

SD_dataset

Katarzyna Otko s202872
Alvils Sture s202586

October 2020

1. A description of the dataset

1.1. What is our data about?

The dataset of our choice is a result of a **Speed Dating Experiment** that was conducted in years 2002 - 2004 by professors of Columbia University for their paper entitled *Gender Differences in Mate Selection: Evidence From a Speed Dating Experiment*. The goal of this two-year experiment was to discover what features and factors play the most important role during first minutes of a date. In others words, what makes us want to see the other person again?

Source of the dataset: <https://data.world/annavmontoya/speed-dating-experiment>

Overall, there were 21 speed dating events (waves), with participants ranging from 12 to 44 in each, which in overall sums up to 551 people. Those subjects were drawn from students in graduate and professional schools at Columbia University. Each date lasted 4 minutes.

The dataset also includes questionnaire data gathered from participants at different stages of the process. Attendees answered questions about their dating habits, self-perception, beliefs on what others may find attractive in a mate, lifestyle, demographics and many more. The process consisted of 4 steps:

- Signup (Time 1) - filling out a survey in order to register for the event;
- Answering questions half way through meeting all potential dates during the night of the event;
- 1st Followup (Time 2) - filling out a survey the day after participating in the event;
- 2nd Followup (Time 3) - filling out a survey 3-4 weeks after participants had been sent their matches.

1.2. Summary of previous analysis of the dataset

We would like to briefly summarize the previously mentioned

```
library(dplyr)
library(ggplot2)
library(reshape2)
library(knitr)
setwd('C:/Users/katin/Desktop/Folder/STUDIA/DTU/Semestr I/Intro to ML/Project I')
SD <- read.csv('Speed Dating Data.csv')

# numdim(SDber of rows and columns
kable(dim(SD))
```

	—
	x
	—
	8378
	195
	—

```
# number of women
length(unique(SD$iid[which(SD$gender == 0)])) # 274
```

```
## [1] 274
```

```
# number of men
length(unique(SD$iid[which(SD$gender == 1)])) # 277
```

```
## [1] 277
```

```
274 + 277
```

```
## [1] 551
```

