Data visualization

Graphs for two categorical variables

Jenine Harris
Brown School

Import and clean the data

```
# import nhanes data and open tidyverse
nhanes.2012 <- read.csv(file = "~/Box/teaching/Teaching/Fall2020/data/nh
library(package = "tidyverse")

# clean gun use variable
nhanes.2012.clean <- nhanes.2012  %>%
  mutate(AUQ300 = na_if(x = AUQ300, y = 7)) %>%
  mutate(AUQ300 = recode_factor(.x = AUQ300,
                                `1` = 'Yes',
                                `2` = 'No')) %>%
  rename(gun.use = AUQ300)

#check data cleaning
summary(object = nhanes.2012.clean$gun.use)
```

```
##  Yes   No NA's
## 1613 3061 4690
```

Types of graphs for two categorical variables

- There are multiple options for graphing two categorical variables.
- The two covered in this video are:
  - Mosaic plots
  - Bar plots

Mosaic plots for two categorical variables

- Mosaic plots which show the relative sizes of groups across two categorical variables.
- The NHANES data set used to demonstrate the waffle plot has many categorical variables that might be useful in better understanding gun ownership.
- One possible relationship to examine would be between sex/gender and gun use.
- The `gun.use` variable is already clean.
- The single sex or gender related variable called `RIAGENDR` in the codebook is described as "Gender of the participant" with categories:
    - 1 = Male
    - 2 = Female
    - . = Missing

```
# check coding of RIAGENDR
table(nhanes.2012$RIAGENDR)
```

```
##
##    1    2
## 4663 4701
```

Recoding gender variable

- There are no missing values; add labels to the two categories and rename the variable `sex`:

```
# recode sex variable
nhanes.2012.clean <- nhanes.2012 %>%
  mutate(AUQ300 = na_if(x = AUQ300, y = 7)) %>%
  mutate(AUQ300 = recode_factor(.x = AUQ300,
                                `1` = 'Yes',
                                `2` = 'No')) %>%
  rename(gun.use = AUQ300) %>%
  mutate(RIAGENDR = recode_factor(.x = RIAGENDR,
                                  `1` = 'Male',
                                  `2` = 'Female')) %>%
  rename(sex = RIAGENDR)

#check recoding
summary(object = nhanes.2012.clean$sex)
```

```
##   Male Female
##   4663   4701
```

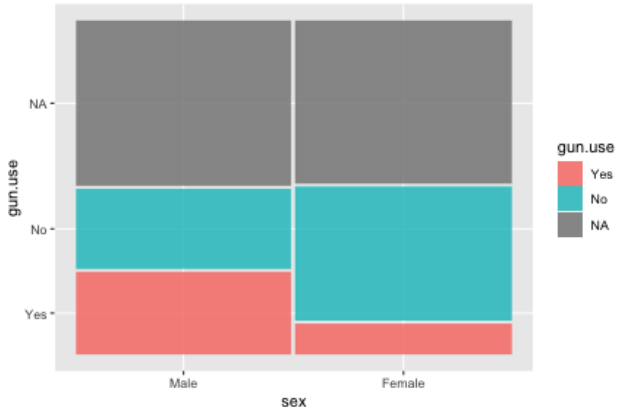
Create a mosaic plot

- The `geom_mosaic()` is not one of the included `geom_` for `ggplot()` so it requires use of the ggmosaic package.
- The `geom_mosaic()` layer is similar to the other `geom_` options, but the variables are added to the aesthetics in the `geom_mosaic()` layer rather than the `ggplot()` layer.

```
# open library
library(package = "ggmosaic")

# mosaic plot of gun use by sex
mosaic.gun.use.sex <- nhanes.2012.clean %>%
  mutate(gun.use, gun.use = na_if(x = gun.use, y = 7)) %>%
  ggplot() +
  geom_mosaic(aes(x = product(gun.use, sex), fill = gun.use))
mosaic.gun.use.sex
```

