Welcome!

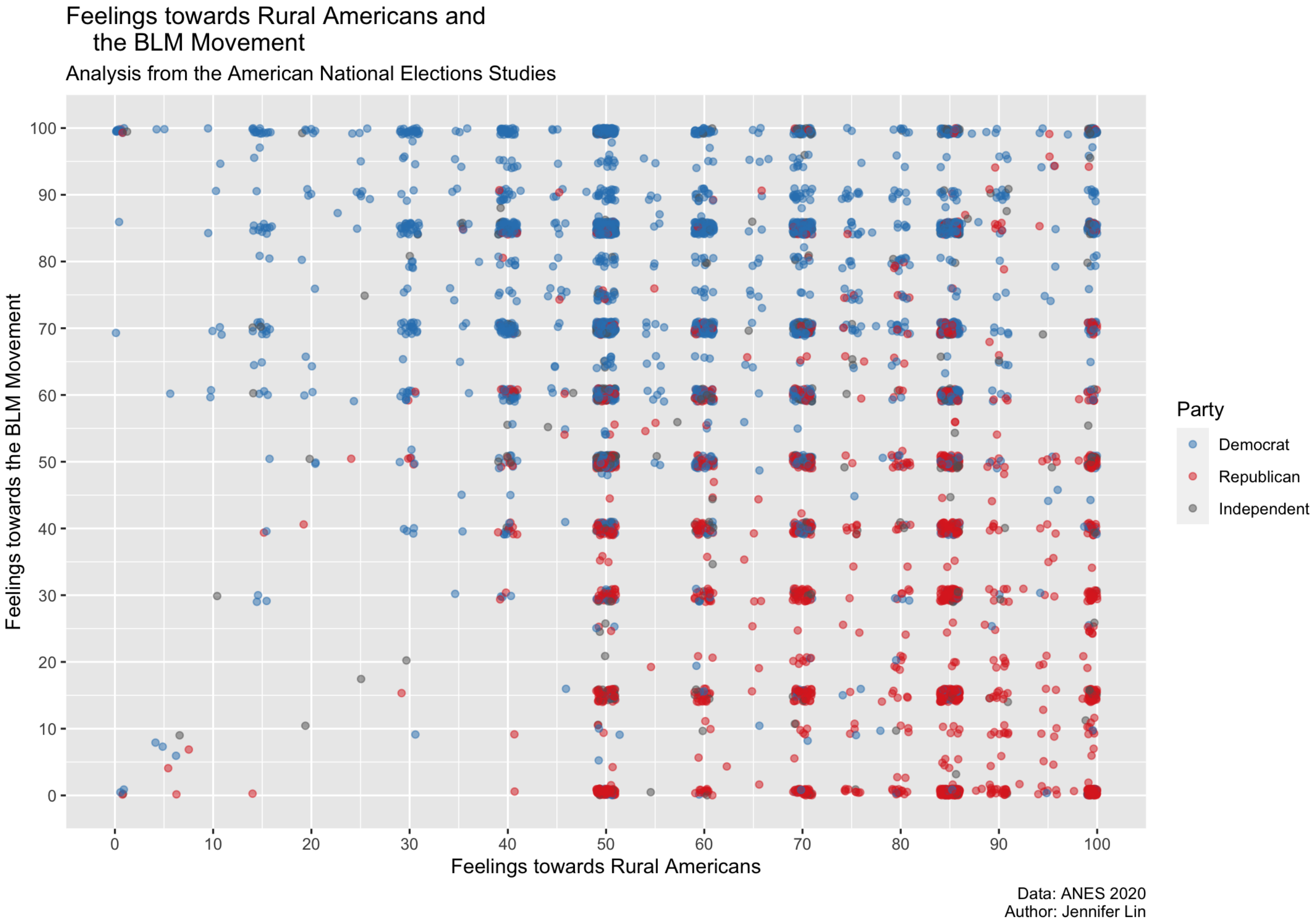
Week 3.3: Themes, Fonts... Ultimate SPICE UP

Jennifer Lin

PS 490: R Workshop

2021-11-20

Last Week


Feelings towards Rural Americans and
the BLM Movement
Analysis from the American National Elections Studies
Feelings towards the BLM Movement
100
90
80
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80
90
100
Feelings towards Rural Americans
Party
Democrat
Republican
Independent
Data: ANES 2020
Author: Jennifer Lin

This Week -- we add Themes!

