Final Report

Tianyi Tan

1. Split Violin Plot

**Features**

The plot visualizes and compares the distribution of the features included. One set of features called ‘self’ were obtained from surveys taken by participants themselves. Another set of features called ‘other’ were obtained from surveys completed by all the dates they met during the speed dating event. Both sets have ratings on Attractive, Sincere, Intelligent, Fun and Ambitious. Both contain ordinal variables on the scale of 1-10 (1=awful, 10=great). The difference is that the rating of others for each participant are averaged rating from all the dates they had using R. Thus, the rating of others might be considered as continuous variables after being averaged. The half violin distribution of each rating for each attribute is displayed and grouped by 5 attributes for comparison.

**Visualization techniques and Design**

Split violin plot is applied to study the difference of the distribution of self-ratings and others’ perceptions grouped by each attribute. It can compare two variables from two aspects grouped together which can make the visual comparison easier compared to individual distribution. The color palette chosen for the split violin plot are divergent colors with different hues to indicate two groups. The distributions are outlined by blue color which show the shape of the distribution. Legend and annotation are added to make the meaning of the color clearer.

**Code efforts and Interactivity**

The plot is produced by using R to pre-process the data and use R package Plotly to plot the split violin plot which can be interactive as well. Codes are attached in the appendix.The plot can be interactive which shows the distribution statistics such as mean and quartile but we chose to put it together with a radar chart showing the different in average to send more direct message to the audience.

**Conclusions and reflection**

When comparing self-ratings and the actual ratings, it shows that the participants generally overestimate themselves. The attribute that they overestimated the most was ​**fun and ambitious** while the one that overestimated the least was **​intelligence**​. It implicates that when people are dating, they might perceive themselves higher than real perception from other which might result in unsuccessful matching. Main take-away for single people trying speed-dating might be save more time to let others know more about you instead of telling jokes that is not funny.

**Future work**

If I had more time, I might incorporate gender to compare the perception difference by gender using violin plot to enrich the analysis.

1. Parallel Coordinate Plot

**Features**

The plot visualizes the pattern of high-dimensional data. 9 features are included: demographics of participants (categorical variables): field of study, race, age, self-perception (ordinal variables) of their attractiveness, sincere, intelligence, fun, ambition and calculated success rate (continuous variables) (matching percentage = match count/total rounds). The matching percentage (success rate) is an important variable to draw conclusion about whether each feature contributes to high or low success rate.

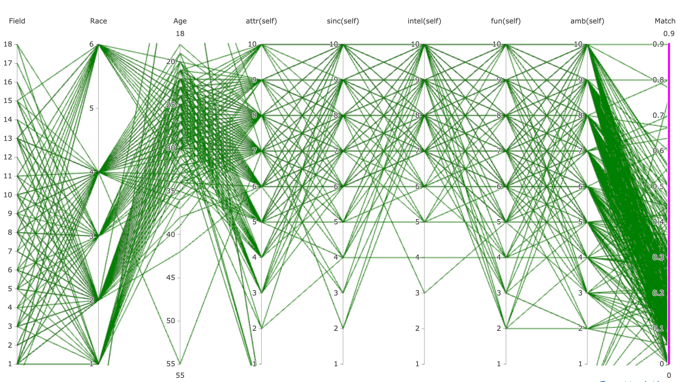
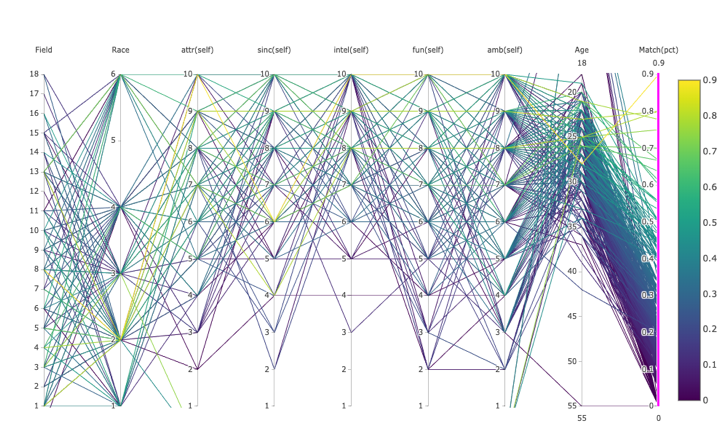
|  |  |
| --- | --- |
| Field Number | Field |
| 1 | Law |
| 2 | Math |
| 3 | Social Science, Psychologist |
| 4 | Medical Science, Pharmaceuticals, and Bio Tech |
| 5 | Engineering |
| 6 | English/Creative Writing/ Journalism |
| 7 | History/Religion/Philosophy |
| 8 | Business/Econ/Finance |
| 9 | Education, Academia |
| 10 | Biological Sciences/Chemistry/Physics |
| 11 | Social Work |
| 12 | Undergrad/undecided |
| 13 | Political Science/International Affairs |
| 14 | Film |
| 15 | Fine Arts/Arts Administration |
| 16 | Languages |
| 17 | Architecture |
| 18 | Other |

