Computing and reporting descriptive statistics

Data cleaning to prepare for making a table

Jenine Harris
Brown School

Bringing in and cleaning the BRFSS data

- Before creating the table, bring in and clean the data as shown in prior videos or text:

```
# import the 2014 BRFSS data
brfss.trans.2014 <- read.csv(file = "~/Box/teaching/Teaching/Fall2020/da

# open tidyverse
library(package = "tidyverse")

# cleaning the TRNSGNDR variable
brfss.2014.cleaned <- brfss.trans.2014 %>%
  mutate(TRNSGNDR = recode_factor(.x = TRNSGNDR,
                                  `1` = 'Male to female',
                                  `2` = 'Female to male',
                                  `3` = 'Gender non-conforming',
                                  `4` = 'Not transgender',
                                  `7` = 'Not sure',
                                  `9` = 'Refused'))%>%
  mutate(PHYSHLTH = na_if(PHYSHLTH, 77)) %>%
  mutate(PHYSHLTH = na_if(PHYSHLTH, 99)) %>%
  mutate(PHYSHLTH = as.numeric(recode(PHYSHLTH, `88` = 0L)))
```

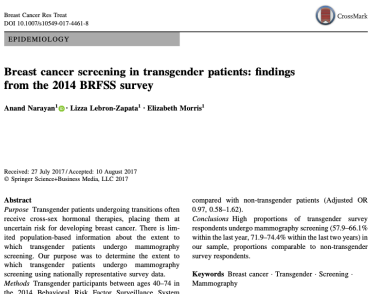
Developing clear tables for reporting descriptive statistics

Clear tables tend to have the following features:

- A title that explains what is in the table
  - The number of observations if possible
  - key pieces of information that describe the sample such as the year of data collection and the data source
  - The units of measurement (people, organizations, etc.)
- Consistent use of the same number of decimal places throughout the table
- Numbers aligned to the right so that the decimal points line up
- Words aligned to the left
- Indentation and shading to differentiate rows or sections
- Limited internal lines
- Clearly labelled rows and columns

Reproducing the transgender health table

- The goal is to reproduce Table 1 from the article:



Breast Cancer Res Treat
DOI 10.1007/s10549-017-4461-8

EPIDEMIOLOGY

Breast cancer screening in transgender patients: findings from the 2014 BRFSS survey

Anand Narayan[1] · Lizza Lebron-Zapata[1] · Elizabeth Morris[1]

Received: 27 July 2017 / Accepted: 10 August 2017

Abstract
Purpose Transgender patients undergoing transitions often receive cross-sex hormonal therapies, placing them at uncertain risk for developing breast cancer. There is limited population-based information about the extent to which transgender patients undergo mammography screening. Our purpose was to determine the extent to which transgender patients undergo mammography screening using nationally representative survey data.
Methods Transgender participants between ages 40–74 in the 2014 Behavioral Risk Factor Surveillance System compared with non-transgender patients (Adjusted OR 0.97, 0.58–1.62).
Conclusions High proportions of transgender survey respondents undergo mammography screening (57.9–66.1% within the last year, 71.9–74.4% within the last two years) in our sample, proportions comparable to non-transgender survey respondents.

Keywords Breast cancer · Transgender · Screening · Mammography

Data cleaning before analysis: creating a smaller data frame

- The table to reproduce does not include all the variables in the data frame exactly as they are
- The data will need some cleaning and recoding before creating the table:

Who is in the table?

- The table contains only those who:
  - answered the transgender status question
  - were in the 40-74 year old age groups
  - and were asked the mammogram question.
- These three variables are in the codebook:
  - `TRNSGNDR`: codebook page 83
  - `_AGEG5YR`: codebook page 108
  - `HADMAM`: codebook page 37

Create a smaller data set with filter()

- The `filter()` functions keeps the `observations` in a data set that meet the set criteria
- In this case, the criteria are having one of the trangender statuses, being in a relevant age group, and being asked the mammogram question.

```
# subset the data set to keep
# transgender status of MtF OR FtM OR Gender non-conforming
# age group higher than group 4 and lower than group 12
# was asked mammogram question
brfss.2014.small <- brfss.2014.cleaned %>%
  filter(TRNSGNDR == 'Male to female'|
           TRNSGNDR == 'Female to male'|
           TRNSGNDR ==  'Gender non-conforming') %>%
  filter(X_AGEG5YR > 4 & X_AGEG5YR < 12) %>%
  filter(!is.na(HADMAM))
```

