Post 2: How to Create a Diverging Bar Chart using ggplot2

Catherine Li

November 5, 2017

Diverging Bar Chart & Data Preparation

Introduction

In class, Professor Gaston Sanchez introduced the concept of ggplot2 as follows:

> The package "ggplot2" is probably the most popular package in R to create beautiful graphics. In contrast to the functions in the base package "graphics", the package "ggplot2" follows a somewhat different philosophy, and it tries to be more consistent and modular as possible.

Motivation: I decided to write Post 2 on diverging bar charts because I wanted to explore the concept of ggplot2 in more depth. While we talked about the main ggplot2 visualizations in class, I wanted to learn about other kinds of visualizations that could be useful in representing various types of data.

Audience: This post is intended for people who are interested in learning more about the ggplot2 package in R. In lecture, we learned the basic uses and functions of ggplot2, including how to create scatterplots, density plots, histograms, barcharts, and regression lines. In my post, I will provide a comprehensive guide to other useful applications of functions within ggplot2 through various examples.

Table of Contents:
1. Basics of ggplot
2. Diverging Bar Chart
3. Other Applications
4. Conclusion
5. References

Basics of ggplot

Let's begin by covering the basics of ggplot2.

- The main function of ggplot2 is `ggplot()`.
  - The argument must be a data frame.
  - The plot is controlled by auxiliary functions.
- These auxiliary functions include:
  - Specify the columns of the data frame that will be used for the graphical elements of the plot: `aes()`
  - Specify the kinds of geometric objects, called geoms, that will be displayed on the plot: e.g. `geom_boxplot()`, `geom_point()`, `geom_bar()`
- To apply each of these functions, use the `+` operator. This action is also called "adding layers".

Install ggplot2

To use the ggplot2 package, you must first install it.

```
# Install ggplot2 (Code is in the line below this. It is commented out because you only need to install the packag
e one time on your computer.)
# install.packages("ggplot2")

# Load ggplot2 package
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

You should load any packages (using the library() function) you plan to use at the top of your script.

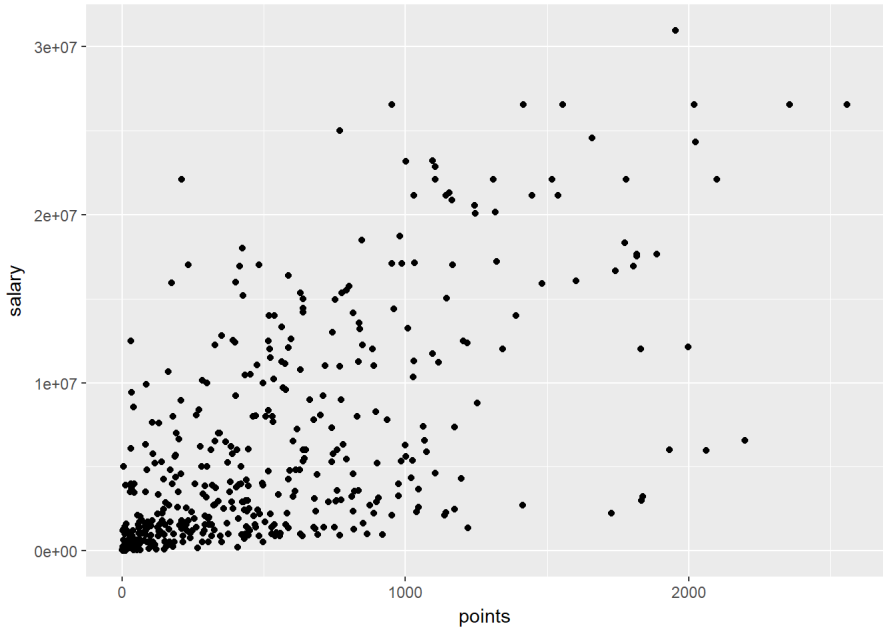
Example: Scatterplot

Here is an example of how to create a scatterplot using data located in the Stat 133 github repository.

