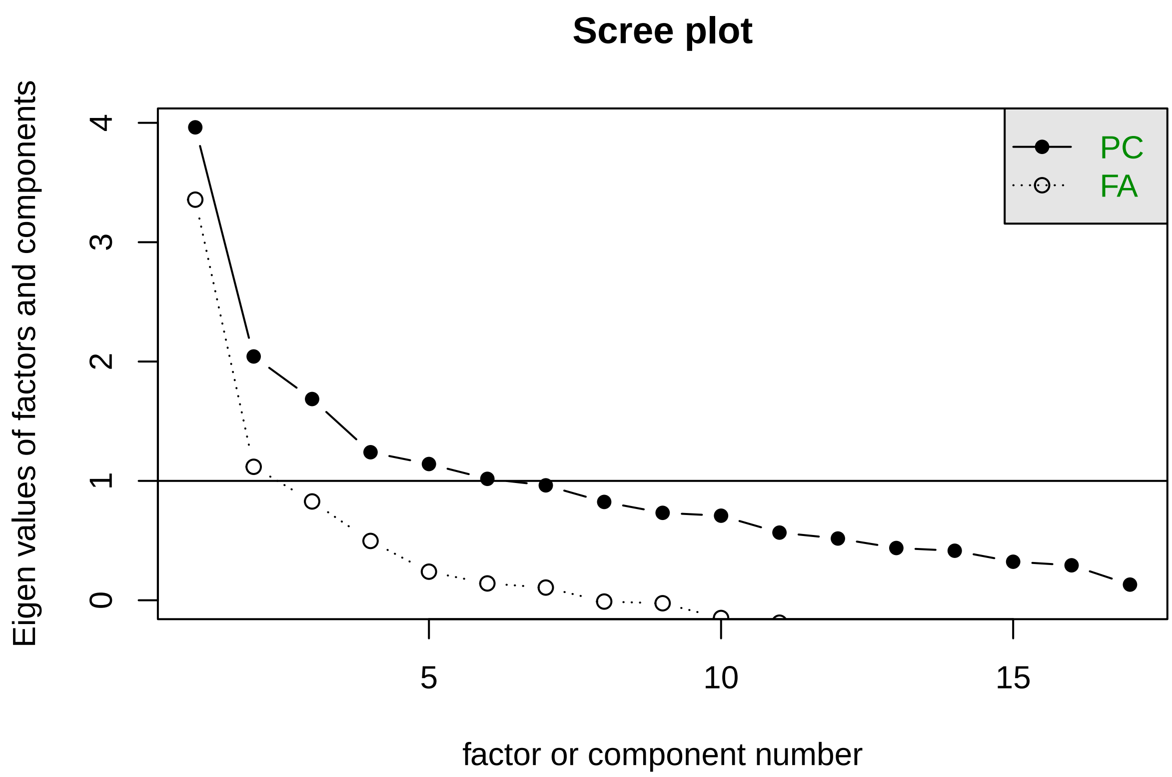
```
statussim_u <- summary(lm(status ~ extraversion + similar, data = means))$r.squared -
             summary(lm(status ~ similar, data = means))$r.squared



similarextra_u <- summary(lm(similar ~ extraversion + status, data = means))$r.squared -
             summary(lm(similar ~ extraversion, data = means))$r.squared


similarstat_u <- summary(lm(similar ~ extraversion + status, data = means))$r.squared -
             summary(lm(similar ~ status, data = means))$r.squared
```

New Direction: Looking into the section on hobbies

We (with the help of Professor Johnson) decided that we may have better luck finding a more parsimonious CFA model with the hobbies section of the data set. After looking into these hobbies it could be seen that they could be put in fewer groups than just the 17 separate items. For example, the hobbies museums, art, and theater all correlate very high with each other so this could be condensed into a singular latent variable called "The arts." Another group of hobbies that all correlated with each other was sports, TV sports, and exercising which could be placed into a singular category of "Physical activity" or "Sporty." For further interpretation of the models we choose and how we choose them look below.

#hobbby CFA We tried to upload our final CFA model from our Final Presentation but it would not let us knit our code with the image.

```
new <- read.csv("Speed Dating Data.csv") %>%
  select( iid, match, sports, tvsports, exercise, dining, museums, art, hiking, gaming, clubbing, readi

new_stats <- new %>%
  select(-iid, -match)

scree(new_stats)
```




```
#look to see which items correlate with each other in a 3 factor model
hobbies_fa <- fa(new_stats,
                 nfactors = 3,
                 fm = "minres",
                     rotate = "oblimin")

hobbies_fa
hobbies_fa$loadings

library(tidyverse)
library(lavaan)

Hobbies.Model <-

'
arts =~ museums + art + reading + theater + concerts + music + yoga + dining
sport =~ sports + tvsports + exercise + hiking + gaming
games =~ clubbing + tv + movies + shopping

'


fit <- cfa(Hobbies.Model, data = new_stats)
summary(fit, fit.measures=TRUE)
lavInspect(fit, what = "std")

#look to see which items go together in a 2 factor model

hobbies2_fa <- fa(new_stats,
```

```
                 nfactors = 2,
                 fm = "minres",
                     rotate = "oblimin")

hobbies2_fa
hobbies2_fa$loadings


Hobbies2.Model <-

'
arts =~ museums + art + reading + theater + concerts + music + yoga + dining + hiking + clubbing + movi
sporty =~ sports + tvsports + exercise + gaming + tv

'


fit2 <- cfa(Hobbies2.Model, data = new_stats)
summary(fit2, fit.measures=TRUE)
lavInspect(fit2, what = "std")
```