Your plot from last week now has better labels and colors. But, there is still more to do to make it ready.

1. Add a theme to change out of the gray background
2. Adjust the font sizes and layouts of various graph backgrounds
3. Add any additional features to spice up your graph as needed

Time permitting, I will also show you some tips on how to google and find answers to your own R problems.

The Theme Layer

Overview

There are 2 different types of theme layer settings

1. Global Themes
2. Specific options

We will cover the global themes before looking at specific theme settings

Global Themes

These themes are ones that you can append to the end of your `ggplot` code for different preset looks.

Classic Options

- `theme_bw()`
- `theme_classic()`
- `theme_light()`
- `theme_linedraw()`

from `ggthemes`

- `theme_tufte()`
- `theme_gdocs()`
- `theme_calc()`

Specific Options

On top of these presets, you can customize your graph using arguments in `theme()`

Options in `theme()` are based on *exactly* what you want to customize

An Example

```
theme(
  plot.title      = element_text(hjust  = 0.5),
  plot.subtitle   = element_text(face = "bold")
  axis.title      = element_text(color ="black"),
  axis.title.y    = element_text(size   = 12),
  axis.text.x     = element_text(angle = 45),
  legend.position = 'bottom'
  legend.title    = element_text(family="serif"),
)
```

Where:

- `hjust`: Left (0), Center (0.5), Right (1) justified
- `color`: Color -- can use names or HEX codes
- `size`: Font size
- `face`: Takes "plain", "italic", "bold", "bold.italic"
- `font`: Font family -- assumes sans serif
- `angle`: Angle of object (0 - 360)

Notice the Pattern

To determine what arguments in `theme()` you need, think about the *specific* graph section you want to change.

These follow the pattern:

```
PART OF THE GRAPH + (.) + WHAT ABOUT THIS PART +
(.) + ANYTHING ELSE?
```

PART OF THE GRAPH

Generally, these are the main parts of the graph

- `plot.*`: Addresses the entire plot
- `axis.*`: Addresses the axis
- `legend.*`: Addresses the legends

WHAT ABOUT THIS PART

These components can include things like:

- `title`
- `subtitle`
- `text`
- `caption`
- `background`
- `position`

All that describe the feature that you are seeking to change

ANYTHING ELSE?