```
# download RData file into your working directory
github <- "https://github.com/ucb-stat133/stat133-fall-2017/raw/master/"
csv <- "data/nba2017-players.csv"
download.file(url = paste0(github, csv), destfile = 'nba2017-players.csv')

# with "base" read.csv()
dat <- read.csv('nba2017-players.csv', stringsAsFactors = FALSE)

# Create a scatterplot
ggplot(data = dat, aes(x = points, y = salary)) +
  geom_point()
```



Now I'm going to move onto graphs that have not been introduced in class.

Diverging Bar Chart

Bar plots effectively display categorical data. Diverging bar plots conveniently compares two categories of values and displays their variations adjacent to each other. The main feature of the chart that allows it to do this is an x axis that ranges from negative to positive values. To distinguish the bars on the negative side from the bars on the positive side, they are set to different colors.
Technically, the values for the bars running to the left are not necessarily negative. They could be positive, just like the values to the right of the center vertical axis. It depends on the kind of data that you would like to compare.

Introducing `geom_bar()`

Key: To get diverging bars instead of one set of bars, your categorical variable must contain 2 categorical variables that changes values at a certain threshold.
The main auxiliary function we will be using is `geom_bar()`. `geom_bar()` can be used to create a bar chart or histogram, but in this case, we will be using it to create a diverging bar chart.

Steps

1. Load Data
2. Data Preparation
3. Graph: Diverging Bar Chart

Step 1: Load Data

To demonstrate how to create a diverging bar chart, I will be using an R built-in data set called `presidential`. `presidential` contains the names of each president, the start and end date of their term, and their party of 11 US presidents from Eisenhower to Obama. Load the data:

```
# Load a built-in R data set: data("dataset_name")
data("presidential")

# Inspect the data set: head(dataset_name)
# Print the first 5 rows.
head(presidential, 5)
```

```
## # A tibble: 5 x 4
##         name      start        end      party
##       <chr>     <date>     <date>      <chr>
## 1 Eisenhower 1953-01-20 1961-01-20 Republican
## 2    Kennedy 1961-01-20 1963-11-22 Democratic
## 3    Johnson 1963-11-22 1969-01-20 Democratic
## 4      Nixon 1969-01-20 1974-08-09 Republican
## 5       Ford 1974-08-09 1977-01-20 Republican
```

```
# Number of columns
ncol(presidential)
```

```
## [1] 4
```

```
# Number of rows
nrow(presidential)
```

```
## [1] 11
```

```
# Learn more about presidential using ?presidential
```

Step 2: Data Preparation

Suppose I want to visualize a timeline of the presidents' terms based on the number of years their term is before or after President Jimmy Carter's term (1977 to 1981).

1. To do this, we will need to add a new column to the `presidential` data frame that contains the number of years.

```
# Create vectors with the start and end dates for each of the presidents' terms.
start_year <- substr(presidential$start, 1, 4)
end_year <- substr(presidential$end, 1, 4)

# Check the class of the above vectors.
class(start_year)
```

```
## [1] "character"
```

```
class(end_year)
```

```
## [1] "character"
```

```
# To use mathematical operators on the years, the class of the start_year and end_year vectors must be numeric. Us
e the as.numeric() function to convert the dates to numbers.
start_year <- as.numeric(start_year)
end_year <- as.numeric(end_year)

# Double check that the vectors are now numeric.
class(start_year)
```

```
## [1] "numeric"
```

```
class(end_year)
```

```
## [1] "numeric"
```

```
# Add columns "Start Year" and "End Year" to the presidential data frame.
presidential$'Start Year' <- start_year
presidential$'End Year' <- end_year
```

2. We want to measure the presidents' terms based on the number of years their term is before or after President Jimmy Carter's term. For the presidents that served before Carter's term, we will use their end years. For the presidents that served after Carter's term, we will use their start years.

For example, "Years from Median" for Eisenhower will be 1977(Carter's start year) - 1961(Eisenhower's start year) = 16. "Years from Median" for Obama will be 2009(Obama's start year) - 1981(Carter's end year) = 28.
This will become more clear when you see the code. To do the above, we will use a for loop.

```
years_from_median = c()

for (i in 1:nrow(presidential)) {
  if (i < 6) {              # Presidents whose terms came before Carter's
    years_from_median[i] = presidential$`Start Year`[i] - 1977
  } else if (i == 6) {      # Carter is in the 6th row of data frame presidential. We will leave this row alone.
    years_from_median[i] = 0
  } else if (i > 6) {       # Presidents whose terms came after Carter's
    years_from_median[i] = presidential$`End Year`[i] - 1981
  }
}
```

Note that for the presidents that served before Carter, their years_from_median value will be negative. This is crucial because we will need two categories when we graph.

3. Add vector years_from_median to presidential in a new column named "Years from Median".

