ETC 3250 Lab 2 2017 - Solutions

Souhaib Ben Taieb

7 August 2017

Import dataset

```
library(readr)
library(plyr)
library(dplyr)
library(tidyr)
library(knitr)
library(ggplot2)  # for graphics
library(gridExtra)


dataset <- tbl_df(read_csv("../../data/speed-dating-data.csv"))

DT <- select(dataset, one_of(c("wave", "iid", "id", "gender", "idg",
                                "match", "samerace", "age_o","race_o",
                                "field_cd", "race", "imprace", "imprelig",
                                "goal", "date", "go_out", "attr1_1",
                                "sinc1_1","intel1_1", "fun1_1",
                                "amb1_1", "shar1_1")))
```

Recode Variable

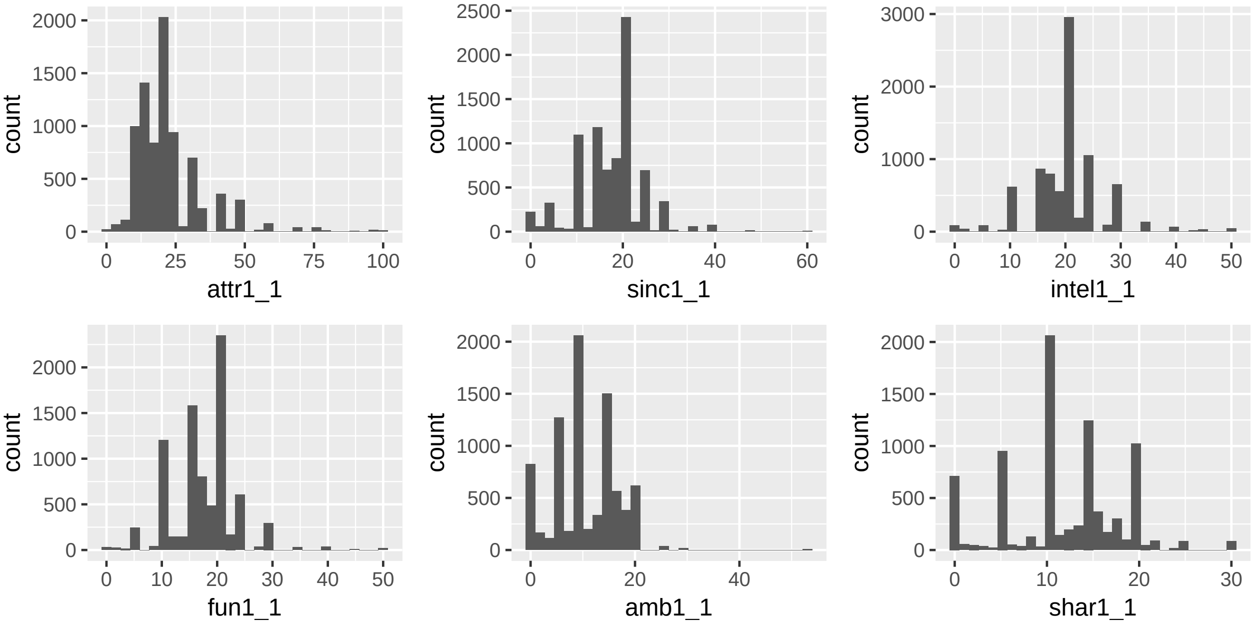
```
# Method 1 : Recode Variable 'Gender'
DT$gender[which(DT$gender == 0)] <- "Female"
DT$gender[which(DT$gender == 1)] <- "Male"
DT$gender <- as.factor(DT$gender)

# Method 2 : Recode Variable 'Match'
DT$match <- as.factor(DT$match)
DT$match <- revalue(DT$match, c("0" = "No", "1" = "Yes"))
```