Missin values

```
NAs <- sapply(SD, function(x) sum(is.na(x)))
sort(NAs[which(NAs > 0)])
```

```
##      id      pid      race  race_o  imprace  imprelig      goal  go_out
##      1       10       63     73      79      79      79      79
## sports tvsports exercise  dining  museums      art  hiking  gaming
##      79      79      79      79      79      79      79      79
## clubbing reading      tv  theater  movies  concerts  music  shopping
##      79      79      79      79      79      79      79      79
##      yoga attr1_1  sinc1_1  intel1_1  attr2_1  sinc2_1  intel2_1  fun2_1
##      79      79      79      79      79      79      79      79
## field_cd pf_o_att pf_o_sin pf_o_int  fun1_1  amb2_1  shar2_1      age
##      82      89      89      89      89      89      89      95
##      date pf_o_fun  amb1_1  exphappy  age_o  attr3_1  sinc3_1  fun3_1
##      97      98      99      101     104     105     105     105
## intel3_1  amb3_1  pf_o_amb  shar1_1  pf_o_sha  career_c  int_corr  attr
##      105     105     107     121     129     138     158     202
##      attr_o  like  like_o  sinc  sinc_o  intel  intel_o  prob
##      212     240     250     277     287     296     306     309
##      prob_o  fun  fun_o  met  met_o  amb  amb_o  satis_2
##      318     350     360     375     385     712     722     915
##      length  sinc1_2  intel1_2  fun1_2  amb1_2  shar1_2  attr3_2  sinc3_2
##      915     915     915     915     915     915     915     915
## intel3_2  fun3_2  amb3_2  attr1_2  numdat_2  shar  shar_o  match_es
##      915     915     915     933     945     1067     1076     1173
## positin1  attr4_1  sinc4_1  intel4_1  fun4_1  amb4_1  shar4_1  attr4_2
##      1846     1889     1889     1889     1889     1889     1911     2603
##      sinc4_2  intel4_2  fun4_2  amb4_2  shar4_2  attr2_2  sinc2_2  intel2_2
##      2603     2603     2603     2603     2603     2603     2603     2603
##      fun2_2  amb2_2  shar2_2  attr5_1  sinc5_1  intel5_1  fun5_1  amb5_1
##      2603     2603     2603     3472     3472     3472     3472     3472
##      attr5_2  sinc5_2  intel5_2  fun5_2  amb5_2  attr1_s  sinc1_s  intel1_s
##      4001     4001     4001     4001     4001     4282     4282     4282
##      fun1_s  amb1_s  shar1_s  attr3_s  sinc3_s  intel3_s  fun3_s  amb3_s
##      4282     4282     4282     4378     4378     4378     4378     4378
## you_call  them_cal  date_3  attr1_3  sinc1_3  intel1_3  fun1_3  amb1_3
##      4404     4404     4404     4404     4404     4404     4404     4404
##      shar1_3  attr3_3  sinc3_3  intel3_3  fun3_3  amb3_3  attr4_3  sinc4_3
##      4404     4404     4404     4404     4404     4404     5419     5419
## intel4_3  fun4_3  amb4_3  shar4_3  attr2_3  sinc2_3  intel2_3  fun2_3
```

```
##      5419      5419      5419      5419      5419      5419      5419      5419
## amb2_3 attr7_3 sinc7_3 intel7_3 fun7_3 amb7_3 shar7_3 shar2_3
##      5419      6362      6362      6362      6362      6362      6362      6362
## attr5_3 sinc5_3 intel5_3 fun5_3 amb5_3 attr7_2 intel7_2 fun7_2
##      6362      6362      6362      6362      6362      6394      6394      6394
## shar7_2 sinc7_2 amb7_2 expnum numdat_3 num_in_3
##      6404      6423      6423      6578      6882      7710
```

```
#filling one missing value in last id row
```

```
SD[which(is.na(SD$id)), 1:2] <- 22
```

```
# filling 10 missing values in pid columns
```

```
SD[which(is.na(SD$pid)), 1:15] # partner's id - 7
```

```
##      iid id gender idg condtn wave round position positin1 order partner pid
## 1756 122 1      1 2      1      5 10      4      NA      6      7 NA
## 1766 123 2      1 4      1      5 10      4      NA     10      7 NA
## 1776 124 3      1 6      1      5 10      4      NA      3      7 NA
## 1786 125 4      1 8      1      5 10      4      NA      8      7 NA
## 1796 126 5      1 10     1      5 10      4      NA      1      7 NA
## 1806 127 6      1 12     1      5 10      4      NA      7      7 NA
## 1816 128 7      1 14     1      5 10      4      NA      9      7 NA
## 1826 129 8      1 15     1      5 10      4      NA      5      7 NA
## 1836 130 9      1 16     1      5 10      4      NA      2      7 NA
## 1846 131 10     1 18     1      5 10      4      NA      4      7 NA
##      match int_corr samerace
## 1756      0     -0.12      0
## 1766      0     -0.29      0
## 1776      0     -0.05      0
## 1786      0      0.15      0
## 1796      0      0.01      0
## 1806      0      0.38      0
## 1816      0     -0.05      0
## 1826      0      0.09      0
## 1836      0     -0.40      0
## 1846      0     -0.14      0
```

```
SD[which(SD$id == 7 & SD$wave == 5), 1:2] # we have to fill these 10 NAs with 128
```

```
##      iid id
## 1807 128 7
## 1808 128 7
## 1809 128 7
## 1810 128 7
## 1811 128 7
## 1812 128 7
## 1813 128 7
## 1814 128 7
## 1815 128 7
## 1816 128 7
```

```
SD[which(is.na(SD$pid)), 'pid'] <- 128
```

```
s = 0
```

```
NAs <- sapply(df, function(x) sum(is.na(x))); NAs
```

```

## Warning in is.na(x): 'is.na()' zastosowane do nie-listy lub nie-wektora typu
## 'symbol'

## Warning in is.na(x): 'is.na()' zastosowane do nie-listy lub nie-wektora typu
## 'symbol'

## Warning in is.na(x): 'is.na()' zastosowane do nie-listy lub nie-wektora typu
## 'symbol'

## Warning in is.na(x): 'is.na()' zastosowane do nie-listy lub nie-wektora typu
## 'symbol'

## Warning in is.na(x): 'is.na()' zastosowane do nie-listy lub nie-wektora typu
## 'language'

##      x df1 df2 ncp log
##      0  0  0  0  0  0

NAS_sum <- sum(NAs); NAS_sum

## [1] 0

DF <- SD[, 70:75]
df <- DF[!complete.cases(DF),]
dim(df); sum(rowSums(df, na.rm = T) == 100) # so there are 42 rows where we can impute 0s

## [1] 121    6
## [1] 42

df[which(df[,1] + df[,2] + df[,3] == 100),]

##      attr1_1 sinc1_1 intel1_1 fun1_1 amb1_1 shar1_1
## 1827      80      10      10      NA      NA      NA
## 1828      80      10      10      NA      NA      NA
## 1829      80      10      10      NA      NA      NA
## 1830      80      10      10      NA      NA      NA
## 1831      80      10      10      NA      NA      NA
## 1832      80      10      10      NA      NA      NA
## 1833      80      10      10      NA      NA      NA
## 1834      80      10      10      NA      NA      NA
## 1835      80      10      10      NA      NA      NA
## 1836      80      10      10      NA      NA      NA

df[which(df[,1] + df[,2] + df[,3] == 100), c(4:6)] <- 0

df[which(df[,1] + df[,2] + df[,3] + df[,4] == 100),]

##      attr1_1 sinc1_1 intel1_1 fun1_1 amb1_1 shar1_1
## 1817      40      20      20      20      NA      NA
## 1818      40      20      20      20      NA      NA
## 1819      40      20      20      20      NA      NA
## 1820      40      20      20      20      NA      NA
## 1821      40      20      20      20      NA      NA
## 1822      40      20      20      20      NA      NA
## 1823      40      20      20      20      NA      NA
## 1824      40      20      20      20      NA      NA
## 1825      40      20      20      20      NA      NA