```
presidential$`Years from Median` <- years_from_median
```

4. Add a column "Before or After" that states whether a president came before or after Carter.

```
presidential$`Before or After` <- ifelse(presidential$`Years from Median` < 0, "before", "after")
```

5. Sort.

```
presidential <- presidential[order(presidential$`Years from Median`), ]
```

6. The last thing we need to do in data preparation is specific to this data set. In the data frame, there are currently two rows named "Bush", which will lead to an incorrect graphing display later on. We need to change the names of the two presidents.

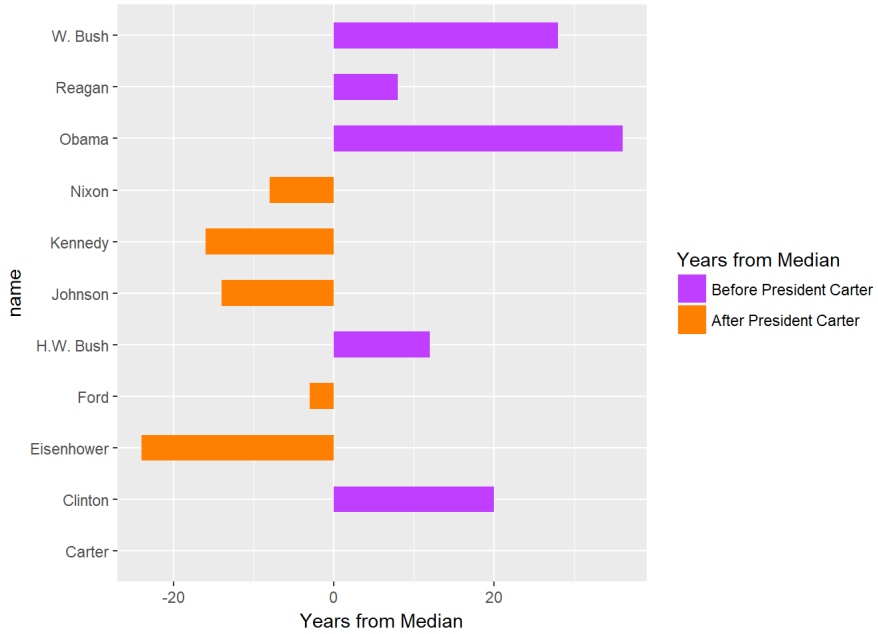
```
# Change "Bush" that appears in an earlier row to "H.W. Bush"
presidential$name[8] <- "H.W. Bush"

# Change "Bush" that appears in a later row to "W. Bush"
presidential$name[10] <- "W. Bush"
```

Step 3: Graph: Diverging Bar Chart

Now that we have all our two categorical variables, we can graph. **I'm going to first show you the final graph and then explain line by line.**

```
ggplot(presidential, aes(x = name, y = `Years from Median`, label = `Years from Median`)) +
    geom_bar(stat = 'identity', aes(fill = `Before or After`), width = 0.5) +
    scale_fill_manual(name = "Years from Median",
                      labels = c("Before President Carter", "After President Carter"),
                      values = c("after" = "#bf3eff", "before" = "#ff7f00")) +
    coord_flip()
```



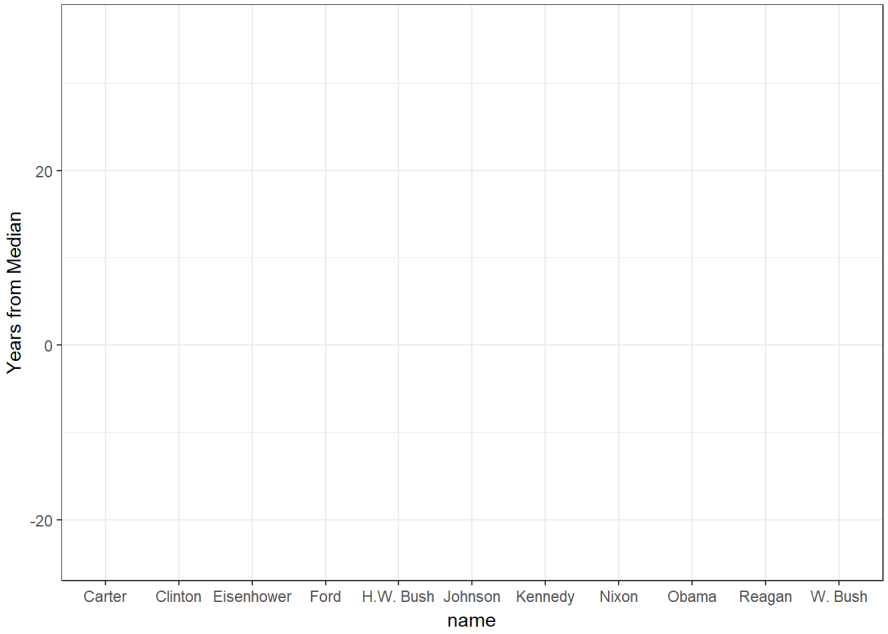

