CS 581 : Final Project - A Dating Experiment

Aravinda Reddy Dandu, Rithvik Reddy Ananth, SriKaavya Toodi, Sowmya Reddy AnthaReddy
12/18/2020

Contents

1 About the dataset

• 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
• The dataset has 195 columns. But we will limit ourselves to a few interesting columns and discard others. It has 8000 rows in 20 different waves. A wave is an event where people match with each other. Each time a person is matched, a record is added.
Primary questions answered in the dataset:

- What do you find most attractive in your date?
- What is your background? (study, undergrad, place, career etc.)
- What are your hobbies? (All numerically rated. No text input)
- Rate the date you are matched with for different attributes
- Rate yourself
- After a few weeks of this experiment, how many contacted you? How many did you contact?
- Rating about the event

Point to be noted is that all the columns in the dataset are numerical and very few have textual inputs. Even for these textual inputs, they are categorical. This makes it statistical easy and accurate.

Data_cleaning_visualization

Group Project

12/16/2020

Selecting required rows

```
rawdat <-
read.csv('/Users/rithvikananth/Documents/R Files/9781789950298/Data for practical examples and exe
         header = T, stringsAsFactors = F)
# remove columns that will not be used
dat <-
rawdat %>%
  select(-id, -idg, -condtn, -round, -position, -positin1, -order,
         -partner, -tuition, -undergra, -mn_sat)
```

Handling NA values

This part of for cleaning the data first. Here some NA values are changed. Instead of dropping them, those who don't sum up to full values are deleted.

```
at00 <-
dat %>%
  select(iid, pid, dec, gender, attr, sinc, intel, fun, amb, shar, like, prob) %>%
  filter(!pid == "NA")

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

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

at00 <-
at00 %>%
  filter(!total == "6000")

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

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


at00 <-
at00 %>%
  filter(!total == "0")

at00 <-
```

```r
at00 %>%
  mutate(pgender = ifelse(gender == 0, 1, 0))
```

Generalizing scale

There are waves in the experiment. In some waves participants are asked to score in 0-10 scale while in some other waves, they are asked to score on 0-100 scale. So, to generalize, all scores to made in the 0-100 scale. What people look for in a match before performing experiment is selected below.

```r
at11<-
dat %>%
  group_by(gender) %>%
  select(iid, gender, attr1_1, sinc1_1, intel1_1, fun1_1, amb1_1, shar1_1) %>%
  unique()

at11[is.na(at11)] <- 0
at11$total <- rowSums(at11[,c("attr1_1", "sinc1_1", "intel1_1",
                              "fun1_1", "amb1_1", "shar1_1")])

at11<-
at11 %>%
  filter(!total == "0")

# Rounding to required scale

at11$attr1_1 <- round(at11$attr1_1/at11$total*100, digits = 2)
at11$sinc1_1 <- round(at11$sinc1_1/at11$total*100, digits = 2)
at11$intel1_1 <- round(at11$intel1_1/at11$total*100, digits = 2)
at11$fun1_1 <- round(at11$fun1_1/at11$total*100, digits = 2)
at11$amb1_1 <- round(at11$amb1_1/at11$total*100, digits = 2)
at11$shar1_1 <- round(at11$shar1_1/at11$total*100, digits = 2)

at11$total <- rowSums(at11[,c("attr1_1", "sinc1_1", "intel1_1",
                              "fun1_1", "amb1_1", "shar1_1")])

at11$total <- round(at11$total, digits = 0)

# What same sex peers are looking for in their partners

at41<-
dat %>%
  group_by(gender) %>%
  select(iid, gender, attr4_1, sinc4_1, intel4_1, fun4_1, amb4_1,
         shar4_1) %>%
  unique()

at41[is.na(at41)] <- 0

at41$total <- rowSums(at41[,c("attr4_1", "sinc4_1", "intel4_1",
                              "fun4_1", "amb4_1", "shar4_1")])

at41<-
```

```r
at41 %>%
  filter(!total == "0")

at41$attr4_1 <- round(at41$attr4_1/at41$total*100, digits = 2)
at41$sinc4_1 <- round(at41$sinc4_1/at41$total*100, digits = 2)
at41$intel4_1 <- round(at41$intel4_1/at41$total*100, digits = 2)
at41$fun4_1 <- round(at41$fun4_1/at41$total*100, digits = 2)
at41$amb4_1 <- round(at41$amb4_1/at41$total*100, digits = 2)
at41$shar4_1 <- round(at41$shar4_1/at41$total*100, digits = 2)

at41$total <- rowSums(at41[,c("attr4_1", "sinc4_1", "intel4_1",
                              "fun4_1", "amb4_1", "shar4_1")])

at41$total <- round(at41$total, digits = 0)

# What opposite sex is looking for in a partner

at21<-
dat %>%
  group_by(gender) %>%
  select(iid, gender, attr2_1, sinc2_1, intel2_1, fun2_1,
         amb2_1, shar2_1) %>%
  unique()

at21[is.na(at21)] <- 0

at21$total <- rowSums(at21[,c("attr2_1", "sinc2_1", "intel2_1",
                              "fun2_1", "amb2_1", "shar2_1")])

at21<-
at21 %>%
  filter(!total == "0")

at21$attr2_1 <- round(at21$attr2_1/at21$total*100, digits = 2)
at21$sinc2_1 <- round(at21$sinc2_1/at21$total*100, digits = 2)
at21$intel2_1 <- round(at21$intel2_1/at21$total*100, digits = 2)
at21$fun2_1 <- round(at21$fun2_1/at21$total*100, digits = 2)
at21$amb2_1 <- round(at21$amb2_1/at21$total*100, digits = 2)
at21$shar2_1 <- round(at21$shar2_1/at21$total*100, digits = 2)

at21$total <- rowSums(at21[,c("attr2_1", "sinc2_1", "intel2_1",
                              "fun2_1", "amb2_1", "shar2_1")])

at21$total <- round(at21$total, digits = 0)

# Twin bar plot - 1

test1 <-
at11 %>%
  group_by(gender) %>%
  summarise(Attractive = mean(attr1_1), Sincere = mean(sinc1_1),
            Intelligent = mean(intel1_1), Fun = mean(fun1_1),
            Ambitious = mean(amb1_1), Interest = mean(shar1_1))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
test1forplot <-
test1 %>%
  select(-gender)

maxmin <- data.frame(
 Attractive = c(36, 0),
 Sincere = c(36, 0),
 Intelligent = c(36, 0),
 Fun = c(36, 0),
 Ambitious = c(36, 0),
 Interest = c(36, 0))

test11 <- rbind(maxmin, test1forplot)

test11male <- test11[c(1,2,4),]
test11female <- test11[c(1,2,3),]

DATA <- data.frame(
  traits = c(colnames(test11)),
  men = as.numeric(test11[4,]),
  women = as.numeric(test11[3,])
)
# Twin bar plot - 2
test4 <-
at41 %>%
  group_by(gender) %>%
  summarise(Attractive = mean(attr4_1), Sincere = mean(sinc4_1),
            Intelligent = mean(intel4_1), Fun = mean(fun4_1),
            Ambitious = mean(amb4_1), Interest = mean(shar4_1))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
test4forplot <-
test4 %>%
  select(-gender)

test41 <- rbind(maxmin, test4forplot)

DATA_1 <- data.frame(
  traits = c(colnames(test41)),
  men = as.numeric(test41[4,]),
  women = as.numeric(test41[3,])
)
# Twin bar plot - 3
test2 <-
at21 %>%
  group_by(gender) %>%
  summarise(Attractive = mean(attr2_1), Sincere = mean(sinc2_1),
            Intelligent = mean(intel2_1), Fun = mean(fun2_1),
            Ambitious = mean(amb2_1), Interest = mean(shar2_1))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
test2forplot <-
test2 %>%
  select(-gender)

test21 <- rbind(maxmin, test2forplot)

DATA_2 <- data.frame(
  traits = c(colnames(test21)),
  men = as.numeric(test21[4,]),
  women = as.numeric(test21[3,])
)
```

Data Cleaning/Preperation - Part-2

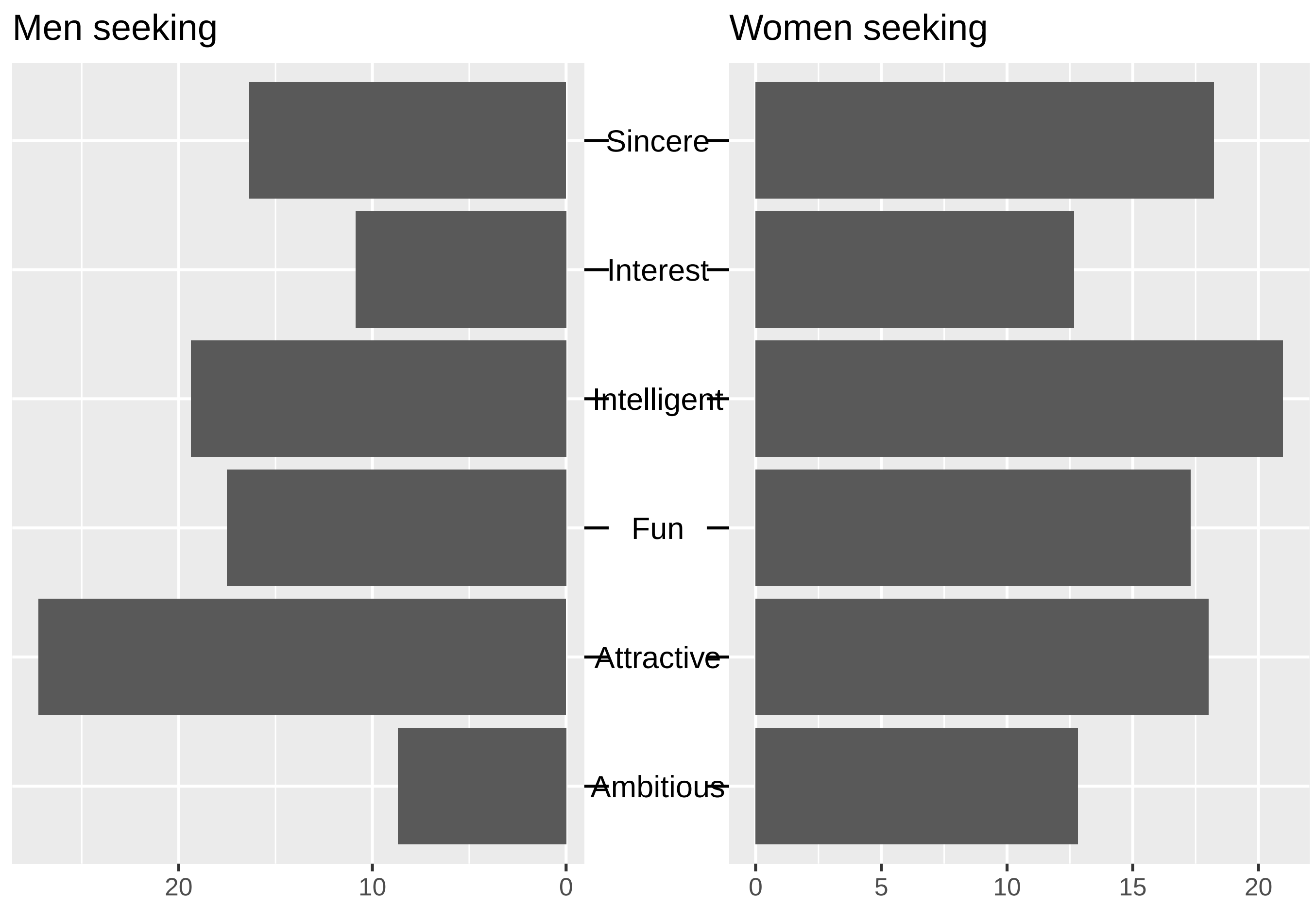
```
data = read.csv(
"/Users/rithvikananth/Documents/R Files/9781789950298/Data for practical examples and exercises/SD
data$gender <- as.factor(data$gender)
data$race <- as.factor(data$race)
data$career_c <- as.factor(data$career_c)
data$dec <- as.factor(data$dec)
data$date <- as.factor(data$date)
data$samerace <- as.factor(data$samerace)

career_data = select(filter(data, !is.na(career_c)), iid,gender, career_c)
career_data <- unique(career_data, by = iid)

# for top men and women
Dating_Data = read.csv("/Users/rithvikananth/Documents/R Files/9781789950298/Data for practical ex

Frequency = table(Dating_Data$iid)
Frequency = data.frame(Frequency)
names(Frequency)= c("iid", "NumOfDates")
Dating_Data = merge(Dating_Data, Frequency, by = "iid")


Pop = aggregate(Dating_Data$dec, list(Dating_Data$pid), sum)
names(Pop)= c("iid", "Popular")
Dating_Data = merge(Dating_Data, Pop, by = "iid")
Dating_Data$PoPRel = Dating_Data$Popular/Dating_Data$NumOfDates * 100

Dating_Data$race[Dating_Data$race==1] = "African"
Dating_Data$race[Dating_Data$race==2] = "European"
Dating_Data$race[Dating_Data$race==3] = "Latino"
Dating_Data$race[Dating_Data$race==4] = "Asian"
Dating_Data$race[Dating_Data$race==5] = "Native"
Dating_Data$race[Dating_Data$race==6] = "Other"

#Subsets with only men and women
Man = subset(Dating_Data, gender == 1)
Woman = subset(Dating_Data, gender == 0)

#Removing the duplicates
```

```
Woman = Woman[order(Woman$PoPRel, decreasing = TRUE),]
index <- which(duplicated(Woman$iid))
Woman = Woman[-index,]
Man = Man[order(Man$PoPRel,decreasing = TRUE),]
index <- which(duplicated(Man$iid))
Man = Man[-index,]
TopWoman = Woman[-c(seq(28,274,1)),]
TopMan = Man[-c(seq(28,277,1)),]
```

What people (grouped by gender) look for in a match?

Above selected data is visualized in below chart.

