Generate a Small Survey Dataset in R

Fan Wang

2020-05-02

Contents

1 Generate A Dataset in R

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools Package, R Code Examples Repository (bookdown site), or Intro Stats with R Repository (bookdown site).

R has a Variety of built-in small datasets for you to experiment with, for example, opening up R, you can just type in: mtcars, and that will show you some variables and observations. The command below shows the first 5 rows of dataset **mtcars**:

```
head(mtcars, 5)
```

There are 52 students in a class. I am interested in how many of you go to games at the University, and how many years each of you has lived in the city of Houston. But I do not have time to ask each of you questions.

- My population is all 52 students in the class.
- Given my limited time and resources, I will gather a sample of just 10 students.

Gathering information regarding game attendance for the 10 students, I can perhaps gain some insights about the population.

1.1 A Random Sample of Students in Class

I can point and pick ten random people, but how do I know I am choosing/selecting randomly? In general, we want to pick a random sample from the population. You don't want your sample of 10 to just be students in the first row or students who happen to be giving eye contacts. We want a random sample that hopefully gives us some representative information about the population of 52 students.

How do you pick random numbers?

Siri, Alexa, Google, . . . , you can get random numbers very easily now. You can download various apps also that allows you to draw a random integer from say between 1 to 4, with an equal chance of drawing any of the four numbers.

In our class: - we have 4 rows of seats - we have 13 columns of seats - there are 52 students in the class together. I will now randomly pick between 1 and 4, 10 times, and randomly pick between 1 and 13 also 10 times.

This gives me ten pairs of row and columns numbers, and these indicate which ten students will be a part of a randomly drawn sample of students.

```
# Setting seed means we will get the same set of random numbers each time
set.seed(12345)
# in R, draw integers between 1 and 4, 10 times
ROW.RAND.DRAWS = sample(1:4, 10, replace = TRUE)
# in R, draw integers between 1 and 13, 10 times
COL.RAND.DRAWS = sample(1:13, 10, replace = TRUE)
# Note that the way we are drawing randomly, we could draw the same individual twice.
rbind(ROW.RAND.DRAWS, COL.RAND.DRAWS)
```

```
##                 [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## ROW.RAND.DRAWS    2    3    4    2    4    4    2    1    3     4
## COL.RAND.DRAWS    8    2    6   11    6    7   10    1    8     7
```

1.2 Create the Dataset Row by Row

R has *base* packages. Many of R's best packages do not come with the base packages. You have to install separately. One of the best packages for data science is tidyverse. **tidyverse** has a great package called tibble, which is a great way to deal with what are called dataframes, where we store our data. **tidyverse** also has dplyr, which is a powerful set of tools for transforming data.

Sequentially, in class, we ask each of the 10 randomly drawn students a number of questions, and record the answers in the dataframe.

```
df <- tibble(ID = integer(), ROW = integer(), COL = integer(),
             gender = factor(),
             years.in.houston = double(),
             major = factor(),
             commute = factor(),
             games.attended = integer())


# Ask Students Questions and Record Answers
df <- add_row(df, ID=1 , ROW=3, COL=1,  gender='MALE',   years.in.houston=21.0,
              major='ECON', commute='YES', games.attended=0)
df <- add_row(df, ID=2 , ROW=4, COL=2,  gender='FEMALE', years.in.houston=21.0,
              major='HEALTH', commute='YES', games.attended=2)
df <- add_row(df, ID=3 , ROW=4, COL=10, gender='MALE',   years.in.houston=22.0,
              major='ECON', commute='YES', games.attended=0)
df <- add_row(df, ID=4 , ROW=4, COL=1,  gender='MALE',   years.in.houston=22.0,
              major='ECON', commute='YES', games.attended=14)
df <- add_row(df, ID=5 , ROW=2, COL=6,  gender='FEMALE', years.in.houston=20.0,
              major='ECON', commute='YES', games.attended=0)
df <- add_row(df, ID=6 , ROW=1, COL=7,  gender='MALE',   years.in.houston=3.0,
              major='PSYCH', commute='YES', games.attended=0)
df <- add_row(df, ID=7 , ROW=2, COL=6,  gender='MALE',   years.in.houston=25.0,
              major='ECON', commute='YES', games.attended=25)
df <- add_row(df, ID=8 , ROW=3, COL=6,  gender='MALE',   years.in.houston=20.0,
              major='CONSUMERSCIENCE', commute= 'YES', games.attended=2)
df <- add_row(df, ID=9 , ROW=3, COL=3,  gender='FEMALE', years.in.houston=5.0,
              major='HUMANRESOURCE', commute='YES', games.attended=0)
df <- add_row(df, ID=10, ROW=4, COL=13, gender='FEMALE', years.in.houston=20.0,
              major='ECON', commute='YES', games.attended=0)
```

```r
# List All Variables in df
str(df)
```

```
## tibble [10 x 8] (S3: tbl_df/tbl/data.frame)
##  $ ID              : num [1:10] 1 2 3 4 5 6 7 8 9 10
##  $ ROW             : num [1:10] 3 4 4 4 2 1 2 3 3 4
##  $ COL             : num [1:10] 1 2 10 1 6 7 6 6 3 13
##  $ gender          : chr [1:10] "MALE" "FEMALE" "MALE" "MALE" ...
##  $ years.in.houston: num [1:10] 21 21 22 22 20 3 25 20 5 20
##  $ major           : chr [1:10] "ECON" "HEALTH" "ECON" "ECON" ...
##  $ commute         : chr [1:10] "YES" "YES" "YES" "YES" ...
##  $ games.attended  : num [1:10] 0 2 0 14 0 0 25 2 0 0
```

```r
# Show full df Table
df
```