```
# This is optional. This just makes the graph and the text darker in color.
theme_set(theme_bw())
```

Now, let's go line by line.

First Line:

- `ggplot()` takes a data frame and `aes()` as parameters.
- `aes()` used within `ggplot()` takes the parameters, `x =`, `y =`, and `labels =`. The x parameter is either character or factor and the y parameter is numeric.

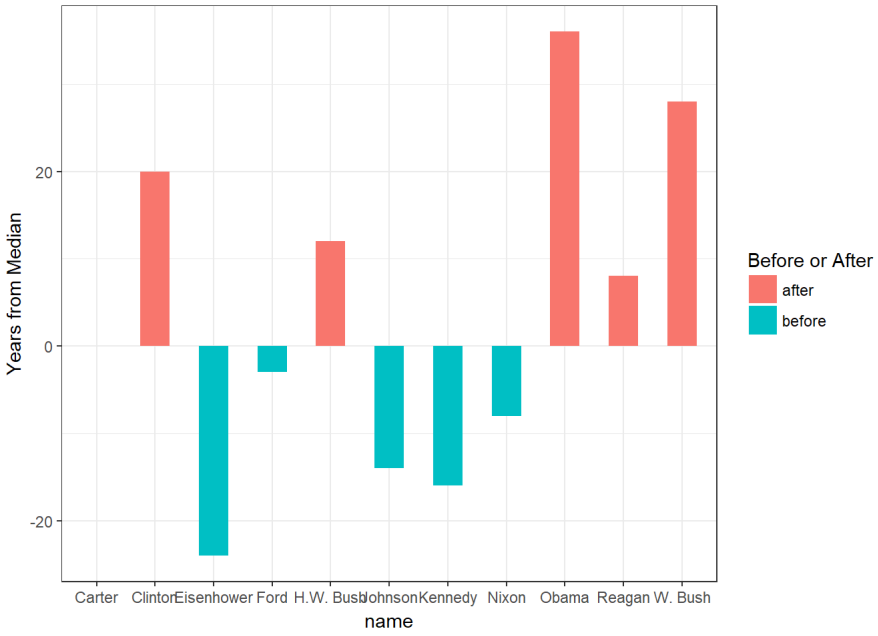
```
ggplot(presidential, aes(x = name, y = `Years from Median`, label = `Years from Median`))
```



Second Line:

- `geom_bar()` takes `stat =`, `aes()`, and `width =` as parameters.
- `stat='identity'` creates bars instead of a histogram.
- `aes()` used within `geom_bar()` takes the `fill=` parameter, which defines how the bars should be filled in. The color of the bars reflect whether or not each president served before or after Carter.
- `width =` defines the width size of each of the bars.

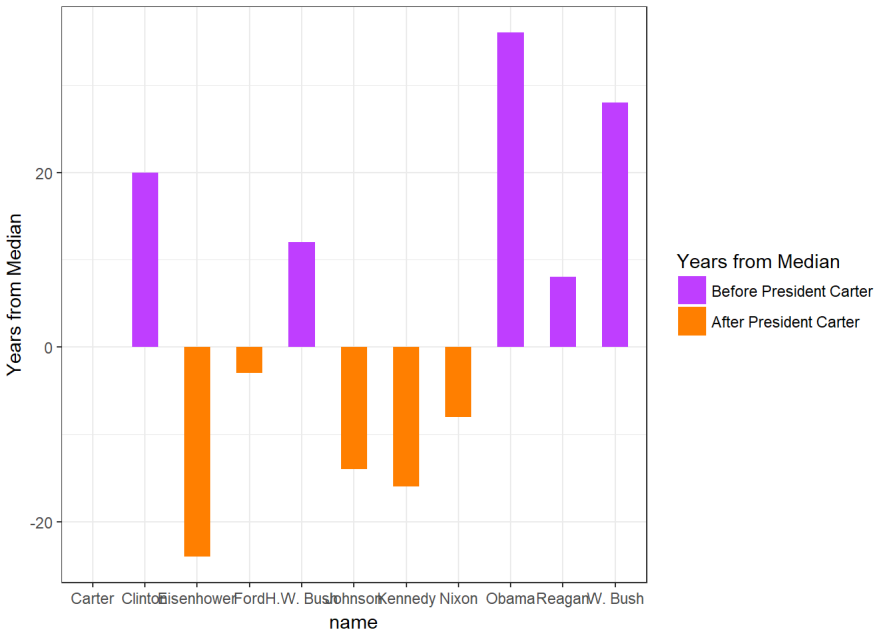
```
ggplot(presidential, aes(x = name, y = `Years from Median`, label = `Years from Median`)) +
    geom_bar(stat = 'identity', aes(fill = `Before or After`), width = 0.5)
```



Third Line:

- `scale_fill_manual()` allows you to create your own discrete scale and specify a set of mappings from the data's levels to aesthetic values.
- `scale_fill_manual()` takes `name =`, `labels =`, and `values =` as parameters.
- `name =` defines the title of the key on the right of the graph. It corresponds to what the colors of the bars are trying to measure. In this case, it is "Years from Median" because we want presidents that came before President Carter to be one color and presidents that came after President Carter to be another color.
- `labels =` labels the two categorical variables. In this case, the two labels are "Before President Carter" and "After President Carter".
- `values =` defines the colors of the categorical values' bars.