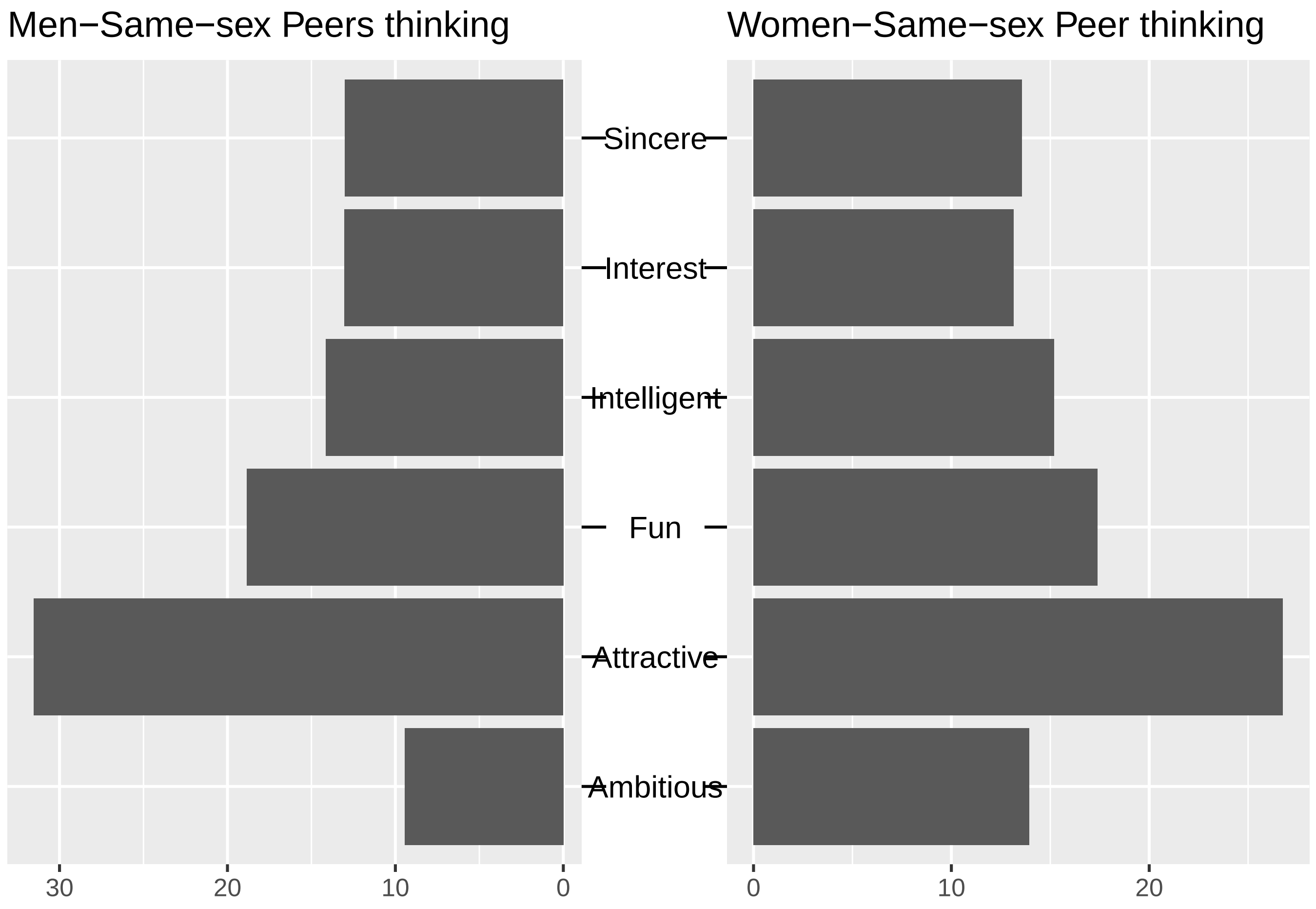
```
g.mid<-ggplot(DATA,aes(x=1,y=traits))+geom_text(aes(label=traits))+
  geom_segment(aes(x=0.94,xend=0.96,yend=traits))+
  geom_segment(aes(x=1.04,xend=1.065,yend=traits))+
  ggtitle("")+
  ylab(NULL)+
  scale_x_continuous(expand=c(0,0),limits=c(0.94,1.065))+
  theme(axis.title=element_blank(),
        panel.grid=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.background=element_blank(),
        axis.text.x=element_text(color=NA),
        axis.ticks.x=element_line(color=NA),
        plot.margin = unit(c(1,-1,1,-1), "mm"))

g1 <- ggplot(data = DATA, aes(x = traits, y = men)) +
  geom_bar(stat = "identity") + ggtitle("Men seeking") +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.margin = unit(c(1,-1,1,0), "mm")) +
  scale_y_reverse() + coord_flip()

g2 <- ggplot(data = DATA, aes(x = traits, y = women)) +xlab(NULL)+
  geom_bar(stat = "identity") + ggtitle("Women seeking") +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        plot.margin = unit(c(1,0,1,-1), "mm")) +
  coord_flip()

gg1 <- ggplot_gtable(ggplot_build(g1))
gg2 <- ggplot_gtable(ggplot_build(g2))
gg.mid <- ggplot_gtable(ggplot_build(g.mid))

grid.arrange(gg1,gg.mid,gg2,ncol=3,widths=c(4/9,1/9,4/9))
```



- For men: Preference of men is mostly for 'Attractiveness' in a woman
- For Women: For women, it's fairly distributed over all the attributes.
- Above visualization is what they want. Next we'll see what they think same sex looks for.

Participants opinion on what their same sex peers are looking for.

```
# what men and women think their sex wants from opposite gender.
g.mid<-ggplot(DATA_1,aes(x=1,y=traits))+geom_text(aes(label=traits))+
  geom_segment(aes(x=0.94,xend=0.96,yend=traits))+
  geom_segment(aes(x=1.04,xend=1.065,yend=traits))+
  ggtitle("")+
  ylab(NULL)+
  scale_x_continuous(expand=c(0,0),limits=c(0.94,1.065))+
  theme(axis.title=element_blank(),
        panel.grid=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.background=element_blank(),
        axis.text.x=element_text(color=NA),
        axis.ticks.x=element_line(color=NA),
        plot.margin = unit(c(1,-1,1,-1), "mm"))

g1 <- ggplot(data = DATA_1, aes(x = traits, y = men)) +
  geom_bar(stat = "identity") + ggtitle("Men-Same-sex Peers thinking") +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.margin = unit(c(1,-1,1,0), "mm")) +
  scale_y_reverse() + coord_flip()

g2 <- ggplot(data = DATA_1, aes(x = traits, y = women)) +xlab(NULL)+
  geom_bar(stat = "identity") + ggtitle("Women-Same-sex Peer thinking") +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        plot.margin = unit(c(1,0,1,-1), "mm")) +
  coord_flip()

gg1 <- ggplot_gtable(ggplot_build(g1))
gg2 <- ggplot_gtable(ggplot_build(g2))
gg.mid <- ggplot_gtable(ggplot_build(g.mid))

grid.arrange(gg1,gg.mid,gg2,ncol=3,widths=c(4/9,1/9,4/9))
```



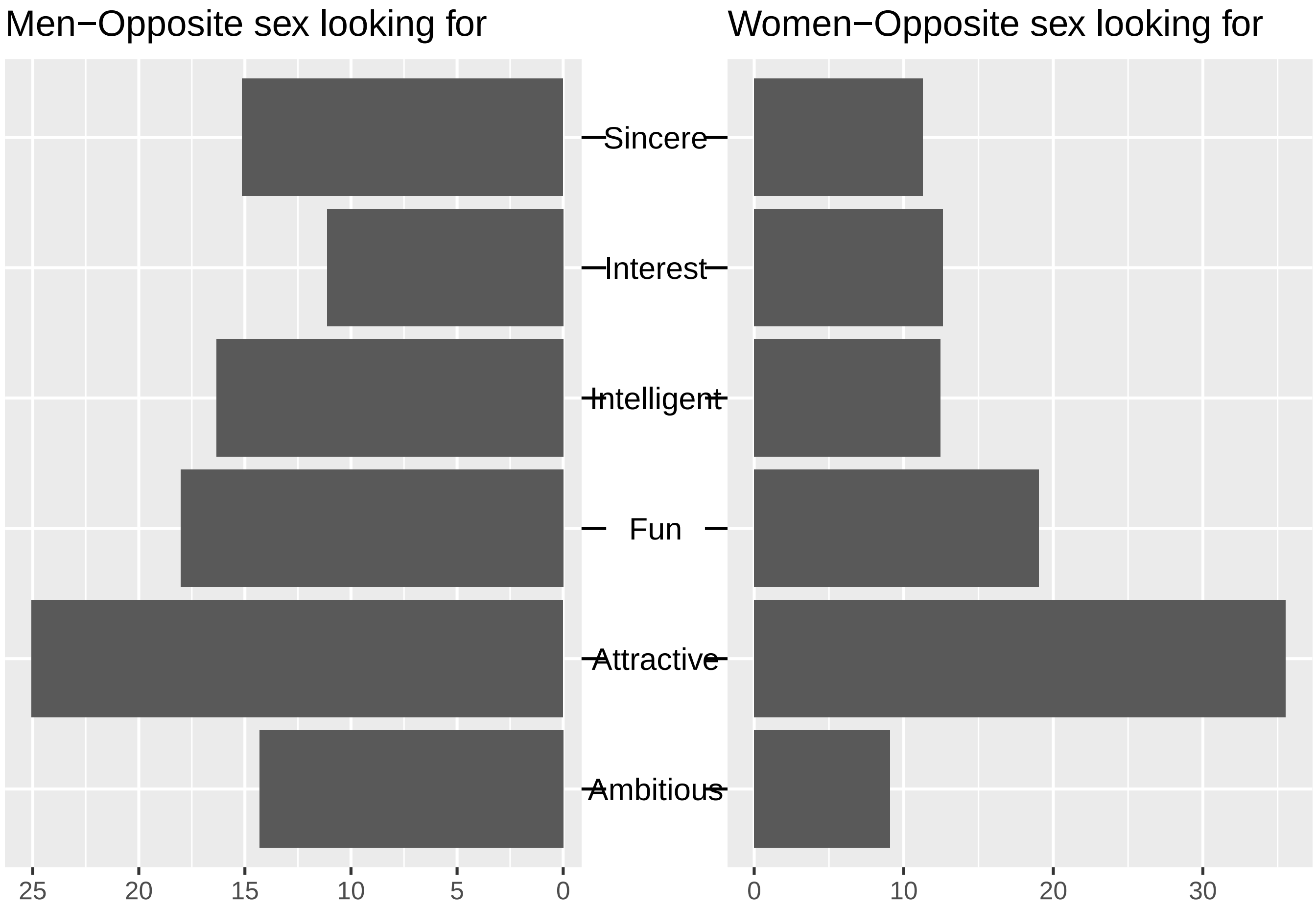
Men−Same−sex Peers thinking

Women−Same−sex Peer thinking

* Both men and women think that their fellow gender people are looking for attractive partners * Men think that their peers are least concerned with women ambitions * When women rate themselves on what they look in their partners, they were looking for someone with all the parameters but they assume that other women are only looking for men who are mostly attractive and fun.

Participants opinion on what the opposite sex is looking for.

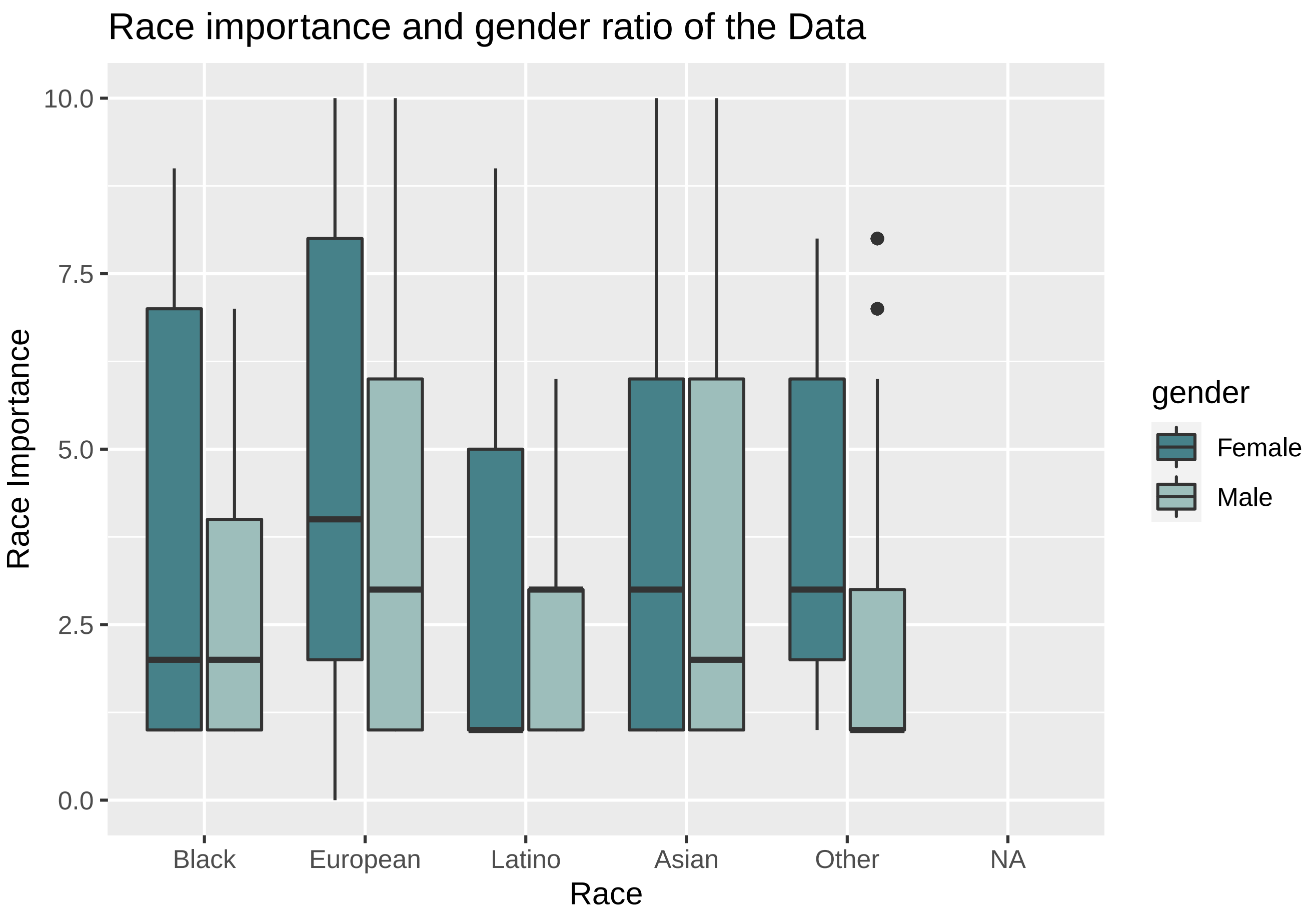
```
# What the opposite sex is looking for
g.mid<-ggplot(DATA_2,aes(x=1,y=traits))+geom_text(aes(label=traits))+
  geom_segment(aes(x=0.94,xend=0.96,yend=traits))+
  geom_segment(aes(x=1.04,xend=1.065,yend=traits))+
  ggtitle("")+
  ylab(NULL)+
  scale_x_continuous(expand=c(0,0),limits=c(0.94,1.065))+
  theme(axis.title=element_blank(),
        panel.grid=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        panel.background=element_blank(),
        axis.text.x=element_text(color=NA),
        axis.ticks.x=element_line(color=NA),
        plot.margin = unit(c(1,-1,1,-1), "mm"))

g1 <- ggplot(data = DATA_2, aes(x = traits, y = men)) +
  geom_bar(stat = "identity") + ggtitle("Men-Opposite sex looking for") +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        plot.margin = unit(c(1,-1,1,0), "mm")) +
  scale_y_reverse() + coord_flip()

g2 <- ggplot(data = DATA_2, aes(x = traits, y = women)) +xlab(NULL)+
  geom_bar(stat = "identity") + ggtitle("Women-Opposite sex looking for") +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        plot.margin = unit(c(1,0,1,-1), "mm")) +
  coord_flip()

gg1 <- ggplot_gtable(ggplot_build(g1))
gg2 <- ggplot_gtable(ggplot_build(g2))
gg.mid <- ggplot_gtable(ggplot_build(g.mid))

grid.arrange(gg1,gg.mid,gg2,ncol=3,widths=c(4/9,1/9,4/9))
```



* Both men and women feel that the opposite gender is looking for an attractive partner * Men also think that women would want them to be fun. * Women have an opinion that men do not give much importance to other attributes apart from being attractive. * Overall women expect men to be good in almost all aspects where as men prefer women who are attractive and have an above average intelligence.

Understanding the demographics of the data.