```

```
## 1826      40      20      20      20      NA      NA
## 1827      80      10      10      0       0       0
## 1828      80      10      10      0       0       0
## 1829      80      10      10      0       0       0
## 1830      80      10      10      0       0       0
## 1831      80      10      10      0       0       0
## 1832      80      10      10      0       0       0
## 1833      80      10      10      0       0       0
## 1834      80      10      10      0       0       0
## 1835      80      10      10      0       0       0
## 1836      80      10      10      0       0       0
```

```
df[which(df[,1] + df[,2] + df[,3] + df[,4] == 100), c(5:6)] <- 0
```

```
df[which(df[,1] + df[,2] + df[,3] + df[,4] +df[, 5] == 100),]
```

```
##      attr1_1 sinc1_1 intell1_1 fun1_1 amb1_1 shar1_1
## 1817      40      20      20      20      0       0
## 1818      40      20      20      20      0       0
## 1819      40      20      20      20      0       0
## 1820      40      20      20      20      0       0
## 1821      40      20      20      20      0       0
## 1822      40      20      20      20      0       0
## 1823      40      20      20      20      0       0
## 1824      40      20      20      20      0       0
## 1825      40      20      20      20      0       0
## 1826      40      20      20      20      0       0
## 1827      80      10      10      0       0       0
## 1828      80      10      10      0       0       0
## 1829      80      10      10      0       0       0
## 1830      80      10      10      0       0       0
## 1831      80      10      10      0       0       0
## 1832      80      10      10      0       0       0
## 1833      80      10      10      0       0       0
## 1834      80      10      10      0       0       0
## 1835      80      10      10      0       0       0
## 1836      80      10      10      0       0       0
## 7829      40      10      30      10      10      NA
## 7830      40      10      30      10      10      NA
## 7831      40      10      30      10      10      NA
## 7832      40      10      30      10      10      NA
## 7833      40      10      30      10      10      NA
## 7834      40      10      30      10      10      NA
## 7835      40      10      30      10      10      NA
## 7836      40      10      30      10      10      NA
## 7837      40      10      30      10      10      NA
## 7838      40      10      30      10      10      NA
## 7839      40      10      30      10      10      NA
## 7840      40      10      30      10      10      NA
## 7841      40      10      30      10      10      NA
## 7842      40      10      30      10      10      NA
## 7843      40      10      30      10      10      NA
## 7844      40      10      30      10      10      NA
## 7845      40      10      30      10      10      NA
## 7846      40      10      30      10      10      NA
```

```
## 7847      40      10      30      10      10      NA
## 7848      40      10      30      10      10      NA
## 7849      40      10      30      10      10      NA
## 7850      40      10      30      10      10      NA
```

```
df[which(df[,1] + df[,2] + df[,3] + df[,4] +df[, 5] == 100), 6] <- 0
```

```
DF[!complete.cases(DF),] <- df
SD[,70:75] <- DF
```

```
# adding one column with explanation for race column (matching index with race names)
race_idx <- unique(SD$race)
race_val <- c('Asian', 'European', 'Other', 'Latino', 'Black', NA)
SD$race_explained <- race_val[match(SD$race, race_idx)]
```

```
# adding one column with explanation for field_cd column (matching index with race names)
# DISCUSS WITH ALVILS IMPUTING DATA INTO field_cd as 9 (because field is Operations Research)
field_idx <- c(1:18, NA)
field_val <- c('Law', 'Math', 'Social Science, Psychologist', 'Medical Science/Pharmaceuticals/Bio Tech',
               'Engineering', 'English/Creative Writing/ Journalism', 'History/Religion/Philosophy',
               'Business/Econ/Finance', 'Education, Academia', 'Biological Sciences/Chemistry/Physics',
               'Social Work', 'Undergrad/undecided', 'Political Science/International Affairs',
               'Film', 'Fine Arts/Arts Administration', 'Languages', 'Architecture', 'Other', 'Other')
SD$field_explained <- field_val[match(SD$field_cd, field_idx)]
```

```
# converting income from string to numeric
SD$income <- as.numeric(gsub(',', '', SD$income, fixed = T))

sum(is.na(SD$income))
```

```
## [1] 4099
```

```
unique(SD$field_cd)
```

```
## [1] 1 2 13 8 5 9 3 11 NA 12 4 7 6 10 14 16 15 17 18
```

```
summary(SD[SD$wave >= 6 & SD$wave <= 9,129:134])
```

```
##      attr1_2      sinc1_2      intell1_2      fun1_2
## Min.   :10.00  Min.    : 5.00  Min.    :13.95  Min.    :11.11
## 1st Qu.:15.38  1st Qu.:16.07  1st Qu.:17.39  1st Qu.:15.69
## Median :16.67  Median :17.65  Median :18.52  Median :17.78
## Mean   :17.45  Mean    :17.36  Mean    :18.79  Mean    :17.34
## 3rd Qu.:19.05  3rd Qu.:19.15  3rd Qu.:20.00  3rd Qu.:18.75
## Max.   :26.32  Max.    :23.81  Max.    :25.00  Max.    :25.00
## NA's   :164    NA's   :164    NA's   :164    NA's   :164
##      amb1_2      shar1_2
## Min.    : 2.50  Min.    : 4.76
## 1st Qu.:12.77  1st Qu.:12.96
## Median :15.38  Median :14.58
## Mean    :14.65  Mean    :14.40
## 3rd Qu.:16.67  3rd Qu.:16.67
## Max.    :22.22  Max.    :22.50
## NA's    :164    NA's    :164
```

```
# Waves 6 - 9:
# attr4_1 - shar4_1 have values between 0 and 10
```

```
# attr2_1 - shar2_1 OK
# attr1_2 - shar1_2 OK
```

```
# Age analysis
sum(is.na(SD$age))
```

```
## [1] 95
```

```
SD[is.na(SD$age), 1:10]
```

