# *Week 3.2: Colors and Labels*

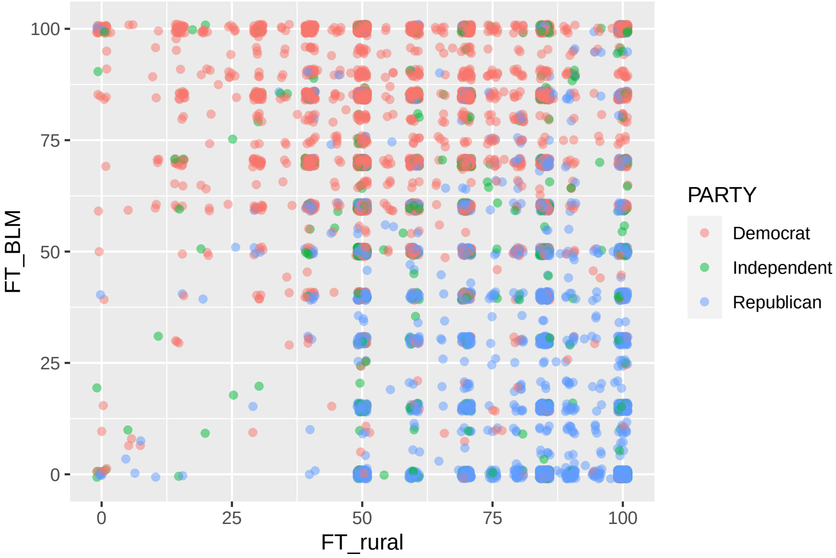
*Jennifer Lin*

*2021-11-10*

So your graph from last week was basic. It shows the general trends of your data but might do so in inefficient ways. Some of these include:

- The labels on the axes do not communicate something meaningful
- The legend is poorly labeled
- R's default colors might not be the most beautiful, especially, if you are coloring/filling by a particular variable
- The default scales might not be as informative

So we will fix those problems today.

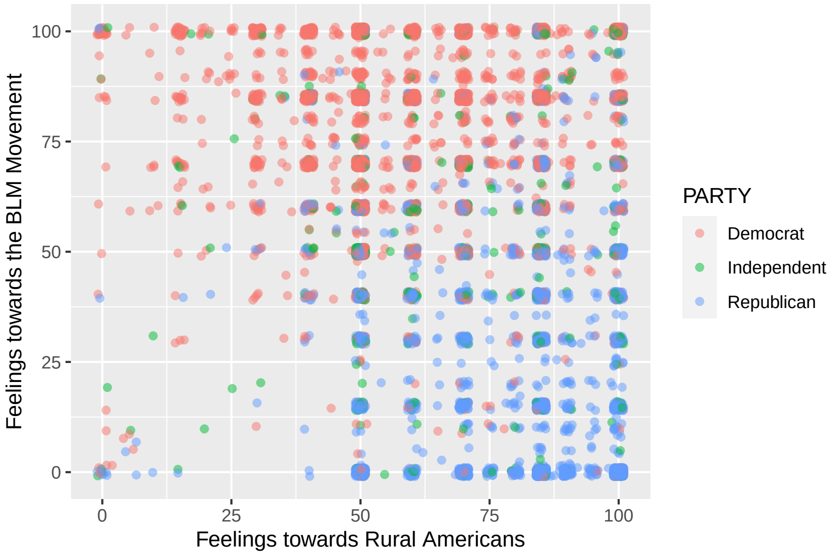


## *The Label Layer*

The label layer allows us to add and change exiting labels for titles, scales and legends.

To change the x and y axis labels, we use `xlab()` and `ylab()` respectively. Within the parentheses, we add the new labels that we want in quotes ("") as such:

```
ANES %>%
  filter(!is.na(PARTY)) %>%
  ggplot(aes(x = FT_rural, y = FT_BLM, color = PARTY))+
  geom_point(position = position_jitter(1, 1), alpha = .5)+
  xlab("Feelings towards Rural Americans")+
  ylab("Feelings towards the BLM Movement")
```

