Computing and reporting descriptive statistics

Using tableone to create 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)))
```

More data cleaning

```
# 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',
                                  `4` = '$35,000 to less than $50,000',
                                  `5` = '$50,000 or more',
                                  `9` = 'Don\'t know/not sure/missing'))
```

Creating a table from the clean data

- Creating well-formatted tables easily is one of the few things that R *does not do very well*.
- The tableone package is the best place to start, even though the tables it creates are not in the easiest format to use for formal reporting.
- The tableone package can create a table that includes all the variables in a data frame and automatically selects descriptive statistics to report based on the variable type.

```
# open tableone after installing it
library(package = "tableone")

# create a basic table
CreateTableOne(data = brfss.2014.small)
```

The output from CreateTableOne()

```
##
##                                                  Overall
##    n                                             222
##    TRNSGNDR (%)
##       Male to female                              77 (35.0)
##       Female to male                             112 (50.9)
##       Gender non-conforming                       31 (14.1)
##    X_AGEG5YR (%)
##       40-44                                       27 (12.2)
##       45-49                                       27 (12.2)
##       50-54                                       32 (14.4)
##       55-59                                       44 (19.8)
##       60-64                                       44 (19.8)
##       65-69                                       24 (10.8)
##       70-74                                       24 (10.8)
##    X_RACE (%)
##       White                                      152 (68.5)
##       Black                                       31 (14.0)
##       Native American                              4 ( 1.8)
##       Asian/Pacific Islander                       6 ( 2.7)
##       Other                                       29 (13.1)
##    X_INCOMG (%)
##       Less than $15,000                           46 (20.7)
##       $15,000 to less than $25,000                44 (19.8)
```

Reviewing the tableone table

A few things to work on to reproduce the table more closely:

- The table only has percentages and not frequencies
- The `HLTHPLN1` variable shows the "Yes" group
- The headings for different sections are not variable names
- The percent signs can be removed from the section headers

Creating labels for the sections

- The `print()` command that goes with `CreateTableOne()` has some options for changing how the table prints, including an option to add variable labels, which will address the third bullet.
- To use this feature, the variables must have labels in the data frame, though.
- To check and see if the variables have labels, use `str(brfss.2014.small)` which shows the *structure* of the data frame.

```
# check the labels for the data frame
str(object = brfss.2014.small)
```

```
## 'data.frame':    222 obs. of  6 variables:
##  $ TRNSGNDR : Factor w/ 3 levels "Male to female",..: 1 1 2 3 1 3 2 1 1 2 ..
##  $ X_AGEG5YR: Factor w/ 7 levels "40-44","45-49",..: 4 7 2 6 3 4 1 2 4 1 ...
##  $ X_RACE   : Factor w/ 5 levels "White","Black",..: 1 1 5 5 1 1 1 1 1 5 ...
##  $ X_INCOMG : Factor w/ 6 levels "Less than $15,000",..: 6 5 6 1 3 2 5 1 5 6
##  $ X_EDUCAG : Factor w/ 4 levels "Did not graduate high school",..: 2 4 4 1
##  $ HLTHPLN1 : Factor w/ 2 levels "Yes","No": 2 1 1 2 2 1 2 1 1 2 ...
```

- The output from `str()` does not include anything that looks like labels.

Add variable labels

- The help documentation shows that the tableone package uses labels created in the labelled package.
- Install the labelled package and review its help documentation.
- Use the `var_label()` function with the `brfss.2014.small` data frame to create variable labels.

```
# add variable labels to print in table
labelled::var_label(x = brfss.2014.small) <- c("Transition status (n = 2
                                               "Age category",
                                               "Race/ethnicity",
                                               "Income category",
                                               "Education category",
                                               "Health insurance?")

