

Session 3: Graphics in R

Welcome!

TA: Jennifer Lin

MMSS 211: Institutions, Rules, & Models in Social Science

2022-04-11

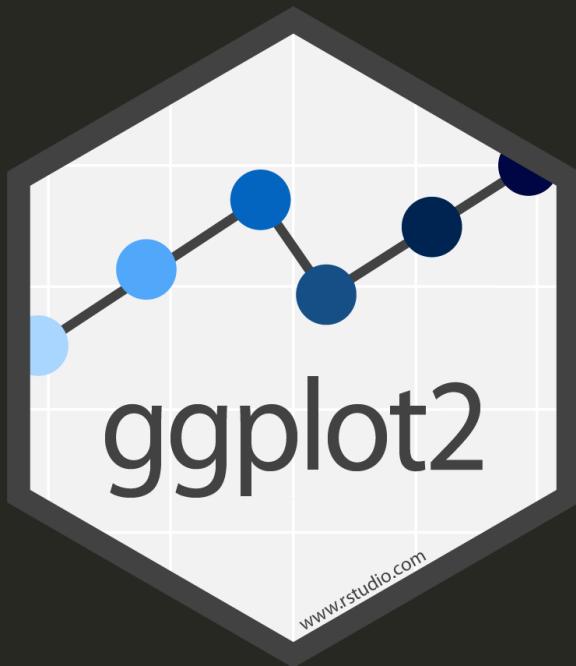
Goals

1. Learn the basics of making graphs in R with `ggplot2`
2. Reinforce fundamental R skills with additional practice on creating vectors and loading datasets
3. Apply the fundamental plotting skills to spatial stat
4. Develop preferences for a publication quality graph

Dimensions of a Good Graph

- One x and one y axis
- Clearly labeled, with titles, axis and legends containing helpful information on graph contents
- Accessible color schemes
- Minimal background, maximal foreground
- Readable fonts and font sizes
- Use graph sections to tell a meaningful story

ggplot2: The Grammar of Graphics



Making a Graph in R is like baking a cake...



... being fully decorated at phase 1 is unreasonable...

... and you need to build it layer by layer

ggplot2 Makes Graphs By LAYERS

Graphs are created by layers such that they can be conceptualized as follows

```
ggplot(data layer)+  
  graph layer +  
  label layer +  
  scale layer +  
  theme layer +  
  others
```

where each layer (roughly) shows what should be there.
Graphs do NOT need to follow a standard order so long as you get your desired outcome

Pipes %>% Meets Their Sibling +

- Components of a ggplot graph are connected with a plus (+) sign.
- You can add as many components in one large ggplot graph chunk as you desire. BUT you need to connect them or R will not interpret your lines as part of the same chunk

The Data Layer

Pew Research Center Data

The American Trends Panel samples a representative group of Americans on their political and religious attitudes.

We load in a cleaned subset from the March 2021 panel.

```
Pew <- read.csv("Pew_March2021.csv")
```

Format for Data Layer

```
ggplot(DATAFRAME,  
       aes(x = X_VAR, y = Y_VAR, fill = FILL_VAR, color = COLOR_VAR))
```

Where

- DATAFRAME = the dataframe that you want to use
- X_VAR = the x-variable
- Y_VAR = the y-variable
- FILL_VAR = the fill variable -- use to FILL graphs that have a space to fill
- COLOR_VAR = the color variable -- use to COLOR lines or points in graphs

NOTE -- you should only include FILL and COLOR arguments IF you want to fill by a variable. If you want a standard color throughout... stay tuned!

Example of Data Layer Input

I am going to demo a plot where I look at the correlation between feelings towards Donald Trump and Joe Biden. Specifically, I want to see if there is a partisan divide between Democrats and Republicans in this interaction.

```
ggplot(Pew,  
       aes(x = FT_Biden, y = FT_Trump, color = pid3)) +
```

The variables that I will be inputting are

- DATAFRAME = Pew (see variable read-in code)
- X_VAR = FT_Biden
- Y_VAR = FT_Trump
- FILL_VAR = DOES NOT APPLY -- This is a scatterplot
- COLOR_VAR = pid3 -- coloring the dots by party

Exercise: Find Variables and Build Data Layer

1. Load the dataset (Code in script)
2. Use `names()` to find the variables and `table()` to explore the variables.
3. Identify one x and one y variable (along with an optional fill or color variable) with which you want to make a graph
4. Without code, determine what kind of graph you want to make
5. Build your data layer accordingly

