Lab Report 2 - Data Handeling

Author - 1826475

Date - 22/10/2020

Introduction: The data

The data considered here is from a speed dating study conducted at the Columbia Business school (Fisman et al. 2006) which collated participant responses to a series of questionnaires before, during, directly following and after the speed dating event. The study consisted of a series of 17 successful events running from 2002 to 2004. The link to the original paper can be found here. The dataset used from the study at Columbia Business school (Fisman et al. 2006) is available from Andrew Gelman's website.

During the speed dating event the participants were asked the question, "How do you think you measure up? Please rate your opinion of your own attributes, on a scale of 1-10 (be honest!)". The attributes considered were *attractive sincere, intelligent, fun* and *ambitious* while the variables used to store them were `attr3_1`, `sinc3_1`, `intel3_1`, `fun3_1` and `amb3_1` respectively.

1 - Constructing our Tibble

When creating a new R Markdown file in RStudio it was saved in a pre-defined *SpeedDating* directory which contained all the necessary files. This was set as th relative directory path to make this piece more reproducible.

There are two key differences between a tibble vs. a data frame, namely printing and subsetting. We shall explore these further in this piece.

The following code load the speed dating raw data into the tibble `SpeedDatingRawData` we can use the R command:

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

Here we're simply calling a specific library of commands `rio` in R which helps us import the dataset as a tibble, a pre-defined class.

2 - Subselect and Print

The following code prints the first 3 rows of the columns 41 to 60:

```
print(SpeedDatingRawData[ ,41:60], n = 3)
```

```
## # A tibble: 8,378 x 20
##   imprace imprelig from  zipcode income  goal  date go_out career career_c
##     <int>    <int> <chr> <chr>   <chr>  <int> <int>  <int> <chr>     <dbl>
## 1       2        4 Chic~ 60,521  69,48~     2     7      1 lawyer       NA
## 2       2        4 Chic~ 60,521  69,48~     2     7      1 lawyer       NA
## 3       2        4 Chic~ 60,521  69,48~     2     7      1 lawyer       NA
## # ... with 8,375 more rows, and 10 more variables: sports <int>,
## #   tvsports <int>, exercise <int>, dining <int>, museums <int>, art <int>,
```

```
## #   hiking <int>, gaming <int>, clubbing <int>, reading <int>
```

The print command for tibbles are designed to show only the first 10 rows and all the columns that can fit on the screen to avoid overloading the software when we print. Here we see that only 10 to the 20 columns selected appear on the screen. This helpful when dealing with large datasets such as this one. However, we have asked it to print only the first 3 rows rather than the usual 10 to keep this section more succinct. Additionally, each column name is accompanied by its type; a helpful reminder.

3 - Subsetting the data

The raw data has 8378 rows and 195 columns but, we aren't always necessarily interested in the entire data. Hence, we tend to subset the data. Here we select the ratings for attractive, sincere, intelligent, fun and ambitious as well as `iid` the unique subject number, and `partner` the partner's subject number within wave the night of event.

The following code uses the command *select* to construct a new data frame, `SpeedData` that subsets the original dataset to contain the variables `iid`, `partner`, `attr3_1`, `sinc3_1`, `intel3_1`, `fun3_1` and `amb3_1`:

```
library(dplyr)
SpeedData <- select(SpeedDatingRawData, iid, partner, attr3_1, sinc3_1, intel3_1, fun3_1, amb3_1)
```

The following code prints the column names of our data frame and thereby verifies our code achieves the goal of updating `SpeedData` to be a subset that contains the desired columns:

```
colnames(SpeedData)
```

```
## [1] "iid"      "partner"  "attr3_1"  "sinc3_1"  "intel3_1" "fun3_1"   "amb3_1"
```

4 - Filter

Each speed date is recorded twice, one from the perspective of the man and the other form the perspective of the woman. This means if an individual participated in x dates his scores are recorded x times. Selecting the scores from each participant's first date only helps us lessen this redundancy. Furthermore, the self-perception generally should not drastically change throughout the event as individuals are unaware of how their partners score them.

`SpeedData` data frame currently has 8378 rows and 7 columns.

The code below updates `SpeedData` to contain only the first speed date of each individual:

```
library(dplyr)
n <- nrow(SpeedData)
SpeedData <- filter(SpeedData, (partner == 1))
dim(SpeedData)
```

```
## [1] 551   7
```

```
cat(round((n-nrow(SpeedData))/n*100, 2), "%")
```

```
## 93.42 %
```

Filtering `SpeedData` such that it contains only the first speed date of each individual has removed 93.42 % of observations leaving us with 551 rows and 7 columns.