Check the new data frame

```
# check the new data frame
summary(object = brfss.2014.small)
```

```
##                       TRNSGNDR     X_AGEG5YR           X_RACE         X_INCOMG
##  Male to female             : 77   Min.   : 5.000   Min.   :1.000   Min.   :1.000
##  Female to male             :113   1st Qu.: 7.000   1st Qu.:1.000   1st Qu.:2.000
##  Gender non-conforming      : 32   Median : 8.000   Median :1.000   Median :4.000
##  Not transgender            :  0   Mean   : 7.986   Mean   :2.054   Mean   :3.685
##  Not sure                   :  0   3rd Qu.: 9.000   3rd Qu.:2.000   3rd Qu.:5.000
##  Refused                    :  0   Max.   :11.000   Max.   :9.000   Max.   :9.000
##
##     X_EDUCAG        HLTHPLN1          HADMAM          X_AGE80
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :40.00
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:50.00
##  Median :3.000   Median :1.000   Median :1.000   Median :57.00
##  Mean   :2.595   Mean   :1.108   Mean   :1.171   Mean   :56.83
##  3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:63.75
##  Max.   :4.000   Max.   :2.000   Max.   :9.000   Max.   :74.00
##
##     PHYSHLTH
##  Min.   : 0.000
##  1st Qu.: 0.000
##  Median : 1.000
##  Mean   : 7.528
##  3rd Qu.:11.000
##  Max.   :30.000
##  NA's   :10
```

Select variables that are in the table

- Now that the data set contained the observations used to create the table, select only the variables used to create the table.
- In addition to transgender status, age categories, and mammogram information, the table contains percentages for race-ethnicity, income category, education category, and health insurance status.
- Altogether, the variables for the table are:
  - `TRNSGNDR`
  - `X_AGEG5YR`
  - `X_RACE`
  - `X_INCOMG`
  - `X_EDUCAG`
  - `HLTHPLN1`
  - `HADMAM`

Use select() to select variables

```
# subset observations and variables
brfss.2014.small <- brfss.2014.cleaned %>%
  filter(TRNSGNDR == 'Male to female'|
           TRNSGNDR == 'Female to male'|
           TRNSGNDR ==  'Gender non-conforming') %>%
  filter(X_AGEG5YR > 4 & X_AGEG5YR < 12) %>%
  filter(!is.na(HADMAM)) %>%
  select(TRNSGNDR, X_AGEG5YR, X_RACE, X_INCOMG, X_EDUCAG, HLTHPLN1, HADM
```

Check the data

```
# check the data
summary(object = brfss.2014.small)
```

```
##                     TRNSGNDR     X_AGEG5YR           X_RACE         X_INCOMG
##  Male to female         : 77   Min.   : 5.000   Min.   :1.000   Min.   :1.000
##  Female to male         :113   1st Qu.: 7.000   1st Qu.:1.000   1st Qu.:2.000
##  Gender non-conforming: 32   Median : 8.000   Median :1.000   Median :4.000
##  Not transgender        :  0   Mean   : 7.986   Mean   :2.054   Mean   :3.685
##  Not sure               :  0   3rd Qu.: 9.000   3rd Qu.:2.000   3rd Qu.:5.000
##  Refused                :  0   Max.   :11.000   Max.   :9.000   Max.   :9.000
##     X_EDUCAG        HLTHPLN1          HADMAM
##  Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :3.000   Median :1.000   Median :1.000
##  Mean   :2.595   Mean   :1.108   Mean   :1.171
##  3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :4.000   Max.   :2.000   Max.   :9.000
```

Fix variable types

- Some of the variables are the wrong data type since R has computed the mean and median for each one when they are all categorical and should all be factor data type.
- There is a variation on `mutate()` that can be used to change all the variables in a data set to factor types.
- The `mutate_all()` function can be used to do something to every variable in a data frame.

```
# change variables to factor data types
brfss.2014.small <- brfss.2014.cleaned %>%
  filter(TRNSGNDR == 'Male to female'|
           TRNSGNDR == 'Female to male'|
           TRNSGNDR ==  'Gender non-conforming') %>%
  filter(X_AGEG5YR > 4 & X_AGEG5YR < 12) %>%
  filter(!is.na(HADMAM)) %>%
  select(TRNSGNDR, X_AGEG5YR, X_RACE, X_INCOMG, X_EDUCAG, HLTHPLN1, HADM
  mutate_all(as.factor)
```

Check the data

```
# check the data
summary(object = brfss.2014.small)
```

```
##                     TRNSGNDR    X_AGEG5YR      X_RACE     X_INCOMG X_EDUCAG HLTHP
##  Male to female         : 77   5 :27     1        :152   1:46     1:24     1:198
##  Female to male         :113   6 :27     2        : 31   2:44     2:86     2: 24
##  Gender non-conforming: 32   7 :32     8        : 11   3:19     3:68
##  Not transgender        :  0   8 :44     7        :  8   4:26     4:44
##  Not sure               :  0   9 :44     5        :  7   5:65
##  Refused                :  0   10:24     4        :  6   9:22
##                              11:24     (Other):  7
##  HADMAM
##  1:198
##  2: 22
##  9:  2
##
##
##
##
```