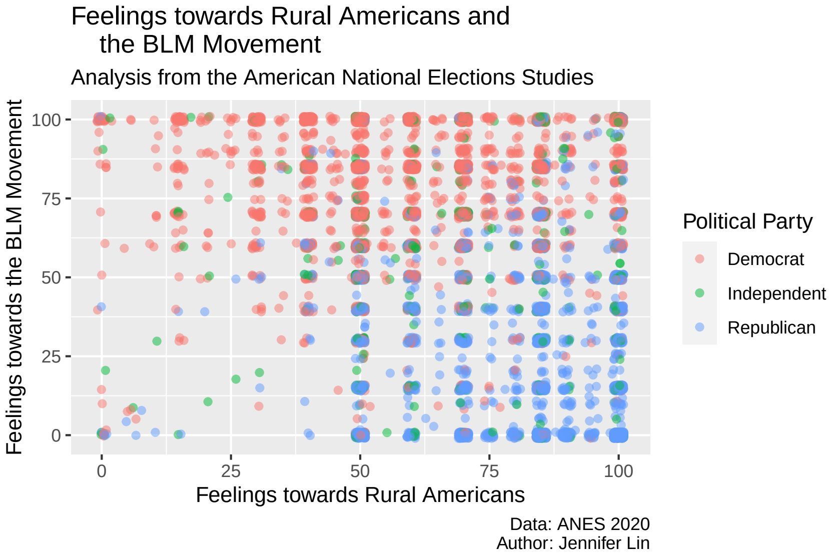


To add a title, subtitle, caption, or legend label to our graph, we can use the options provided by `labs()`. It generally takes the following form:

```
labs(
  color    = "",
  fill     = "",
  title    = "",
  subtitle = "",
  caption  = ""
)
```

- The `color/fill` option labels the legend and it is dependent on whether you used COLOR or FILL in the data layer
- `title` adds a main title to the top of the plot
- `subtitle` adds a subtitle to the top of the plot
- `caption` adds a caption to the bottom of the plot

```
ANES %>%
  filter(!is.na(PARTY)) %>%
  ggplot(aes(x = FT_rural, y = FT_BLM, color = PARTY))+
  geom_point(position = position_jitter(1, 1), alpha = .5)+
  xlab("Feelings towards Rural Americans")+
  ylab("Feelings towards the BLM Movement")+
  labs(
    color = "Political Party",
    title = "Feelings towards Rural Americans and
    the BLM Movement",
    subtitle = "Analysis from the American National Elections Studies",
    caption = "Data: ANES 2020
    Author: Jennifer Lin"
  )
```



A few things to note:

1. Since it is a scatterplot, and I used color in the data layer, my label option for my legend is `color`
2. When you press `ENTER` to break a line in the quotes, it inserts a line break in the outcome.

### *Exercise: Adding labels to your graph*

1. Revisit your code from last class. Copy the results that you have from the data and graph layers onto the code for this week.
2. Using `xlab()`, `ylab()` and `labs()` to add a x axis label, y axis label, title, subtitle, legend label and optional caption.

## *The Scale Layer*

An additional thing that you might want or need to do is adjust the scales. By default, `ggplot` adjusts the scales based on the best fit of the graph from the data presented. However, sometimes, you want to override these settings.

To start, if you want to adjust the limits on continues x or y axes, you can use `xlim()` or `ylim()` with your desired start and end encased in the parentheses.

However, if you want to adjust the scales and legends for continuous and categorical variables, `ggplot`also contains a variety of `scales` functions to help you do just that. These come in the format of `scale_[SOMETHING]_[SOMEHOW]()` and their arguments vary depending on the thing you are scaling and how you are doing it.

Most scales take the following arguments

- `name` = Name the thing you are scaling
- `breaks` = Locate where you want to break it
- `labels` = Assign each break point a name
- `limits` = Set upper and lower bounds (if applicable)

Here are some examples:

- `scale_x_continuous()`
  - `[SOMETHING]` = x-axis
  - `[SOMEHOW]` = continuously
- `scale_fill_manual()`
  - `[SOMETHING]` = shape fill
  - `[SOMEHOW]` = manually
- `scale_colour_brewer(palette = "[PALETTE NAME]")`
  - `[SOMETHING]` = color
  - `[SOMEHOW]` = Using the R Color Brewer palette

Here are some examples for how to apply these scales:

### SCALE COLOR MANUAL

*If you are using a `fill`, this situation is the same thing, just change `color` to `fill`*

So I want to introduce my own colors to the mix since the R default colors are completely trash. I can do that with `scale_color_manual()`

```
scale_color_manual(
  name = "Party",
  breaks = c("Democrat",  "Republican", "Independent"),
  values = c("Democrat" = "#3182bd", "Republican" = "#de2d26", "Independent" = "#636363")
)
```

Here, I am using the `name`, `breaks` and `values` arguments

- `name` changes my legend label name from the `labs()` argument earlier (or from defaults)
- `breaks` set the categories for the legend. NOTICE that your legend will appear in this order
- `values` allows you to set colors manually. You can list the color (“red”, “blue” etc) or use HEX codes (but keep them in quotes!)