The Graph Layer

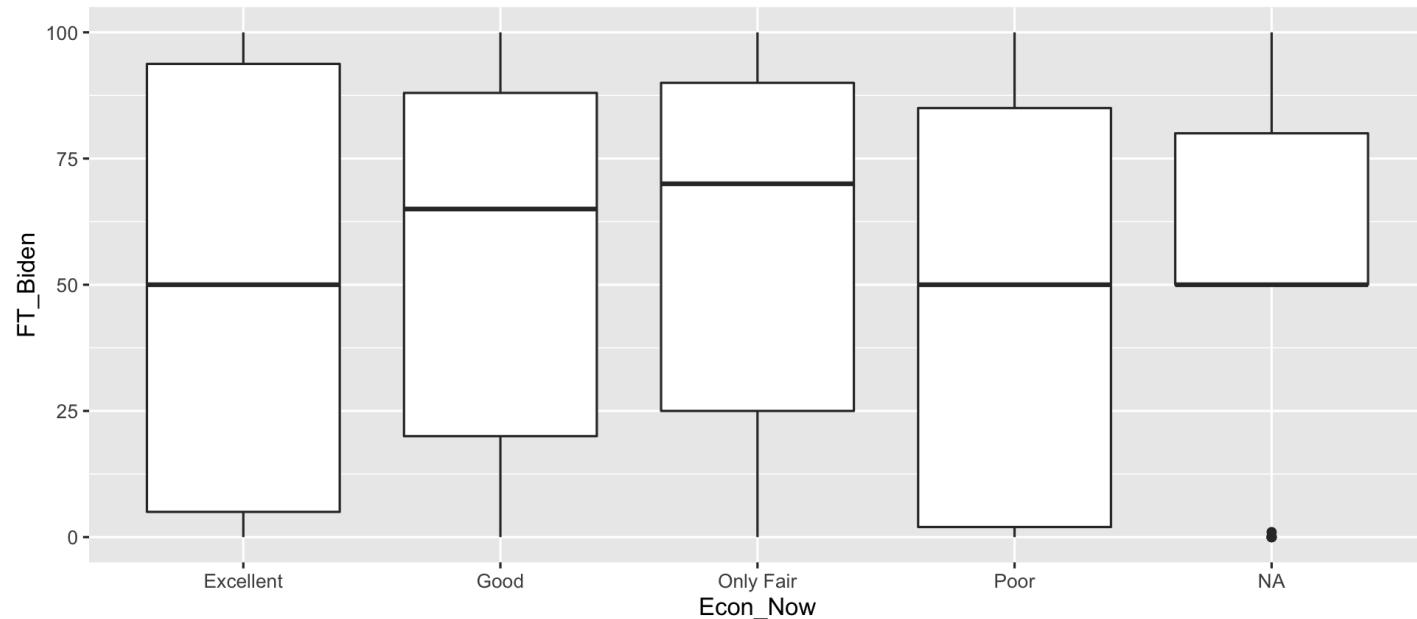
Basic Graphs

Operator	Description
geom_line()	line graph
geom_point()	scatterplot
geom_bar()	bar plot
geom_histogram()	histogram
geom_boxplot()	boxplot
geom_violin()	violin plot
geom_sf()	maps with shapefiles

Notice that it all starts with geom_

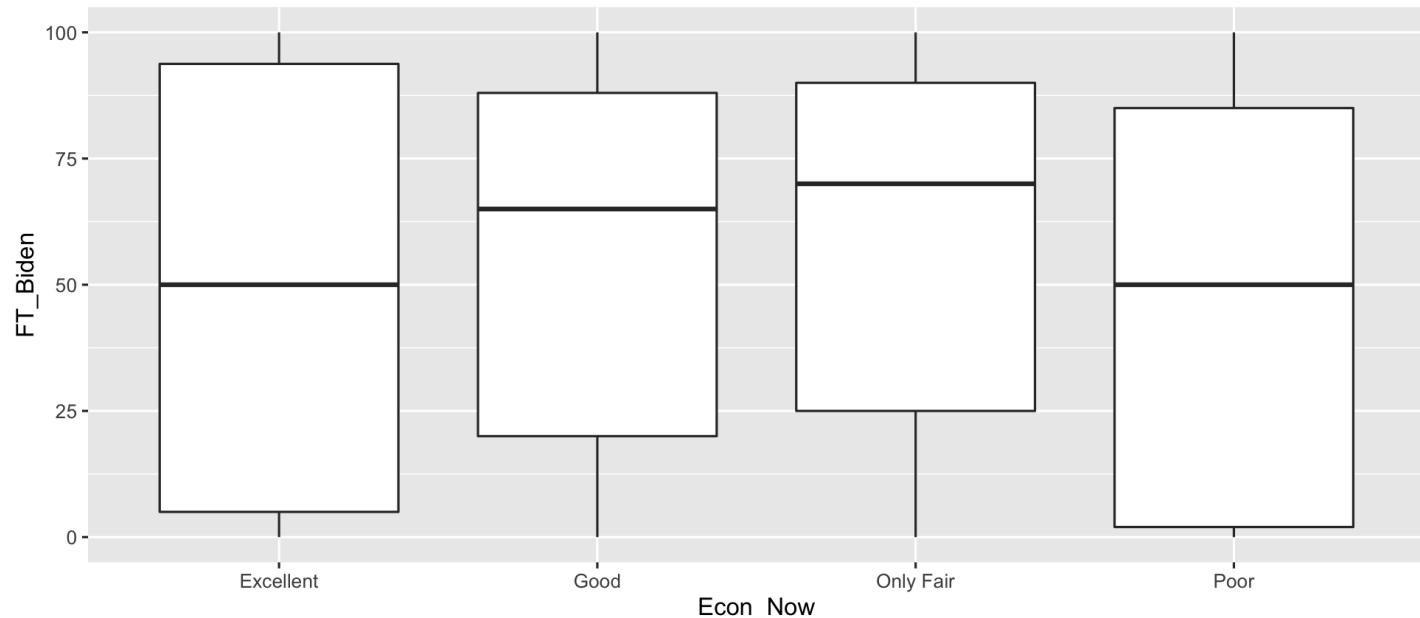
A Basic Box plot

```
ggplot(Pew, aes(x = Econ_Now, y = FT_Biden)) +  
  geom_boxplot()
```