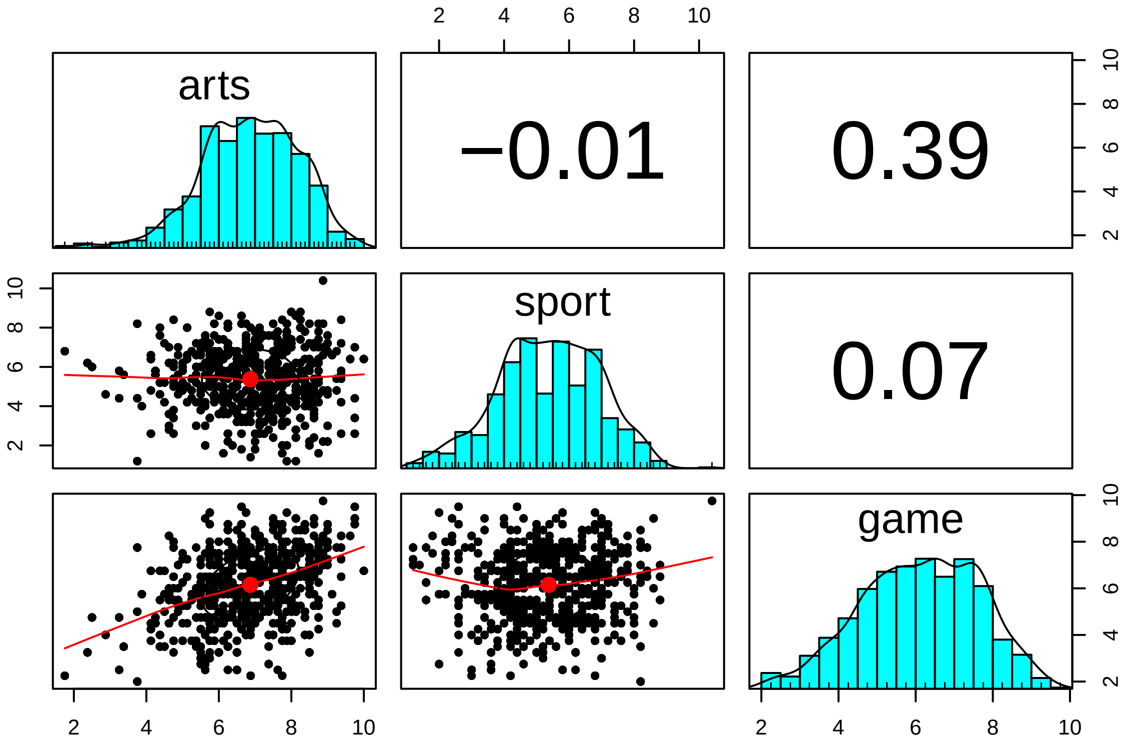
Interpreting the Two Models

The first model we can look at is a 3 factor model. From the scree plot it seems like 4 or 5 factors can be counted for the model before factors start going below 0 for Eigen value. However, having this many different latent variables would possibly be too much considering there are only 17 items and each latent variable would not have many items. By completing a CFA model for the proposed 3 factor model it can be we grouped the items into an arts, sports, and gaming categories. The CFI value is .641 which is lower than the suggested .95 or higher, the RMSEA is .127 which is much higher than the suggested .05 or lower, and finally the SRMR has a value of .106 which is higher than the suggested value of .05 or lower. From these models it can be seen that a more parsimonious model may include more or less factors than 3.

The second model looked at 2 factors instead of 3. When this CFA model was completed it can be seen that 2 is not enough factors since the CFI value was .823 which is lower than the previous model. The RMSEA is .080 and the SRMR is .083, which makes both values higher than the previous model, and therefore not as reliable. From these models it can be seen that a more parsimonious model should include more than 3 factors.