Adding labels to variables

```
# add labels to factor variables
brfss.2014.small <- brfss.2014.cleaned %>%
  filter(TRNSGNDR == 'Male to female'|
           TRNSGNDR == 'Female to male'|
           TRNSGNDR ==  'Gender non-conforming') %>%
  filter(X_AGEG5YR > 4 & X_AGEG5YR < 12) %>%
  filter(!is.na(HADMAM)) %>%
  select(TRNSGNDR, X_AGEG5YR, X_RACE, X_INCOMG, X_EDUCAG, HLTHPLN1, HADM
  mutate_all(as.factor) %>%
  mutate(X_AGEG5YR = recode_factor(.x = X_AGEG5YR,
                                   `5` = '40-44',
                                   `6` = '45-49',
                                   `7` = '50-54',
                                   `8` = '55-59',
                                   `9` = '60-64',
                                   `10` = '65-69',
                                   `11` = '70-74')) %>%
  mutate(X_INCOMG = recode_factor(.x = X_INCOMG,
                                  `1` = 'Less than $15,000',
                                  `2` = '$15,000 to less than $25,000',
                                  `3` = '$25,000 to less than $35,000',
                                  `4` = '$35,000 to less than $50,000',
                                  `5` = '$50,000 or more',
                                  `9` = 'Don\'t know/not sure/missing'))
  mutate(X_EDUCAG = recode_factor(.x = X_EDUCAG,
                                  `1` = 'Did not graduate high school',
                                  `2` = 'Graduated high school',
```

Check the labels

```
#check the work so far
summary(object = brfss.2014.small)
```

```
##                    TRNSGNDR   X_AGEG5YR      X_RACE
##  Male to female        : 77   40-44:27   1       :152
##  Female to male        :113   45-49:27   2       : 31
##  Gender non-conforming: 32   50-54:32   8       : 11
##  Not transgender       :  0   55-59:44   7       :  8
##  Not sure              :  0   60-64:44   5       :  7
##  Refused               :  0   65-69:24   4       :  6
##                               70-74:24   (Other):  7
##                            X_INCOMG                                    X_EDUC
##  Less than $15,000            :46   Did not graduate high school          :2
##  $15,000 to less than $25,000:44   Graduated high school                  :8
##  $25,000 to less than $35,000:19   Attended College/Technical School      :6
##  $35,000 to less than $50,000:26   Graduated from College/Technical School:4
##  $50,000 or more              :65
##  Don't know/not sure/missing :22
##
##  HLTHPLN1  HADMAM
##  Yes:198   1:198
##  No : 24   2: 22
##            9:  2
##
##
##
##
```

Collapsing categories for the race variable

- Review the BRFSS codebook page 106:
    - White only, non-Hispanic
    - Black only, non-Hispanic
    - American Indian or Alaskan Native only, Non-Hispanic
    - Asian only, non-Hispanic
    - Native Hawaiian or other Pacific Islander only, Non-Hispanic
    - Other race only, non-Hispanic
    - Multiracial, non-Hispanic
    - Hispanic
    - Don't know/Not sure/Refused
- Check the percentages for the race variable:

```
prop.table(x = table(brfss.2014.small$X_RACE))
```

```
##
##           1           2           3           4           5           7           8
## 0.68468468 0.13963964 0.01801802 0.02702703 0.03153153 0.03603604 0.04954955
##           9
## 0.01351351
```

Compare the codebook to the table

The table in the 2017 manuscript included the following categories:

- White
- Black
- Native American
- Asian/Pacific Islander
- Other

Mapping the the categories in the codebook into the categories in the table

- Category 1 (White only, non-Hispanic) from the BRFSS data was labeled as *White* in the table
- Category 2 (Black only, non-Hispanic) from the BRFSS data was labeled as *Black* in the table
- Category 3 (American Indian or Alaskan Native only, Non-Hispanic) from BRFSS was *Native American*
- Category 4 (Asian only, non-Hispanic) from BRFSS was *Asian/Pacific Islander*
- Due to a mistake in labeling in the paper, categories 5, 6, 7, and 8 from BRFSS were *Other* in the table