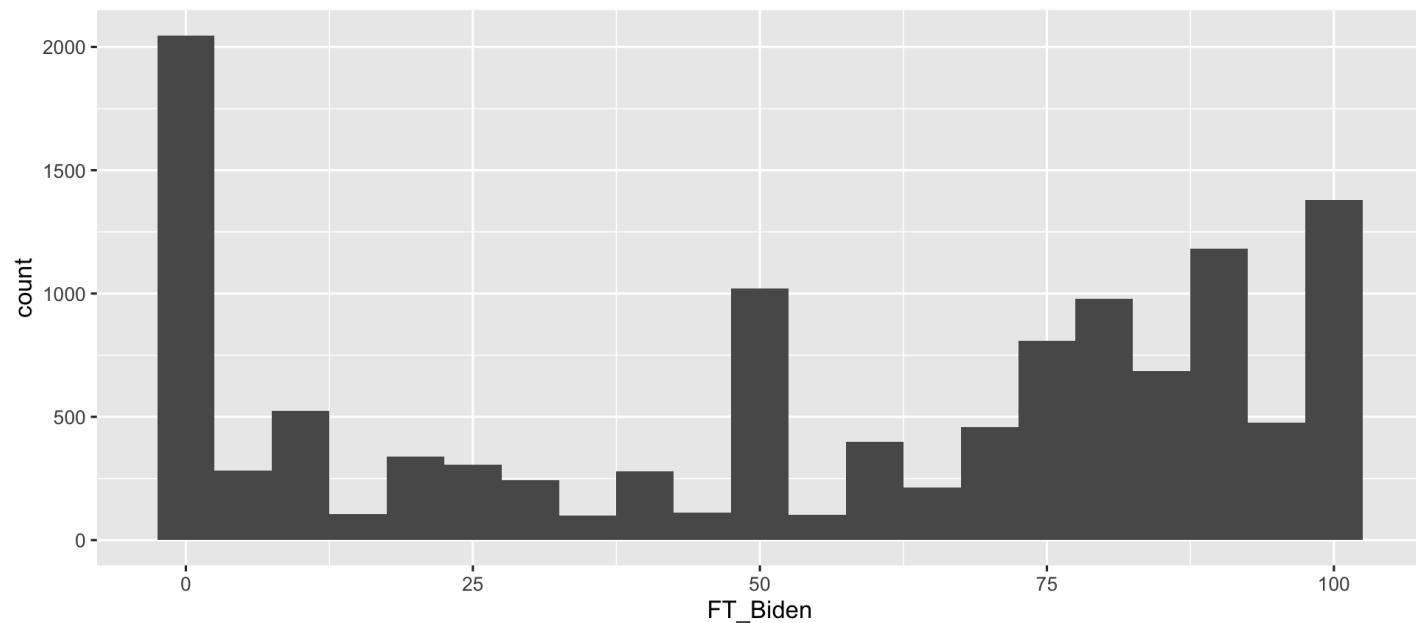
A Basic Box Plot integrating dplyr

```
Pew %>%
  filter(!is.na(Econ_Now)) %>%
  ggplot(aes(x = Econ_Now, y = FT_Biden)) +
  geom_boxplot()
```



A Basic Histogram

```
ggplot(Pew, aes(x = FT_Biden)) +  
  geom_histogram(binwidth = 5)
```



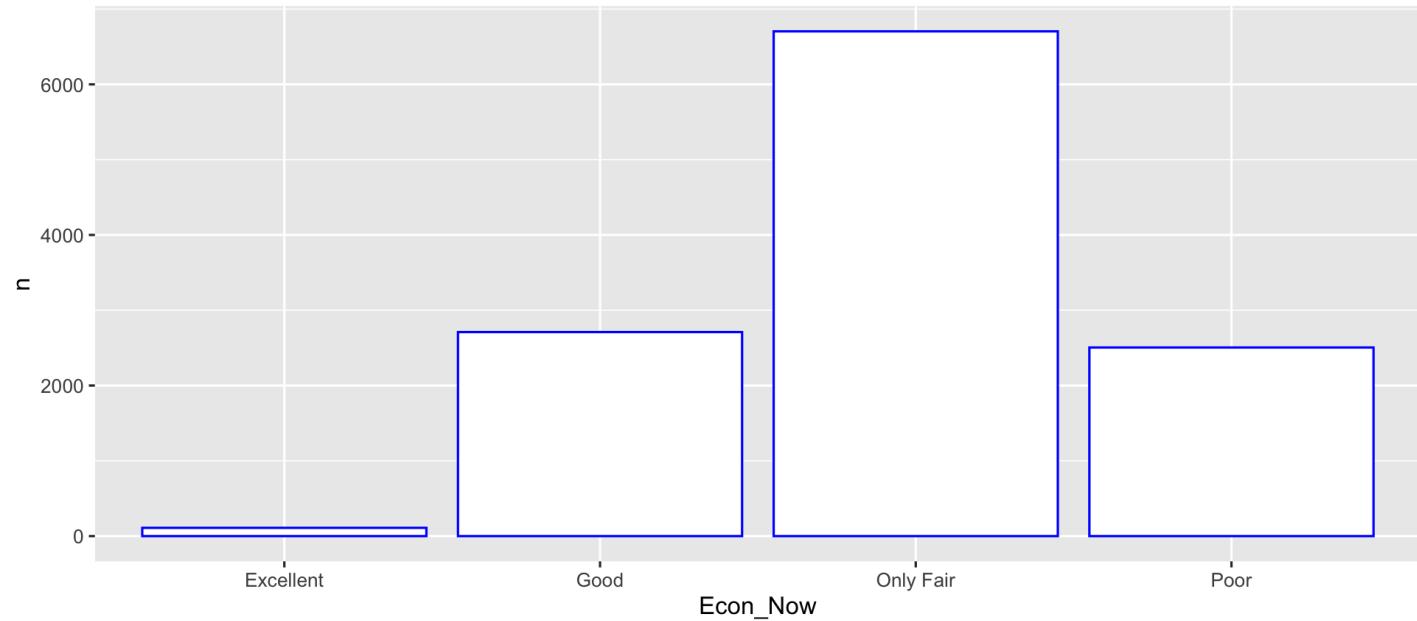
A Basic Bar Graph

First, you need to prep the data using dplyr.

```
Econ_Now <- Pew %>%
  group_by(Econ_Now) %>%
  summarise(
    n = n(),
    .groups = 'keep'
  ) %>%
  filter(!is.na(Econ_Now))
```