##	iid	id	gender	idg	condtn	wave	round	position	positin1	order	
## 829	58	3		0	5	1	3	10	7	NA	9
## 830	58	3		0	5	1	3	10	7	NA	5
## 831	58	3		0	5	1	3	10	7	NA	10
## 832	58	3		0	5	1	3	10	7	NA	1
## 833	58	3		0	5	1	3	10	7	NA	6
## 834	58	3		0	5	1	3	10	7	NA	4
## 835	58	3		0	5	1	3	10	7	NA	3
## 836	58	3		0	5	1	3	10	7	NA	7
## 837	58	3		0	5	1	3	10	7	NA	2
## 838	58	3		0	5	1	3	10	7	NA	8
## 839	59	4		0	7	1	3	10	8	NA	10
## 840	59	4		0	7	1	3	10	8	NA	6
## 841	59	4		0	7	1	3	10	8	NA	1
## 842	59	4		0	7	1	3	10	8	NA	2
## 843	59	4		0	7	1	3	10	8	NA	7
## 844	59	4		0	7	1	3	10	8	NA	5
## 845	59	4		0	7	1	3	10	8	NA	4
## 846	59	4		0	7	1	3	10	8	NA	8
## 847	59	4		0	7	1	3	10	8	NA	3
## 848	59	4		0	7	1	3	10	8	NA	9
## 1817	129	8		1	15	1	5	10	6	NA	7
## 1818	129	8		1	15	1	5	10	9	NA	10
## 1819	129	8		1	15	1	5	10	7	NA	8
## 1820	129	8		1	15	1	5	10	1	NA	2
## 1821	129	8		1	15	1	5	10	8	NA	9
## 1822	129	8		1	15	1	5	10	2	NA	3
## 1823	129	8		1	15	1	5	10	5	NA	6
## 1824	129	8		1	15	1	5	10	3	NA	4
## 1825	129	8		1	15	1	5	10	10	NA	1
## 1826	129	8		1	15	1	5	10	4	NA	5
## 1867	136	6		0	8	1	6	5	5	5	3
## 1868	136	6		0	8	1	6	5	5	5	5
## 1869	136	6		0	8	1	6	5	5	5	1
## 1870	136	6		0	8	1	6	5	5	5	2
## 1871	136	6		0	8	1	6	5	5	5	4
## 5005	339	8		1	16	1	13	10	1	1	1
## 5006	339	8		1	16	1	13	10	5	5	5
## 5007	339	8		1	16	1	13	10	4	4	4
## 5008	339	8		1	16	1	13	10	6	6	6
## 5009	339	8		1	16	1	13	10	3	3	3
## 5010	339	8		1	16	1	13	10	9	9	9
## 5011	339	8		1	16	1	13	10	2	2	2
## 5012	339	8		1	16	1	13	10	10	10	10
## 5013	339	8		1	16	1	13	10	7	7	7