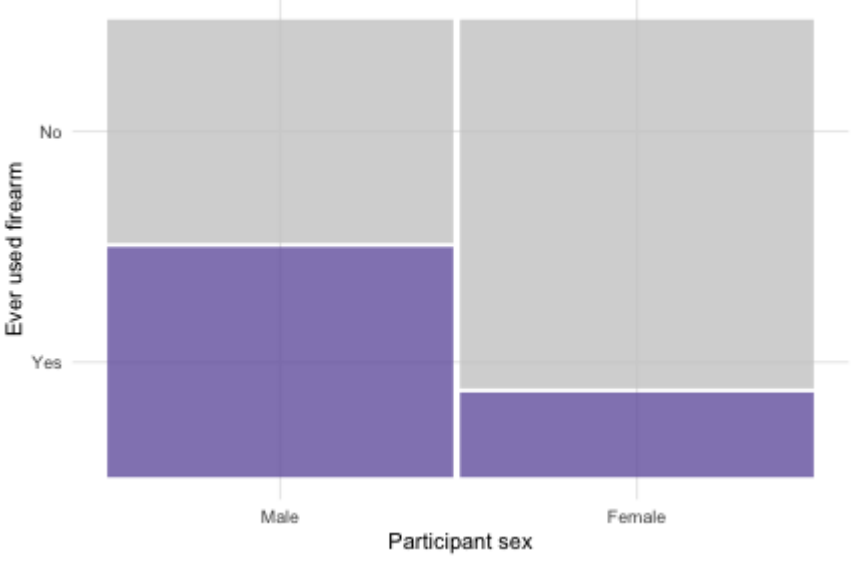
Examining the plot



- The resulting graph shows boxes representing the proportion of males and females who have used a gun and those who have not.
- There are a few things to fix to make the graph more clearly convey the difference in gun use between males and females in this sample:
    - remove the `NA` category
    - add useful labels to the axes
    - remove the legend
    - change the colors to highlight the difference more clearly
    - change the theme so the graph is less cluttered

Formatting the graph

```
# formatted mosaic plot of sex and gun use
# mosaic gun use by sex
mosaic.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot() +
  geom_mosaic(aes(x = product(gun.use, sex), fill = gun.use)) +
  labs(x = "Participant sex", y = "Ever used firearm") +
  scale_fill_manual(values=c("#7463AC", "gray80"),
                    guide = FALSE) +
  theme_minimal()
mosaic.gun.use.sex
```



Bar plots for two categorical variables

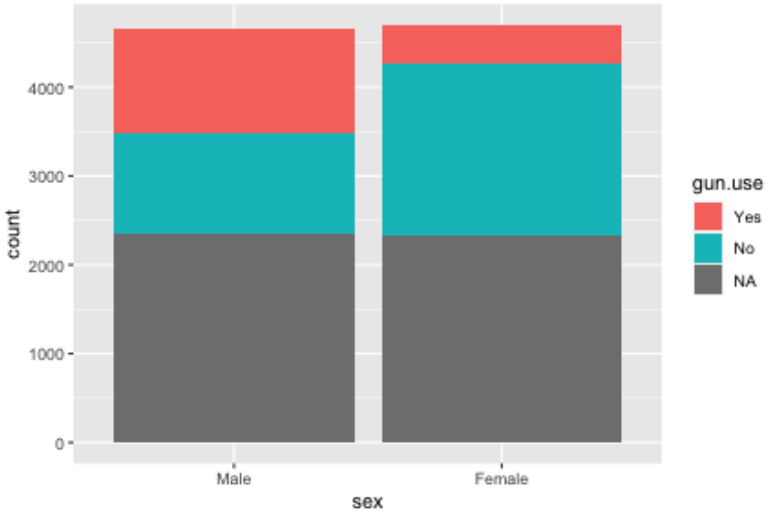
- Mosaic plots are ok for variables with a small number of categories like `gun.use` but using a mosaic plot for variables with many categories is not useful.
- Mosaic plots have some similarity to pie charts because it is hard to tell the relative sizes of some boxes apart, especially when there are more than a few.
- Bar graphs tend to be preferred over mosaic plots fo demonstrating the relationship between two categorical variables.
- Bar graphs showing frequencies across groups can take two formats: (1) stacked, or (2) grouped.
    - Like pie charts, stacked bar graphs show parts of a whole.
    - Also like pie charts, if there are many groups or parts that are similar in size, the stacked bar graph is difficult to interpret and *not* recommended.
- Grouped bar plots are usually the best option.

geom_bar vs. geom_col for bar plots

- Stacked and grouped bar plots could be created with `ggplot()`, and there are two types of `geom_` that work:
  - `geom_bar()`
  - `geom_col()`
- `geom_bar()` is used to display the number of cases in each group (parts of a whole)
- `geom_col()` is used to display actual values like like means and percentages rather than parts of a whole and is often used when graphs are created from summary statistics (rather than raw data)
- For example, use `geom_bar()` to show gun use by sex

```
# stacked bar graph
stack.gun.use.sex <- nhanes.2012.clean %>%
  ggplot(aes(x = sex, fill = gun.use)) +
  geom_bar()
stack.gun.use.sex
```