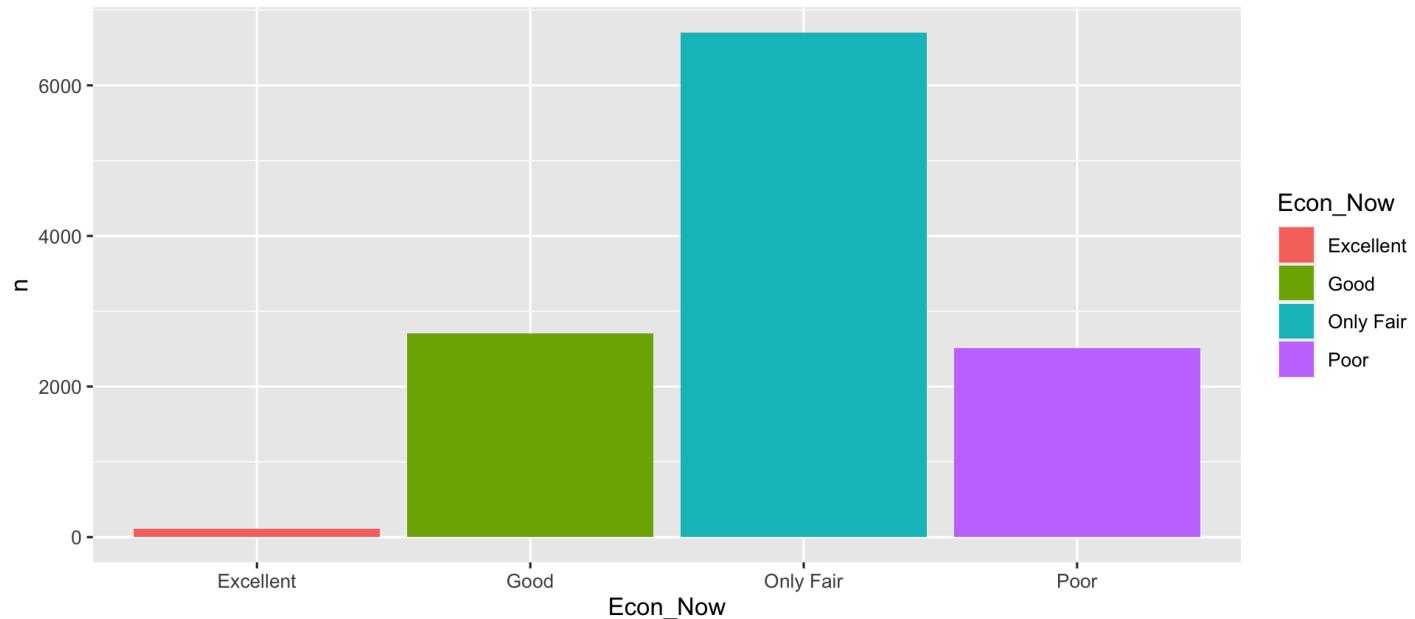
A Basic Bar Graph -- static color

```
ggplot(Econ_Now, aes(x = Econ_Now, y = n))+
  geom_bar(stat = "identity", position = position_dodge(),
           color = "blue", fill = "white")
```



A Basic Bar Graph -- Variable color

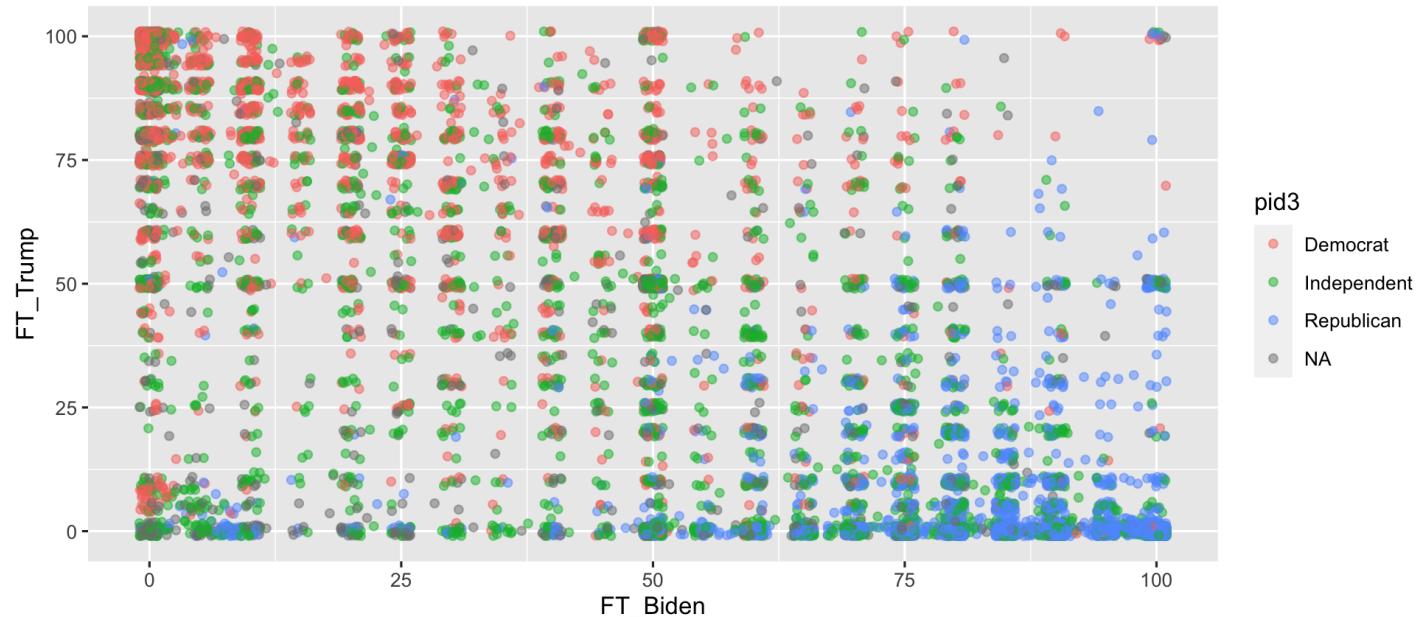
```
ggplot(Econ_Now, aes(x = Econ_Now, y = n, fill = Econ_Now))+  
  geom_bar(stat = "identity", position = position_dodge())
```