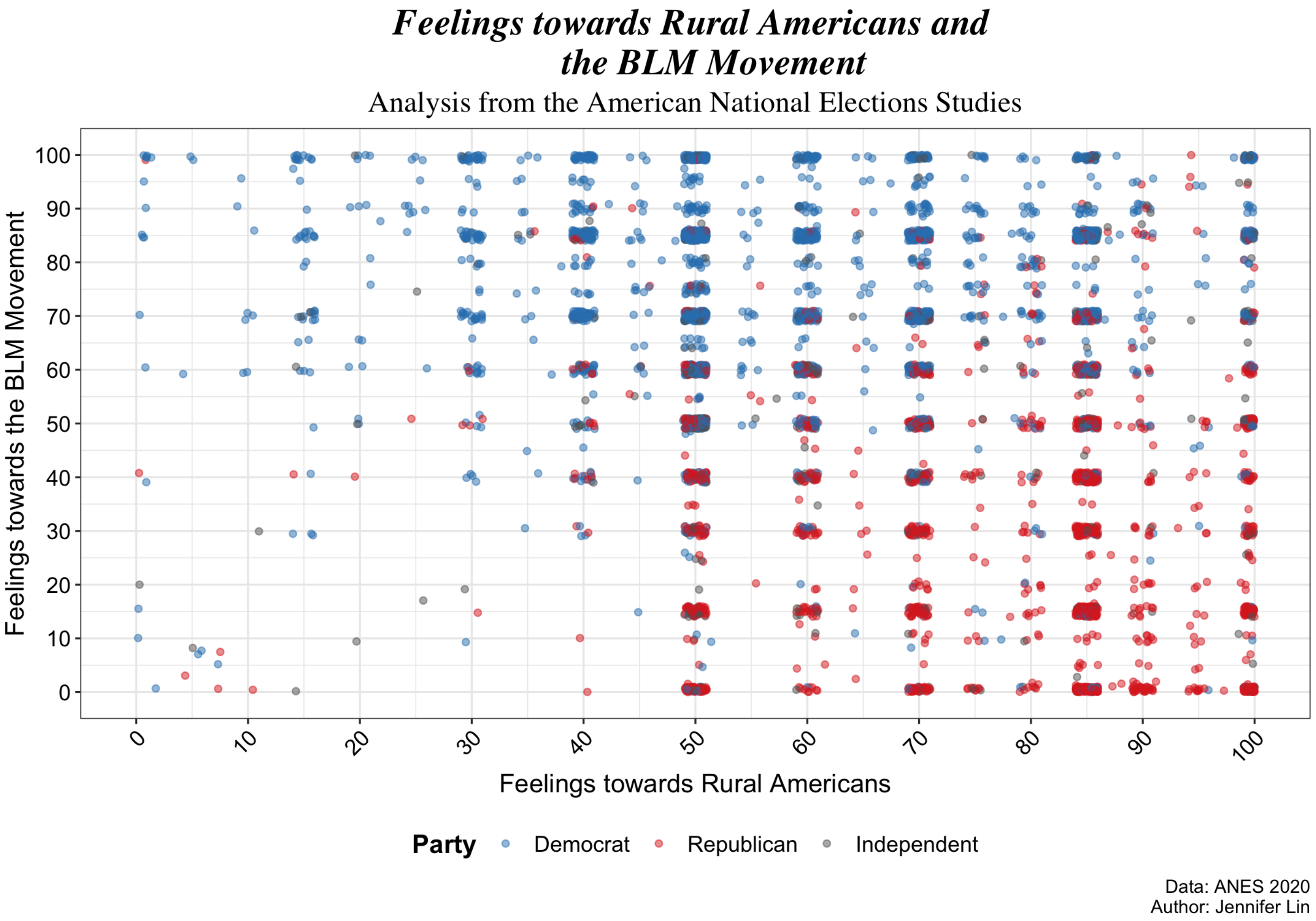
What if you want to make changes to just one axis?

Usually, you can append `.x` or `.y` to the argument.