Exploring Data

```
glimpse(DT)
# Observations: 8,378
# Variables: 22
# $ wave     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
# $ iid      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,...
# $ id       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,...
# $ gender   <fctr> Female, Female, Female, Female, Female, Female, Fema...
# $ idg      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3,...
# $ match    <fctr> No, No, Yes, Yes, Yes, No, No, No, Yes, No, No, No, ...
# $ samerace <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1,...
# $ age_o    <int> 27, 22, 22, 23, 24, 25, 30, 27, 28, 24, 27, 22, 22, 2...
# $ race_o   <int> 2, 2, 4, 2, 3, 2, 2, 2, 2, 2, 2, 2, 4, 2, 3, 2, 2, 2,...
# $ field_cd <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
# $ race     <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, 2, 2,...
# $ imprace  <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,...
# $ imprelig <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5,...
# $ goal     <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1,...
# $ date     <int> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 5, 5, 5, 5, 5, 5, 5, 5,...
# $ go_out   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
# $ attr1_1  <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 45, 45, 45, 4...
# $ sinc1_1  <dbl> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 5, 5, 5, 5, 5...
# $ intel1_1 <dbl> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 25, 25, 25, 2...
# $ fun1_1   <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 20, 20, 20, 2...
# $ amb1_1   <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 0, 0, 0, 0, 0...
# $ shar1_1  <dbl> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 5, 5, 5, 5, 5...
dim(DT)
# [1] 8378   22
head(DT)
# # A tibble: 6 x 22
#    wave   iid    id gender   idg  match samerace age_o race_o field_cd
#   <int> <int> <int> <fctr> <int> <fctr>    <int> <int>  <int>    <dbl>
# 1     1     1     1 Female     1     No        0    27      2        1
# 2     1     1     1 Female     1     No        0    22      2        1
# 3     1     1     1 Female     1    Yes        1    22      4        1
# 4     1     1     1 Female     1    Yes        0    23      2        1
# 5     1     1     1 Female     1    Yes        0    24      3        1
# 6     1     1     1 Female     1     No        0    25      2        1
# # ... with 12 more variables: race <int>, imprace <int>, imprelig <int>,
# #   goal <int>, date <int>, go_out <int>, attr1_1 <dbl>, sinc1_1 <dbl>,
# #   intel1_1 <dbl>, fun1_1 <dbl>, amb1_1 <dbl>, shar1_1 <dbl>
# tail(DT)
# str(DT)
summary(DT)
#       wave            iid              id            gender    
#  Min.   : 1.00   Min.   :  1.0   Min.   : 1.00   Female:4184  
#  1st Qu.: 7.00   1st Qu.:154.0   1st Qu.: 4.00   Male  :4194  
#  Median :11.00   Median :281.0   Median : 8.00                
#  Mean   :11.35   Mean   :283.7   Mean   : 8.96                
#  3rd Qu.:15.00   3rd Qu.:407.0   3rd Qu.:13.00                
#  Max.   :21.00   Max.   :552.0   Max.   :22.00                
#                                  NA's   :1                    
#       idg         match          samerace          age_o      
#  Min.   : 1.00   No :6998   Min.   :0.0000   Min.   :18.00  
#  1st Qu.: 8.00   Yes:1380   1st Qu.:0.0000   1st Qu.:24.00  
#  Median :16.00              Median :0.0000   Median :26.00  
#  Mean   :17.33              Mean   :0.3958   Mean   :26.36  
#  3rd Qu.:26.00              3rd Qu.:1.0000   3rd Qu.:28.00  
#  Max.   :44.00              Max.   :1.0000   Max.   :55.00  
#                                              NA's   :104    
#      race_o         field_cd           race          imprace
#  Min.   :1.000   Min.   : 1.000   Min.   :1.000   Min.   : 0.000
#  1st Qu.:2.000   1st Qu.: 5.000   1st Qu.:2.000   1st Qu.: 1.000
#  Median :2.000   Median : 8.000   Median :2.000   Median : 3.000
#  Mean   :2.757   Mean   : 7.662   Mean   :2.757   Mean   : 3.785
#  3rd Qu.:4.000   3rd Qu.:10.000   3rd Qu.:4.000   3rd Qu.: 6.000
#  Max.   :6.000   Max.   :18.000   Max.   :6.000   Max.   :10.000
#  NA's   :73      NA's   :82       NA's   :63      NA's   :79
#     imprelig           goal            date           go_out
#  Min.   : 1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
#  1st Qu.: 1.000   1st Qu.:1.000   1st Qu.:4.000   1st Qu.:1.000
#  Median : 3.000   Median :2.000   Median :5.000   Median :2.000
#  Mean   : 3.652   Mean   :2.122   Mean   :5.007   Mean   :2.158
#  3rd Qu.: 6.000   3rd Qu.:2.000   3rd Qu.:6.000   3rd Qu.:3.000
#  Max.   :10.000   Max.   :6.000   Max.   :7.000   Max.   :7.000
#  NA's   :79       NA's   :79      NA's   :97      NA's   :79
#     attr1_1          sinc1_1         intel1_1         fun1_1
#  Min.   :  0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
#  1st Qu.: 15.00   1st Qu.:15.00   1st Qu.:17.39   1st Qu.:15.00
#  Median : 20.00   Median :18.18   Median :20.00   Median :18.00
#  Mean   : 22.51   Mean   :17.40   Mean   :20.27   Mean   :17.46
#  3rd Qu.: 25.00   3rd Qu.:20.00   3rd Qu.:23.81   3rd Qu.:20.00
#  Max.   :100.00   Max.   :60.00   Max.   :50.00   Max.   :50.00
#  NA's   :79       NA's   :79      NA's   :79      NA's   :89
#      amb1_1         shar1_1
#  Min.   : 0.00   Min.   : 0.00
#  1st Qu.: 5.00   1st Qu.: 9.52
#  Median :10.00   Median :10.64
#  Mean   :10.68   Mean   :11.85
#  3rd Qu.:15.00   3rd Qu.:16.00
#  Max.   :53.00   Max.   :30.00
#  NA's   :99      NA's   :121

# Tabulating Variable
table(Gender = DT$gender, Match = DT$match)
#         Match
# Gender     No  Yes
#   Female 3494  690
#   Male   3504  690
table(Gender = DT$gender, Same_Race = DT$samerace)
#         Same_Race
# Gender      0    1
#   Female 2526 1658
#   Male   2536 1658
table(Go_Out = DT$go_out, Match = DT$match)
#       Match
# Go_Out   No  Yes
#      1 2103  507
#      2 2511  479
#      3 1660  289
#      4  393   57
#      5  145   19
#      6   86   13
#       7   36    1
table(Race = DT$race, Partner_Race = DT$race_o)
#      Partner_Race
# Race     1    2    3    4    6
#    1    18  238   35  103   22
#    2   238 2724  363 1091  271
#    3    35  363   52  159   48
#    4   103 1091  159  480  133
#    6    22  271   48  133   42
```