##	5014	339	8	1	16	1	13	10	8	8	8
##	5015	340	9	1	18	1	13	10	1	1	9
##	5016	340	9	1	18	1	13	10	5	5	3
##	5017	340	9	1	18	1	13	10	4	4	2
##	5018	340	9	1	18	1	13	10	6	6	4
##	5019	340	9	1	18	1	13	10	3	3	1
##	5020	340	9	1	18	1	13	10	9	9	7
##	5021	340	9	1	18	1	13	10	2	2	10
##	5022	340	9	1	18	1	13	10	10	10	8
##	5023	340	9	1	18	1	13	10	7	7	5
##	5024	340	9	1	18	1	13	10	8	8	6
##	5115	346	6	0	11	2	14	18	10	10	7
##	5116	346	6	0	11	2	14	18	10	10	1
##	5117	346	6	0	11	2	14	18	10	10	16
##	5118	346	6	0	11	2	14	18	10	10	18
##	5119	346	6	0	11	2	14	18	10	10	14
##	5120	346	6	0	11	2	14	18	10	10	17
##	5121	346	6	0	11	2	14	18	10	10	10
##	5122	346	6	0	11	2	14	18	10	10	8
##	5123	346	6	0	11	2	14	18	10	10	12
##	5124	346	6	0	11	2	14	18	10	10	6
##	5125	346	6	0	11	2	14	18	10	10	9
##	5126	346	6	0	11	2	14	18	10	10	4
##	5127	346	6	0	11	2	14	18	10	10	5
##	5128	346	6	0	11	2	14	18	10	10	15
##	5129	346	6	0	11	2	14	18	10	10	11
##	5130	346	6	0	11	2	14	18	10	10	13
##	5131	346	6	0	11	2	14	18	10	10	2
##	5132	346	6	0	11	2	14	18	10	10	3
##	7477	512	4	0	7	2	21	22	7	7	16
##	7478	512	4	0	7	2	21	22	7	7	13
##	7479	512	4	0	7	2	21	22	7	7	6
##	7480	512	4	0	7	2	21	22	7	7	15
##	7481	512	4	0	7	2	21	22	7	7	12
##	7482	512	4	0	7	2	21	22	7	7	5
##	7483	512	4	0	7	2	21	22	7	7	17
##	7484	512	4	0	7	2	21	22	7	7	22
##	7485	512	4	0	7	2	21	22	7	7	4
##	7486	512	4	0	7	2	21	22	7	7	19
##	7487	512	4	0	7	2	21	22	7	7	7
##	7488	512	4	0	7	2	21	22	7	7	2
##	7489	512	4	0	7	2	21	22	7	7	18
##	7490	512	4	0	7	2	21	22	7	7	3
##	7491	512	4	0	7	2	21	22	7	7	11
##	7492	512	4	0	7	2	21	22	7	7	8
##	7493	512	4	0	7	2	21	22	7	7	14
##	7494	512	4	0	7	2	21	22	7	7	21
##	7495	512	4	0	7	2	21	22	7	7	1
##	7496	512	4	0	7	2	21	22	7	7	10
##	7497	512	4	0	7	2	21	22	7	7	9
##	7498	512	4	0	7	2	21	22	7	7	20

```
age_df <- subset(SD, !duplicated(SD[,1])) %>%
  filter(!is.na(age)) %>%
```



```

group_by(wave, gender) %>%
  summarize(Average_age = mean(age))

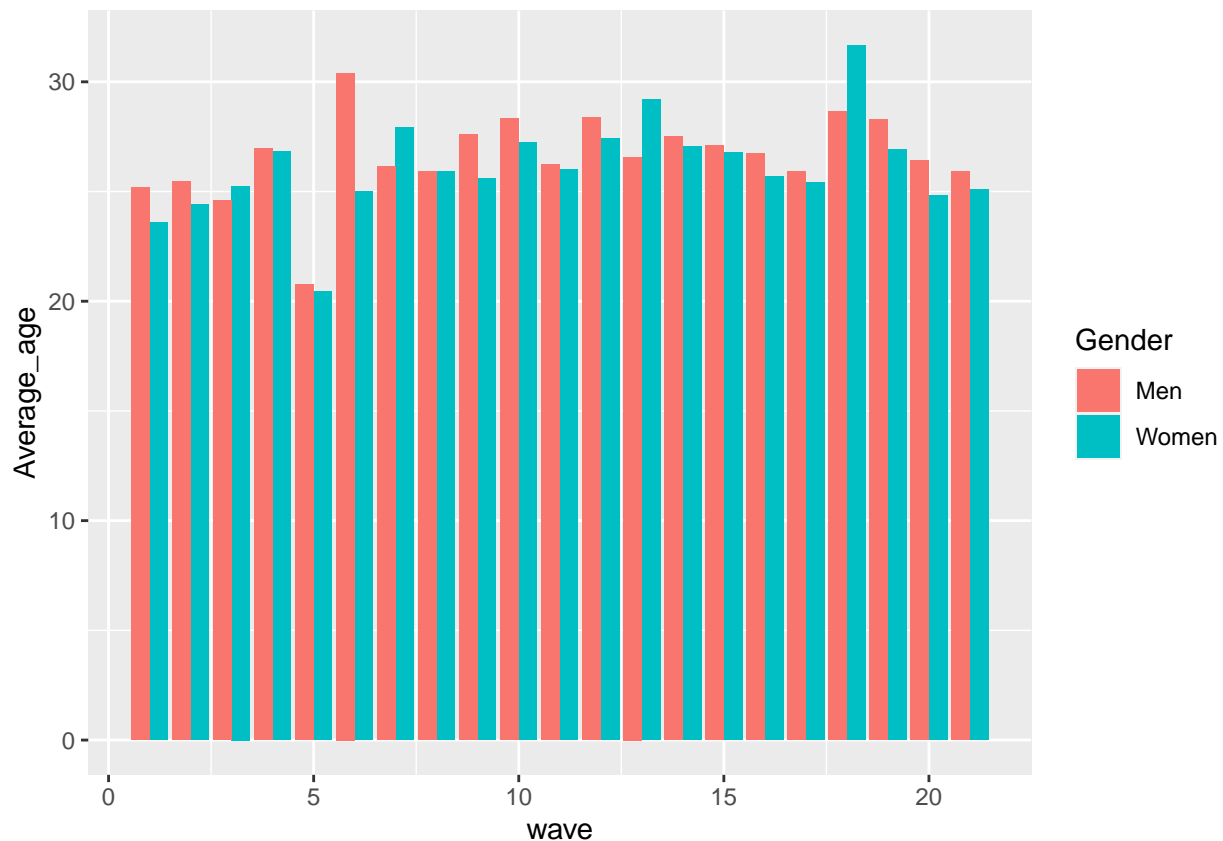
## `summarise()` regrouping output by 'wave' (override with ` `.groups` argument)
SD %>% nrow()

## [1] 8378
nrow(SD)

## [1] 8378
age_df$gender <- ifelse(age_df$gender == 0, 'Women', 'Men')

# Mean age per wave
age_df %>% ggplot(aes(x = wave, y = Average_age, fill = gender)) +
  geom_bar(stat = 'identity', position = 'dodge') +
  scale_fill_discrete(name = "Gender")