### SCALE X CONTINUOUS

Now suppose I do not like the fact that the axis labels of my graph are too far apart, and I want to manipulate the continuous scale so that it shows more breaks.

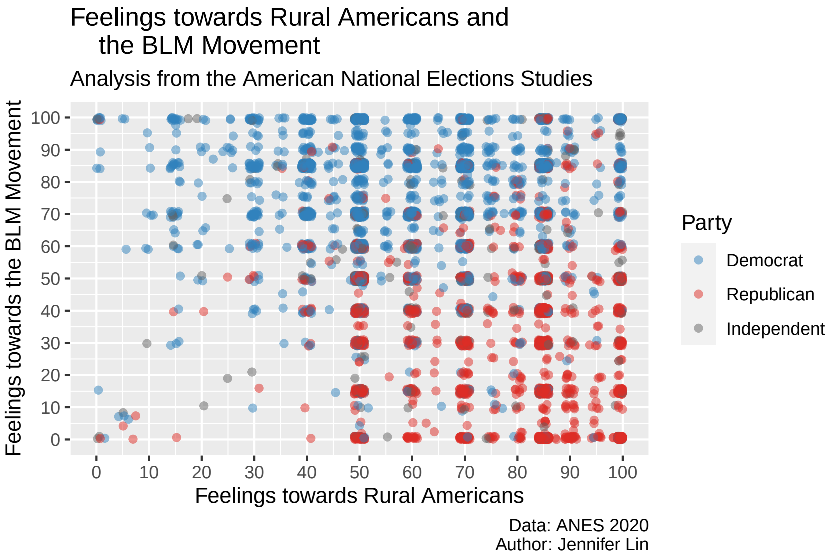
```
  scale_x_continuous(
    breaks = seq(0, 100, 10),
    limits = c(0, 100)
  )+
```

Here, I use the `breaks` and `limits` options

- `breaks` tell R the numbers to include on the axis and how frequently I want to scale to add a new tick for a number. Since I am lazy, I am not writing all 10 numbers. Rather, I use `seq(BEGINNING, END, BY)` to count.
- `limits` tell R to cut the scale off at a certain point. In this case, I am telling it to go for the full scale (0-100)

Here is how everything looks on the graph:

```
ANES %>%
  filter(!is.na(PARTY)) %>%
  ggplot(aes(x = FT_rural, y = FT_BLM, color = PARTY))+
  geom_point(position = position_jitter(1, 1), alpha = .5)+
  xlab("Feelings towards Rural Americans")+
  ylab("Feelings towards the BLM Movement")+
  labs(
    color = "Political Party",
    title = "Feelings towards Rural Americans and
    the BLM Movement",
    subtitle = "Analysis from the American National Elections Studies",
    caption = "Data: ANES 2020
    Author: Jennifer Lin"
  )+
  scale_color_manual(
    name = "Party",
    breaks = c("Democrat",  "Republican", "Independent"),
    values = c("Democrat" = "#3182bd", "Republican" = "#de2d26", "Independent" = "#636363")
  )+
  scale_x_continuous(
    breaks = seq(0, 100, 10),
    limits = c(0, 100)
  )+
  scale_y_continuous(
    breaks = seq(0, 100, 10),
    limits = c(0, 100)
  )
```



## *Exercise: Changing your scales*

1. Using the graph that you created in the last exercise, change the scales on your continuous variable
2. Change the color of your fills or lines, either by variable, or revisit your graph layer and change it globally
3. CHALLENGE: Go online and find a HEX code generator and add a color that is not an R default color (i.e. your color should include a #).

## *Your Submission to the Lab Assignment for this week*

1. Export the graph as a PDF using the "Export" button on the upper left hand corner of the plot window.
2. Upload your PDF AND the code, with your answers to the questions
3. Don't worry about the background or the font sizes. We will work on that next week.