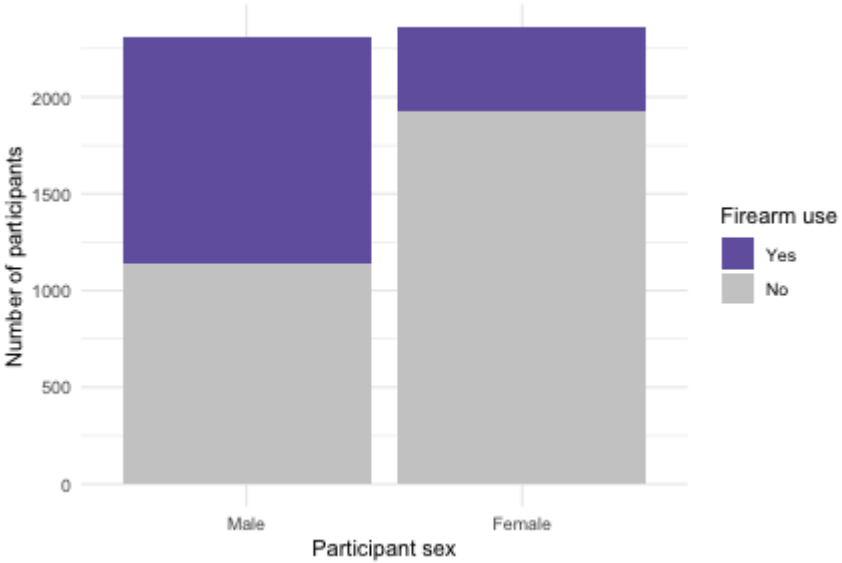
Examining the geom_bar plot



- The plot shows boxes representing the proportion of males and females who have ever used a gun or not used a gun.
- Like the mosaic plot, there are a few things to fix to make it more clearly convey the difference in gun use between males and females.
    - remove the `NA` values from the bars
    - fix the titles
    - use the minimal theme

Formatting the geom_bar plot

```
# formatted stacked bar graph
stack.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = sex, fill = gun.use)) +
  geom_bar() +
  theme_minimal() +
  labs(x = "Participant sex", y = "Number of participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    name = "Firearm use")
stack.gun.use.sex
```

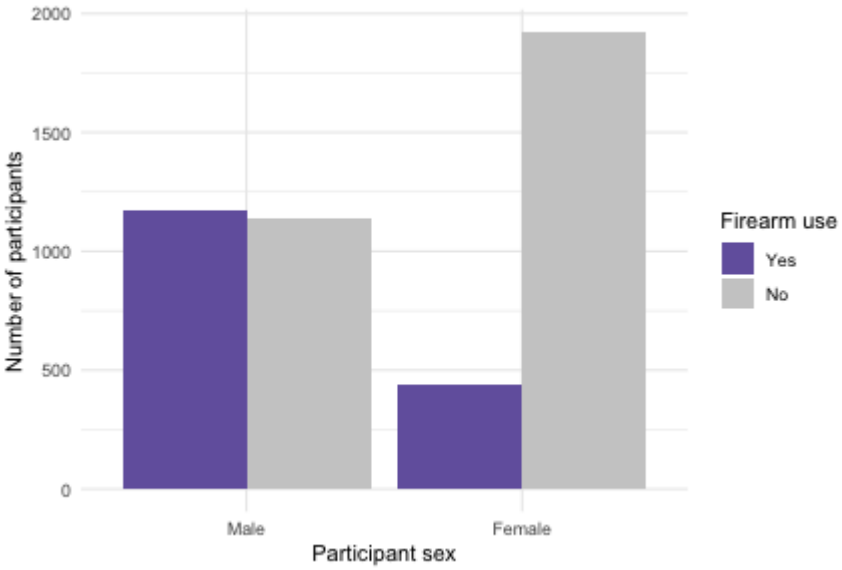


Change the geom_bar plot to a grouped plot

- The `position =` option for the `geom_bar()` layer is the place to specify whether the bars should be stacked or grouped.
- The default is stacked, so to get grouped add `position = "dodge"` to the `geom_bar()` layer.

```
# formatted grouped bar graph
group.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = sex, fill = gun.use)) +
  geom_bar(position = "dodge") +
  theme_minimal() +
  labs(x = "Participant sex", y = "Number of participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    name = "Firearm use")
group.gun.use.sex
```