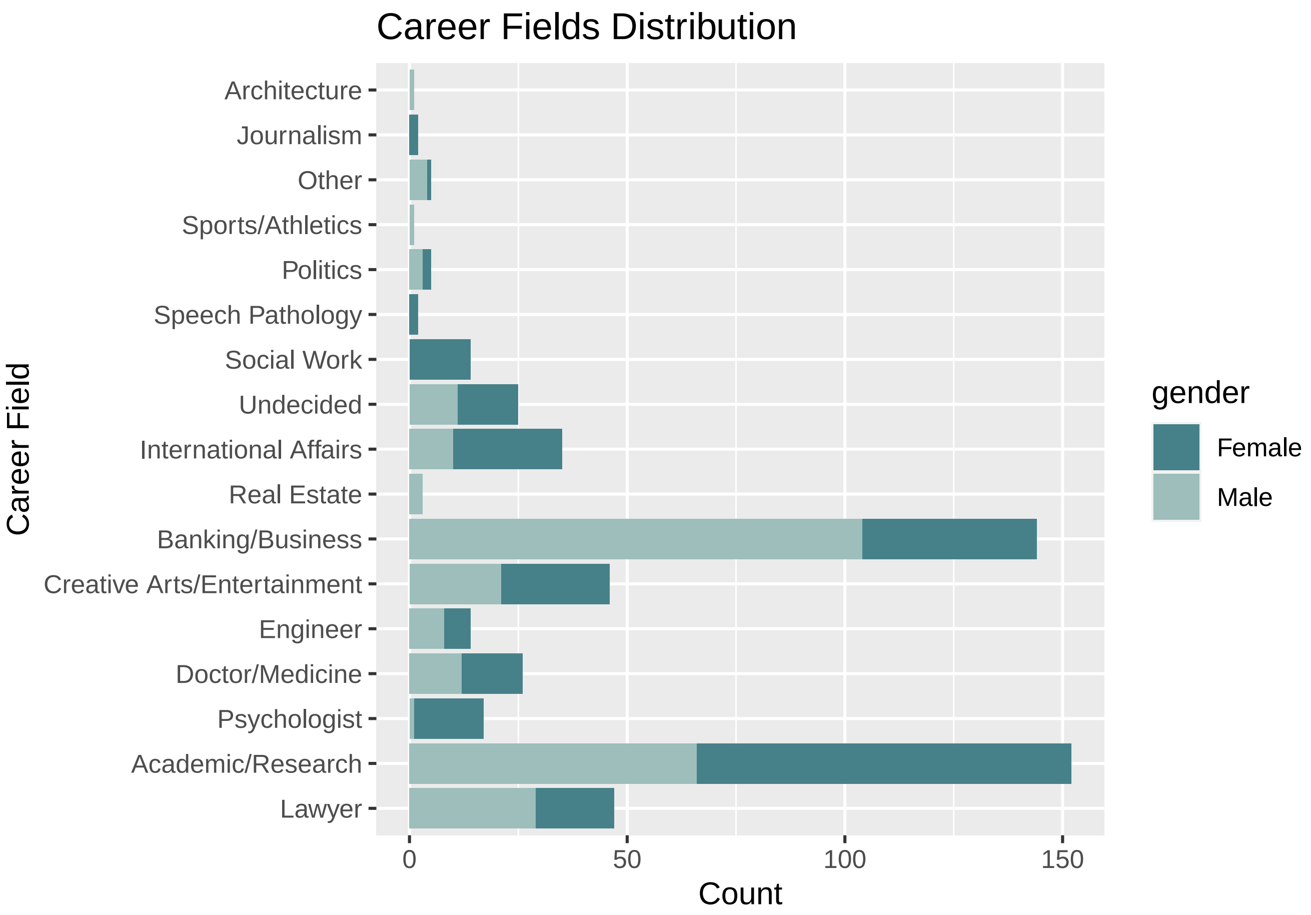
```
ggplot(select(data , race, imprace, gender),
       aes(x=race, y=imprace, fill=gender),
       labels = c("Race", "Race Importance")) +
  geom_boxplot()+
  scale_x_discrete(labels=c("Black", "European", "Latino", "Asian", "Other")) +
  scale_fill_manual(values = c("#468189", "#9DBEBB"),
                    labels = c("Female", "Male")) +
  labs(title= "Race importance and gender ratio of the Data",
       y="Race Importance", x = "Race")
```



- In the given data set we can see that Europeans are the highest among all the races.
- In this survey apart from Asians, the number of females are more compared men.
- There are twice as many as Latino and Black females compared to males.

Career fields of people in the data set

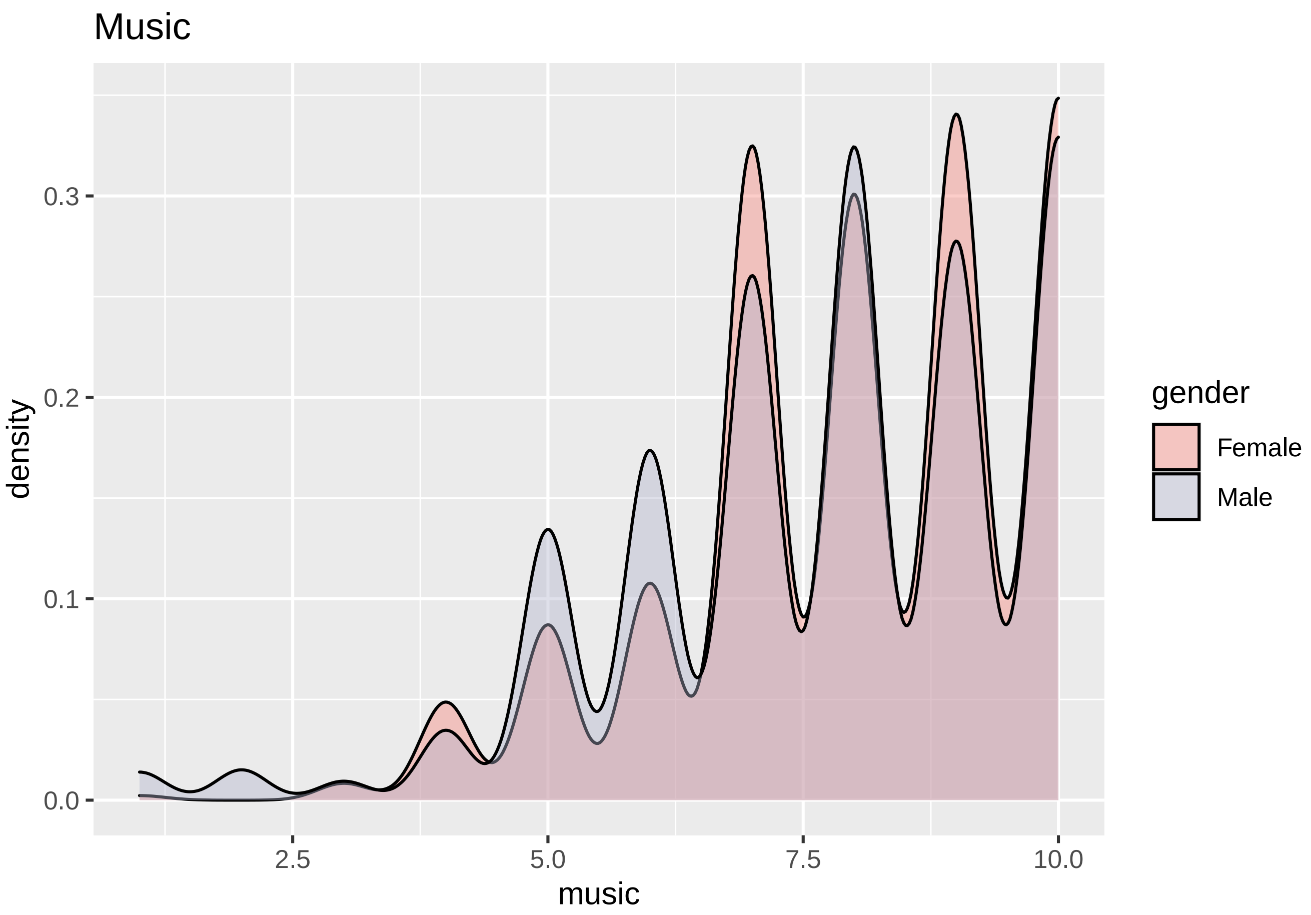
```
career_labels <- c("Lawyer", "Academic/Research", "Psychologist",
                   "Doctor/Medicine", "Engineer", "Creative Arts/Entertainment",
                   "Banking/Business", "Real Estate", "International Affairs",
                   "Undecided", "Social Work", "Speech Pathology", "Politics",
                   "Sports/Athletics", "Other", "Journalism", "Architecture")

ggplot(data = career_data) +
  geom_bar(aes(career_c, fill=gender)) +
  scale_x_discrete(label = career_labels) + coord_flip() +
  labs(title = "Career Fields Distribution",
       x = "Career Field", y = "Count")    +
  scale_fill_manual(values = c("#468189", "#9DBEBB"),
                    labels = c("Female", "Male"))
```



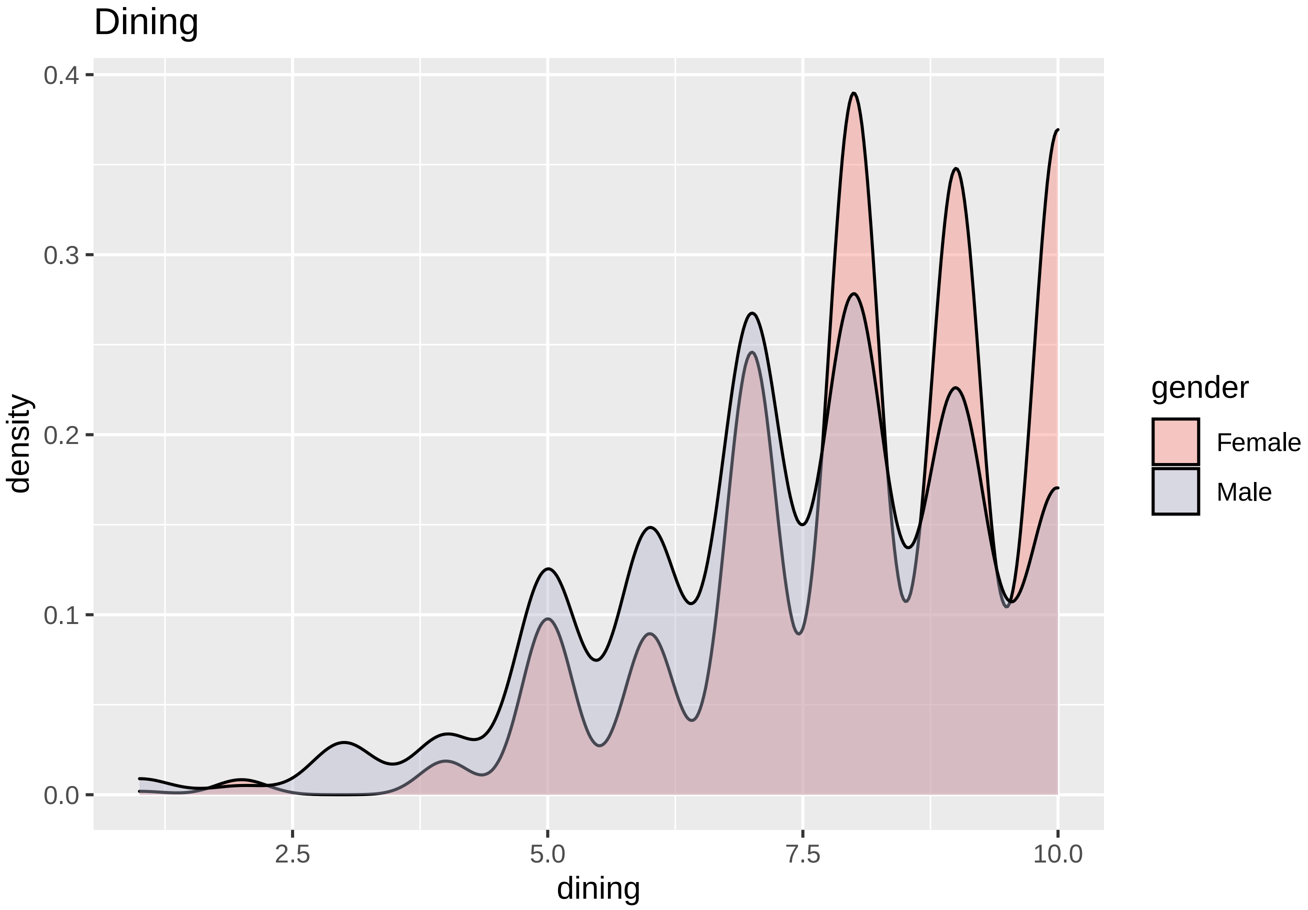
- More number of people are into academic research, followed by business/ banking and lawyers.
- Females dominate in academic research and males in business/banking.
- Few interesting observations is that there are no females who are pursuing a career in architecture, real estate and sports in our data set. Vice-versa, there are no males in journalism, speech pathology.

Interests in various habits of the people in the dataset.

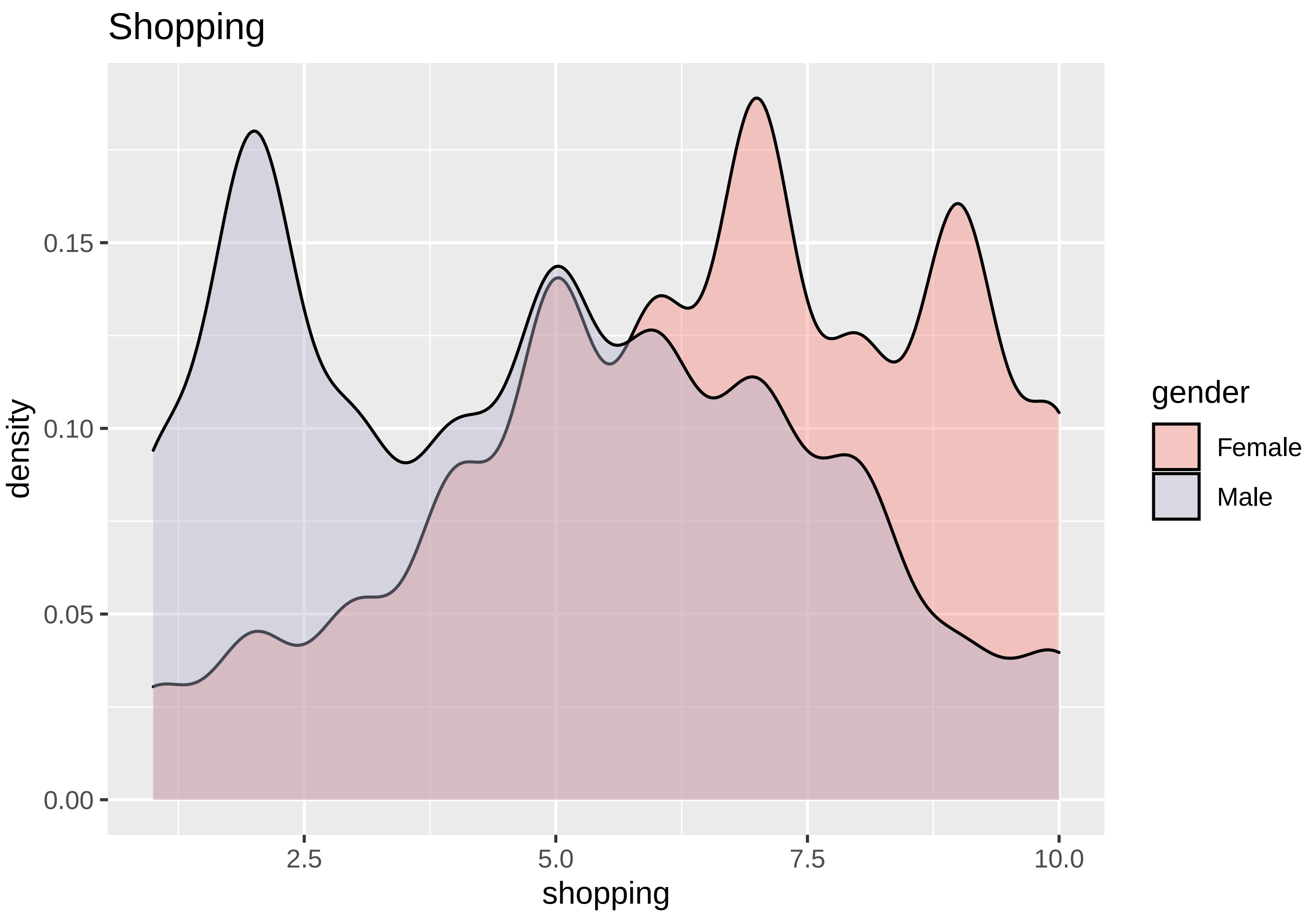
```
# habits list overlapping

habit_list =  colnames(data)[51:67]

ggplot(na.omit(select(data, gender, music)), aes(x=music, fill=gender))+
  geom_density(alpha=0.4) + ggtitle("Music")+
  scale_fill_manual(values = c("#f88379", "#B0B2CC"),
                    labels = c("Female", "Male"))
```