|  |  |
| --- | --- |
| Race Number | Race |
| 1 | Black/African American |
| 2 | European/Caucasian-American |
| 3 | Latino/Hispanic American |
| 4 | Asian/Pacific Islander/Asian-American |
| 5 | Native American |
| 6 | Other |

**Visualization techniques and Design**

I have also applied parallel coordinate plot for 9 features. It is usually used for high-dimension data to show the patterns of complex relationships. The parallel coordinate enables more features to be included in a clearer way and also plot is interactive produced by R code. By pulling the feature around vertically and selecting different values for each different feature, it is a lot of fun and also help identify patterns even with many instances and features involved.

The color palette chosen for the plot were originally sequential colors with different hues and saturation to indicate the percentage of matching. However, the best sequential color in Plotly is still making the graph more complicated and the messages a bit unclear for audience. Following the advice from Professor, we decide to change the color to pure green which is most sensitive to human cones and serves the purpose of highlighting the lines selected for comparison.



**Code efforts and Interactivity**

The plot is produced by using R to pre-process the data and use R package Plotly to plot which is interactive. Codes are attached in the appendix. It is important to use the interactive plot to show the pattern by pulling different columns and selecting different options for each column. By interactively playing with the plot, it can help us answer some interesting questions. It can be also related to the split violin plot that I created and complement each other. This plot takes a deep step to analyze at individual level and take a broader view with more features and observe how they interact with each other.

**Conclusions and reflection**

This plot shows interesting patterns of demographics, self-perceptions and dating success by answering the following three questions: (static plot is shown here for illustration and interactive plot is attached as an HTML file)

1. Are people of different races perceiving themselves differently in general?

Four of the six options of races are analyzed which have enough instances to shown patterns and specified ethnicity. By pulling ‘race’ to the first vertical line and select different values of this feature, the four graphs show the patterns of 4 different races. In terms of the general trend of different attributes, there are similarities among different races. For example, people of different races have the most diversities in self-perceived ambitions while most of the people of different races rate themselves above 5 for intelligence. We might conclude that people of different races might have similar perception of themselves.

|  |  |
| --- | --- |
| Black/African American | European/Caucasian-American |
| Latino/Hispanic American | Asian/Pacific Islander/Asian-American |