Examining the grouped plot

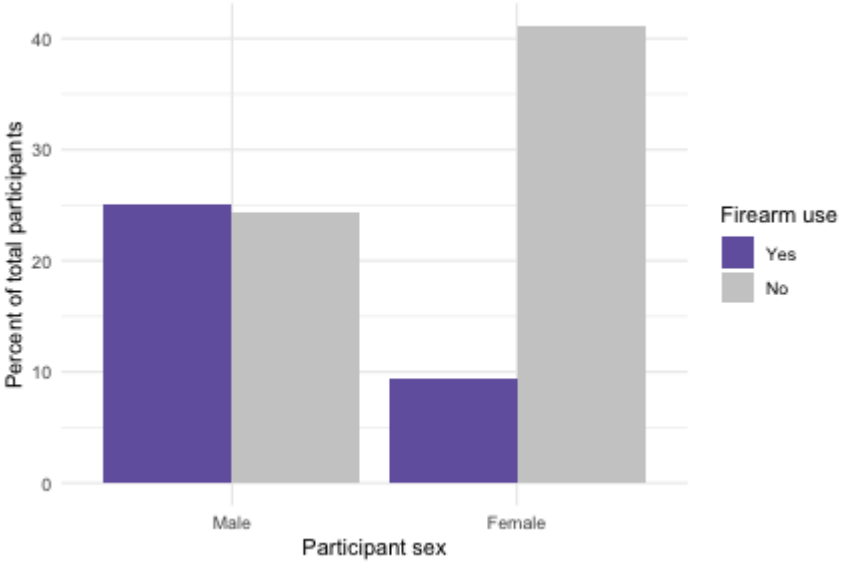


Using percentages rather than frequencies

- Sometimes percentages are more useful than frequencies for a bar graph.
- To change to percentages, use *special variables* to add a percent calculation to the y-axis in the `ggplot()`

```
# formatted grouped bar graph with percents
group.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  ggplot(aes(x = sex, fill = gun.use,
             y = 100*(..count..)/sum(..count..))) +
  geom_bar(position = "dodge") +
  theme_minimal() +
  labs(x = "Participant sex", y = "Percent of total participants") +
  scale_fill_manual(values = c("#7463AC", "gray80"),
                    name = "Firearm use")
group.gun.use.sex
```

Examining the percentages graph & fixing it

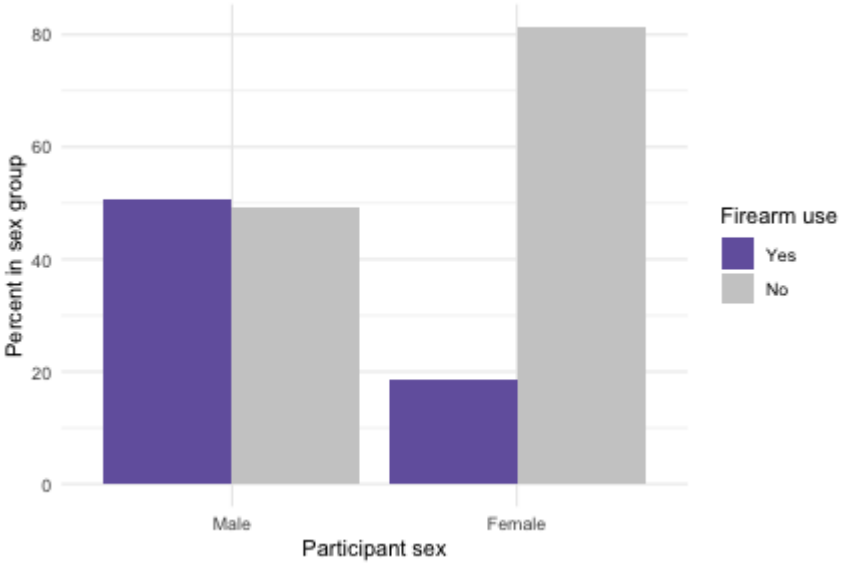


- Note that all the bars together added up to 100%.
- This isn't quite right for comparing males to females since there could be more males than females overall or vice versa.
- Instead try chaging the percentages so that they add up to 100% *within each group* using some additional tidyverse code.

Computing percents for a bar plot with geom_col

```
# formatted grouped bar graph with percents
group.gun.use.sex <- nhanes.2012.clean %>%
  drop_na(gun.use) %>%
  group_by(gun.use, sex) %>%                          # make groups of gu
  count() %>%                                         # count how many ar
  group_by(sex) %>%                                   # pick the variable
  mutate(percent = 100*(n/sum(n))) %>%                # compute percents
  ggplot(aes(x = sex, fill = gun.use,
             y = percent)) +                          # use new values fr
  geom_col(position = "dodge") +
  theme_minimal() +
  labs(x = "Participant sex",
       y = "Percent in sex group") +
  scale_fill_manual(values = c("#7463AC",
                               "gray80"),
                    name = "Firearm use")
group.gun.use.sex
```

Examine the new bar plot



Examining all the plot options

```
# plot all three options together
gridExtra::grid.arrange(mosaic.gun.use.sex,
                        stack.gun.use.sex,
                        group.gun.use.sex,
                        nrow = 2)
```