Data Visualization

Objectives:

1. Make comparison plots with `ggplot`.

Download our data from Google Sheets

Open the group Google Sheet. Select File -> Download -> Comma Separated Values (.csv). (R plays nice with CSV files, but not always with Excel files.) Copy the data you downloaded into your `data` folder for this project.

Open the data in R

Create a new Quarto file.

Read the data into `R` with `read.csv`.

```
county_data <- read.csv("data/Case study data_10-30-23.csv")
```

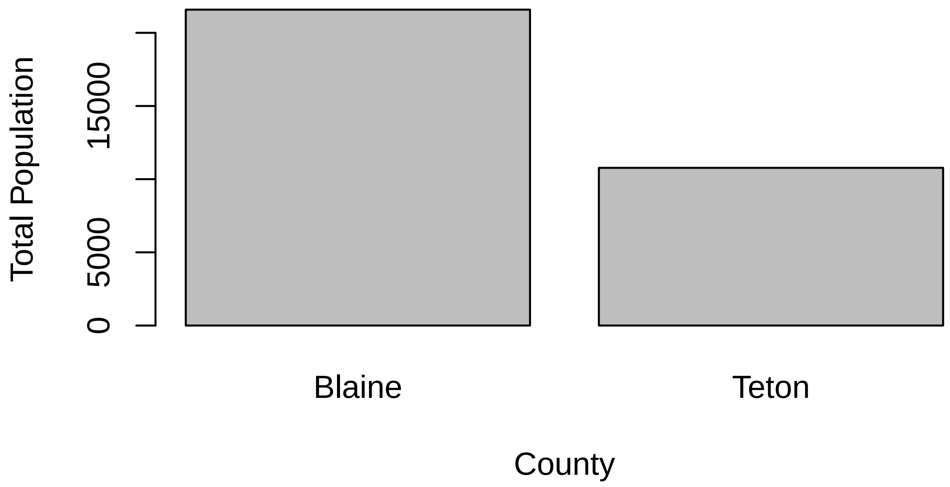
Making Figures

I'll show you one example of a simple figure and an advanced figure. You can decide what kinds of figures (simple or difficult, what kind of summary you want to show) and I will work with you to make it happen!

If you want to work with me to create the figure(s) you want but don't feel ready to code, make a drawing on a piece of paper of what you want to show and we will make the figure together.

There are two different ways to make bar graphs. In these two figures, I will compare the population of the two counties.

```
county_pop_plot_data <- county_data[, c("County", "Total_population")]
barplot(height=county_pop_plot_data$Total_population, #height of bars
        names.arg=county_pop_plot_data$County, #names of bars
        xlab="County", #x-axis label
        ylab="Total Population") #y-axis label
```



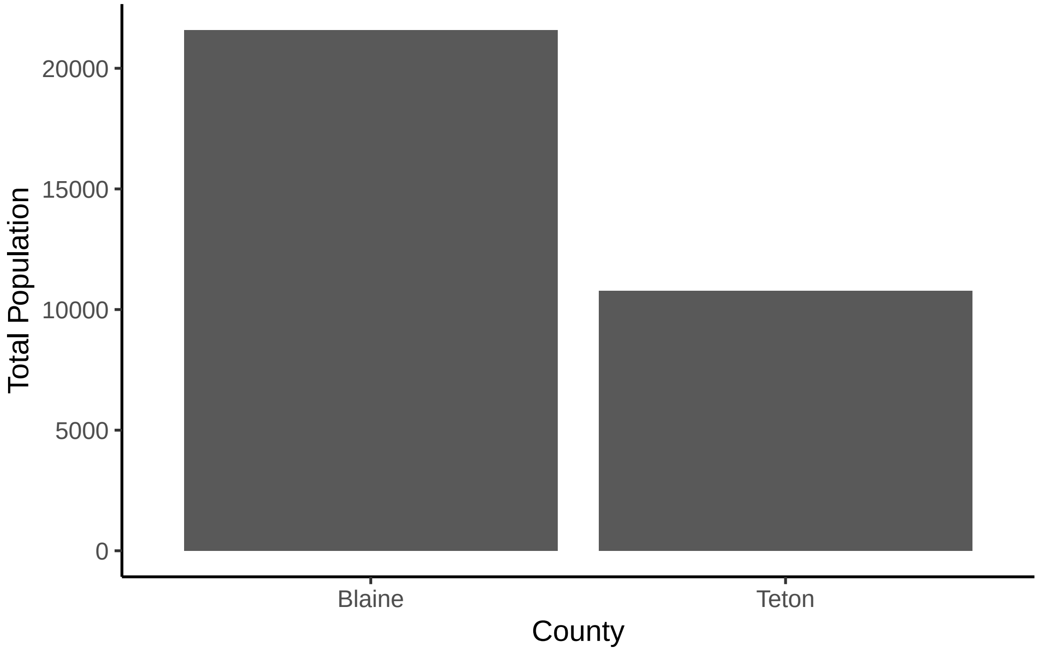
I prefer the `ggplot2` library, a standard figure making library that many scientists use. If you'd like a complete tutorial on this package, you can follow the tutorial at https://datacarpentry.github.io/R-ecology-lesson/visualizing-ggplot.html. There are a lot of plotting functions that we can't use with our data, so this tutorial may be useful in your future work.