Recoding race

- Add to the code so far:

```
mutate(X_RACE = recode_factor(.x = X_RACE,
                              `1` = 'White',
                              `2` = 'Black',
                              `3` = 'Native American',
                              `4` = 'Asian/Pacific Islander',
                              `5` = 'Other',
                              `6` = 'Other',
                              `7` = 'Other',
                              `8` = 'Other',
                              `9` = 'Other'))
```

- Run all the code and check the work:

Check the work

```
# check the work
summary(object = brfss.2014.small)
```

```
##                    TRNSGNDR   X_AGEG5YR                       X_RACE
##  Male to female        : 77   40-44:27   White                   :152
##  Female to male        :113   45-49:27   Black                   : 31
##  Gender non-conforming: 32   50-54:32   Native American         :  4
##  Not transgender       :  0   55-59:44   Asian/Pacific Islander:  6
##  Not sure              :  0   60-64:44   Other                   : 29
##  Refused               :  0   65-69:24
##                               70-74:24
##                          X_INCOMG                                    X_EDUC
##  Less than $15,000           :46   Did not graduate high school          :2
##  $15,000 to less than $25,000:44   Graduated high school                 :8
##  $25,000 to less than $35,000:19   Attended College/Technical School     :6
##  $35,000 to less than $50,000:26   Graduated from College/Technical School:4
##  $50,000 or more             :65
##  Don't know/not sure/missing :22
##
##  HLTHPLN1
##  Yes:198
##  No : 24
##
##
##
##
##
##
```

Recoding problematic values

- Notice there are 222 observations in the data frame and 220 in Table 1 from the paper.
- Since the table only contained percentages, review the percentages to see if you can find where the problem is.
- Percentages are produced with the `prop.table()` command, which needs a `table()` as input. To get a table of transgender status percentages, use:

```
# get percents for TRNSGNDER
prop.table(x = table(brfss.2014.small$TRNSGNDR))
```

```
##
##         Male to female       Female to male Gender non-conforming
##              0.3468468            0.5090090            0.1441441
##       Not transgender             Not sure                Refused
##              0.0000000            0.0000000            0.0000000
```

- Thee values are slightly different from those in the original table.
- This is likely due to the 2 observation difference.
- Using some data sleuthing, you would find this difference was because two observations where the `HADMAM` variable was coded as *9*, or *Refused*, were dropped before computing percentages of the transgender variable but were kept for computing the percentages of all the other variables.

Fixing a tricky data management problem

- One way to fix it is to code `TRNSGNDR` to `NA` for when the `HADMAM` variable is category 9, which is the code for "Refused" using the `if_else()` function.
- The `if_else()` function takes three arguments.
  - The first argument is a logical statement (or condition) that must be either `TRUE` or `FALSE`.
  - The second argument is `true =`. This is where you tell R what to do if the statement from the first argument is `TRUE`.
  - The third argument, `false =` is what you want to happen if the statement from the first argument is `FALSE`.
- The second and third arguments have to be the same data type.
- Goal for this code: "For each person in the data set, if that person's value in `HADMAM` was *not* equal to 9, then leave their `TRNSGNDR` value as it is (do nothing). For everyone else that *does* have a value of 9 in `HADMAM`, change their `TRNSGNDR` value to be `NA`".

Use if_else() to correct coding error (and add droplevels())

```
# complete data management code
brfss.2014.small <- brfss.2014.cleaned %>%
  filter(TRNSGNDR == 'Male to female'|
           TRNSGNDR == 'Female to male'|
           TRNSGNDR ==  'Gender non-conforming') %>%
  filter(X_AGEG5YR > 4 & X_AGEG5YR < 12) %>%
  filter(!is.na(HADMAM)) %>%
  mutate(TRNSGNDR = if_else(condition = HADMAM != 9,
                            true = TRNSGNDR,
                            false = factor(NA))) %>%
  select(TRNSGNDR, X_AGEG5YR, X_RACE,X_INCOMG, X_EDUCAG, HLTHPLN1) %>%
  mutate_all(as.factor) %>%
  mutate(X_AGEG5YR = recode_factor(.x = X_AGEG5YR,
                                   `5` = '40-44',
                                   `6` = '45-49',
                                   `7` = '50-54',
                                   `8` = '55-59',
                                   `9` = '60-64',
                                   `10` = '65-69',
                                   `11` = '70-74')) %>%
  mutate(X_INCOMG = recode_factor(.x = X_INCOMG,
                                  `1` = 'Less than $15,000',
                                  `2` = '$15,000 to less than $25,000',
                                  `3` = '$25,000 to less than $35,000',
```

Check the work

```
#check the work
prop.table(x = table(brfss.2014.small$TRNSGNDR))
```

```
##
##         Male to female         Female to male Gender non-conforming
##              0.3500000              0.5090909              0.1409091
```