Data Wrangling

```
p1 <- ggplot(aes(attr1_1), data = DT) + geom_histogram()
p2 <- ggplot(aes(sinc1_1), data = DT) + geom_histogram()
p3 <- ggplot(aes(intel1_1), data = DT) + geom_histogram()
p4 <- ggplot(aes(fun1_1), data = DT) + geom_histogram()
p5 <- ggplot(aes(amb1_1), data = DT) + geom_histogram()
p6 <- ggplot(aes(shar1_1), data = DT) + geom_histogram()
grid.arrange(p1,p2,p3,p4,p5,p6,nrow=2, ncol=3) #put multiple plots together using grid.arrange() from l
```



You can use the %>% operator with standard R functions as well as your own functions. The rules are simple: the object on the left hand side is passed as the first argument to the function on the right hand side.

- **data %>% function** is the same as **function(my.data)**
- **data %>% function(arg = value)** is the same as **function(data, arg = value)**

```
# Example 1
DT %>% dim
```

```
# [1] 8378   22
dim(DT)
# [1] 8378   22

# Example 2
s1 <- subset(DT, gender == "Male")
s1[1:5,1:6]
# # A tibble: 5 x 6
#    wave   iid    id gender   idg  match
#   <int> <int> <int> <fctr> <int> <fctr>
# 1     1    11     1   Male     2     No
# 2     1    11     1   Male     2     No
# 3     1    11     1   Male     2     No
# 4     1    11     1   Male     2     No
# 5     1    11     1   Male     2     No

s2 <- DT %>% subset(gender == "Male")
s2[1:5, 1:6]
# # A tibble: 5 x 6
#    wave   iid    id gender   idg  match
#   <int> <int> <int> <fctr> <int> <fctr>
# 1     1    11     1   Male     2     No
# 2     1    11     1   Male     2     No
# 3     1    11     1   Male     2     No
# 4     1    11     1   Male     2     No
# 5     1    11     1   Male     2     No
```

```
# Example 1 : (same as function table() )
DT %>% select(imprace) %>% group_by(imprace) %>% tally()
# # A tibble: 12 x 2
#    imprace     n
#      <int> <int>
#  1       0     8
#  2       1  2798
#  3       2   954
#  4       3   983
#  5       4   510
#  6       5   657
#  7       6   524
#  8       7   543
#  9       8   663
# 10       9   409
# 11      10   250
# 12      NA    79
table(DT$imprace)
#
#    0    1    2    3    4    5    6    7    8    9   10
#    8 2798  954  983  510  657  524  543  663  409  250

d1 <- select(DT,imprace)
d2 <- group_by(d1,imprace)

# Example 2 : Compute the average and standard deviation of particular group
```

```
DT %>% filter(race %in% c("2", "3")) %>%
  group_by(race) %>%
  summarise(m=mean(attr1_1, na.rm = TRUE), s = sd(attr1_1, na.rm = TRUE)) %>% kable(digits = 1)
```

race	m	s
2	23.3	12.9
3	21.6	13.6

Confirm the number of males and females in each wave given in the documentation is correct

- To compute some statistic for each group individually, rather than for the data set as a whole, we can use **aggregate** function from library **dplyr**
- **aggregate(y~x,data,function)**
- function(x) length(unique(x)) : defining new function that comes from R
- y ~ x : y is numeric data to be split into groups according to x variable

```
aggregate(id ~ gender + wave , DT, function(x) length(unique(x)))
#    gender wave id
# 1  Female    1 10
# 2    Male    1 10
# 3  Female    2 19
# 4    Male    2 16
# 5  Female    3 10
# 6    Male    3 10
# 7  Female    4 18
# 8    Male    4 18
# 9  Female    5  9
# 10   Male    5 10
# 11 Female    6  5
# 12   Male    6  5
# 13 Female    7 16
# 14   Male    7 16
# 15 Female    8 10
# 16   Male    8 10
# 17 Female    9 20
# 18   Male    9 20
# 19 Female   10  9
# 20   Male   10  9
# 21 Female   11 21
# 22   Male   11 21
# 23 Female   12 14
# 24   Male   12 14
# 25 Female   13 10
# 26   Male   13  9
# 27 Female   14 20
# 28   Male   14 18
# 29 Female   15 18
# 30   Male   15 19
# 31 Female   16  6
# 32   Male   16  8
# 33 Female   17 10
# 34   Male   17 14
# 35 Female   18  6
# 36   Male   18  6
# 37 Female   19 15
# 38   Male   19 15
# 39 Female   20  6
# 40   Male   20  7
# 41 Female   21 22
# 42   Male   21 22
# function(x) length(unique(x)) : defining new function that comes from R
# y ~ x : y is numeric data to be split into groups according to x variable
```

How many people have participated to the speed dating experiment?