A Basic Scatterplot

My graph from the data layer...

```
ggplot(Pew,  
       aes(x = FT_Biden, y = FT_Trump, color = pid3))+  
  geom_point(position = position_jitter(1, 1), alpha = 0.5)
```



Exercise: Adding a basic graph

1. From your data layer code earlier, add a + to the end if it is not already there.
2. Using the graphs explored in this section, construct a basic graph and color accordingly

The Label Layer

Changing X and Y axis labels

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:

Adding Title, Legend Names and Subtitles

To add a title, subtitle, caption, or legend label to our graph, we can use the options provided by `labs()`

```
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

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.

```
ggplot(Pew,
       aes(x = FT_Biden, y = FT_Trump, color = pid3))+
  geom_point(position = position_jitter(1, 1), alpha = 0.5) +
  labs(
    title = "Feelings Towards Donald Trump and Joe Biden",
    subtitle = "March 2021",
    caption = "Data: Pew Research Center  
American Trends Panel Wave 84  
Author: Jennifer Lin",
    color = "Political Party"
  ) +
  xlab("Feelings towards Joe Biden") +
  ylab("Feelings towards Donald Trump")
```

Feelings Towards Donald Trump and Joe Biden

March 2021



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

Changing the Colors in Our Plots in R

`ggplot` comes with a lot of `scale_*_*`() functions. These help us override defaults.

Most `ggplot` scales come in the format of
`scale_[SOMETHING]_[SOMEHOW]`()

- [SOMETHING] -- What do you want to scale? Color, the x-axis (x), the fill, y-axis (y)?
- [SOMEHOW] -- How do you want it to start the rescale process? Transformed, gradients, manual?

Arguments

Most scales take the following arguments

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

Common Scale Uses

- `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**

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.

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

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

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)  
) +
```

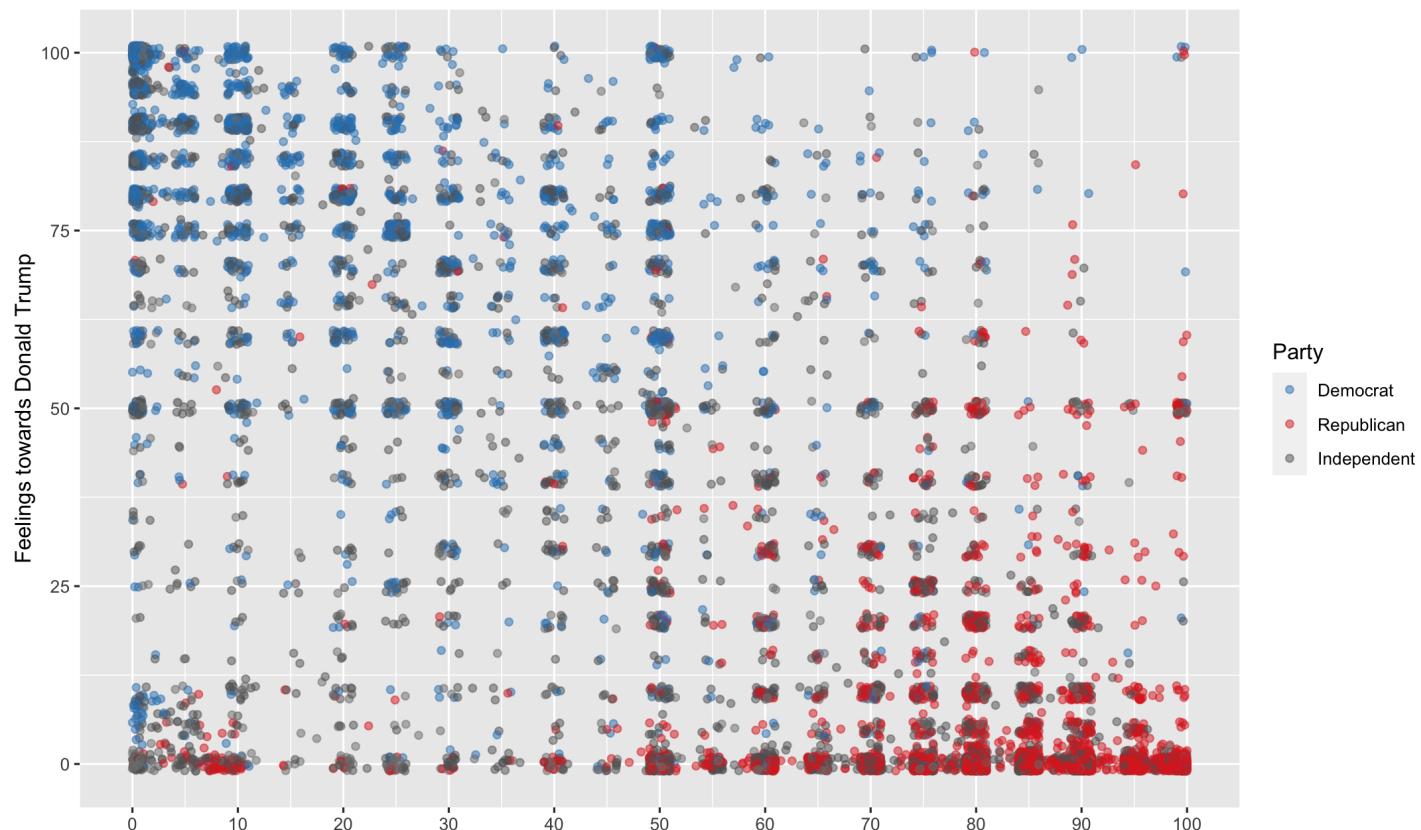
Here, I use the `breaks` and `limits` options