# check data frame for labels
str(object = brfss.2014.small)
```

```
## 'data.frame':    222 obs. of  6 variables:
##  $ TRNSGNDR : Factor w/ 3 levels "Male to female",..: 1 1 2 3 1 3 2 1 1 2 ..
##   ..- attr(*, "label")= chr "Transition status (n = 220)"
##  $ X_AGEG5YR: Factor w/ 7 levels "40-44","45-49",..: 4 7 2 6 3 4 1 2 4 1 ...
##   ..- attr(*, "label")= chr "Age category"
##  $ X_RACE   : Factor w/ 5 levels "White","Black",..: 1 1 5 5 1 1 1 1 1 5 ...
##   ..- attr(*, "label")= chr "Race/ethnicity"
##  $ X_INCOMG : Factor w/ 6 levels "Less than $15,000",..: 6 5 6 1 3 2 5 1 5 6
```

Add the labels to the CreateTableOne output

```
# create a basic table as an object
trans.hc.table <- CreateTableOne(data = brfss.2014.small)

# use print to show table with labels
print(x = trans.hc.table, varLabels = TRUE)
```

```
##
##                                                  Overall
##   n                                              222
##   Transition status (n = 220) (%)
##      Male to female                               77 (35.0)
##      Female to male                              112 (50.9)
##      Gender non-conforming                        31 (14.1)
##   Age category (%)
##      40-44                                        27 (12.2)
##      45-49                                        27 (12.2)
##      50-54                                        32 (14.4)
##      55-59                                        44 (19.8)
##      60-64                                        44 (19.8)
##      65-69                                        24 (10.8)
##      70-74                                        24 (10.8)
##   Race/ethnicity (%)
##      White                                       152 (68.5)
##      Black                                        31 (14.0)
```

Remove the frequencies from the table

- The help documentation shows that adding `format = "p"` to the `print()` function will result in just printing the percentages.
- The help documentation also shows that the percent (%) symbols can be removed with `explain = FALSE`.

```
# use print to show table with labels and percent
print(x = trans.hc.table,
      varLabels = TRUE,
      format = "p",
      explain = FALSE)
```

```
##
##                                                Overall
##   n                                             222
##   Transition status (n = 220)
##      Male to female                             35.0
##      Female to male                             50.9
##      Gender non-conforming                      14.1
##   Age category
##      40-44                                      12.2
##      45-49                                      12.2
##      50-54                                      14.4
```

Restrict categories

The last thing to fix is to show the "Yes" category for the health insurance variable, however, this is not an option without a bunch of additional data management.

- The table is ready to use in a report or manuscript:

```
# use print to show table with labels and percent
print(x = trans.hc.table,
      varLabels = TRUE,
      format = "p",
      explain = FALSE)
```

```
##
##                                                  Overall
##   n                                               222
##   Transition status (n = 220)
##      Male to female                               35.0
##      Female to male                               50.9
##      Gender non-conforming                        14.1
##   Age category
##      40-44                                        12.2
##      45-49                                        12.2
##      50-54                                        14.4
##      55-59                                        19.8
##      60-64                                        19.8
##      65-69                                        10.8
##      70-74                                        10.8
```

That was a lot of work!

All the data cleaning and formatting steps in R may feel like a lot of work, however:

- Transcribing the numbers can (and does) result in errors
- Small changes to table contents can be made more easily
- Code can be reused, so developing new tables will take less time after the first one is complete

Adding a continuous variable to the table

- Go back to the data management code and add the `PHYSHLTH` variable to the `select()` list.
- `PHYSHLTH` shows the number of days of poor physical health in the last 30 days, so it is not categorical.
- Remove the `mutate_all(as.factor) %>%` function since the `recode_factor()` worked to change these variables anyhow.

```
# 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(HADMAM != 9, TRNSGNDR, factor(NA))) %>%
  select(TRNSGNDR, X_AGEG5YR, X_RACE,X_INCOMG,
         X_EDUCAG, HLTHPLN1, PHYSHLTH) %>%
  mutate(X_AGEG5YR = recode_factor(.x = X_AGEG5YR,
                                   `5` = '40-44',
                                   `6` = '45-49',
                                   `7` = '50-54',
                                   `8` = '55-59',
```

Add a label for the PHYSHLTH variable

To include `PHYSHLTH` in the table, add a variable label for it to the labelling code:

```
# add variable labels to print in table
labelled::var_label(x = brfss.2014.small) <- c("Transition status (n = 2
                                               "Age category",
                                               "Race/ethnicity",
                                               "Income category",
                                               "Education category",
                                               "Health insurance?",
                                               "Days/month poor physical