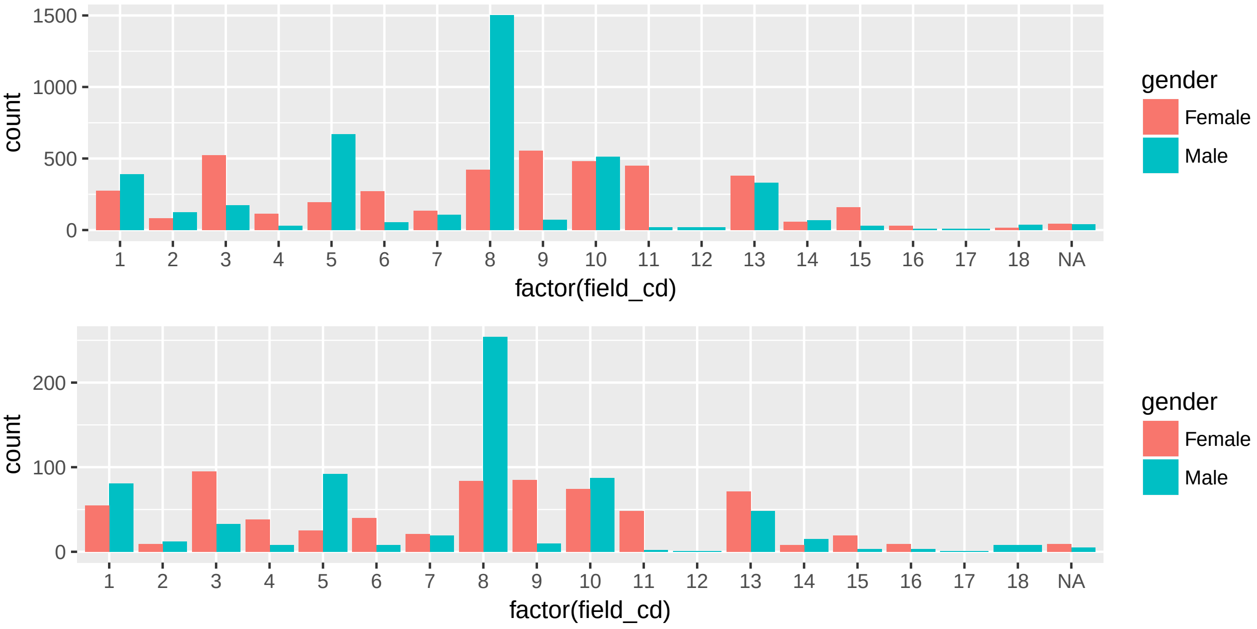
```
length(unique(DT$iid))
# [1] 551
```

How many dates each person has participated to? Compute a summary of these numbers