5 - Remove 'partner'

The following code removes the column`partner` from `SpeedData`as it no longer adds any information to our data following our filtering step:

```
library(dplyr)
SpeedData <- select(SpeedData, -partner)
colnames(SpeedData)
```

```
## [1] "iid"      "attr3_1"  "sinc3_1"  "intel3_1" "fun3_1"   "amb3_1"
```

6 - Basic Data Checks

Here it seems wise to do some basic data checks on our data frame. The function `summary` in R provides a summary of each variable in the data frame.

```
summary(SpeedData)
```

```
##       iid           attr3_1          sinc3_1          intel3_1
##  Min.   :  1.0   Min.   : 2.000   Min.   : 2.000   Min.   : 3.000
##  1st Qu.:139.5   1st Qu.: 6.000   1st Qu.: 8.000   1st Qu.: 8.000
##  Median :277.0   Median : 7.000   Median : 8.000   Median : 8.000
##  Mean   :276.8   Mean   : 7.092   Mean   : 8.286   Mean   : 8.386
##  3rd Qu.:414.5   3rd Qu.: 8.000   3rd Qu.: 9.000   3rd Qu.: 9.000
##  Max.   :552.0   Max.   :10.000   Max.   :10.000   Max.   :10.000
##                  NA's   :9        NA's   :9        NA's   :9
##      fun3_1           amb3_1
##  Min.   : 2.000   Min.   : 2.000
##  1st Qu.: 7.000   1st Qu.: 7.000
##  Median : 8.000   Median : 8.000
##  Mean   : 7.701   Mean   : 7.577
##  3rd Qu.: 9.000   3rd Qu.: 9.000
##  Max.   :10.000   Max.   :10.000
##  NA's   :9        NA's   :9
```

Looking the across variables we notice that `attr3_1`, the variable for attractiveness has the lowest value across the board, highlighting perhaps the hesitance for individuals to give high marks for personal attractiveness. On the other hand, `intel3_1`, the variable for intelligence has scored the highest across the board which is not too surprising considering all participants in the study were graduate students at the University of Columbia. Finally, we notice there are 9 missing variables across the 5 variables, this tends to indicate all missing values occur for the same participants.

7 - NA's

After inspecting the dataset, it becomes clear that all the missing values across the 5 attributes do in fact stem from the same 9 participants. This means we do not lose any information for any attribute by simply excluding these participants from the study, it simply makes our dataset smaller, and thereby justifies our decision to remove all missing values. The following code removes all `NA` values from our tibble `SpeedData`.

```
# missing observations
na <- nrow(SpeedData)
SpeedData <- na.omit(SpeedData)
cat(round((na-nrow(SpeedData))/n*100, 2), "%")
```

```
## 0.11 %
```

Removing observations with missing data from `SpeedData` has removed just 0.11 % of the observations therefore we are omitting a very small amount of observations.

8 - AttributeSums

It could be interesting to look at the total score for each participant. So, the following code adds up all the attributes in each row and stores it as the vector `AttributeSums`.

```
AttributeSums <- rowSums(SpeedData[,-1])
```

Here is the summary of the total scores.

```
summary(AttributeSums)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   36.00   39.50   39.04   42.00   50.00
```

9 - Normalization

Normalization here allows us to change the values of the different variables in our dataset in order to bring therm onto a common scale, without affecting the differences in the value ranges across variables.

The code below normalizes `AttributeSums`, the total scores to add up to a total of 100:

```
SpeedData <- mutate_at(SpeedData, .vars = vars(attr3_1:amb3_1), .funs = (~.*100/AttributeSums))
```

As a measure, we can check the correctness of our normalization using the code below which shows the sum of each row is **100**. We first add up all the attributes in each row and stores it as the vector `Attr_NormSum`. Then, utilizing an **if loop** we ask *R* to print *TRUE* if each row sums up to **100**.

```
Attr_NormSum <- rowSums(SpeedData[,-1])
if(sum(Attr_NormSum) == 100*length(Attr_NormSum)){
  print(TRUE)
} else {
  print(FALSE)
}
```

```
## [1] TRUE
```

10 - Export

For the purpose of further analysis involving the dataset we have produced we will export `SpeedData` as a *.csv* file .

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

Reference

Fisman, R., S. S. Iyengar, E. Kamenica, and I. Simonson. 2006. "Gender Differences in Mate Selection: Evidence from a Speed Dating Experiment." The Quarterly Journal of Economics 121 (2). MIT Press: 673–97.