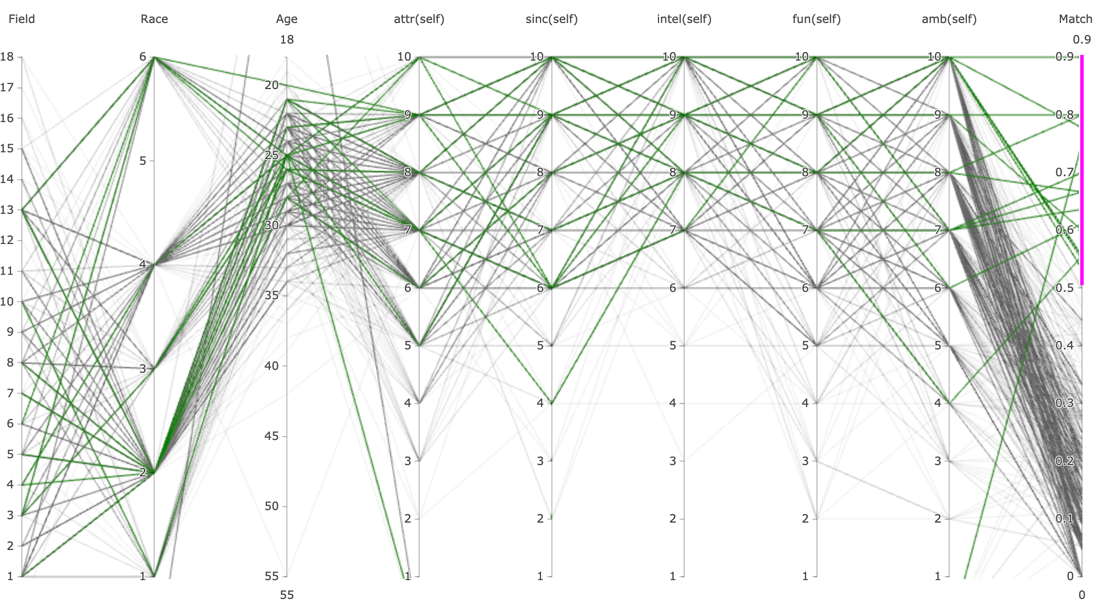
1. Are people from certain field of study more successful in dating?

By pulling field to the first vertical line and selecting matching percent over 50%, the graph shows that participants of field of study from 1 to 13 all have high matching percent. The participant with 90% matching rate studies (field 8) business/econ/finance. Participants with over 80% matching rates also study (field 1,3,4,8,13) Law, Social Science, Medical Science, Business and Political Science/International Affairs. People who are good at communication or studying science might win themselves a date. By selecting the matching rate ranging from 0 to 10%, we can see that no particular field of study might result in low matching rate. Thus, never blame your major!

|  |  |
| --- | --- |
| Matching Rate | Interactive Plot Screen Shot |
| 50% to 90% |  |
| 90% |  |
| 70% - 90% |  |
| 0-10% |  |

1. Are people who have a high self-perception for different attributes more successful in dating?

Using the same graph with high matching rate selected (50% to 90%), we can observe that participants who have matching rate over 50% tends to rate themselves high in all 5 attributes. Thus, confident people or people with high value in the five attributes tend to have successful matching in speed dating. Self-perceived intelligence ratings are at least 7 shown in the graph which means people thinks themselves or are really smart tend to secure a date while pattern is not as clear for ambition.



**Future work**

If I had more time, I will try to add more relevant features such as income level and other’s perception to explore more patterns from more aspects.

**Code**

1. Split-Violin Plot

---

title: " Split-Violin Plot"

output: word\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

library(fmsb)

library(dplyr)

library(tibble)

library(stringr)

library(ggplot2)

library(grid)

library(gridBase)

library(scales)

```

## R Markdown

```{r Load Data and Cleaning }

setwd("~/Dropbox/Homework/DSC465/HW3")

rawdat <-

read.csv('Speed Dating Data.csv', header = T, stringsAsFactors = F)

```

```{r remove variables that will not be used}

dat <-

rawdat %>%

select(-id, -idg, -condtn, -round, -position, -positin1, -order, -partner, -tuition, -undergra, -mn\_sat)

```

```{r Clean Data}

#Ratings by other ppl and self-rating

at00 <-

dat %>%

select(iid, pid, age, field\_cd,gender,race,) %>%

filter(!pid == "NA")

#drop rows where all attributes were rated NA (col 4-9)

#Since in the instructions it clearly outlined that not attributes will be discussed during

#the couple's meetings, we cannot do a full NA drop in the data. The workaround here is we will assign all NA values to 1000, and drop the rows if all the attributes add up to 5000. Rows with 1 or 2 NAs will add up to be less than 6000 and will not be dropped. Finally the rows with 1000 will be converted back to NA.

at00[is.na(at00)] <- 1000

at00$total <- rowSums(at00[,c("attr\_o", "sinc\_o", "intel\_o", "fun\_o", "amb\_o")])

at00$total1 <-rowSums(at00[,c("attr3\_1","sinc3\_1","intel3\_1","fun3\_1","amb3\_1")])

at00 <-

at00 %>%

filter(!total == "5000")

at00 <-

at00 %>%

filter(!total1 == "5000")

at00[at00 == "1000"] <- NA

at00$total <- rowSums(at00[,c("attr\_o", "sinc\_o", "intel\_o", "fun\_o", "amb\_o")], na.rm=TRUE)

table(at00$total)

#A total of 0 means all entries are 0, which constitutes missing data and row is dropped

at00$total1 <- rowSums(at00[,c("attr3\_1","sinc3\_1","intel3\_1","fun3\_1","amb3\_1")], na.rm=TRUE)

table(at00$total1)

at00 <-

at00 %>%

filter(!total == "0")

#Finally, it is important to realize that the attributes are evaluated for the opposite gender.