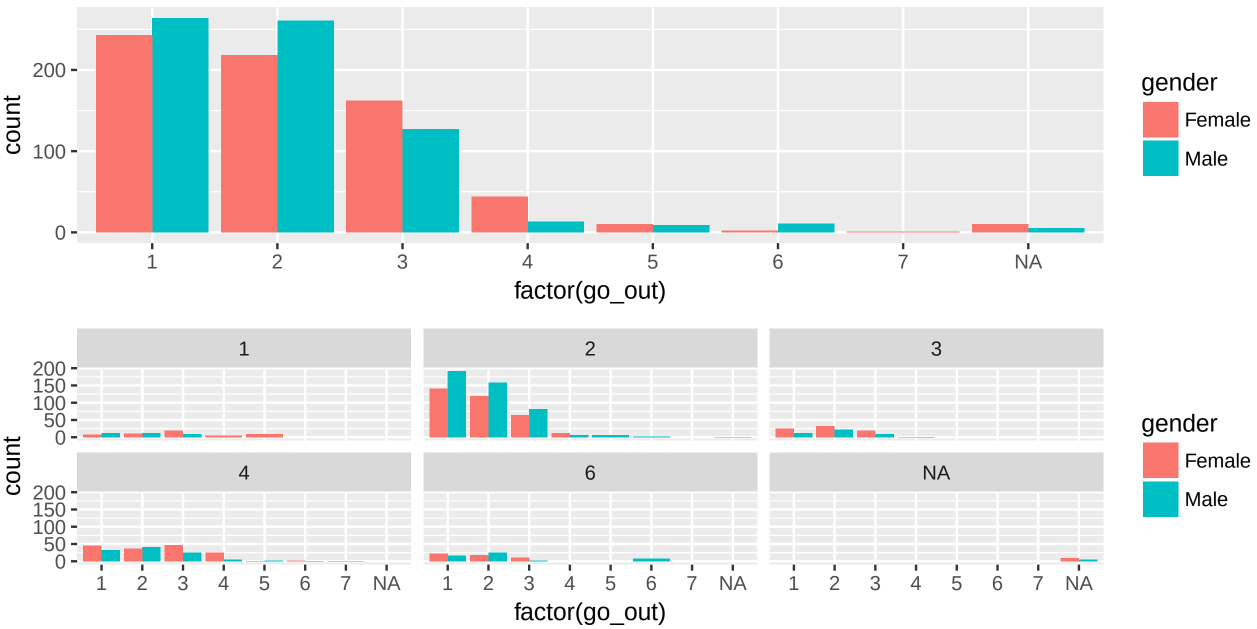
```
 DT.date <- dataset[,c("wave","iid","id","order","pid")]

DT.date.tally <- DT.date %>%
        select(wave, iid, order) %>%
        group_by(wave, iid) %>%
        tally(order)

DT.date.tally
# # A tibble: 551 x 3
# # Groups:   wave [?]
#     wave   iid     n
#    <int> <int> <int>
#  1     1     1    55
#  2     1     2    55
#  3     1     3    55
#  4     1     4    55
#  5     1     5    55
#  6     1     6    55
#  7     1     7    55
#  8     1     8    55
#  9     1     9    55
# 10     1    10    55
# # ... with 541 more rows

DT.date.summary <- DT.date %>%
        select(wave,iid,order) %>%
        group_by(wave,iid) %>%
  summarise(m=mean(order,na.rm=TRUE), s=sd(order,na.rm=TRUE))

DT.date.summary
# # A tibble: 551 x 4
# # Groups:   wave [?]
#      wave   iid     m       s
#     <int> <int> <dbl>   <dbl>
#  1      1     1   5.5 3.02765
#  2      1     2   5.5 3.02765
#  3      1     3   5.5 3.02765
#  4      1     4   5.5 3.02765
#  5      1     5   5.5 3.02765
#  6      1     6   5.5 3.02765
#  7      1     7   5.5 3.02765
#  8      1     8   5.5 3.02765
#  9      1     9   5.5 3.02765
# 10      1    10   5.5 3.02765
# # ... with 541 more rows
```

Visualization

```
##  Field of Study , Gender
p1 <- ggplot(data = DT,aes(x = factor(field_cd), fill = gender))+
  geom_bar(stat="count", position = position_dodge())

p2 <- ggplot(data = subset(DT, as.character(DT$match) == "Yes"), aes(x = factor(field_cd), fill = gende
  geom_bar(stat = "count", position = position_dodge())

grid.arrange(p1, p2, nrow=2, ncol=1)
```



```
## Frequency of Going Out, Gender, Race
p1 <- ggplot(data=subset(DT,as.character(DT$match)=="Yes"),
             aes(x=factor(go_out),fill=gender)) +
  geom_bar(stat="count",position = position_dodge())

p2 <- ggplot(data=subset(DT,as.character(DT$match)=="Yes"),
             aes(x=factor(go_out),fill=gender)) +
  geom_bar(stat="count",position = position_dodge())  +
  facet_wrap(~ race)

grid.arrange(p1, p2, nrow = 2, ncol = 1)
```


count
NA
factor(go_out)
gender
Female
Male
NA
count
factor(go_out)
gender
Female
Male