Install the `ggplot2` library and read it into `R`.

```
install.packages("ggplot2")
```

```
library(ggplot2)
```

```
ggplot(aes(x=County, # column on the x-axis
           y=Total_population), # column on the y axis
       data=county_data) + # data source
  geom_col() + # add columns/bars
  labs(x="County", # x-axis label
       y="Total Population") + # y-axis label
  theme_classic() # add a nice looking theme
```



To make other plots, we have to format the data in the way `ggplot` likes. With data in the same category (like farm size), we need to make the column names into row labels. Here's an example. (You may need to `install.packages` the packages `dplyr` and/or `tidyr`).

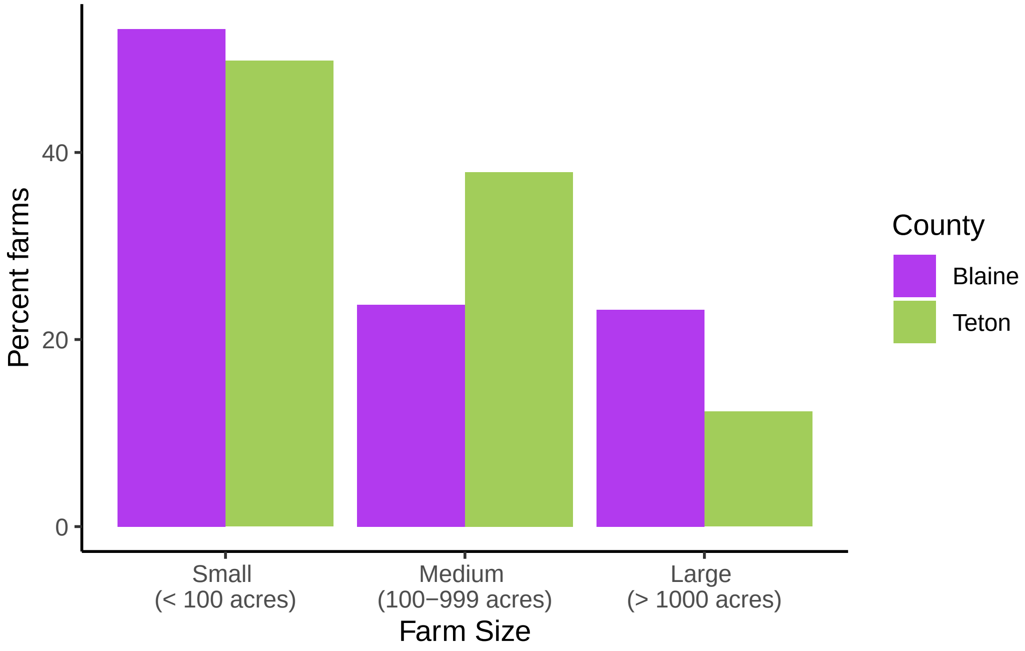
```
library(dplyr)
library(tidyr)

farm_size_plot_data <- county_data %>%
  # select columns
  select(County, Percent_farms_large, Percent_farms_mid, Percent_farms_small) %>%
  # pivot columns to rows
             # choose column range with a colon
             # from Percent_farms_large to Percent_farms_small
  pivot_longer(cols = Percent_farms_large:Percent_farms_small,
             # column name where the column labels go
             names_to = "Farm_size",
             # column name where the values stay
             values_to = "Percent_farms")
```

```
# print data
farm_size_plot_data
```

```
# A tibble: 6 x 3
  County Farm_size           Percent_farms
  <chr>  <chr>                       <dbl>
1 Blaine Percent_farms_large          23.2
2 Blaine Percent_farms_mid            23.7
3 Blaine Percent_farms_small          53.2
4 Teton  Percent_farms_large          12.3
5 Teton  Percent_farms_mid            37.9
6 Teton  Percent_farms_small          49.8
```

Now we can make the figure. Try changing some of the arguments in the "customize" code sections to see what changes. I always have fun playing with colors (a full list of colors in `R` is available at https://r-charts.com/colors/). All the "customize" sections are optional and can be deleted.

```
# make figure
        # specify the columns that match the x, y, and color fill
ggplot(aes(x = Farm_size, y = Percent_farms, fill = County),
       # specify the data object
       data = farm_size_plot_data) +
  # create bars that are next to each other
  geom_col(position = "dodge") +
                   # customize the x-axis order
  scale_x_discrete(limits = c("Percent_farms_small",
                              "Percent_farms_mid",
                              "Percent_farms_large"),
                   # customize the x-axis labels
                   # \n creates a new line
                   labels = c("Small \n(< 100 acres)",
                              "Medium \n(100-999 acres)",
                              "Large \n(> 1000 acres)")) +
  # customize the fill colors
  scale_fill_manual(values = c("darkorchid2", "darkolivegreen3")) +
  # customize the axis names
  labs(x = "Farm Size", y = "Percent farms") +
  # customize with a ggplot theme to make it look pretty
  theme_classic()
```


40
20
0
Percent farms
Small
(< 100 acres)
Medium
(100−999 acres)
Large
(> 1000 acres)
Farm Size
County
Blaine
Teton

Statement of original and referenced work:

The entirety of this module is original work authored by Carolyn Koehn.

License