- `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)

```
ggplot(Pew,
       aes(x = FT_Biden, y = FT_Trump, color = pid3))+
  geom_point(position = position_jitter(1, 1), alpha = 0.5)+
  labs(
    title = "Feelings Towards Donald Trump and Joe Biden",
    subtitle = "March 2021",
    caption = "Data: Pew Research Center  
American Trends Panel Wave 84  
Author: Jennifer Lin",
    color = "Political Party"
  )+
  xlab("Feelings towards Joe Biden")+
  ylab("Feelings towards Donald Trump")+
  scale_color_manual(
    name = "Party",
    breaks = c("Democrat", "Republican", "Independent"),
    values = c("Democrat" = "#3182bd", "Republican" = "#de2d26", ''))
  +
  scale_x_continuous(
    breaks = seq(0, 100, 10),
    limits = c(0, 100)
)
```

Feelings Towards Donald Trump and Joe Biden

March 2021



Data: Pew Research Center
American Trends Panel Wave 84
Author: Jennifer Lin

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 #).

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.

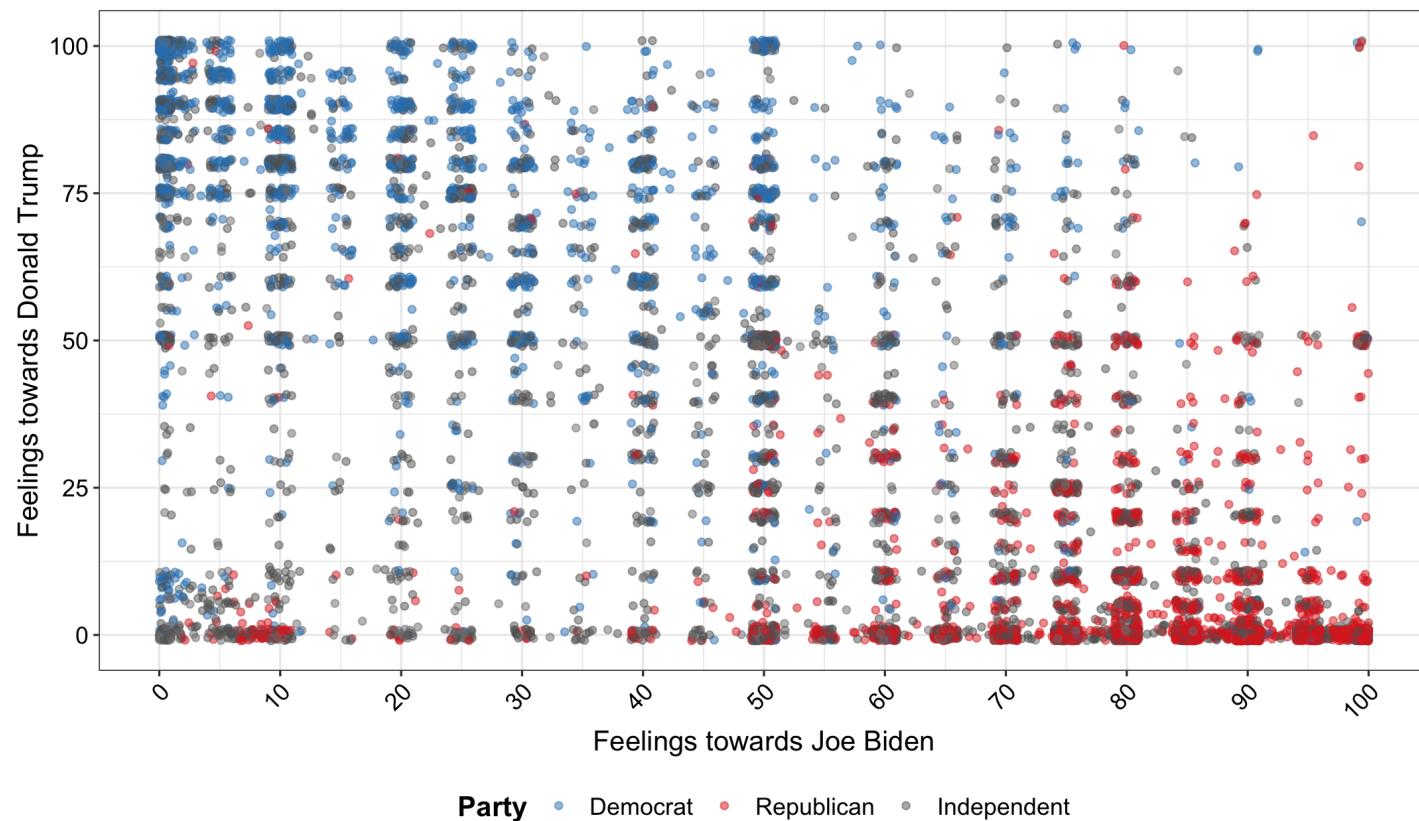
In Practice

```
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"))
)
```

```
ggplot(Pew,
       aes(x = FT_Biden, y = FT_Trump, color = pid3))+
  geom_point(position = position_jitter(1, 1), alpha = 0.5)+
  labs(
    title = "Feelings Towards Donald Trump and Joe Biden",
    subtitle = "March 2021",
    caption = "Data: Pew Research Center  
American Trends Panel Wave 84  
Author: Jennifer Lin",
    color = "Political Party"
) +
  xlab("Feelings towards Joe Biden") +
  ylab("Feelings towards Donald Trump") +
  scale_color_manual(
    name = "Party",
    breaks = c("Democrat", "Republican", "Independent"),
    values = c("Democrat" = "#3182bd", "Republican" = "#de2d26", "Independent" = "#4c78a8")
) +
  scale_x_continuous(
    breaks = seq(0, 100, 10),
    limits = c(0, 100)
) +
  theme_bw() +
  theme(
    plot.title = element_text(
      hjust = 0.5, size = 20, colour = "black",
      face = "bold.italic", family = "serif"
    )
  )
```

