Lab2

1512005

16 October 2017

1. Initially use import to store the speed dating data.

```
library(rio)
SpeedDatingRawData <- import("./SpeedDatingRawData.csv", setclass = "tibble")
```

2.

```
print(SpeedDatingRawData[,1:20], n=5) #could also do print(data[1:5,1:20])
```

```
## # A tibble: 8,378 x 20
##     iid    id gender   idg condtn  wave round position positin1 order
##   <int> <int>  <int> <int>  <int> <int> <int>    <int>    <int> <int>
## 1     1     1      0     1      1     1    10        7       NA     4
## 2     1     1      0     1      1     1    10        7       NA     3
## 3     1     1      0     1      1     1    10        7       NA    10
## 4     1     1      0     1      1     1    10        7       NA     5
## 5     1     1      0     1      1     1    10        7       NA     7
## # ... with 8,373 more rows, and 10 more variables: partner <int>,
## #   pid <int>, match <int>, int_corr <dbl>, samerace <int>, age_o <int>,
## #   race_o <int>, pf_o_att <dbl>, pf_o_sin <dbl>, pf_o_int <dbl>
```

3. Creating a new dataframe.

```
library(dplyr)
SpeedData <- select(SpeedDatingRawData,
                    iid, wave, partner, gender, attr1_1, sinc1_1, intel1_1, fun1_1, amb1_1, shar1_1)
colnames(SpeedData) #Prints column names of SpeedData
```

```
##  [1] "iid"      "wave"     "partner"  "gender"   "attr1_1"  "sinc1_1"
##  [7] "intel1_1" "fun1_1"   "amb1_1"   "shar1_1"
```

4. Filter SpeedData.

```
SpeedData <- filter(SpeedData,
                    wave != "6" & wave != "7" & wave != "8" & wave != "9", partner == 1)
#Remove waves 6:9, and ensure we only take the first partner of each person
```

5. Remove columns.

```
SpeedData$wave <- NULL
SpeedData$partner <- NULL   #Defining columns to be NULL will remove them from the dataframe
```

6.

```
summary(SpeedData)
```

```
##       iid             gender          attr1_1          sinc1_1
##  Min.   :  1.0   Min.   :0.0000   Min.   :  0.00   Min.   : 0.00
##  1st Qu.:113.0   1st Qu.:0.0000   1st Qu.: 15.00   1st Qu.:10.00
##  Median :328.0   Median :1.0000   Median : 20.00   Median :19.00
##  Mean   :298.2   Mean   :0.5033   Mean   : 24.17   Mean   :17.14
##  3rd Qu.:440.0   3rd Qu.:1.0000   3rd Qu.: 30.00   3rd Qu.:20.00
##  Max.   :552.0   Max.   :1.0000   Max.   :100.00   Max.   :60.00
##                                   NA's   :6        NA's   :6
##     intel1_1         fun1_1          amb1_1          shar1_1
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.000   Min.   : 0.00
##  1st Qu.:17.00   1st Qu.:12.25   1st Qu.: 5.000   1st Qu.: 5.00
##  Median :20.00   Median :18.09   Median :10.000   Median :10.00
##  Mean   :20.47   Mean   :17.35   Mean   : 9.909   Mean   :11.25
##  3rd Qu.:25.00   3rd Qu.:20.00   3rd Qu.:15.000   3rd Qu.:15.00
##  Max.   :50.00   Max.   :50.00   Max.   :53.000   Max.   :30.00
##  NA's   :6       NA's   :7       NA's   :8        NA's   :9
```

Looking at the average of all the attributes, attractiveness had the largest mean at 24.17 and joint largest median at 20 points (intelligence also has 20 for median and 20.47 for mean). This suggests that people consider attractiveness and intelligence significant factors, compared to the other options, when considering a potential dating partner.
However, one person supposedly put all 100 points into attractiveness, which could distort the mean. The mean and median of ambition and shared interests are significantly lower than for the other attributes - median is 10 for both and mean is 9.91 and 11.25 respectively. This shows people seem to value these less than the other attributes, with 50% of people putting these attributes between 5 and 15.