Multiple regression and correlation for Hobbie CFA 1

```
hob <- new_stats %>%
  mutate(arts = (museums + art + reading + theater + concerts + music + yoga + dining) / 8) %>%
  mutate(sport = (sports + tvsports + exercise + hiking + gaming) / 5) %>%
  mutate(game = (clubbing + tv + movies + shopping) / 4) %>%
  select(arts, sport, game)


pairs.panels(hob)
```




```
library(lm.beta)

multirega <- lm.beta(lm(arts ~ sport + game, data = hob))
summary(multirega)
```

```
##
## Call:
## lm(formula = arts ~ sport + game, data = hob)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8750 -0.8272  0.0640  0.8828  2.9572
##
## Coefficients:
##              Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  4.937667           NA   0.070997  69.548  < 2e-16 ***
## sport       -0.029483    -0.034141   0.008760  -3.365 0.000768 ***
## game         0.339831     0.388127   0.008882  38.260  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.233 on 8296 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:   0.15,  Adjusted R-squared:  0.1498
## F-statistic: 732.3 on 2 and 8296 DF,  p-value: < 2.2e-16
```

```
multiregb <- lm.beta(lm(sport ~ arts + game, data = hob))
summary(multiregb)
```

```
##
## Call:
```

```
## lm(formula = sport ~ arts + game, data = hob)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4502 -1.0643  0.0642  1.1578  4.8206
##
## Coefficients:
##             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept)  5.17937           NA    0.09634  53.759  < 2e-16 ***
## arts        -0.04624     -0.03994    0.01374  -3.365 0.000768 ***
## game         0.08313      0.08199    0.01203   6.909 5.23e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.544 on 8296 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:  0.00579,    Adjusted R-squared:  0.00555
## F-statistic: 24.16 on 2 and 8296 DF,  p-value: 3.46e-11
```

```
multiregc <- lm.beta(lm(game ~ arts + sport, data = hob))
summary(multiregc)
```

```
##
## Call:
## lm(formula = game ~ arts + sport, data = hob)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1256 -1.0009  0.0971  0.9573  3.6484
##
## Coefficients:
##             Estimate Standardized Std. Error t value Pr(>|t|)
## (Intercept) 2.748654           NA   0.097224  28.271  < 2e-16 ***
## arts        0.441352     0.386433   0.011536  38.260  < 2e-16 ***
## sport       0.068830     0.069786   0.009962   6.909 5.23e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.405 on 8296 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:  0.1538, Adjusted R-squared:  0.1535
## F-statistic: 753.6 on 2 and 8296 DF,  p-value: < 2.2e-16
```

```
#correlation

hob1 <- lm(sport ~ arts,
                data = hob)
hob1
```

```
##
## Call:
## lm(formula = sport ~ arts, data = hob)
##
## Coefficients:
## (Intercept)         arts
##    5.438981    -0.009611
```

```r
summary(hob1)
```

```
##
## Call:
## lm(formula = sport ~ arts, data = hob)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2029 -1.1549  0.0343  1.2139  5.0463
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.438981   0.088965  61.136   <2e-16 ***
## arts        -0.009611   0.012712  -0.756     0.45
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.549 on 8297 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:  6.889e-05,  Adjusted R-squared:  -5.163e-05
## F-statistic: 0.5716 on 1 and 8297 DF,  p-value: 0.4497
```

```r
hob2 <- lm(arts ~ game,
     data = hob)
hob2
```

```
##
## Call:
## lm(formula = arts ~ game, data = hob)
##
## Coefficients:
## (Intercept)         game
##      4.7915       0.3378
```

```r
summary(hob2)
```

```
##
## Call:
## lm(formula = arts ~ game, data = hob)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8962 -0.8219  0.0598  0.8842  2.9382
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.791496   0.056197   85.26   <2e-16 ***
## game        0.337841   0.008868   38.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.234 on 8297 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:  0.1489, Adjusted R-squared:  0.1488
## F-statistic:  1451 on 1 and 8297 DF,  p-value: < 2.2e-16
```

```r
hob3 <- lm(sport ~ game,
           data = hob)
hob3
```

```
##
## Call:
## lm(formula = sport ~ game, data = hob)
##
## Coefficients:
## (Intercept)         game
##      4.9578       0.0675
```

```r
summary(hob3)
```

```
##
## Call:
## lm(formula = sport ~ game, data = hob)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2809 -1.0784  0.0397  1.2034  4.7840
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.95779    0.07038  70.442  < 2e-16 ***
## game         0.06750    0.01111   6.078 1.27e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.545 on 8297 degrees of freedom
##   (79 observations deleted due to missingness)
## Multiple R-squared:  0.004433,   Adjusted R-squared:  0.004313
## F-statistic: 36.94 on 1 and 8297 DF,  p-value: 1.27e-09
```

Ethical Reflection after Analysis

Participants were informed of the study before taking part in the speed dating exercise. They had knowledge of how the data would be used, as well as knowledge of the confidentiality, which is present in the dataset. No names or identifying data are present. People could use the analysis of the data to try to manipulate people based on what results prove to be effective for getting a match. However, I do not think that this is of significant concern. We have made sure to test our results for differences between genders. We have not tested for the differences between different racial groups because of the added complexity that those models would entail, however in the original data collection they were sure to take measure of both gender and race for all participants.

What does need to be stated is that this data collection was done in the years 2002-2004. Since then society has changed vastly and one of these changes is the greater acceptance of homosexual couples. This data collection only looked at heterosexual couples and therefore is not truly representative of all people's relationships and kinds of thoughts that could go towards a partner. Questions such as "What do you think the opposite sex looks for?" are too broad considering it is not asking if they are talking about the heterosexual opposite sex or homosexual. It also needs to be stated that while participants did consent to have their data used for this dataset, they might not have thought that 20 years later students would be using this data for a project. However, besides the lack consideration towards non-heterosexual individuals, the data is fairly ethical considering the information gathered is confidential and participants did have to consent to have their responses recorded.