Feelings Towards Donald Trump and Joe Biden

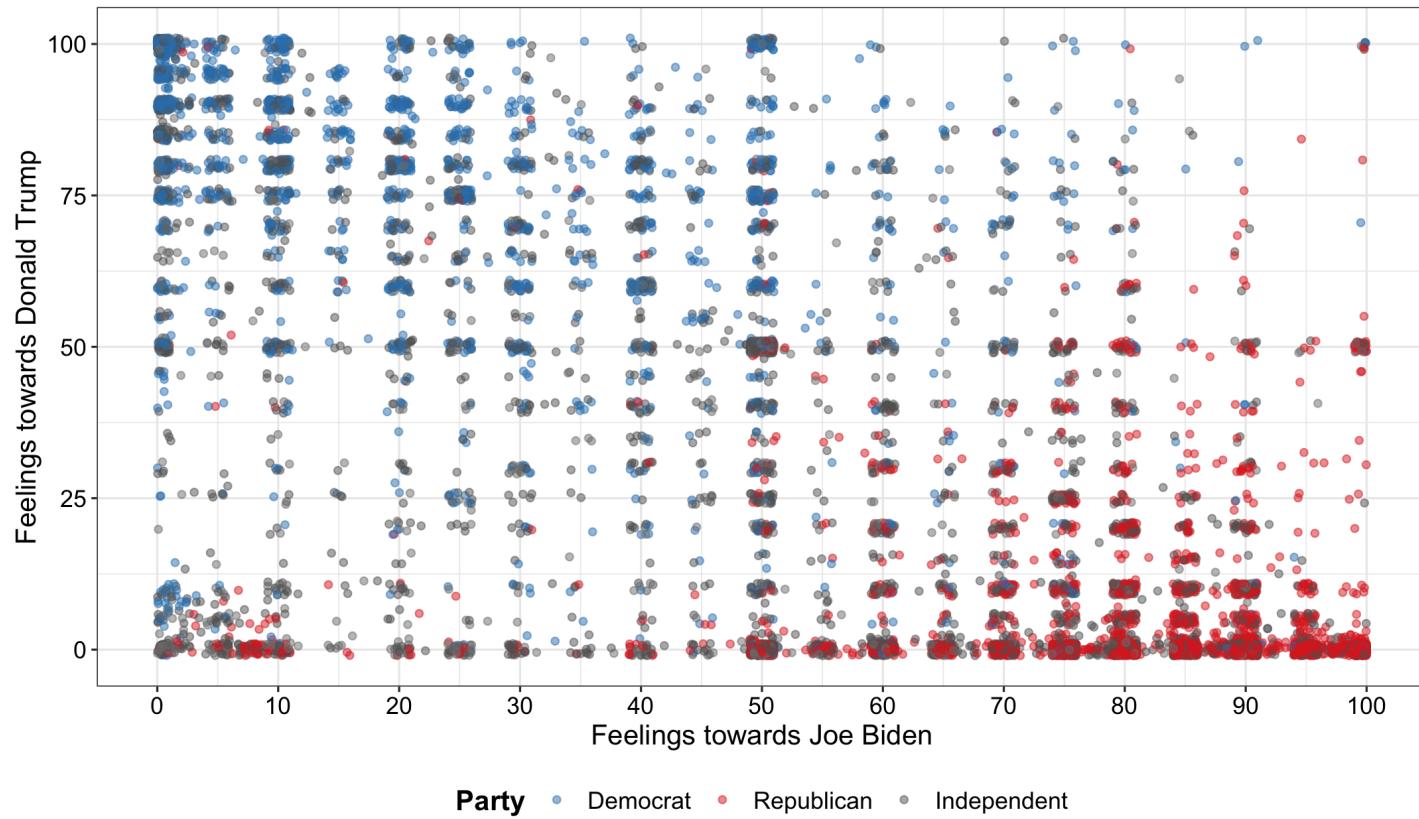
March 2021



Data: Pew Research Center
American Trends Panel Wave 84
Author: Jennifer Lin

Feelings Towards Donald Trump and Joe Biden

March 2021



Data: Pew Research Center
American Trends Panel Wave 84
Author: Jennifer Lin

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

Application

Spatial Data

Data

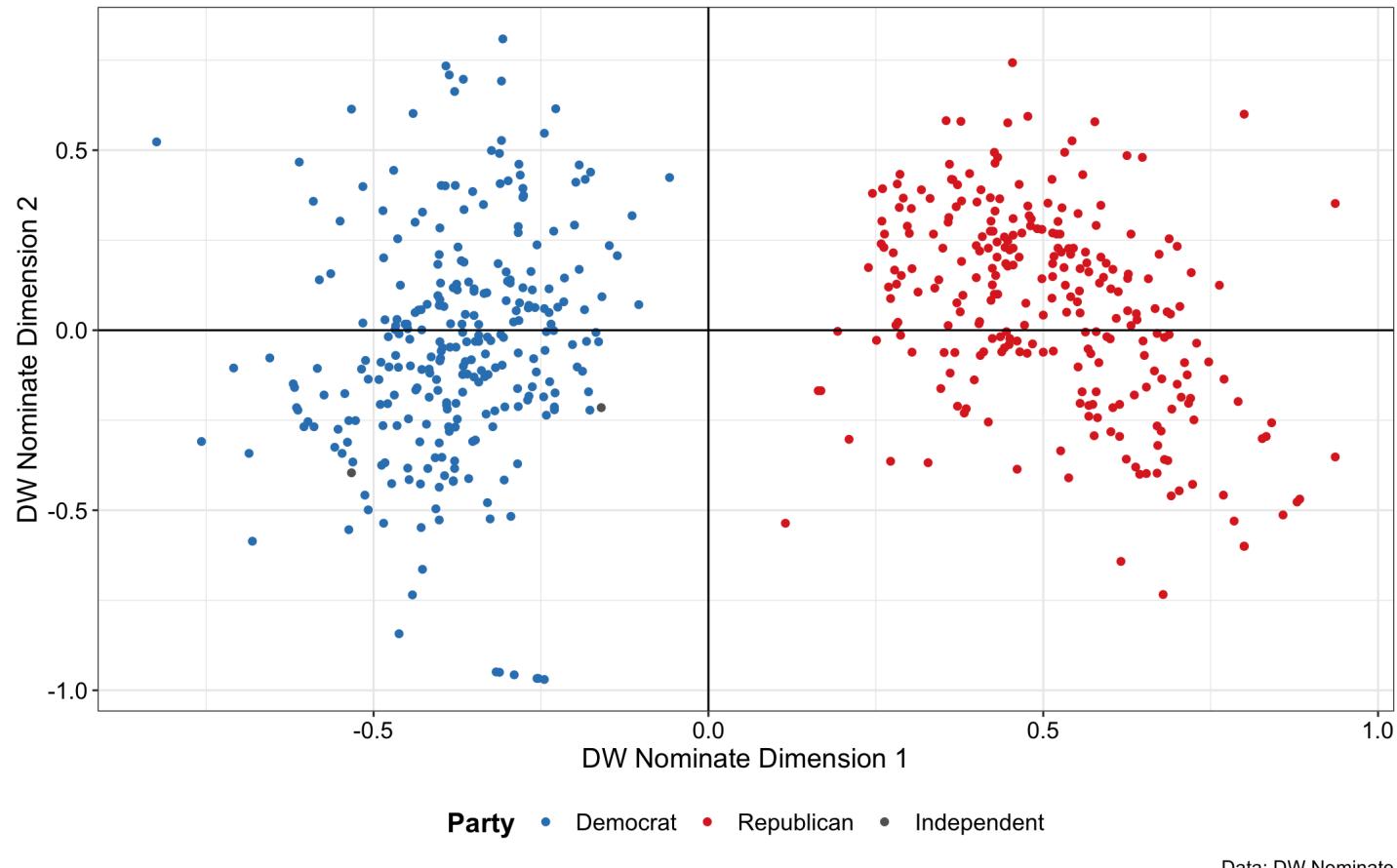
- DW Nominate places members of Congress on a one-dimensional ideological scale where -1 is the most liberal and 1 is the most conservative
- We can download the data or access it through a URL

```
url <- "https://voteview.com/static/data/"
extension <- "out/members/HS117_members.csv"
data_url <- paste0(url, extension)

N117 <- read.csv(data_url) %>%
  filter(chamber != "President") %>%
  mutate(
    party = case_when(
      party_code == 100 ~ "Democrat",
      party_code == 200 ~ "Republican",
      TRUE ~ "Independent"
    )
  )
```

```
ggplot(N117,
       aes(x = nominate_dim1, y = nominate_dim2,
           color = party))+
  geom_point()+
  geom_hline(yintercept=0)+
  geom_vline(xintercept=0)+
  scale_color_manual(
    name = "Party",
    breaks = c(
      "Democrat", "Republican", "Independent"),
    values = c(
      "Democrat" = "#3182bd",
      "Republican" = "#de2d26",
      "Independent" = "#636363"))
  )+
  xlab("DW Nominate Dimension 1")+
  ylab("DW Nominate Dimension 2")+