```



```

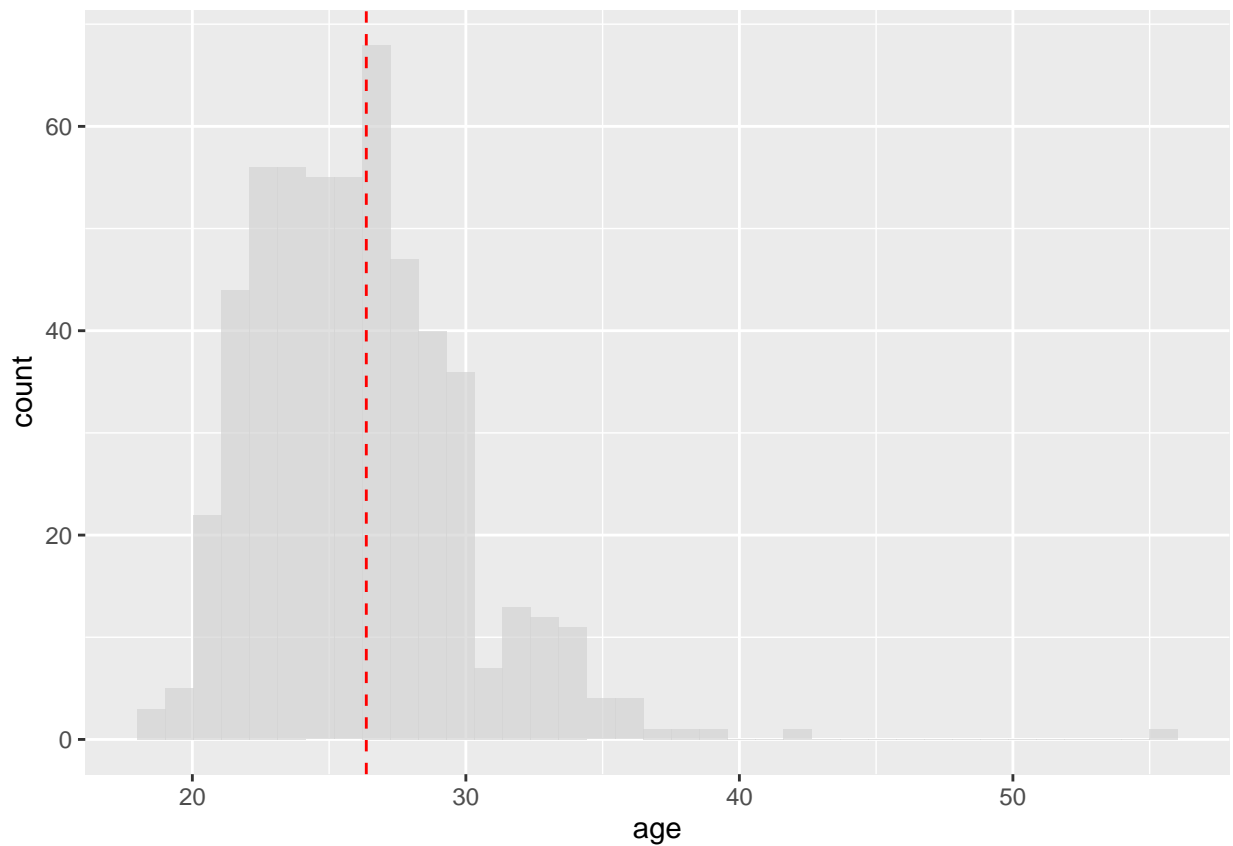
age_df <- subset(SD, !duplicated(SD$id), select = c(id, gender, age)) %>%
  filter(!is.na(age)) %>%
  mutate(mean = mean(age))
age_df$gender <- ifelse(age_df$gender == 0, 'Women', 'Men')

# Histogram of age
max(unique(age_df$age)) - min(unique(age_df$age)) # number of bins

## [1] 37