```
ggplot(presidential, aes(x = name, y = `Years from Median`, label = `Years from Median`)) +
    geom_bar(stat = 'identity', aes(fill = `Before or After`), width = 0.5) +
    scale_fill_manual(name = "Years from Median",
                      labels = c("Before President Carter", "After President Carter"),
                      values = c("after" = "#bf3eff", "before" = "#ff7f00"))
```

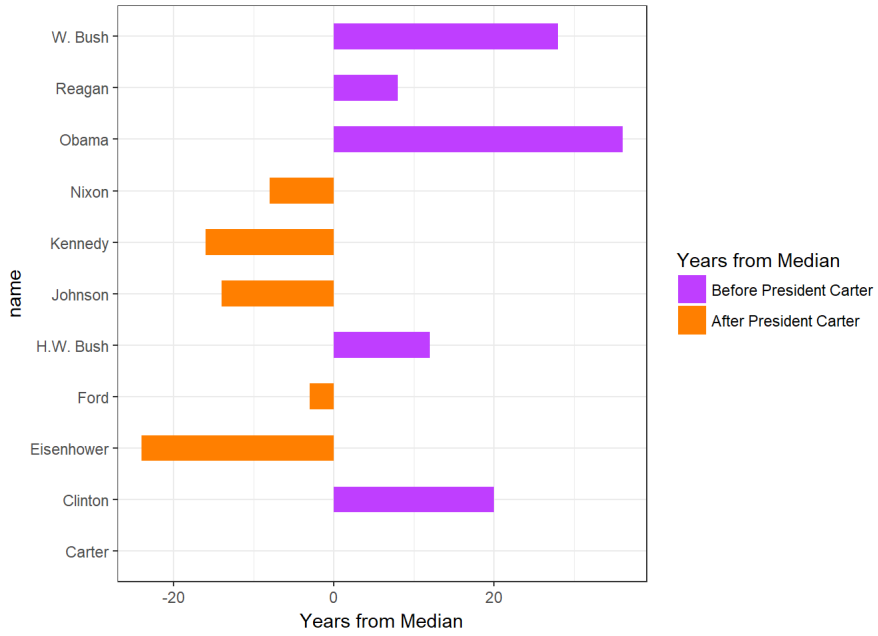


Fourth Line:

- `coord_flip()` rotates the axes, aka turns the bar chart to the side. Instead of vertical bars, the bars are now horizontal. The horizontal layout allows for easy comparison of the bars. As you can see, the names of the presidents are no longer clustered together (like in the

previous plots).