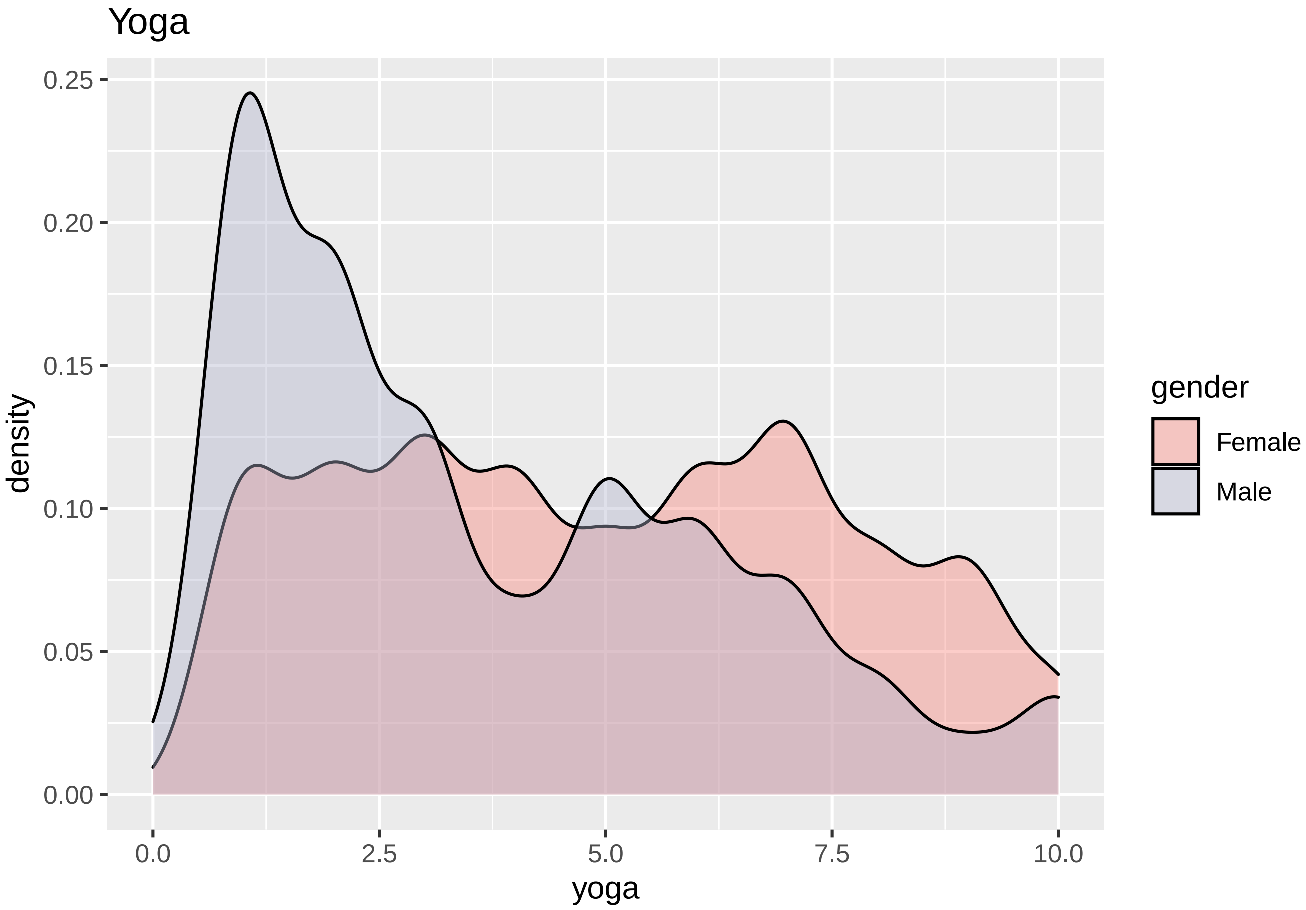
```
ggplot(na.omit(select(data, gender, dining)), aes(x=dining, fill=gender))+
  geom_density(alpha=0.4) + ggtitle("Dining")+
  scale_fill_manual(values = c("#f88379", "#B0B2CC"),
                    labels = c("Female", "Male"))
```



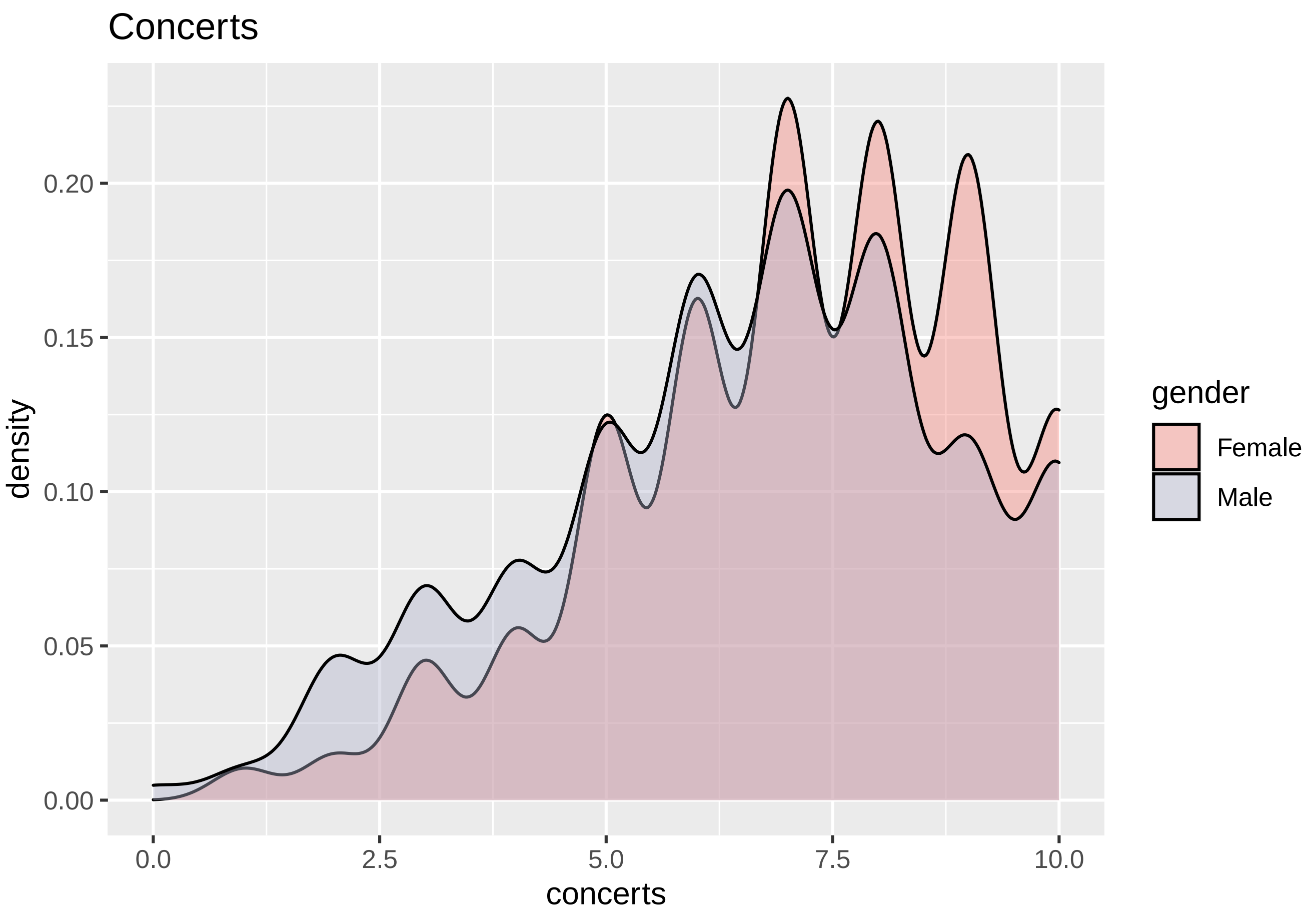
```
ggplot(na.omit(select(data, gender, shopping)), aes(x=shopping, fill=gender))+
  geom_density(alpha=0.4) + ggtitle("Shopping")+
  scale_fill_manual(values = c("#f88379", "#B0B2CC"),
                    labels = c("Female", "Male"))
```



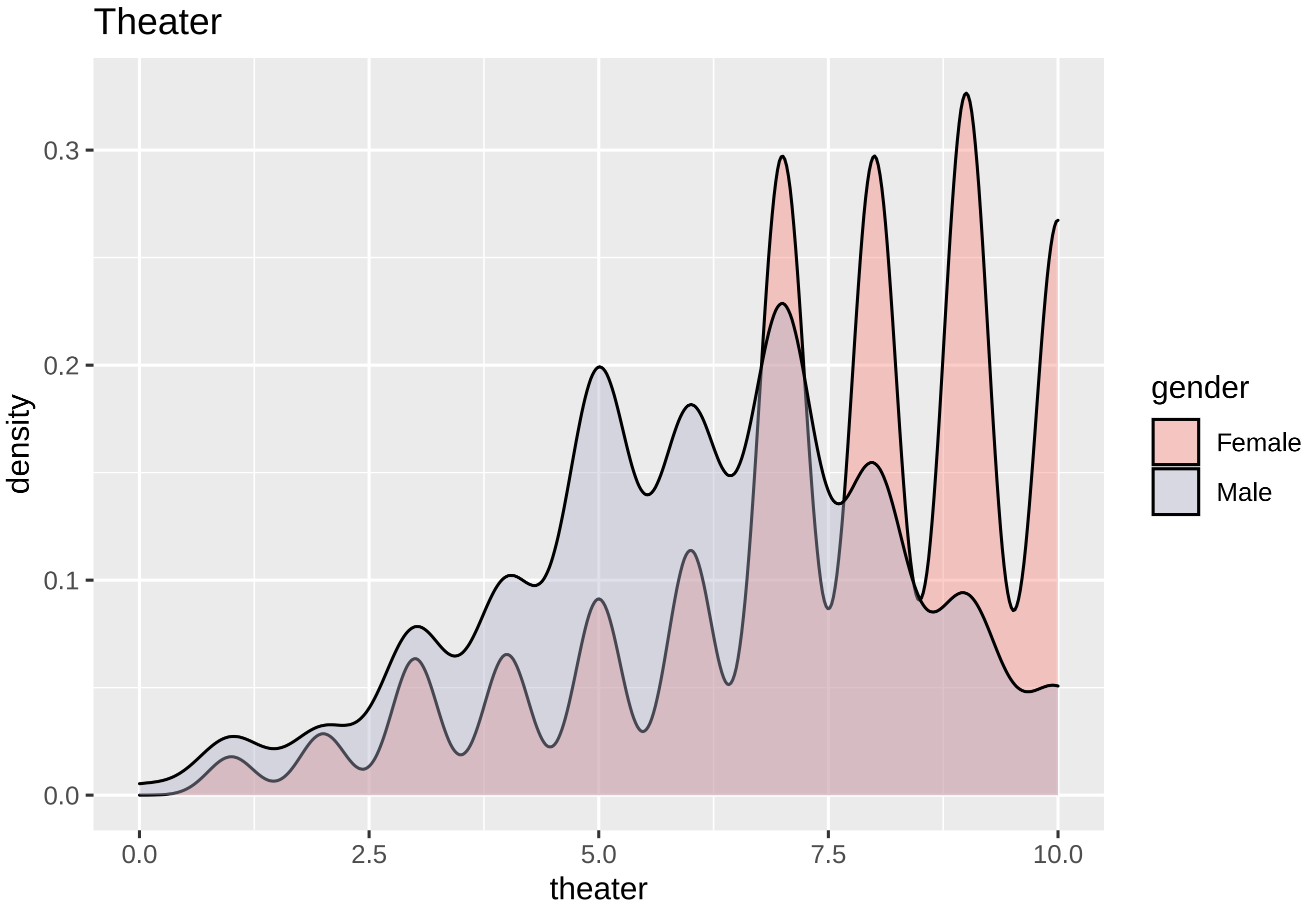
```
ggplot(na.omit(select(data, gender, yoga)), aes(x=yoga, fill=gender))+
  geom_density(alpha=0.4) + ggtitle("Yoga")+
  scale_fill_manual(values = c("#f88379", "#B0B2CC"),
                    labels = c("Female", "Male"))
```



```
ggplot(na.omit(select(data, gender, concerts)), aes(x=concerts, fill=gender))+
  geom_density(alpha=0.4) + ggtitle("Concerts")+
  scale_fill_manual(values = c("#f88379", "#B0B2CC"),
                    labels = c("Female", "Male"))
```



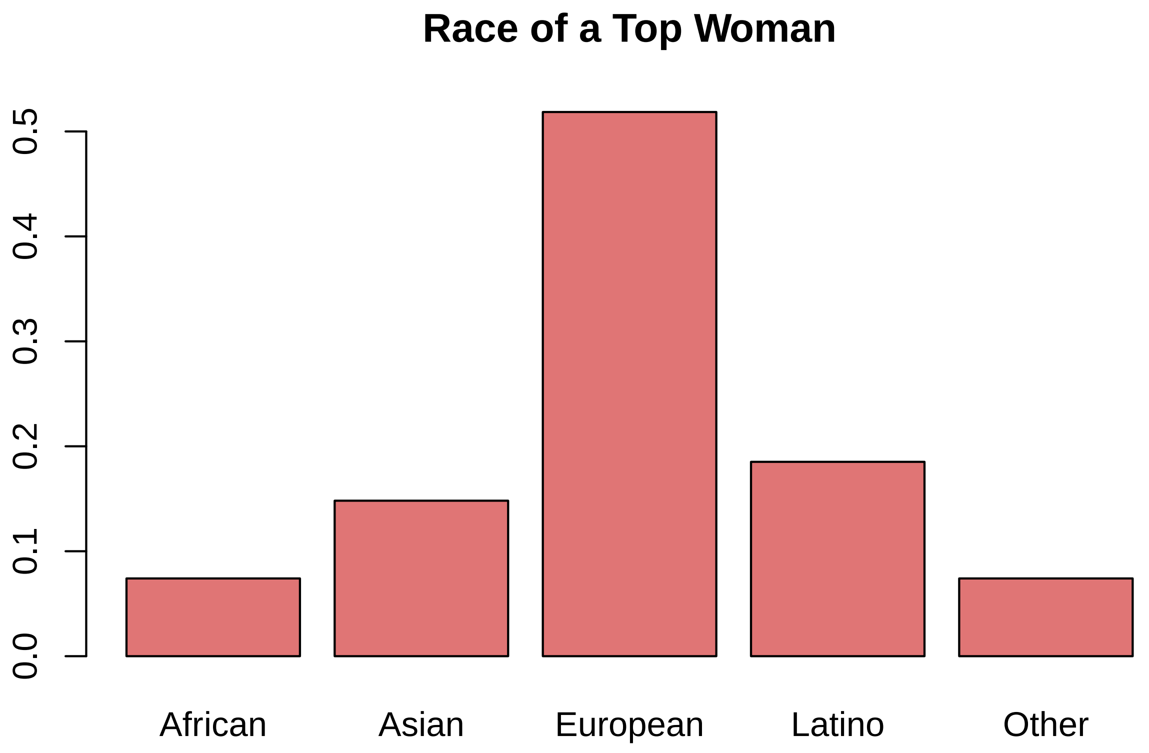
```
ggplot(na.omit(select(data, gender, theater)), aes(x=theater, fill=gender))+
  geom_density(alpha=0.4) + ggtitle("Theater")+
  scale_fill_manual(values = c("#f88379", "#B0B2CC"),
                    labels = c("Female", "Male"))
```



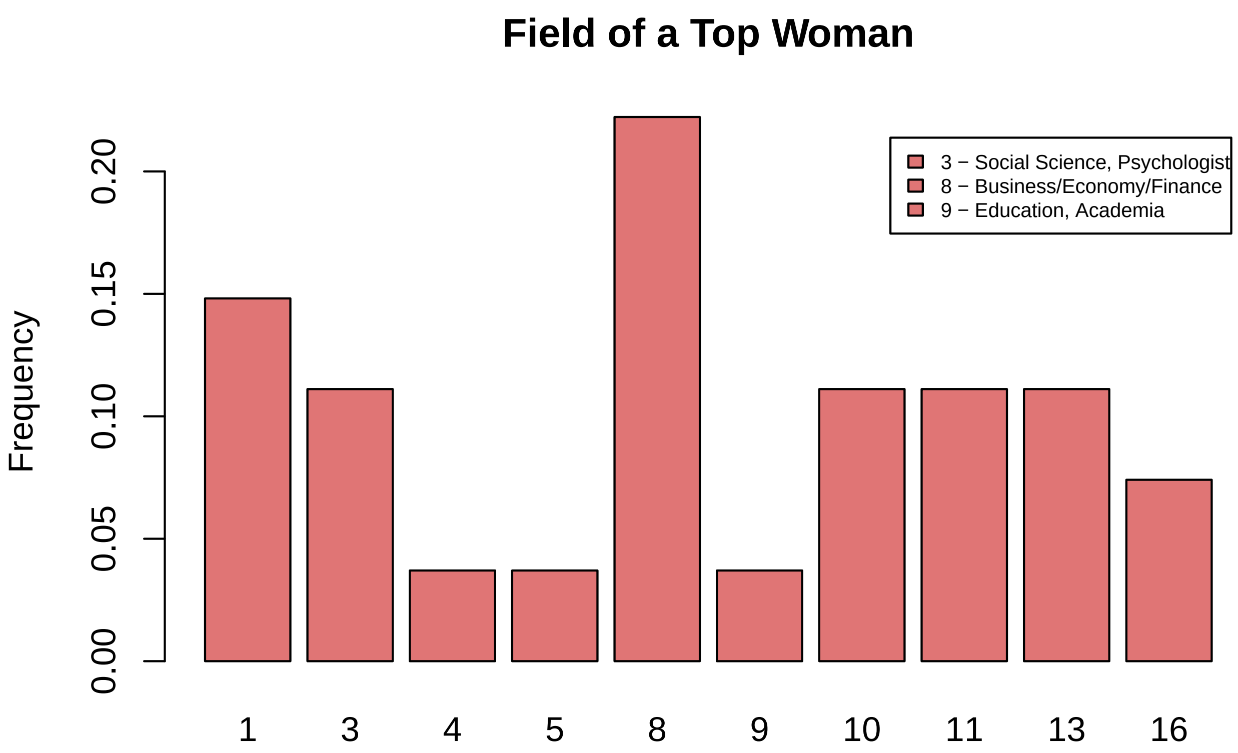
* We can see that there are many similarities in habits such as music, dining, and concerts. * The contrast is seen clearly in shopping, yoga, theater habits for male and female.