1.3 Analyze the Dataset and Create Additional Variables

Based on the variables we gathered, we can generate some additional variables. Specifically, we will generate dummy variables for games.attended, and major. Let's Generate Basic Sample Statistics. Summary will

```r
# Generate Binary Variable of Having attended any games or not
df$games.any <- ifelse(df$games.attended>0, 1, 0)
df$games.any <- factor(df$games.any, levels = c(1,0), labels = c('Has.Attended', 'Never.Attended'))

# Generate Binary Variable Econ or Not, 1 if major is ECON, 0 otherwise
df$econ <- ifelse(df$major=='ECON', 1, 0)
df$econ <- factor(df$econ, levels = c(1, 0), labels = c('ECON', 'Not.Econ'))

# new data set with the two additional variables
df
```

We have a continuous/quantitative variable for games.attended, based on which we generated a binary/dummy variable of games.any to indicate if someone has attended any games or not.

- What fraction of students in the sample have attended games?
  - 60 percent of students never attended any games
- What is the average number of games students attended?
  - But on average students attended 4.3 games (because one student attended 25 games, and another student in the sample attended 10 games)

Suppose you are running the sports program, and each attendee of games pays the same ticket price, if you only care about total revenue, you only care about the average of 4.3 games attended per student. It does not matter if 1 student attended 10 games, or 10 students attended 1 game each, you get the same total revenue, and the same game per student attended.

But if you are trying to generate school spirit from the football program, you might care more about the 60 percent number, which you might think is too high. Too many students never going to any games.

If someone is writing an article about the popularity of football at the University, they can use the 4.3 number to say that it is really popular, or the 60 non-attendance ratio to say it is not that popular. So you see when we read news reports that have data, or when others provide us with statistics, we have to be careful and think if what they report are showing the full picture. The same simple attendance variable, calculated in two ways, shows two pictures of how popular football is.

```r
# Using the dplyr package, we can find the fraction of individuals in factor variables
# Fraction of individuals who attended any games
df %>%
```

```
  group_by(games.any) %>%
  summarise (freq = n()) %>%
  mutate(fraction = freq / sum(freq))
```

```
# Average Game Attendance Overall, and for each sub-group, conditional on attending any or no attendanc
# This creates a column that has overall average for games.attended, showing in every row:
df %>% summarise(games.attended.avg =mean(games.attended))
# This creates a column that has overall average for games.attended, showing in every row:
df %>% group_by(games.any) %>% summarise(games.attended.avg = mean(games.attended))
```

```
# To show both overall average and the group specific average together
#   the first mutate adds to df a column that has the overall average, same value every row
#   then when we group by, we can calculate within group average for games.attended
#   we can also take the average of the avg var, which is the same for all rows, so grouped averages ar
df %>%
   mutate(avg=mean(games.attended)) %>%
   group_by(games.any) %>%
   summarise(avg.group=mean(games.attended),avg.overall=mean(avg))
```

If we drew a different set of 10 people from the class, we would likely get different statistics than what we have in the current random sample. We want to make inference based on the sample about the population knowing that each set of random draws could give different sample statistics. Sample statistics is not population statistics. If our samples are smaller than 10, our sample statistics are likely to be even further away from population statistics.

1.4 Store the data

A standard format for data storage is CSV (comma separated files). The file has texts and can be opened by any text editor. Each row of text corresponds to a row in the table above, which is an observation in the dataset. Commas separate values for each variable. The first row stores the list of column variable names, also separated by commas.

```
# This File is written in a subfolder survey of folder Stat4Econ
# In Stat4Econ, there is another folder called data
# Typing in ../ means go back to the master folder
# /data/ means go to the parallel subfolder data and store our csv file there
write_csv(df , path = 'data/classsurvey.csv')
```