#Another column for the partner is generated

at00 <-

at00 %>%

mutate(pgender = ifelse(gender == 0, 1, 0))

```

```{r Data Clean Step 2}

# How do you think you measure up?

#Please rate your opinion of your own attributes, on a scale of 1-10 (be honest!):

#take related attributes with iid and gender into new data frame

at31<-

dat %>%

select(iid, gender, attr3\_1, sinc3\_1, intel3\_1, fun3\_1, amb3\_1) %>%

unique()

#Next, we would like to turn all NA into 0, but before this, we check if any entries in iid or gender is NA to prevent mislabels

sum(is.na(at31$iid))

sum(is.na(at31$gender))

# no entries in iid or gender is NA

at31[is.na(at31)] <- 0

#Add column to check if total of attributions add up to 100

at31$total <- rowSums(at31[,c("attr3\_1", "sinc3\_1", "intel3\_1", "fun3\_1", "amb3\_1")])

table(at31$total)

#A total of 0 means all entries are missing and row is dropped

at31<-

at31 %>%

filter(!total == "0")

```

```{r Self-perceived vs Others' Perception}

at00[is.na(at00)] <- 0

Others <-

at00 %>%

group\_by(iid) %>%

summarise(Attractive = mean(attr\_o), Sincere = mean(sinc\_o), Intelligent = mean(intel\_o), Fun = mean(fun\_o), Ambitious = mean(amb\_o))

Self <-

at00 %>%

select(iid,attr3\_1, sinc3\_1, intel3\_1, fun3\_1, amb3\_1) %>%

unique()

# merge the two table (left outer join)

Others\_Self <- merge(x=Others, y=Self, by = "iid", all.x=TRUE)

```

```{r}

library(data.table)

setnames(Others\_Self, old=c("attr3\_1", "sinc3\_1", "intel3\_1", "fun3\_1", "amb3\_1"), new=c("Attr(Self)", "Sinc(Self)", "Intel(Self)", "Fun(Self)", "Amb(Self)"))

```

```{r}

library(dplyr)

library(tidyr)

library(plotly)

plot\_data <- Others\_Self %>%

gather(variable, value, -iid)

plot\_data <-

plot\_data %>%

mutate(self = ifelse(variable == 'Attr(Self)' |variable == 'Sinc(Self)'|variable == 'Fun(Self)'|variable == 'Amb(Self)'|variable == 'Intel(Self)', "Self", "Others"))

```

```{r}

Sys.setenv("plotly\_username"="ttanjocelyn")

Sys.setenv("plotly\_api\_key"="OXWKncxtmPfJWNwhCmcu")

chart\_link <- api\_create(p, filename = "violin-split")

chart\_link

```

```{r Split Violin Plot}

splitplot <- plot\_data

splitplot$variable[splitplot$variable == 'Attr(Self)'] <- "Attractive"

splitplot$variable[splitplot$variable == 'Sinc(Self)'] <- "Sincere"

splitplot$variable[splitplot$variable == 'Intel(Self)'] <- "Intelligent"

splitplot$variable[splitplot$variable == 'Fun(Self)'] <- "Fun"

splitplot$variable[splitplot$variable == 'Amb(Self)'] <- "Ambitious"

```

```{r}

df <- splitplot

p <- df %>%

plot\_ly(type = 'violin') %>%

add\_trace(

x = ~variable[df$self == "Self"],

y = ~value[df$self == "Self"],

legendgroup = 'Yes',

scalegroup = 'Yes',

name = 'Self',

side = 'negative',

box = list(

visible = T

),

meanline = list(

visible = T

),

line = list(

color = 'blue'

)