```

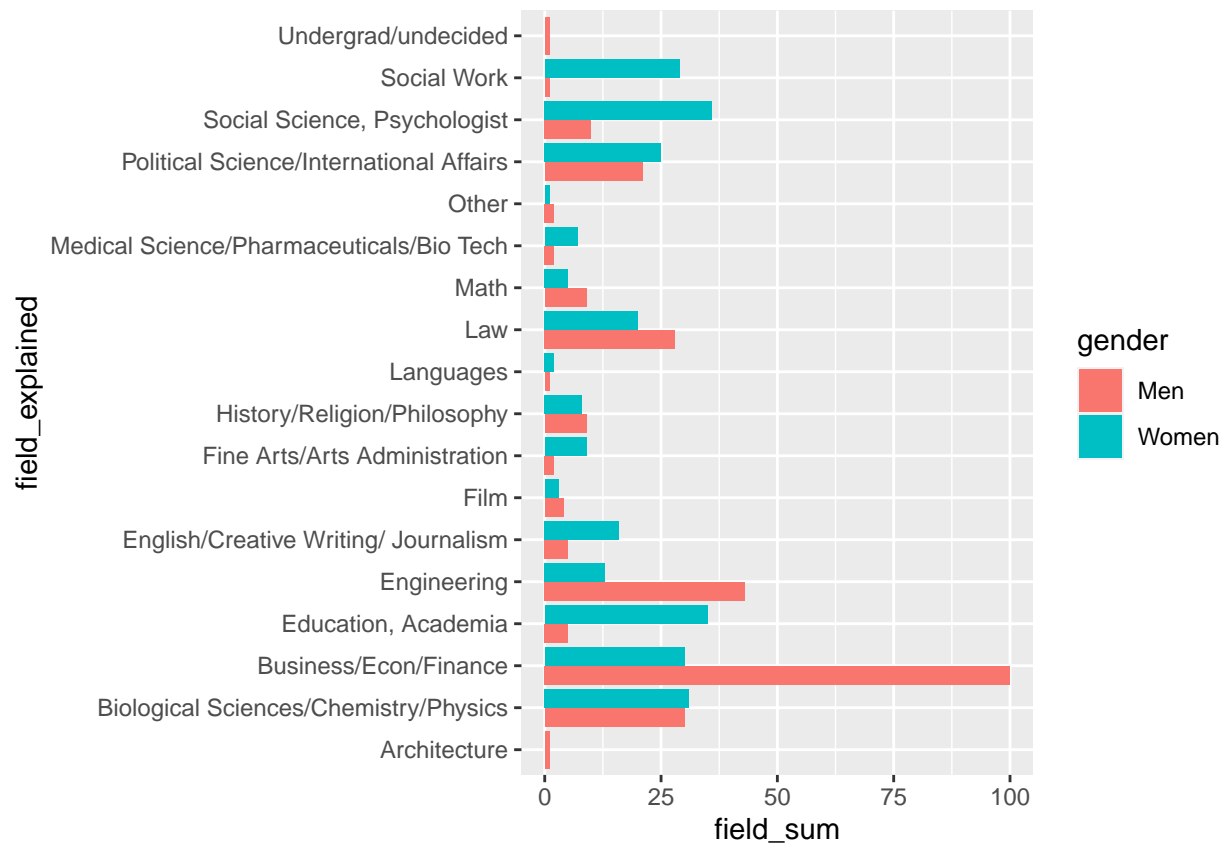
```
age_df %>% ggplot(aes(x = age)) +
  geom_histogram(bins = 37, fill = 'lightgrey', position = 'identity', alpha = .7) +
  geom_vline(aes(xintercept = mean), col = 'red', linetype = 'dashed')
```



```
# Field analysis
field_df <- subset(SD, !duplicated(SD$iid)) %>%
  filter(!is.na(field_cd)) %>%
  group_by(field_explained, gender) %>%
  summarize(field_sum = n())

## `summarise()` regrouping output by 'field_explained' (override with `.groups` argument)
field_df$gender <- ifelse(field_df$gender == 0, 'Women', 'Men')

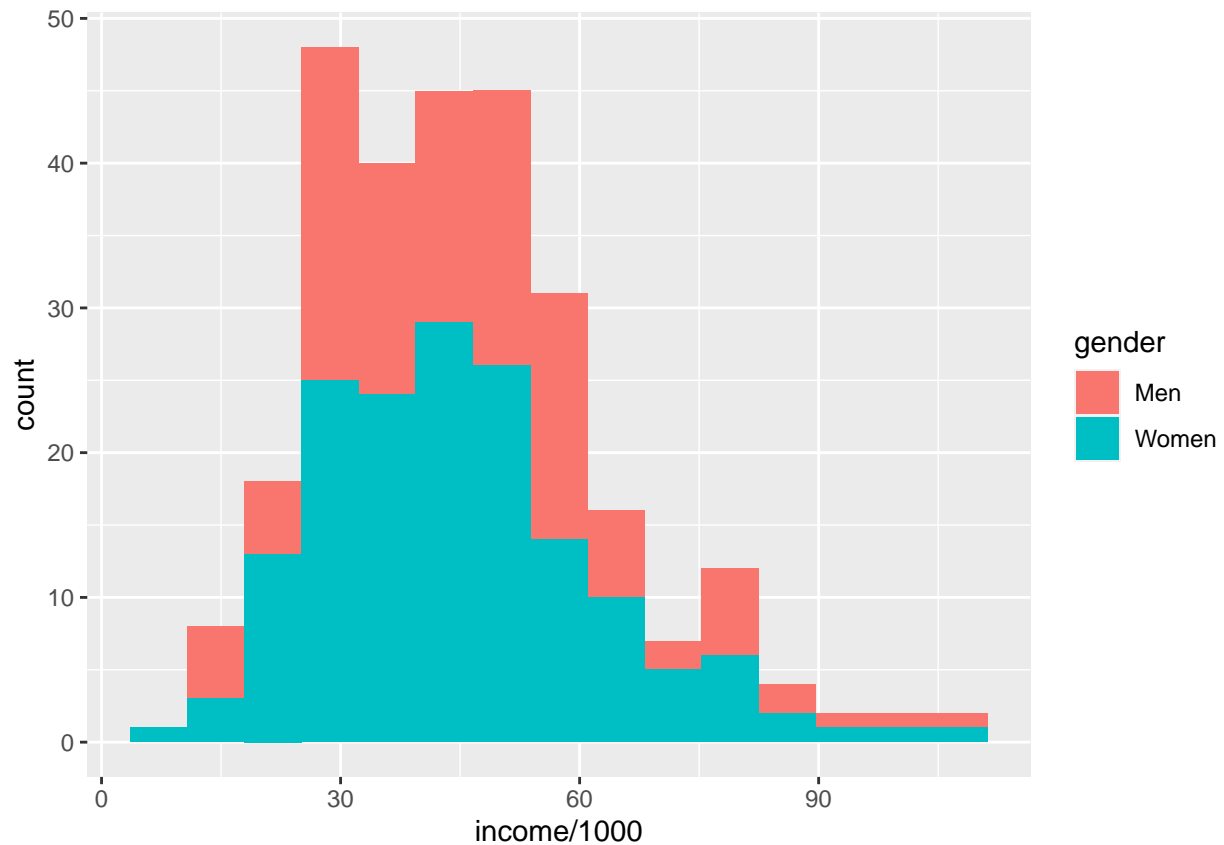
field_df %>% ggplot(aes(x = field_explained, y = field_sum, fill = gender)) +
  geom_bar(stat = 'identity', position = 'dodge') +
  coord_flip()
```



```
# Income
income_df <- subset(SD, !duplicated(SD$iid)) %>%
  filter(!is.na(income))

income_df$gender <- ifelse(income_df$gender == 0, 'Women', 'Men')

income_df %>% ggplot(aes(x = income/1000, fill = gender)) +
  geom_histogram(bins = 15)
```