Top Men and Women – Most desirable (people who got a second date)

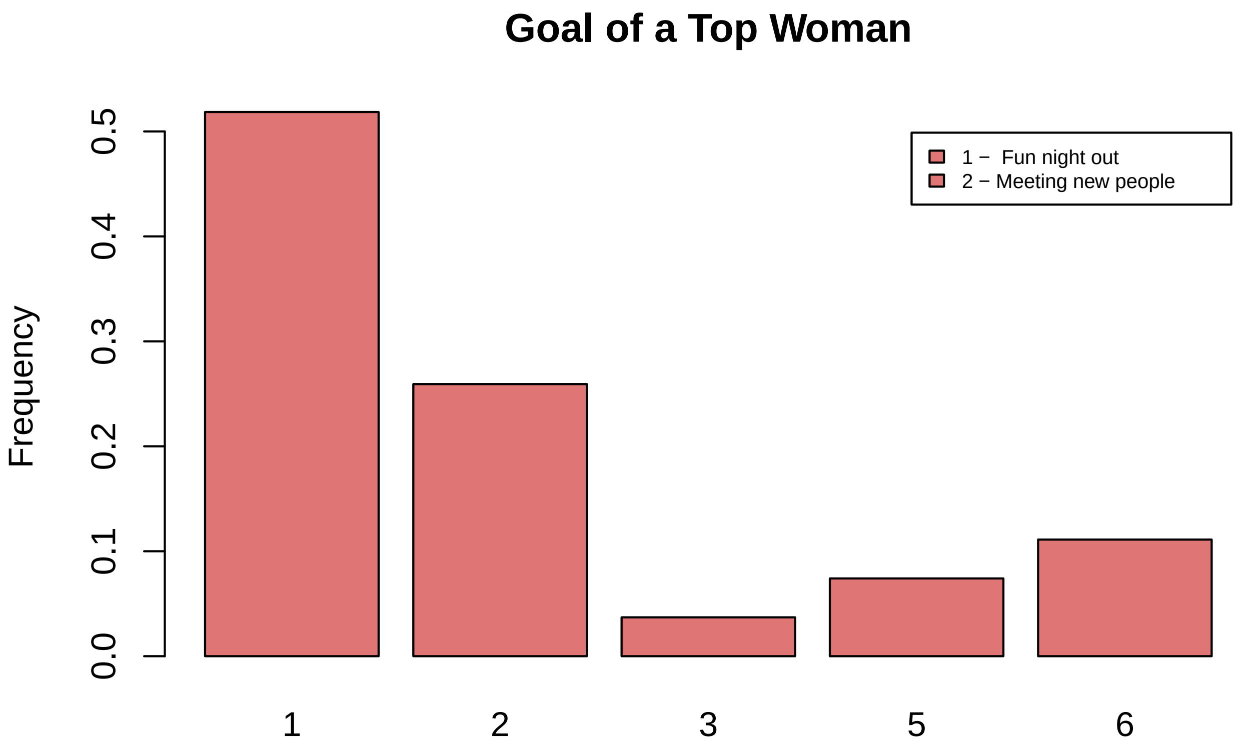
```
# Top Women
# Race
barplot(col=rgb(0.8,0.1,0.1,0.6), prop.table(table(TopWoman$race)),
        main="Race of a Top Woman")
```



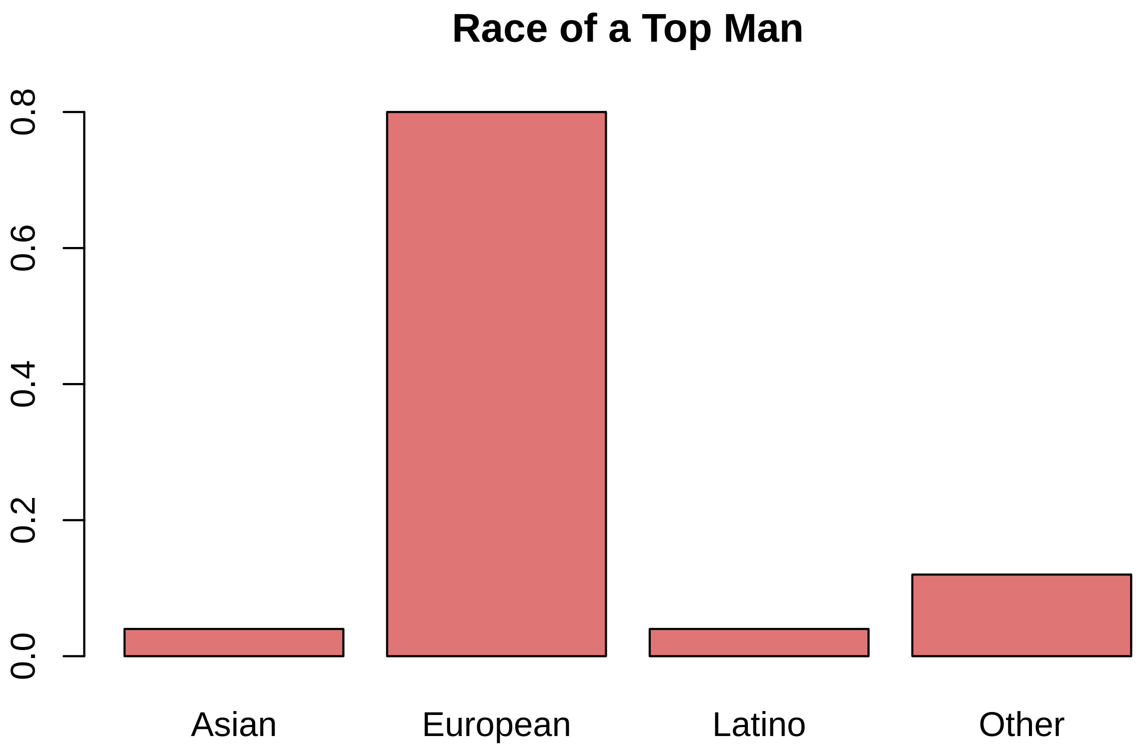
```
# Career field
barplot(col=rgb(0.8,0.1,0.1,0.6), prop.table(table(TopWoman$field_cd)),
        main="Field of a Top Woman",
        legend.text = c("3 - Social Science, Psychologist",
                        "8 - Business/Economy/Finance", "9 - Education, Academia "
        ), args.legend = list(text.width = 3.3, xjust = 1, cex = 0.58),
        ylab = "Frequency")
```



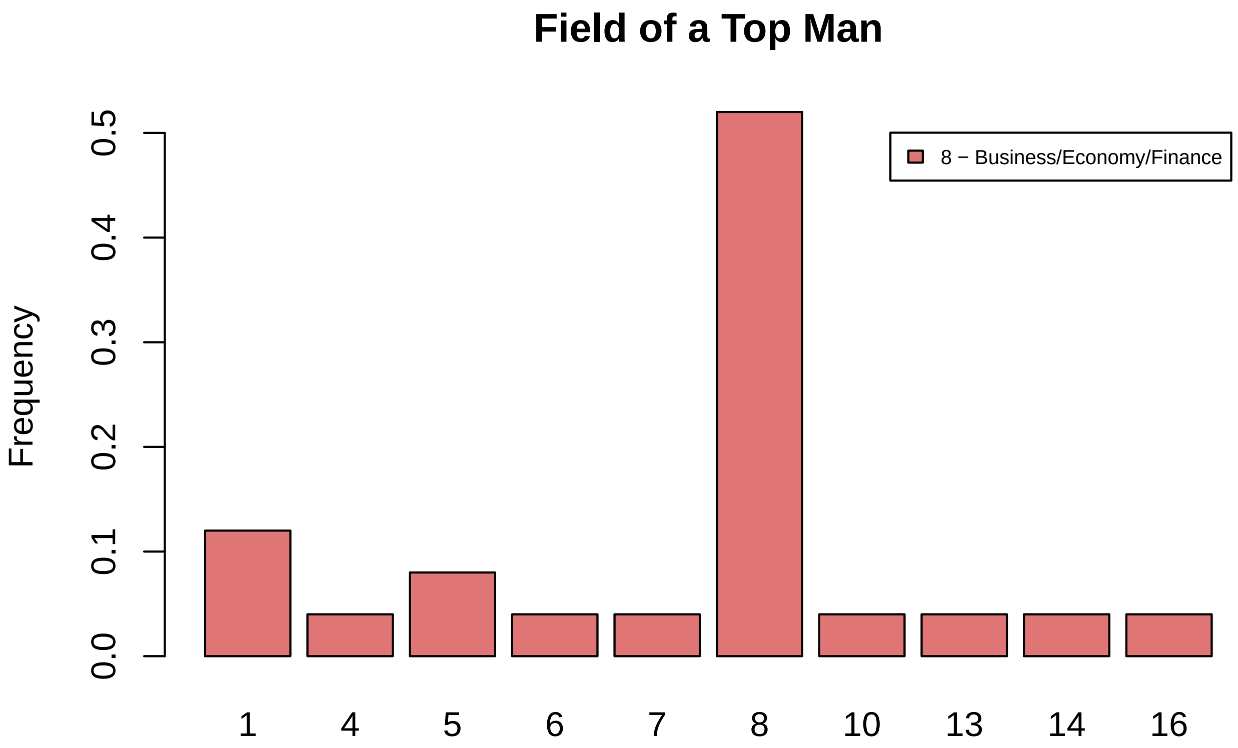
```r
# goal in a date
barplot(col=rgb(0.8,0.1,0.1,0.6), prop.table(table(TopWoman$goal)),
        main="Goal of a Top Woman",
        legend.text = c("1 -  Fun night out" , "2 - Meeting new people"
        ), args.legend = list(text.width = 1.5, xjust = 1, cex = 0.58),
        ylab = "Frequency")
```



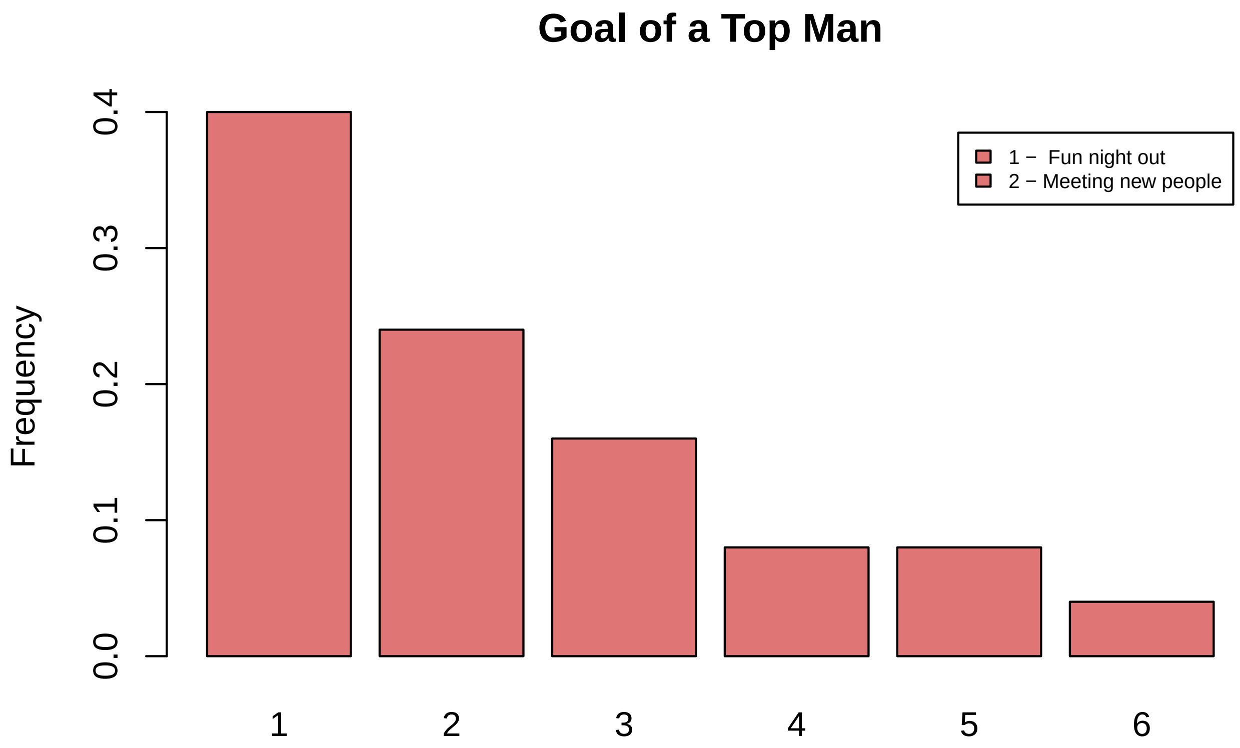
```
# Top Men
# race
barplot(col=rgb(0.8,0.1,0.1,0.6), prop.table(table(TopMan$race)),
        main="Race of a Top Man")
```



```r
#career field
barplot(col=rgb(0.8,0.1,0.1,0.6), prop.table(table(TopMan$field_cd)),
        main="Field of a Top Man",
        legend.text = c("8 - Business/Economy/Finance"
        ), args.legend = list(text.width = 3.3, xjust = 1, cex = 0.58),
        ylab = "Frequency")
```



```
# Goal in a date
barplot(col=rgb(0.8,0.1,0.1,0.6), prop.table(table(TopMan$goal)),
        main="Goal of a Top Man",
        legend.text = c("1 -  Fun night out" , "2 - Meeting new people"
        ), args.legend = list(text.width = 1.5, xjust = 1, cex = 0.58),
        ylab = "Frequency")
```



Conclusions for Top men and Women:

- European men are most desired and Asian men are least desired.
- The goal of top men and women are to have a fun night out and to meet new people.
- Top women are pursuing a career in business/finance/banking and Top men are also pursuing a career in business/finance/banking by large margin.
- European women are most desired and African women are least desired.
- Top women are almost equally distributed in different type of career fields, but top men are predominantly in business/finance/banking.

EM_cluster and overestimation

Group Project

12/16/2020

Selecting required rows

How are interests,race and age co-related with positive decision from a person?