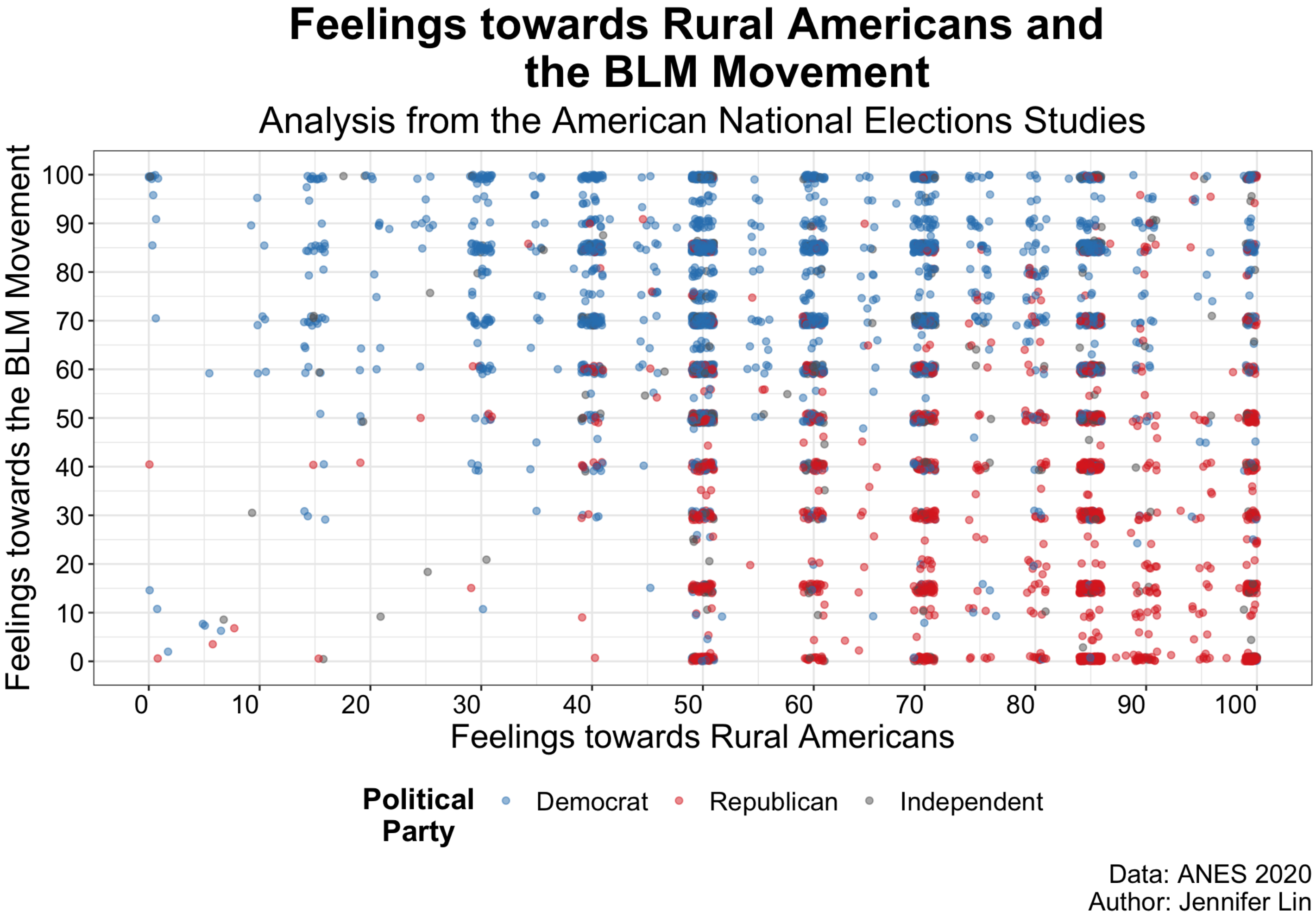
In Practice

```
theme_bw()+
theme(
  plot.title         = element_text(
    hjust = 0.5, size = 20, colour="black",
    face = "bold.italic", family="serif"),
  plot.subtitle      = element_text(
    hjust = 0.5, size = 16,
    colour="black", family="serif"),
  legend.title       = element_text(
    hjust = 0.5, size = 14,
    colour="black", face = "bold"),
  plot.caption       = element_text(size = 10, colour="black"),
  axis.title         = element_text(size = 14, colour="black"),
  axis.text.x        = element_text(
    size = 12, colour="black",
    angle = 45, hjust = 1),
  axis.text.y        = element_text(size = 12, colour="black"),
  legend.position    = 'bottom',
  legend.direction   = "horizontal",
  legend.text        = element_text(size = 12, colour="black")
)
```

Putting it All Together



Publication Quality Graph



Exercise: Adding Themes to your Plots

1. Take your code from last week and add it into this week's script
2. Add a global theme
3. Add specific theme options to adjust your titles, axis labels and legend

Other Components

Some Other Components

- `facet_grid()` and `facet_wrap()` can group your data by a designated grouping variable

- `coord_flip()` changes what is on your x-axis to y-axis and vice versa

Addressing your own R Problems

Think Back to Week 1...

In the first exercise, when you were deciding what graph to make, I asked you to first think about it *without code*.

This was deliberate. Because often we want to do things in R but do not have all the skills for it

Not having the skills should not stop you from making something.

I did not cover everything thee is to know about `ggplot`

There is so much to do with any of the tidyverse packages that a short course does not cover all that you will need

GOOGLE TO THE RESCUE!

Google is often your best friend when it comes to solving R problems, and especially Stack Overflow.

However, the challenge is knowing what to google. (Finding words can be hard.)

Tips for Googling R problems.

1. Draw (on paper) your desired end result and find words around that
2. Take advantage of related searches
3. Use the "Ask Question" feature and have Stack Overflow AI help you
4. Post your own Question on Stack Overflow.

Exercise: Stack Overflow

Here are some common R challenges that I have used stack overflow for. Pick one, do a stack overflow search and apply it to your graph

1. Make the legend 2 column
2. Make the legend run vertical/horizontal
3. Add space between the axis text and the plot
4. Or ANYTHING that you think can help make your graph look nicer

Your Submission to the Lab Assignment for this week

1. A finalized, publication ready PDF version of your graph and the corresponding script file
2. Your stack overflow search and the answer that you used -- you can write this as part of your script file in the form of a comment. State your question and insert the URL for the thread.
3. In a few words, discuss how you added your stack overflow findings to your plot.