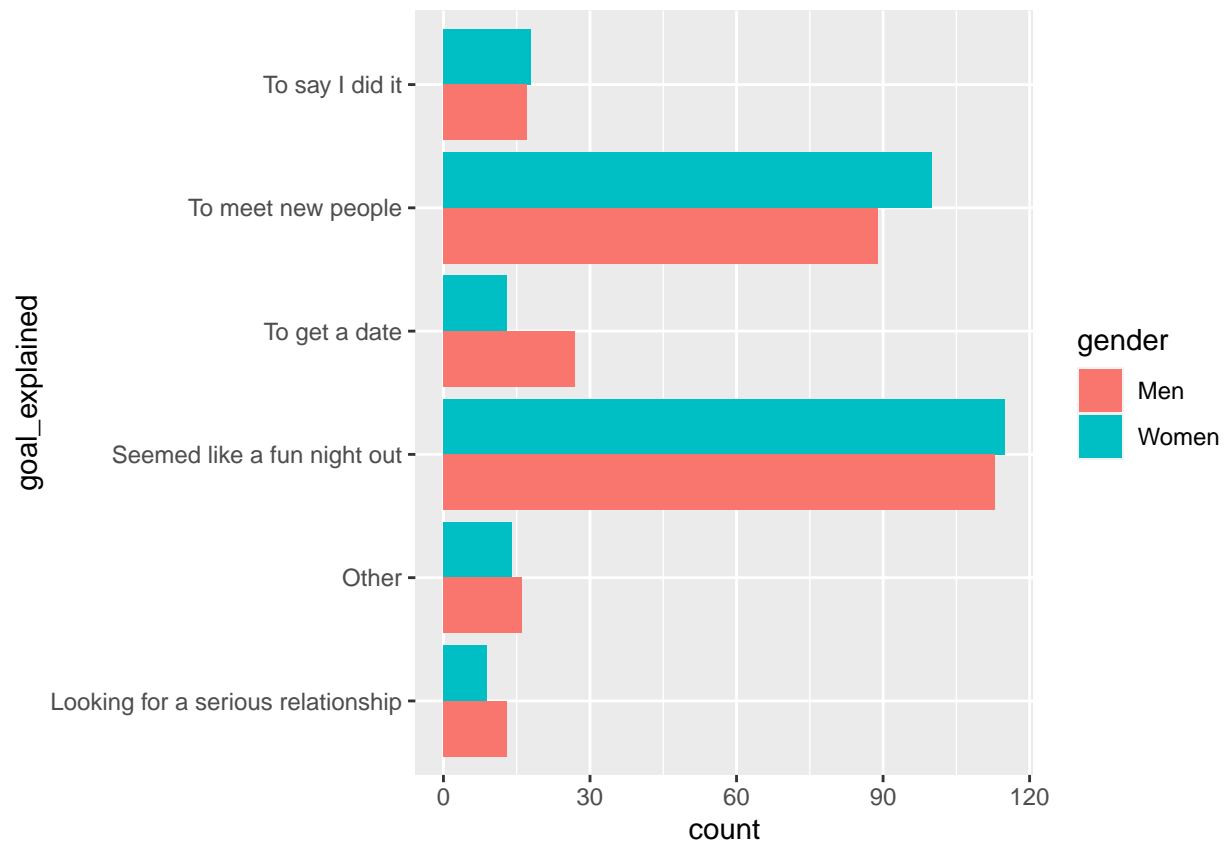


```
# Purpose
goal_df <- subset(SD, !duplicated(SD$iid)) %>%
  filter(!is.na(goal)) %>%
  group_by(goal, gender) %>%
  summarise(count = n())

## `summarise()` regrouping output by 'goal' (override with `.groups` argument)
goal_df$gender <- ifelse(goal_df$gender == 0, 'Women', 'Men')

goal_idx <- unique(goal_df$goal)
goal_val <- c('Seemed like a fun night out', 'To meet new people', 'To get a date',
             'Looking for a serious relationship', 'To say I did it', 'Other')
goal_df$goal_explained <- goal_val[match(goal_df$goal, goal_idx)]

goal_df %>% ggplot(aes(x = goal_explained, y = count, fill = gender)) +
  geom_bar(stat = 'identity', position = 'dodge') +
  coord_flip()
```



```
# Importance of features for men/women
```

PCA

```
colnames(SD) [51:67]
```

```
## [1] "sports" "tvsports" "exercise" "dining" "museums" "art"
## [7] "hiking" "gaming" "clubbing" "reading" "tv" "theater"
## [13] "movies" "concerts" "music" "shopping" "yoga"
```

```
Y <- subset(SD, !duplicated(SD$id), select = c(51, 56:60, 63, 65, 66))
Y <- t(apply(Y, 1, '-', colMeans(Y, na.rm = T)))
```

```
s <- svd(Y[complete.cases(Y),])
diagS <- s$d
rho <- diagS^2/sum(diagS^2)
```

```
threshold = 0.9
```

```
xlimits <- c(1, ncol(Y));
plot(rho,
     type='o',
     main="Variance explained by principal components",
     xlab="Principal components",
     ylab="Variance explained",
```

```

xlim=xlimits,
ylim=c(0,1),
col='blue')
lines(cumsum(rho), type='o', col='orange')
lines(xlimits, c(threshold, threshold), lty='dashed')

legend("right", # Define position
      legend=c("Cumulative", "Individual", "Threshold"), # Set strings for legend
      col=c("orange", "blue", "black"), lty=c(1,1,2), # Match appearance of lines
      cex=1.5, bg='lightblue') # Setup how the box looks (cex controls size)

```

Variance explained by principal components