```
ggplot(presidential, aes(x = name, y = `Years from Median`, label = `Years from Median`)) +
    geom_bar(stat = 'identity', aes(fill = `Before or After`), width = 0.5) +
    scale_fill_manual(name = "Years from Median",
                      labels = c("Before President Carter", "After President Carter"),
                      values = c("after" = "#bf3eff", "before" = "#ff7f00")) +
    coord_flip()
```

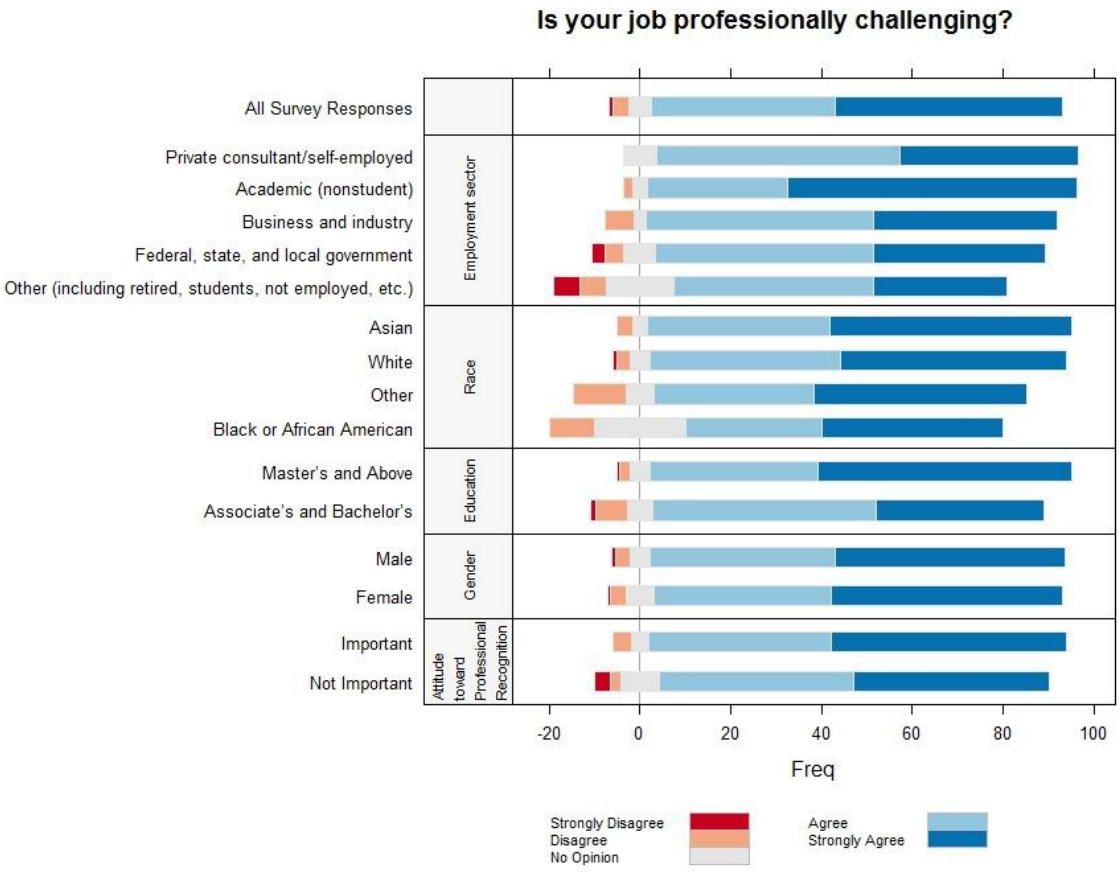


Why diverging bar chart?: You may be wondering why a diverging bar chart would be beneficial in this case as possosed to a regular bar chart. The horizontal diverging bars make it very clear what categorical variables you are trying to measure. If your data set contains positive and negative values, this visualization is ideal because the axis in the middle of the graph usually serves as `x = 0`.

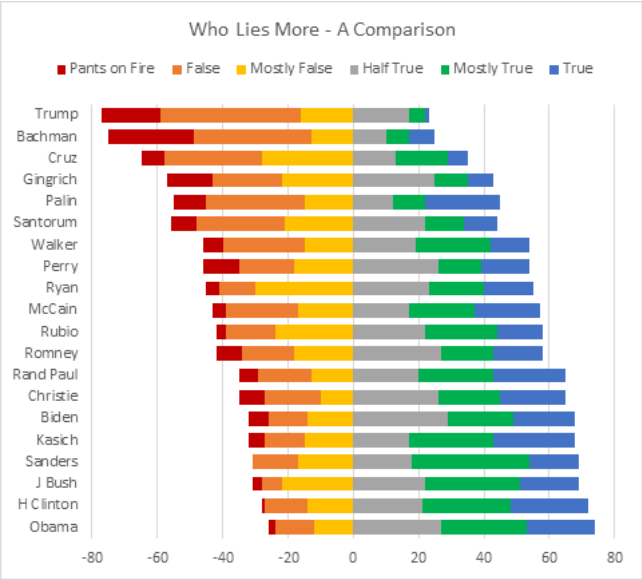
Other Applications

You may be wondering what kind of data sets a diverging bar chart could be useful for. Here are some examples:

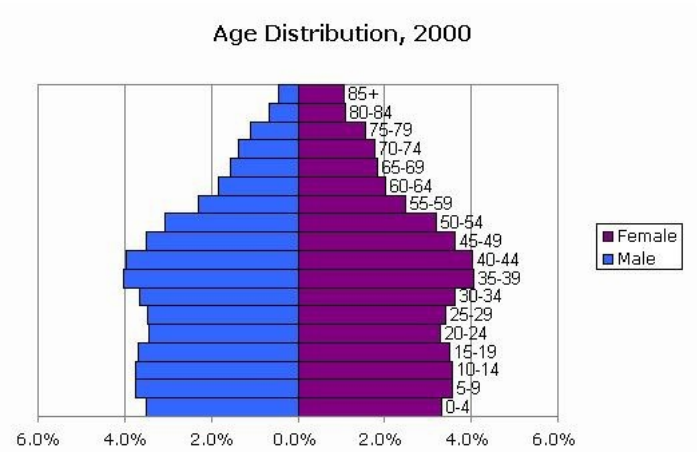
1. The diverging bar chart is typically used to represent data collected in the survey, where there are more than two categories being measured. Each individual bar represents the quantity of people that responded a certain way. This display is called a Lickert scale.