# check data frame for labels
str(object = brfss.2014.small)
```

```
## 'data.frame':    222 obs. of  7 variables:
##  $ TRNSGNDR : Factor w/ 3 levels "Male to female",..: 1 1 2 3 1 3 2 1 1 2 ..
##   ..- attr(*, "label")= chr "Transition status (n = 220)"
##  $ X_AGEG5YR: Factor w/ 7 levels "40-44","45-49",..: 4 7 2 6 3 4 1 2 4 1 ...
##   ..- attr(*, "label")= chr "Age category"
##  $ X_RACE   : Factor w/ 5 levels "White","Black",..: 1 1 5 5 1 1 1 1 1 5 ...
##   ..- attr(*, "label")= chr "Race/ethnicity"
##  $ X_INCOMG : Factor w/ 6 levels "Less than $15,000",..: 6 5 6 1 3 2 5 1 5 6
##   ..- attr(*, "label")= chr "Income category"
```

Try the table code again

Then, Leslie copied the table code and ran it to see what happens:

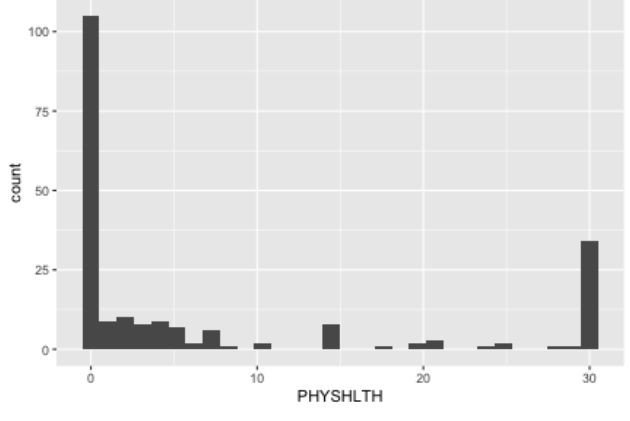
```
# create a basic table as an object
trans.hc.table <- CreateTableOne(data = brfss.2014.small)

# use print to show table with labels
print(x = trans.hc.table, varLabels = TRUE)
```

```
##
##                                                  Overall
##    n                                              222
##    Transition status (n = 220) (%)
##       Male to female                               77 (35.0)
##       Female to male                              112 (50.9)
##       Gender non-conforming                        31 (14.1)
##    Age category (%)
##       40-44                                        27 (12.2)
##       45-49                                        27 (12.2)
##       50-54                                        32 (14.4)
##       55-59                                        44 (19.8)
##       60-64                                        44 (19.8)
##       65-69                                        24 (10.8)
##       70-74                                        24 (10.8)
##    Race/ethnicity (%)
##       White                                       152 (68.5)
##       Black                                        31 (14.0)
##       Native American                               4 ( 1.8)
```

Changing the measures of central tendency & spread

- The mean and standard deviation were added to the table!
- The mean and standard deviation are only good measures for the continuous variables when the variables are normally distributed.
- Examine the distribution of `PHYSHLTH` with a histogram to see if this is appropriate

```
# make a histogram of
brfss.2014.small %>%
  ggplot(aes(x = PHYSHLTH)) +
  geom_histogram()
```



Changing the table stats

- The documentation for the `print()` options for a table show an argument for `nonnormal =` to specify which numeric variables are not normally distributed.

```
# use print to show table
print(x = trans.hc.table,
      varLabels = TRUE,
      nonnormal = 'PHYSHLTH')
```

```
##
##                                                    Overall
##   n                                                 222
##   Transition status (n = 220) (%)
##      Male to female                                  77 (35.0)
##      Female to male                                 112 (50.9)
##      Gender non-conforming                           31 (14.1)
##   Age category (%)
##      40-44                                           27 (12.2)
##      45-49                                           27 (12.2)
##      50-54                                           32 (14.4)
##      55-59                                           44 (19.8)
##      60-64                                           44 (19.8)
##      65-69                                           24 (10.8)
##      70-74                                           24 (10.8)
##   Race/ethnicity (%)
##      White                                          152 (68.5)
##      Black                                           31 (14.0)
```