To answer this we have constructed a correlation map between interests, race of the person and their partner along with their positive response percentage.

Now to identify if there are any patterns or similarites between them, we have used EM algorithm to classify the data into 3 cluster. Expectation–maximization (EM) algorithm is an iterative method to find (local) maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. Under interests I have taken “goal,date,sports,tvsports,dining,museums,art,hiking,gaming,clubbing,reading,tv,theater,movies,concerts,music,shopping
fields of the data on a sclae of 1 to 10 a person rates their interest in each field.

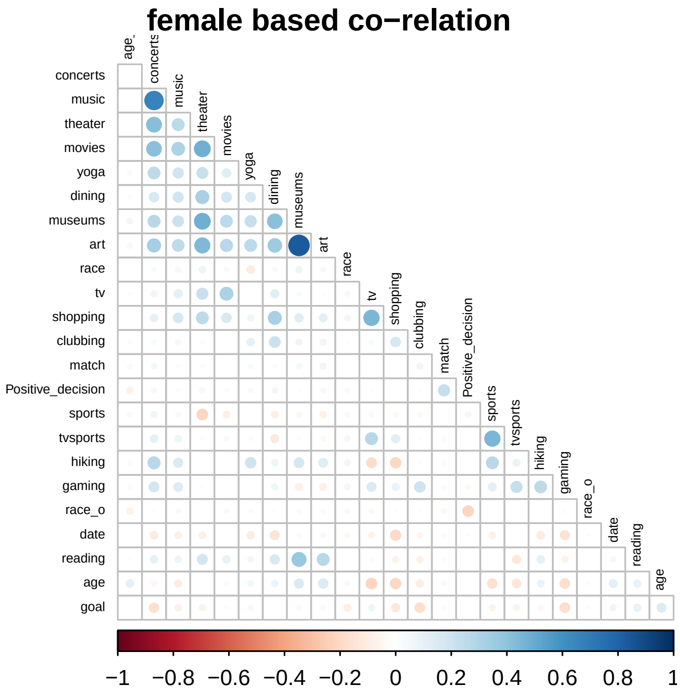
```
pre_cleaned_data_frame <-
  raw_data %>%
  dplyr::select(pid,gender,race,race_o,age,age_o,goal,date,sports,tvsports,dining,museums,art,hiki

test2 <-
  partial_cleaned_data %>%
  group_by(pid, gender) %>%
  summarise(Positive_decision = round(mean(dec), digits = 3))
```

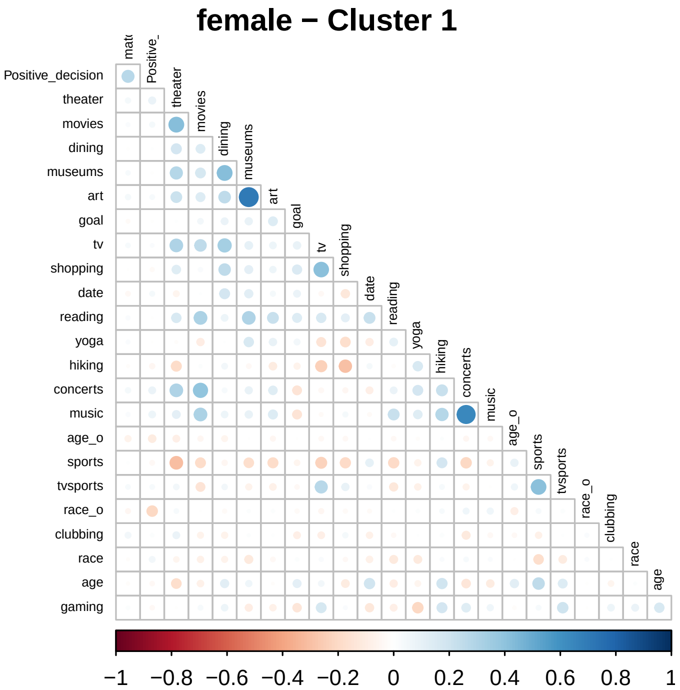
```
## `summarise()` regrouping output by 'pid' (override with `.groups` argument)
```

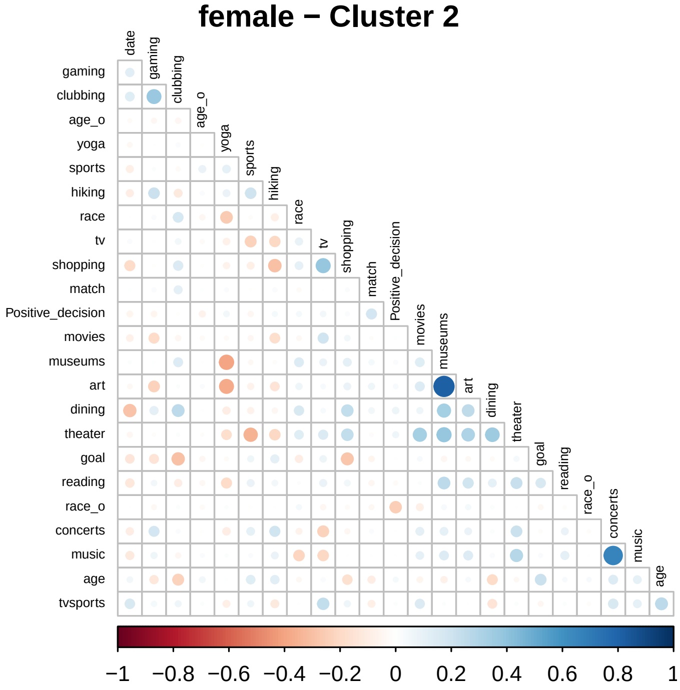
```
test3 <-merge(pre_cleaned_data_frame, test2, by.x=c("pid", "gender"), by.y=c("pid", "gender"))

pre_cleaned_female_data <-test3[ which(test3$gender==0), ]
female_data <- pre_cleaned_female_data  %>% dplyr::select(-pid,-gender)
pre_cleaned_male_data <-test3[ which(test3$gender==1), ]
male_data <- pre_cleaned_male_data  %>% dplyr::select(-pid,-gender)

EM_algorithm <- function(data ,gender){
  x <- cor(data)
  corrplot(x, type = "lower",method = "circle", tl.cex = 0.5, tl.col = 'black',
         title = paste0(gender ," based co-relation"),
         order = "hclust", diag = FALSE, mar=c(0,0,1,0))

  # Use Kmeans to generate 3 clusters
  comps <- kmeans(data, 3)$cluster
  # A temporary dataframe to store cluster information
  temp = cbind(data, comps)
  head(temp)
  # Defining each partition
  data1 = temp[temp[, "comps"] == 1,]
```

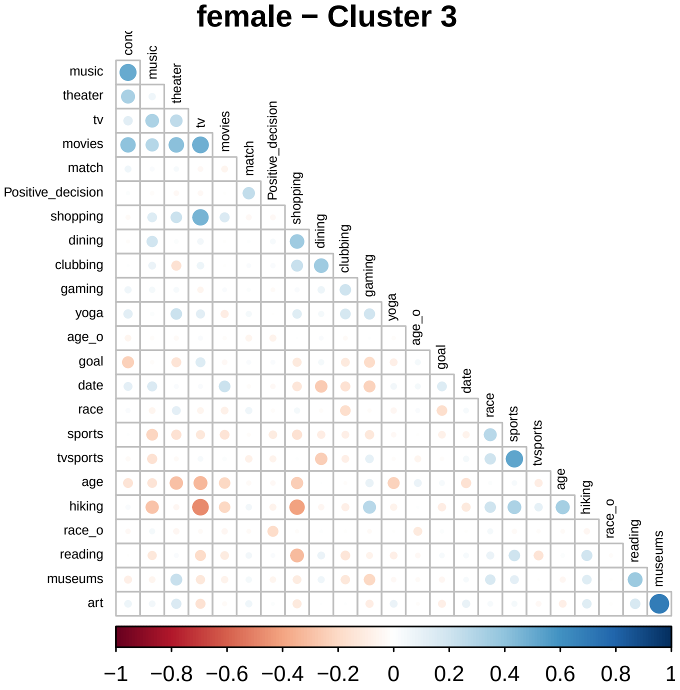
```
data2 = temp[temp[, "comps"] == 2,]
data3 = temp[temp[, "comps"] == 3,]

data1 = subset(data1, select = -c(comps) )
data2 = subset(data2, select = -c(comps) )
data3 = subset(data3, select = -c(comps) )

# Initial parametars for mu and covariance
mu1 = colMeans(data1)
mu2 = colMeans(data2)
mu3 = colMeans(data3)

cov1 = cov(data1)
cov2 = cov(data2)
cov3 = cov(data3)

# Initial parameters for mix probabilities
pi1 <- sum(comps==1)/length(comps)
pi2 <- sum(comps==2)/length(comps)
pi3 <- sum(comps==3)/length(comps)

# Converting data into matrix format
data = data.matrix(data)

# Defining functions to sum
sum.finite <- function(x) {
  colSums(x)
}

normal_sum.finite <- function(x) {
  sum(x[is.finite(x)])
}

# Initilizing log likelihood value
Q <- 0
Q[2] <- normal_sum.finite(log(pi1)+log(dmvnorm(data, mu1, cov1)))
+ normal_sum.finite(log(pi2)+log(dmvnorm(data, mu2, cov2)))
+ normal_sum.finite(log(pi3)+log(dmvnorm(data, mu3, cov3)))

k <- 2

# Looping until convergence
while (abs(Q[k]-Q[k-1]) >= 1e-6) {
  # E step
  # Computing denominator values
  comp1 <- pi1 * dmvnorm(data, mu1, cov1)
  comp2 <- pi2 * dmvnorm(data, mu2, cov2)
  comp3 <- pi3 * dmvnorm(data, mu3, cov3)

  comp.sum <- comp1 + comp2 + comp3
  # Assigning responsibility values
  p1 <- comp1/comp.sum
  p2 <- comp2/comp.sum
```

```
    p3 <- comp3/comp.sum


    # M step
    # Re calculating parameter values using formula
    pi1 <- normal_sum.finite(p1) / dim(data)[1]
    pi2 <- normal_sum.finite(p2) / dim(data)[1]
    pi3 <- normal_sum.finite(p3) / dim(data)[1]


    # Calculating mean values

    mu1 <- sum.finite(p1 * data) / normal_sum.finite(p1)
    mu2 <- sum.finite(p2 * data) / normal_sum.finite(p2)
    mu3 <- sum.finite(p3 * data) / normal_sum.finite(p3)


    # Calculating covariance matrices
    cov1 <- (( t(sweep(data,2, mu1)) %*% (p1 * sweep(data,2, mu1)) ) / normal_sum.finite(p1))
    cov2 <- (( t(sweep(data,2, mu2)) %*% (p2 * (sweep(data,2, mu2))) ) / normal_sum.finite(p2))
    cov3 <- (( t(sweep(data,2, mu3)) %*% (p3 * (sweep(data,2, mu3))) ) / normal_sum.finite(p3))


    k <- k + 1
    # Changing log-likelihood
    Q[k] <- sum(log(comp.sum))
  }

  x <- cor(data1)
  corrplot(x, type = "lower",method = "circle", tl.cex = 0.5, tl.col = 'black',
         title = paste0(gender ," - Cluster 1"),
         order = "hclust", diag = FALSE, mar=c(0,0,1,0))
  x <- cor(data2)
  corrplot(x, type = "lower", method = "circle", tl.cex = 0.5, tl.col = 'black',
         title = paste0(gender ," - Cluster 2"),
         order = "hclust", diag = FALSE, mar=c(0,0,1,0))
  x <- cor(data3)
  corrplot(x, type = "lower", method = "circle", tl.cex = 0.5, tl.col = 'black',
         title = paste0(gender ," - Cluster 3"),
         order = "hclust", diag = FALSE, mar=c(0,0,1,0))

}

female_data <- tidyr::drop_na(female_data)
male_data <- tidyr::drop_na(male_data)

EM_algorithm(female_data,"female")
```

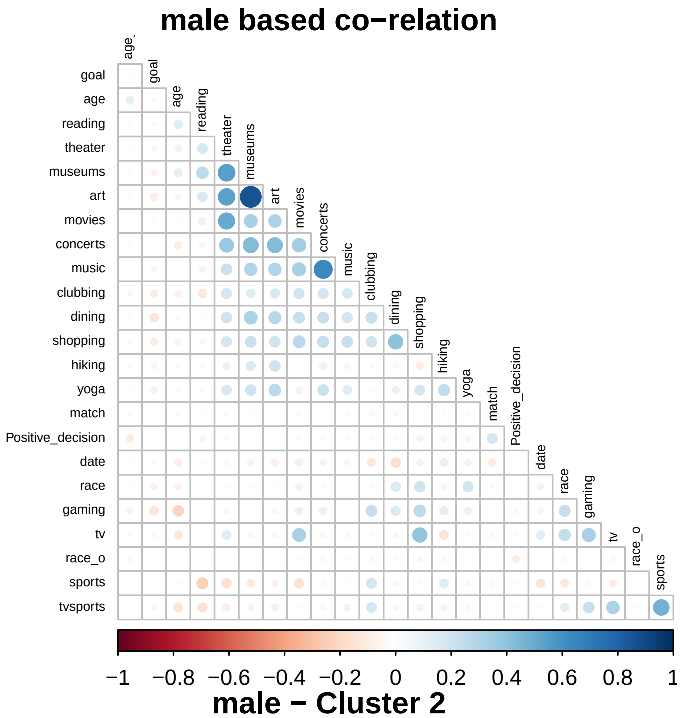




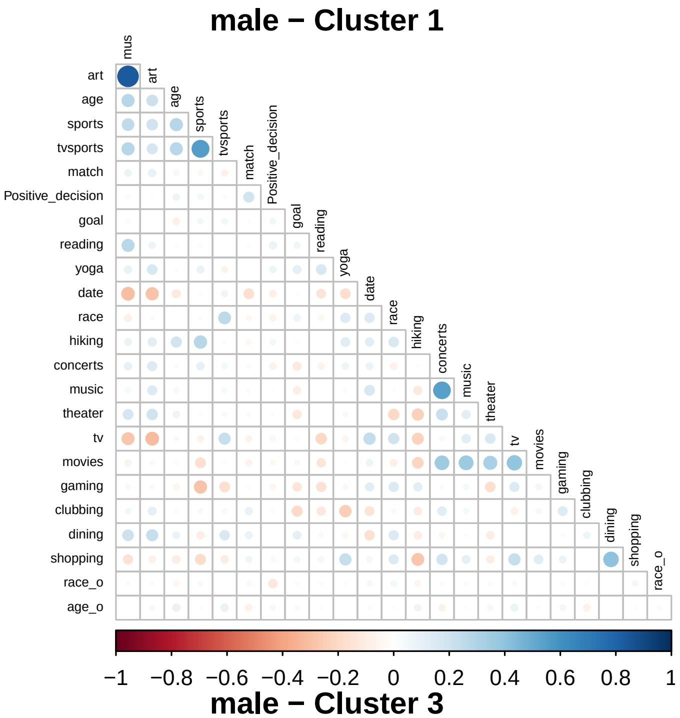




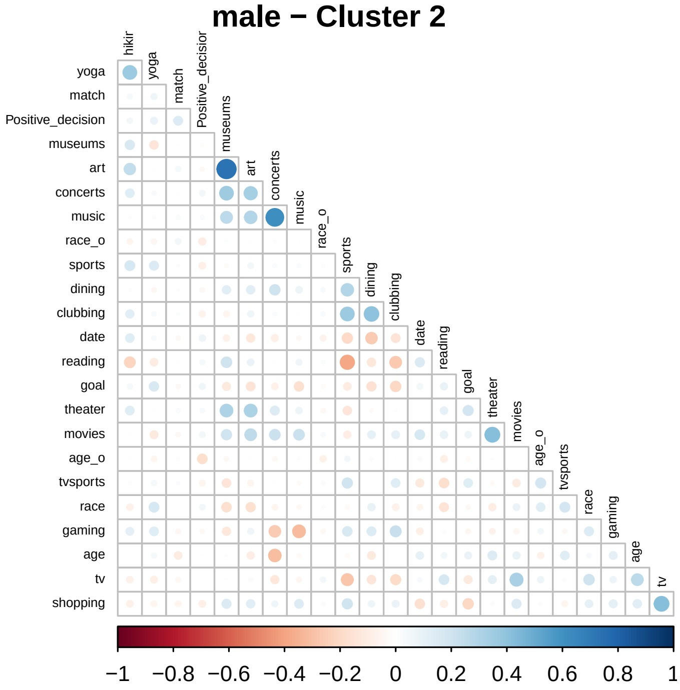
```
EM_algorithm(male_data,"male")
```



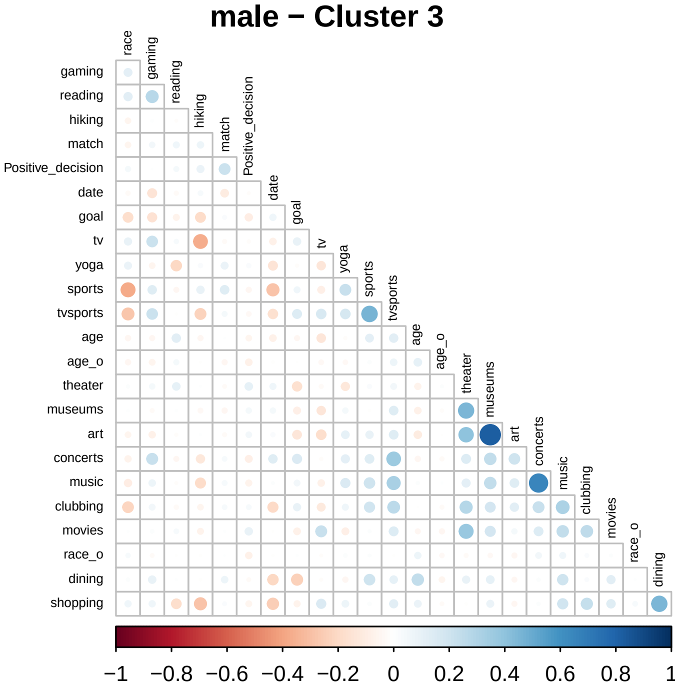
male based co−relation



male − Cluster 1



male − Cluster 2



male − Cluster 3

Inferences based on results:

Although, all the 3 clusters have varied interest co-relation between them, we can see for women race of the opposite gender(race_o) has the highest negative correlation with positive decision from their partner. This means that they are more interested in people of other races.

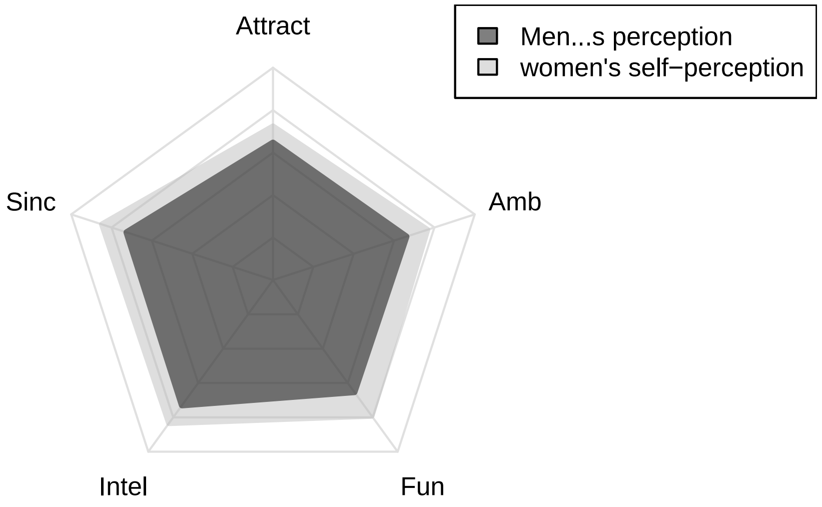
Men in general have negative co-relation with their partners age and gender. We can see that the data with common interests have been clustered together. For instance, cluster 3 in males has people who are more interested in concerts,music,reading,art,muesum and theater. While, cluster 2 has people more interested in shopping and dining.

```
male_data = partial_cleaned_data[partial_cleaned_data$gender =="1", ]

female_data = partial_cleaned_data[partial_cleaned_data$gender =="0", ]
```

```
#Female
Attract = c ( (mean(female_data$attr3_1 , na.rm = T)) ,( mean(female_data$attr_o , na.rm = T)))
Sinc = c ( (mean(female_data$sinc3_1 , na.rm = T)) ,( mean(female_data$sinc_o , na.rm = T)))
Intel  = c ( (mean(female_data$intel3_1 , na.rm = T)) ,( mean(female_data$intel_o , na.rm = T)))
Fun   = c ( (mean(female_data$fun3_1 , na.rm = T)) ,( mean(female_data$fun_o , na.rm = T)))
Amb = c ( (mean(female_data$amb3_1 , na.rm = T)) ,( mean(female_data$amb_o , na.rm = T)))
df = data.frame(Attract ,Sinc, Intel, Fun, Amb)

df = t(df)
df = as.data.frame(df)

colnames(df)[1]= "women_self_perception"
colnames(df)[2]= "mens_perception"

radar_women = t(df)
radar_women = as.data.frame(radar_women)

maxmin <- data.frame(
  Attract = c(10, 0),
  Sinc = c(10, 0),
  Intel = c(10, 0),
  Fun = c(10, 0),
  Amb = c(10, 0))
radar_women_max <- rbind(maxmin, radar_women)

radarchart(radar_women_max,
           pty = 32,
           axistype = 0,
           pcol = c(adjustcolor("gray", 0.5), adjustcolor("black", 0.5)),
           pfcol = c(adjustcolor("gray", 0.5), adjustcolor("black", 0.5)),
           plty = 1,
           plwd = 3,
           cglty = 1,
           cglcol = "gray88",
           centerzero = TRUE,
           seg = 5,
           vlcex = 0.75,
           palcex = 0.75)


legend("topright",
       c("Men’s perception", "women's self-perception"), box.lty=1, box.lwd=1,cex = 0.75,
       fill = c(adjustcolor("black", 0.5), adjustcolor("gray", 0.5)))
```

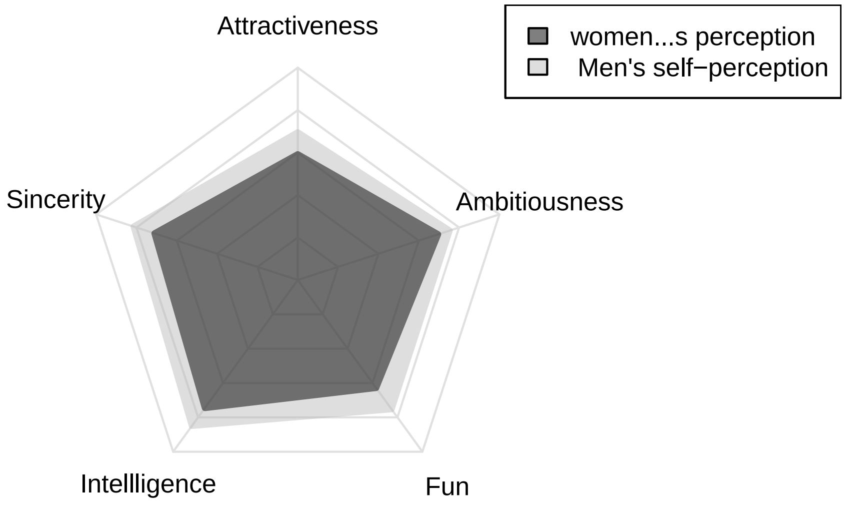
```
## Warning in strwidth(legend, units = "user", cex = cex): conversion failure on
## 'Men’s perception' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in strwidth(legend, units = "user", cex = cex): conversion failure on
## 'Men’s perception' in 'mbcsToSbcs': dot substituted for <80>

## Warning in strwidth(legend, units = "user", cex = cex): conversion failure on
## 'Men’s perception' in 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x, y, ...): conversion failure on 'Men’s perception' in
## 'mbcsToSbcs': dot substituted for <e2>
## Warning in text.default(x, y, ...): conversion failure on 'Men’s perception' in
## 'mbcsToSbcs': dot substituted for <80>
```

```
## Warning in text.default(x, y, ...): conversion failure on 'Men’s perception' in
## 'mbcsToSbcs': dot substituted for <99>
```



```
col <- c( "Attract" ,"Sinc", "Intel", "Fun", "Amb")

#Male

Attractiveness = c ( (mean(male_data$attr3_1 , na.rm = T)) ,( mean(male_data$attr_o , na.rm = T)))

Sincerity = c ( (mean(male_data$sinc3_1 , na.rm = T)) ,( mean(male_data$sinc_o , na.rm = T)))
Fun    = c ( (mean(male_data$fun3_1 , na.rm = T)) ,( mean(male_data$fun_o , na.rm = T)))
Ambitiousness = c ( (mean(male_data$amb3_1 , na.rm = T)) ,( mean(male_data$amb_o , na.rm = T)))
Intellligence  = c ( (mean(male_data$intel3_1 , na.rm = T)) ,( mean(male_data$intel_o , na.rm = T
df_m = data.frame(Attractiveness ,Sincerity, Intellligence, Fun, Ambitiousness)
df_m = t(df_m)
df_m = as.data.frame(df_m)
colnames(df_m)[1]= "men_self_perception"
colnames(df_m)[2]= "womens_perception"

radar_men = t(df_m)
radar_men = as.data.frame(radar_men)
maxmin <- data.frame(
  Attractiveness = c(10, 0),
  Sincerity = c(10, 0),
  Intellligence = c(10, 0),
  Fun = c(10, 0),
  Ambitiousness = c(10, 0))
radar_men_max <- rbind(maxmin, radar_men)

radarchart(radar_men_max,
           pty = 32,
           axistype = 0,
           pcol = c(adjustcolor("gray", 0.5), adjustcolor("black", 0.5)),
           pfcol = c(adjustcolor("gray", 0.5), adjustcolor("black", 0.5)),
           plty = 1,
           plwd = 3,
           cglty = 1,
           cglcol = "gray88",
           centerzero = TRUE,
           seg = 5,
           vlcex = 0.75,
           palcex = 0.75)

legend("topright",
       c("women’s perception", " Men's self-perception"), box.lty=1, box.lwd=1,cex = 0.75,
       fill = c(adjustcolor("black", 0.5), adjustcolor("gray", 0.5)))
```

```
## Warning in strwidth(legend, units = "user", cex = cex): conversion failure on
## 'women’s perception' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in strwidth(legend, units = "user", cex = cex): conversion failure on
## 'women’s perception' in 'mbcsToSbcs': dot substituted for <80>

## Warning in strwidth(legend, units = "user", cex = cex): conversion failure on
## 'women’s perception' in 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x, y, ...): conversion failure on 'women’s perception'
## in 'mbcsToSbcs': dot substituted for <e2>

## Warning in text.default(x, y, ...): conversion failure on 'women’s perception'
## in 'mbcsToSbcs': dot substituted for <80>

## Warning in text.default(x, y, ...): conversion failure on 'women’s perception'
## in 'mbcsToSbcs': dot substituted for <99>
```



We can see that both men and women over-estimate themselves in all the attributes. Women over-estimate themselves mostly in fun and least in attractiveness attributes. Suprisingly, men also over-estimate their fun the most, however they over-estimate themselves least in ambitiousness.

Predicting_match

Group Project

12/16/2020

```
library(fmsb)
library(dplyr)
library(tibble)
library(stringr)
library(ggplot2)
library(grid)
library(gridBase)
library(scales)
library(tidyverse)
library(randomForest)
library(xgboost)
```

Goal

Goal is to predict match between two persons given a set of attributes between them. In the experiment, we have data of match between persons and qualities of the persons involved in match. Using these details, we'll build a model which tries to predict a match.

Creating Pairs data

Form the given dataset, we create the data frame for pairs. This data frame should have

1) Male attributes(Details, Interests, Data given in survey etc.)
2) Male attributes(Details, Interests, Data given in survey etc.)
3) Match. This is Boolean 0 or 1

```
raw_data <- read.csv('D:/Study/Stat/Final_project/Datasets/Speed_Dating/Speed Dating Data.csv', he

# Selecting required features
use_features = c("iid", "gender", "wave", "pid", "match", "samerace", "age_o", "race_o",
                 "pf_o_att", "pf_o_sin", "pf_o_int","pf_o_fun", "pf_o_amb", "pf_o_sha",
                 "age", "field_cd", "race", "imprace", "imprelig", "goal", "date",
                 "go_out", "sports",
                 "tvsports", "exercise", "dining", "museums", "art", "hiking",
                 "gaming", "clubbing",
                 "reading", "tv", "theater", "movies", "concerts", "music",
                 "shopping", "yoga", "exphappy",
                 "attr1_1", "sinc1_1", "intel1_1", "fun1_1", "amb1_1", "shar1_1",
                 "attr2_1", "sinc2_1",
                 "intel2_1", "fun2_1", "amb2_1", "shar2_1", "attr3_1", "sinc3_1",
                 "fun3_1", "intel3_1", "amb3_1")



data = raw_data[use_features]

mdata_df = data[data[,'gender']==1,]
fdata_df = data[data[,'gender']==0,]

# selecting male features. This will be matched with pis for female

cmfeatures = c('iid', 'pid', 'match', 'samerace', 'age', 'field_cd', 'race',
               'imprace', 'imprelig', 'goal', 'date', 'go_out',
              'sports', 'tvsports', 'exercise', 'dining', 'museums', 'art',
              'hiking', 'gaming', 'clubbing',
              'reading', 'tv', 'theater', 'movies', 'concerts', 'music',
              'shopping', 'yoga', 'exphappy',
              'attr1_1', 'sinc1_1', 'intel1_1', 'fun1_1', 'amb1_1', 'shar1_1',
              'attr2_1', 'sinc2_1',
              'intel2_1', 'fun2_1', 'amb2_1', 'shar2_1', 'attr3_1', 'sinc3_1',
              'fun3_1', 'intel3_1', 'amb3_1')

new_mdata = mdata_df[cmfeatures]

cffeatures = c(cmfeatures[1] , cmfeatures[c(5:length(cmfeatures))])

new_fdata = fdata_df[cffeatures]

# Merging data using pid and iid columns
pair_data = merge_data = merge(x=new_mdata, y=new_fdata, by.x="pid", by.y="iid")


# Storing data in csv for future use
write.csv(pair_data, file = 'D:/Study/Stat/Final_project/Datasets/Speed_Dating/Pairs_Data.csv', ro
```

```
pairs_data <- read.csv('D:/Study/Stat/Final_project/Datasets/Speed_Dating/Pairs.csv', header = T,
```

Now we use this data and apply different models for this.