) %>%

add\_trace(

x = ~variable[df$self == "Others"],

y = ~value[df$self == "Others"],

legendgroup = 'No',

scalegroup = 'No',

name = 'Others',

side = 'positive',

box = list(

visible = T

),

meanline = list(

visible = T

),

line = list(

color = 'green'

)

) %>%

layout(

xaxis = list(

title = ""

),

yaxis = list(

title = "",

zeroline = F

),

violingap = 0,

violingroupgap = 0,

violinmode = 'overlay'

)

p

```

1. Interactive Parallel Coordinate Plot

---

1. title: " Interactive Parallel Coordinate Plot”

output: word\_document

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = TRUE)

library(fmsb)

library(dplyr)

library(tibble)

library(stringr)

library(ggplot2)

library(grid)

library(gridBase)

library(scales)

library(psych)

library(plotly)

```

## R Markdown

```{r Load Data and Cleaning }

setwd("~/Dropbox/Homework/DSC465/HW3")

rawdat <-

read.csv('Speed Dating Data.csv', header = T, stringsAsFactors = F)

```

```{r remove variables that will not be used}

dat <-

rawdat %>%

select(-id, -idg, -condtn, -round, -position, -positin1, -order, -partner, -tuition, -undergra, -mn\_sat)

```

```{r Prepare the Features}

at00 <-

dat %>%

select(iid, pid, age, field\_cd,race,attr3\_1,sinc3\_1,intel3\_1,fun3\_1,amb3\_1,attr1\_1,sinc1\_1,intel1\_1,fun1\_1,amb1\_1,shar1\_1,match) %>%

filter(!pid == "NA") %>%

```

```{r Calculate Match Percentage}

match\_df <-

at00%>%

group\_by(iid) %>%

summarise(matchcount = sum(match)/n())

```

```{r Remove Duplicates}

features\_df <-

at00 %>%

select(iid,age,field\_cd,race,attr3\_1,sinc3\_1,intel3\_1,fun3\_1,amb3\_1,attr1\_1,sinc1\_1,intel1\_1,fun1\_1,amb1\_1,shar1\_1) %>%

unique()

```

```{r Merge Dataset}

df\_all <-merge(x = features\_df, y = match\_df, by = "iid", all = TRUE)

```

```{r}

df\_all$highmatch <- ifelse(as.numeric(df\_all$matchcount) > 0.14, 1,0)

```

```{r}

describe(df\_all)

```

```{r Plot}

p <- df\_all %>%

plot\_ly(width = 1000, height = 600) %>%

add\_trace(type = 'parcoords',

#line = list(color = ~matchcount,

#colorscale = "Viridis",

#showscale = TRUE

#),

line = list(color = 'green',showscale = TRUE),

dimensions = list(

list(range = c(1,18),

tickvals=c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18),

label = 'Field', values = ~field\_cd)

list(range = c(1,6),

tickvals=c(1,2,3,4,5,6),

label = 'Race', values = ~race),

list(range = c(1,10),

tickvals=c(1,2,3,4,5,6,7,8,9,10),

label = 'attr(self)', values = ~attr3\_1),

list(range = c(55,18),

label = 'Age', values = ~age)

list(range = c(1,10),

tickvals=c(1,2,3,4,5,6,7,8,9,10),

label = 'sinc(self)', values = ~sinc3\_1),

list(range = c(1,10),

tickvals=c(1,2,3,4,5,6,7,8,9,10),

label = 'intel(self)', values = ~intel3\_1 ),

list(range = c(1,10),

tickvals=c(1,2,3,4,5,6,7,8,9,10),

label = 'fun(self)', values = ~fun3\_1 ),

list(range = c(1,10),

tickvals=c(1,2,3,4,5,6,7,8,9,10),

label = 'amb(self)', values = ~amb3\_1 ),

list(range = c(~min(matchcount),~max(matchcount)),

constraintrange = c(0,1),

label = 'Match%', values = ~matchcount)

)

)

p

```

```{r}

# Create a shareable link to your chart

# Set up API credentials: https://plot.ly/r/getting-started

Sys.setenv("plotly\_username"="ttanjocelyn")

Sys.setenv("plotly\_api\_key"="OXWKncxtmPfJWNwhCmcu")

chart\_link = api\_create(p, filename="parcoords-advanced")

chart\_link

```