Color, Legend, and Themes

2022-07-17

Import Survey Data

```
# install.packages("RColorBrewer")
# install.packages("ggthemes")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# the package that works well with ggplot that provides aesthetically pleasing color palettes
library(RColorBrewer)
library(ggthemes)

cces <- read_csv("cces_sample_coursera.csv")
```

```
## Rows: 1000 Columns: 25
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (25): caseid, region, gender, educ, edloan, race, hispanic, employ, mars...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

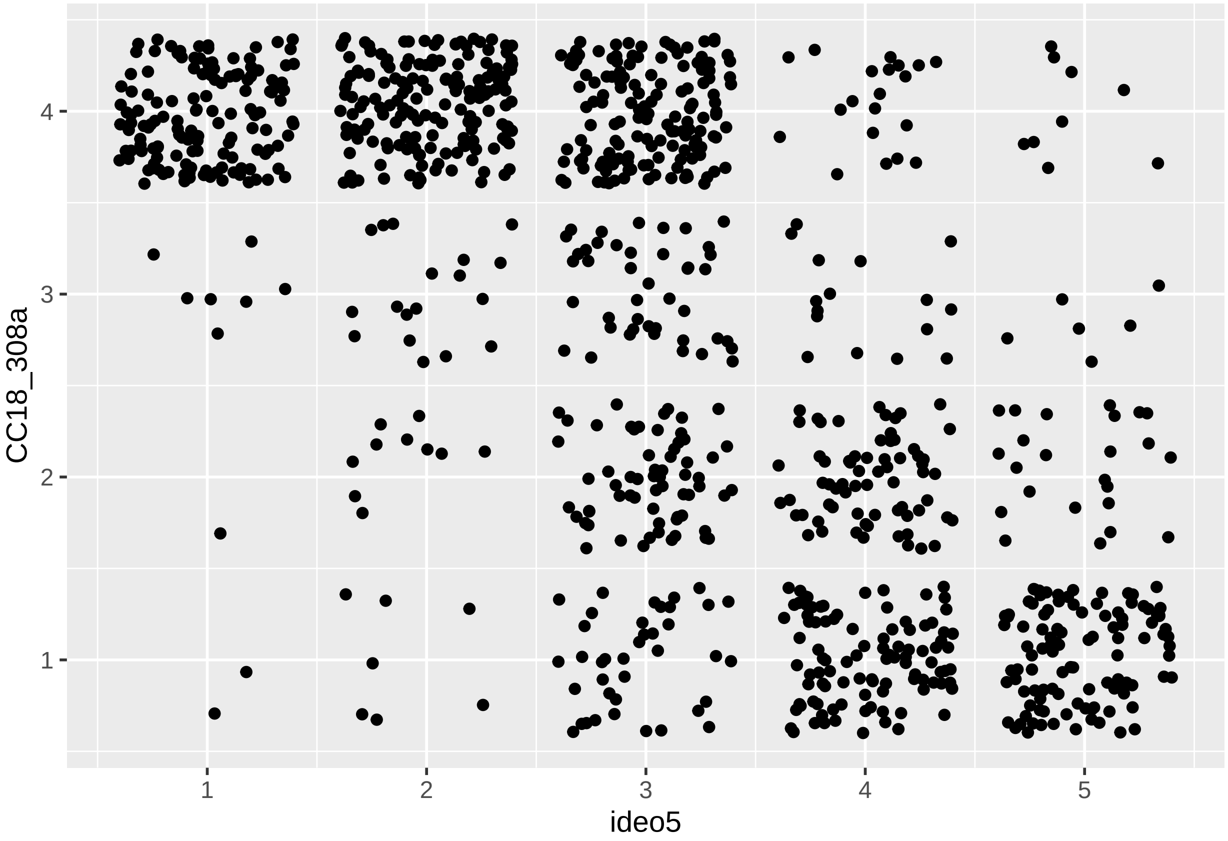
Make a Small Data Set with Variables