Logistic Regression

Using regression to find match between two people. Here input will be attributes of both men and women. Output will be Match(Boolean)

```
# Code for Regression


# Sigmoid for squashing output between 0 and 1
sigmoid <- function(z){1/(1+exp(-z))}
```

```r
# Cost function. Using cross-entropy
cost <- function(theta, X, y){
  m <- length(y)
  h <- sigmoid(X %*% theta)
  J <- (t(-y)%*%log(h)-t(1-y)%*%log(1-h))/m
  J
}


# Gradient function to be given to optim
grad <- function(theta, X, y){
  m <- length(y)

  h <- sigmoid(X%*%theta)
  grad <- (t(X)%*%(h - y))/m
  grad
}


# Main Logistic Regression function
logisticReg <- function(X, y, max_iters){
  X <- na.omit(X)
  y <- na.omit(y)
  # Adding Bias to first column
  X <- mutate(X, bias =1)
  X <- as.matrix(X[, c(ncol(X), 1:(ncol(X)-1))])
  y <- as.matrix(y)
  theta <- matrix(rep(0, ncol(X)), nrow = ncol(X))
  # Using optim function to descend
  costOpti <- optim(theta, fn = cost, gr = grad, X = X, y = y, control=list(maxit=max_iters))
  return(costOpti$par)
}

# To get probability using regressor
logisticProb <- function(theta, X){
  X <- na.omit(X)
  X <- mutate(X, bias =1)
  X <- as.matrix(X[,c(ncol(X), 1:(ncol(X)-1))])
  return(sigmoid(X%*%theta))
}

# To round probabilities >0.5 to 1 and vice-versa
logisticPred <- function(prob){
  return(round(prob, 0))
}


x = pairs_data[, -length(pairs_data)]
y = pairs_data[, length(pairs_data)]
theta <- logisticReg(x, y, 10000)
probZ <- logisticProb(theta, x)
Z <- logisticPred(probZ)
```

```
# Probabilities
print('Sample probabilities predicted are below')
```

```
## [1] "Sample probabilities predicted are below"
```

```
head(probZ)
```

```
##             [,1]
## [1,] 0.08071825
## [2,] 0.19010033
## [3,] 0.11701937
## [4,] 0.25113912
## [5,] 0.18383262
## [6,] 0.24465286
```

```
# Classified match
print('Sample predicted match')
```

```
## [1] "Sample predicted match"
```

```
head(Z)
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
## [4,]    0
## [5,]    0
## [6,]    0
```

```
Z = as.numeric(Z)

# Confusion matrix
print('Confusion matrix')
```

```
## [1] "Confusion matrix"
```

```
table(Z, y)
```

```
##    y
## Z      0    1
##   0 3340  641
##   1    5   13
```

We can see from the confusion matrix that many of the matches are classified as non-matches. Although the prediction accuracy is good, as there are few matches, logistic regression is not able to predict match properly. So, we tried below different methods.

```
dummy_sep <- rbinom(nrow(pairs_data), 1, 0.2)     # Create dummy indicator

train_data <- pairs_data[dummy_sep == 0, ]
test_data <- pairs_data[dummy_sep == 1, ]


X = train_data[, -length(train_data)]
#
Y = train_data[, length(train_data)]

X = data.matrix(X)
Y = data.matrix(Y)

bst <- xgboost(data = data.matrix(x),
               label = data.matrix(y),
               eta = 0.1,
               max_depth = 15,
               nround=100,
               subsample = 0.5,
               colsample_bytree = 0.5,
               seed = 1,
               objective = "reg:logistic",
)
```

```
## Warning in xgb.train(params, dtrain, nrounds, watchlist, verbose = verbose, :
## xgb.train: ‘seed‘ is ignored in R package. Use ‘set.seed()‘ instead.
```

```
## [1]  train-rmse:0.474457
## [2]  train-rmse:0.452092
## [3]  train-rmse:0.431405
## [4]  train-rmse:0.412277
## [5]  train-rmse:0.395542
## [6]  train-rmse:0.380412
## [7]  train-rmse:0.367397
## [8]  train-rmse:0.355147
## [9]  train-rmse:0.344378
## [10] train-rmse:0.335202
## [11] train-rmse:0.326511
## [12] train-rmse:0.317875
## [13] train-rmse:0.309874
## [14] train-rmse:0.302639
## [15] train-rmse:0.296189
## [16] train-rmse:0.290007
## [17] train-rmse:0.284710
## [18] train-rmse:0.279477
## [19] train-rmse:0.274682
## [20] train-rmse:0.269786
## [21] train-rmse:0.265260
## [22] train-rmse:0.261362
## [23] train-rmse:0.257245
## [24] train-rmse:0.252797
## [25] train-rmse:0.249304
## [26] train-rmse:0.245749
## [27] train-rmse:0.242368
## [28] train-rmse:0.238758
## [29] train-rmse:0.235541
## [30] train-rmse:0.233048
## [31] train-rmse:0.230158
## [32] train-rmse:0.227534
## [33] train-rmse:0.224306
## [34] train-rmse:0.222028
## [35] train-rmse:0.219326
## [36] train-rmse:0.216226
## [37] train-rmse:0.214061
## [38] train-rmse:0.211968
## [39] train-rmse:0.208927
## [40] train-rmse:0.206392
## [41] train-rmse:0.204011
## [42] train-rmse:0.201471
## [43] train-rmse:0.199600
## [44] train-rmse:0.197459
## [45] train-rmse:0.195388
## [46] train-rmse:0.193105
## [47] train-rmse:0.190598
## [48] train-rmse:0.188803
## [49] train-rmse:0.186557
## [50] train-rmse:0.184627
## [51] train-rmse:0.182573
## [52] train-rmse:0.180169
## [53] train-rmse:0.178014
## [54] train-rmse:0.176150
## [55] train-rmse:0.174341
## [56] train-rmse:0.172341
## [57] train-rmse:0.170457
## [58] train-rmse:0.168425
## [59] train-rmse:0.166805
## [60] train-rmse:0.164740
## [61] train-rmse:0.162829
## [62] train-rmse:0.161221
## [63] train-rmse:0.159187
## [64] train-rmse:0.157461
## [65] train-rmse:0.155859
## [66] train-rmse:0.154492
## [67] train-rmse:0.152954
## [68] train-rmse:0.151585
## [69] train-rmse:0.150089
## [70] train-rmse:0.148696
## [71] train-rmse:0.147118
## [72] train-rmse:0.145628
## [73] train-rmse:0.143901
## [74] train-rmse:0.142283
## [75] train-rmse:0.140993
## [76] train-rmse:0.139519
## [77] train-rmse:0.137942
## [78] train-rmse:0.135925
## [79] train-rmse:0.134307
## [80] train-rmse:0.132906
## [81] train-rmse:0.131909
## [82] train-rmse:0.130605
## [83] train-rmse:0.129300
## [84] train-rmse:0.127729
## [85] train-rmse:0.126609
## [86] train-rmse:0.125120
## [87] train-rmse:0.123838
## [88] train-rmse:0.122445
## [89] train-rmse:0.121510
## [90] train-rmse:0.120313
## [91] train-rmse:0.119063
## [92] train-rmse:0.117852
## [93] train-rmse:0.116830
## [94] train-rmse:0.115840
## [95] train-rmse:0.114543
## [96] train-rmse:0.113680
## [97] train-rmse:0.112254
## [98] train-rmse:0.111196
## [99] train-rmse:0.110201
## [100]    train-rmse:0.109469
```

```
test_y = data.matrix(test_data[, length(pairs_data)])
test_x = data.matrix(test_data[, -length(pairs_data)])
test_y = as.numeric(test_y)
test_pred = predict(bst, test_x)
pred_labels = ifelse(test_pred > 0.5, 1, 0)


# Confusion matrix

print('Confusion matrix')
```

```
## [1] "Confusion matrix"
```

```
table(pred_labels, test_y)
```

```
##            test_y
## pred_labels   0   1
##           0 668   0
##           1   0 135
```

```
# Accuracy

print('Accuracy')
```

```
## [1] "Accuracy"
```

```
sum(pred_labels==test_y) * 100 /length(test_y)
```

```
## [1] 100
```

```
# Important features
features = xgb.importance(colnames(test_x), model = bst)$Feature

imp_features = xgb.importance(colnames(test_x), model = bst)$Feature[c(1:20)]

print('Important features are')
```

```
## [1] "Important features are"
```

```
print(imp_features)
```

```
##  [1] "shar1_1"    "age"        "sinc1_1"    "amb1_1_f"   "fun1_1_f"
##  [6] "shar1_1_f"  "field_cd_f" "hiking_f"   "fun2_1_f"   "sports_f"
## [11] "hiking"     "intel2_1"   "tvsports"   "intel2_1_f" "imprace_f"
## [16] "age_f"      "attr2_1"    "intel1_1"   "clubbing"   "attr1_1_f"
```

```
least_imp_features = features[c((length(features)-5): length(features))]
print('Least important features are')
```

```
## [1] "Least important features are"
```

```
least_imp_features
```

```
## [1] "goal_f"    "attr3_1"   "go_out_f"  "sinc3_1_f" "theater"   "samerace"
```

Results

Above, we can see that XGB classifier predicts well and has a nice confusion matrix.

Important attributes to determine a match are:

1) Fun1_1_f - How funny a female wants male to be?
2) Clubbing - How interested is a male in clubbing is?
3) Exercise_f - How interested in exercise is the female?
4) att2_1_f - How much does the female wants the male to be?
5) intel1_1 - How much intelligent does the male wants female to be?

While these are the top 5 features, few other contribute to the successful prediction as well.

Least important attributes as analysed using this dataset are:

1) race: Race of the male participant
2) go_out: How interested is male in going out
3) intel3_1 and intel3_1f: How intelligent does a person think of themselves
4) goal_f: Primary goal of female participant
5) samerace: Whether participants are of the same race

How likely to get a second date?

Goal:

How likely to get a second date?

The goal is to Predict probability of getting a second date based on the information collected after speed dating session. In a speed dating session Data was gathered from participants.During the session ,the attendees would have a four minute “first date” with every other participant of the opposite gender .After the session ,all participants were asked to rate their date on six attributes:Attractiveness,Sincerity,Intelligence,Fun,Ambition,Shared Interests and also asked would they like to see their date again. By considering the ratings given by participant to the date we can predict the probability of getting a chance for second date using logistic regression.

```
# importing all necessary libraries and reading the data
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------
```

```
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
## -- Conflicts ---------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(broom)
speed_dating <- read.csv("D:\\Study\\Stat\\Final_project\\Datasets\\Speed_Dating\\Speed Dating Dat
                         header = T, stringsAsFactors = F)
#head(speed_dating)
```

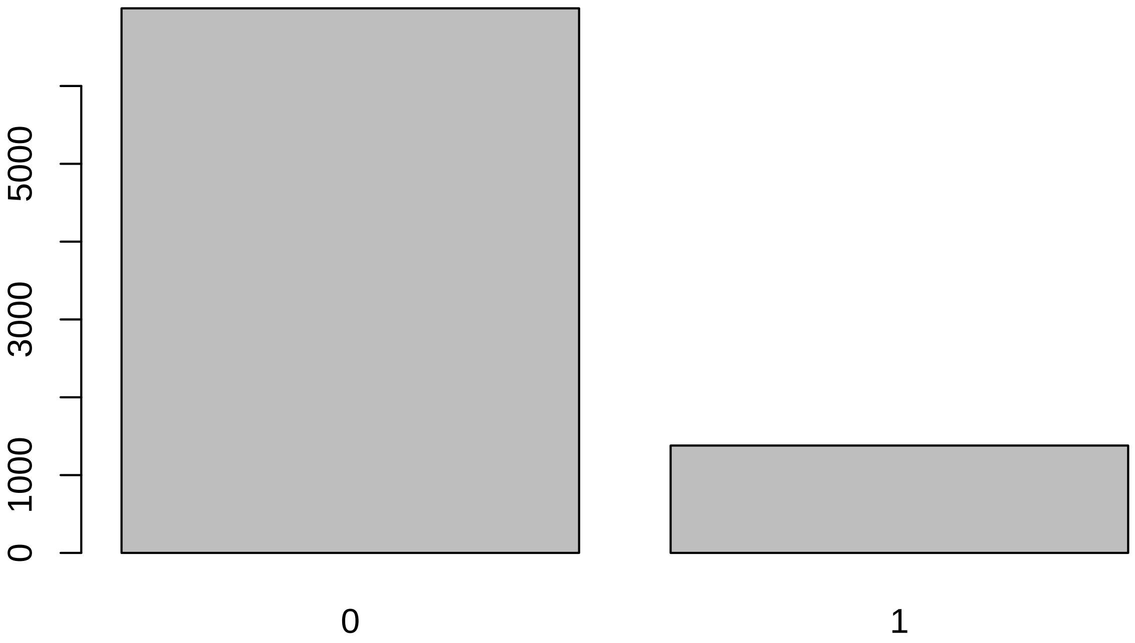
```
# visualizing match column.
count(speed_dating, "match")
```

```
##   "match"    n
## 1   match 8378
```

```
table(speed_dating$match)
```

```
##
##    0    1
## 6998 1380
```

```
barplot(table(speed_dating$match))
```



After the speed dating session we can observe that only 20% of people found a match . It is likely that only 20% of people will get a chance for second date.

Logistic regression:

We are using logistic regression to predict how likely a person can get second date by training the model based on ratings given by partner in first date. By passing certain set of attribute ratings the output will be the probality to get second date.

```
#sigmoid for squashing output between o and 1
sigmoid <- function(z) {
  1 / (1 + exp(-z))
}

#cost function.using cross-entropy
cost <- function(theta, X, y) {
```

```r
  m <- length(y)

  h <- sigmoid(X %*% theta)
  J <- (t(-y) %*% log(h) - t(1 - y) %*% log(1 - h)) / m
  J
}

#gradient function to be given to optim
grad <- function(theta, X, y) {
  m <- length(y)

  h <- sigmoid(X %*% theta)
  grad <- (t(X) %*% (h - y)) / m
  grad
}
# code to perform logistic regression
logisticRegression <- function(X, y) {
  # removing NA values from data
  val <- na.omit(cbind(y, X))
  # adding bias term and then converting into matrix
  X <- mutate(val[,-1], bias = 1)
  X <- as.matrix(X[, c(ncol(X), 1:(ncol(X) - 1))])
  y <- as.matrix(val[, 1])
  #calculating theta
  theta <- matrix(rep(0, ncol(X)), nrow = ncol(X))
  #using optim function to descend
  costOpti <-
    optim(matrix(rep(0, 8), nrow = 8), cost, grad, X = X, y = y)
  return(costOpti$par)
}
logisticProb<-function(p,x){
  val <- na.omit( X)
  # adding bias term and then converting into matrix
  X <- mutate(val[,-1], bias = 1)
  X <- as.matrix(X[, c(ncol(X), 1:(ncol(X) - 1))])
  return(sigmoid(X %*% p))
}
# adding bias term and then converting into matrix
logisticPrediction <- function(model, X) {
  X <- na.omit(X)
  X <- mutate(X, bias = 1)
  X <- as.matrix(X[, c(ncol(X), 1:(ncol(X) - 1))])
  return(sigmoid(X %*%model ))
}
```

```r
# selecting required columns from data: we require ratings
# given by partner in first date to train the model.
speed_dating.df <-
  select(speed_dating,
         match,
         attr_o,
         sinc_o,
         intel_o,
         fun_o,
         amb_o,
         like_o,
         shar_o)
print(head(speed_dating.df))
```

```
##   match attr_o sinc_o intel_o fun_o amb_o like_o shar_o
## 1     0      6      8       8     8     8      7      6
## 2     0      7      8      10     7     7      8      5
## 3     1     10     10      10    10    10     10     10
## 4     1      7      8       9     8     9      7      8
## 5     1      8      7       9     6     9      8      7
## 6     0      7      7       8     8     7      7      7
```

```r
# performing logistic regression for selected data
speed_dating.X <- speed_dating.df[,-1]
speed_dating.y <- speed_dating.df[, 1]
# performing logistic regression for selected data
model <- logisticRegression(speed_dating.X,speed_dating.y)
# calculating probability for how likely a person with
# certain attributes rating can get a chance for second date.
input_constraints <-
  expand.grid(
    attr = seq(6, 9),
    sinc = 7,
    intel_o = 8,
    fun = 9,
    amb = 8,
    like = seq(6,9),
    shar= 6
  )
second_date <- logisticPrediction(model, input_constraints)
second_date
```

```
##            [,1]
##  [1,] 0.1219409
##  [2,] 0.1431970
##  [3,] 0.1674517
##  [4,] 0.1948802
##  [5,] 0.1718418
##  [6,] 0.1998169
##  [7,] 0.2310754
##  [8,] 0.2656008
##  [9,] 0.2366594
## [10,] 0.2717243
## [11,] 0.3098755
## [12,] 0.3508031
## [13,] 0.3165798
## [14,] 0.3579336
## [15,] 0.4015157
## [16,] 0.4467123
```

conclusion:

By passing certain set of attribute ratings for a particular person we got the probability of getting a second date with the help of ratings given by partner in first date. From results we can observe that for different combination of attributes the probability to get a second date changes. based on results we can conclude that being likeable is more important for securing a second date and even we can observe that maintaining good ratings for every attribute increases the chances to secure second date. Therefore, we cannot conclude that based on which attribute a person's second date completely depends on, All the attributes play a key role.

6 References

https://www.kaggle.com - Ray Fisman and Sheena Iyengar