2. A variation of the diverging bar chart is the diverging **stacked bar chart**, where the horizontal bars represent multiple responses rather than just one. For the graph below, the positive responses extend to the right of the vertical zero axis and the negative responses extend to the left of the vertical zero axis. The representation of the public's opinion of the politicians is now easily comparable.



3. Another common use of the diverging stacked bar chart is to display a population's age distribution. This is called a population pyramid. The example below shows the age distribution of the 2000 U.S. population broken down by gender using data from the 2000 U.S. Census.



Conclusion

Take-home message: Here is a crash course to the important information in this post.

- The main function of ggplot2 is `ggplot()`.
- Diverging bar charts:
  - conveniently compare two categories of values and displays their variations adjacent to each other
  - are extremely useful when you want to compare two categorical variables
- Your categorical variable must contain 2 categorical variables that changes values at a certain threshold.
- The main auxiliary function we will be using is `geom_bar()`

References

1. The Complete ggplot2 Tutorial - Part1 | Introduction To ggplot2
2. Top 50 ggplot2 Visualizations - The Master List (With Full R Code)
3. Graphics with ggplot2
4. ggplot2 package
5. Lab 5: First contact with dplyr and ggplot2
6. Lab 6: More dplyr, ggplot2, and files' stuff
7. Diverging Bar Plots
8. Diverging Stacked Bar Charts
9. Plotting Likert Scales
10. Diverging stacked bar charts