  labs(
    title = "Ideological Scores from Two Dimensions",
    subtitle = "From the 117th Congress",
    caption = "Data: DW Nominate"
  )+
  theme_bw()+
  theme(
    plot.title          = element_text(
      hjust = 0.5, size = 20, colour="black", face = "bold"),
    plot.subtitle        = element_text(
```

Ideological Scores from Two Dimensions From the 117th Congress



```
ggplot(N117, aes(x = nominate_dim1, y = 0, color = party))+
  geom_point(position = position_jitter(.1, .5))+  
  geom_hline(yintercept=0)+  
  xlim(-1, 1)+  
  facet_wrap(~chamber, scales = "free_x", ncol = 1)+  
  scale_color_manual(  
    name = "Party",  
    breaks = c(  
      "Democrat", "Republican", "Independent"),  
    values = c(  
      "Democrat" = "#3182bd",  
      "Republican" = "#de2d26",  
      "Independent" = "#636363"))  
)+  
  xlab("DW Nominate Dimension 1") +  
  labs(  
    title = "Ideological Scores from One Dimension",  
    subtitle = "From the 117th Congress",  
    caption = "Data: DW Nominate"  
)+  
  theme_bw() +  
  theme(  
    plot.title = element_text(  
      hjust = 0.5, size = 20, colour="black", face = "bold"),  
    plot.subtitle = element_text(  
      hjust = 0.5, size = 16, colour="black", face = "bold"),  
    legend.title = element_text(  
      hjust = 0.5, size = 16, colour="black", face = "bold"))
```

Ideological Scores from One Dimension From the 117th Congress