7.

```
SpeedData <- na.omit(SpeedData)
nrow(SpeedData)
```

```
## [1] 440
```

440 Observations remain.

8. Change gender to factor.

```
SpeedData <- mutate(SpeedData, gender = factor(gender, levels = c("0", "1"), labels = c("female", "male
#can check using head(SpeedData)
nrow(subset(SpeedData, gender == "female"))
```

```
## [1] 218
```

```
nrow(subset(SpeedData, gender == "male"))
```

```
## [1] 222
```

There are 218 female and 222 male.

9. Applying rowSums.

```
#select which columns to apply rowSums to
AttributeSums <- transmute(SpeedData,
                    rowSums(select(SpeedData, attr1_1, sinc1_1, intel1_1, fun1_1, amb1_1, shar1_1)))
```

```
#check if any rows do not add to 100, then check how many don't
AttributeSums[AttributeSums != 100,]
```

```
nrow(AttributeSums[AttributeSums != 100,])
```

```
## [1] 16
```

This shows that 16 people did not allocate 100 points. They may have allocated more or less. (Can easily be checked using $>$ and $<$).

To normalize the data we must find any rows which do not sum to 100, divide all attributes in that row by what they do sum to (to change to a percentage), then multiply by 100 to get them between 0 and 100.

10. Normalise the data.

```
#this for loop will check if a row sums to 100, if it doesn't, it will perform the operation described
for(i in 1:nrow(SpeedData)){
  if(rowSums(SpeedData[i,3:8]) != 100){
    SpeedData[i,3]<- (SpeedData[i,3]/max(SpeedData[i,3]))*100
    SpeedData[i,4]<- (SpeedData[i,4]/max(SpeedData[i,4]))*100
    SpeedData[i,5]<- (SpeedData[i,5]/max(SpeedData[i,5]))*100
    SpeedData[i,6]<- (SpeedData[i,6]/max(SpeedData[i,6]))*100
    SpeedData[i,7]<- (SpeedData[i,7]/max(SpeedData[i,7]))*100
    SpeedData[i,8]<- (SpeedData[i,8]/max(SpeedData[i,8]))*100
  }
  return(SpeedData)
}
```

11.

```
export(SpeedData, "Lab2Data.csv")
```

12. Tidying data.

```
messy <- import("./MessyData.csv", setclass = "tibble")
```

The messy dataset contains 3 key variables; Gender, Current Studies, Career Intentions. In tidy data, each observation or total counts should be in a row, with each variable forming a column. This dataset is messy because each observation (or counts) does not form a row, and multiple variables are stored in single columns, e.g. "Business - Law" are two different variables, but are in one column.

13. Using tidyr.

```
library(tidyr)

#Code will gather the number of people into their current studies and career plans
new.messy <- gather(messy, key = Current.Studies, value = Number_of_People,
  'Business - Law', 'Law - Law', 'MBA - Law', 'Business - Banking', 'Law - Banking', 'MBA - Banking')

#Will separate what they currently study and their future career plans
new.messy <- separate(new.messy, col = "Current.Studies", into = c("Current.Study", "Career"))

#We could remove those variables which no people are members of,
#however argument could be made that this is removing information from the data

print(new.messy)
```

```
## # A tibble: 12 x 4
##    Gender Current.Study  Career Number_of_People
##  *  <chr>         <chr>   <chr>            <int>
##  1 female      Business     Law                0
##  2   male      Business     Law                0
##  3 female           Law     Law               12
##  4   male           Law     Law               17
##  5 female           MBA     Law                0
##  6   male           MBA     Law                1
##  7 female      Business Banking                3
##  8   male      Business Banking               19
##  9 female           Law Banking                0
## 10   male           Law Banking                0
## 11 female           MBA Banking                1
## 12   male           MBA Banking               20
```