```
plot_data <- select(cces, "CC18_308a", "ideo5", "educ", "faminc_new", "employ") %>% drop_na()
```

Visualize the Data

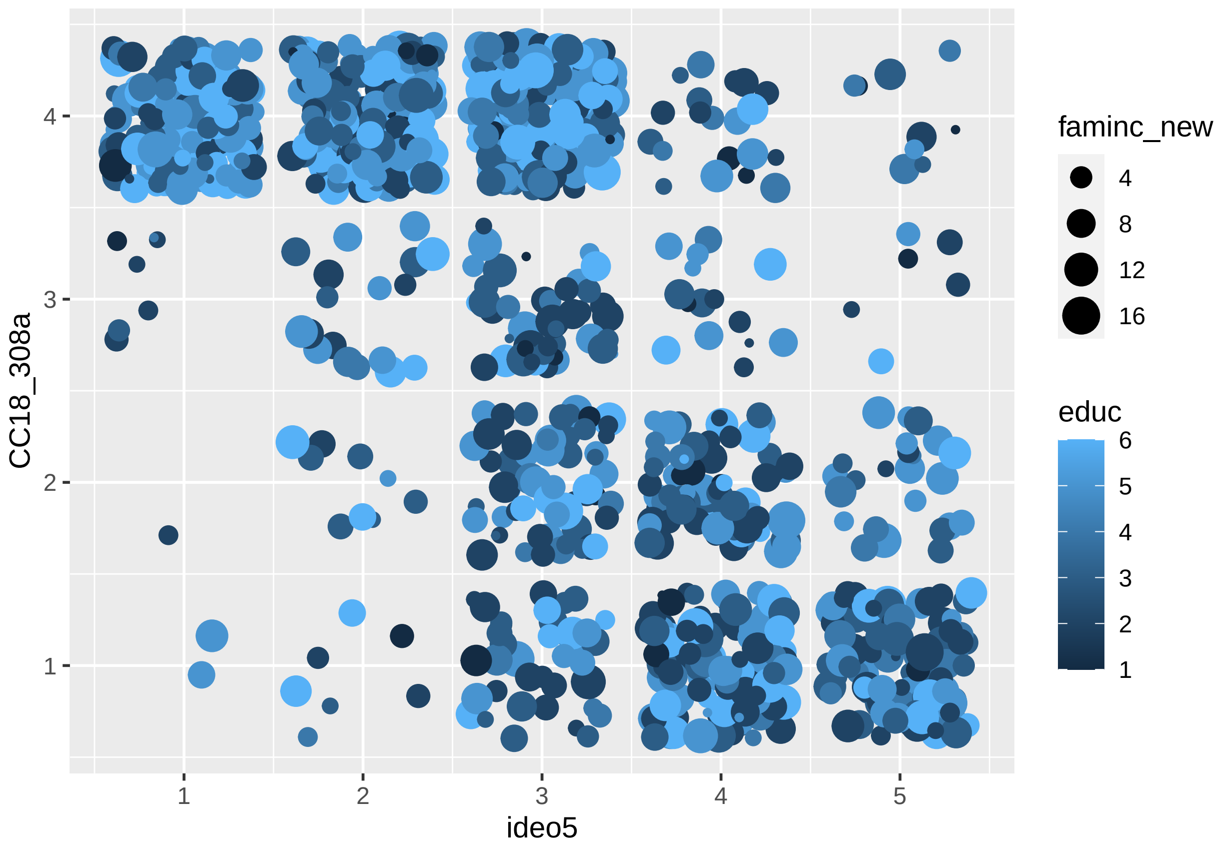
Basic Scatter Plot with Jitter

```
ggplot(plot_data, aes(y = CC18_308a, x = ideo5))+
  geom_jitter()
```

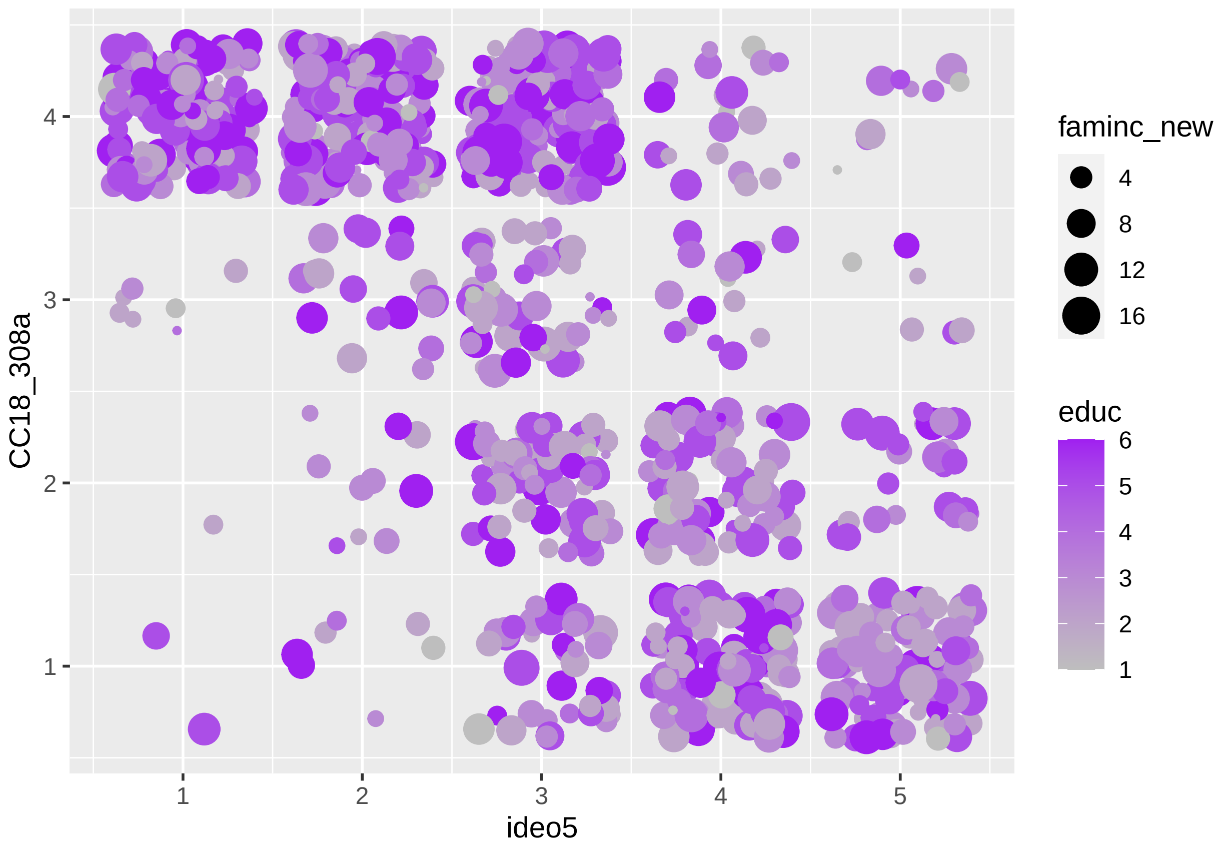


Scatter Plot with Automatic Color Gradient

```
ggplot(plot_data, aes(y = CC18_308a, x = ideo5, color = educ, size = faminc_new))+
  geom_jitter()
```



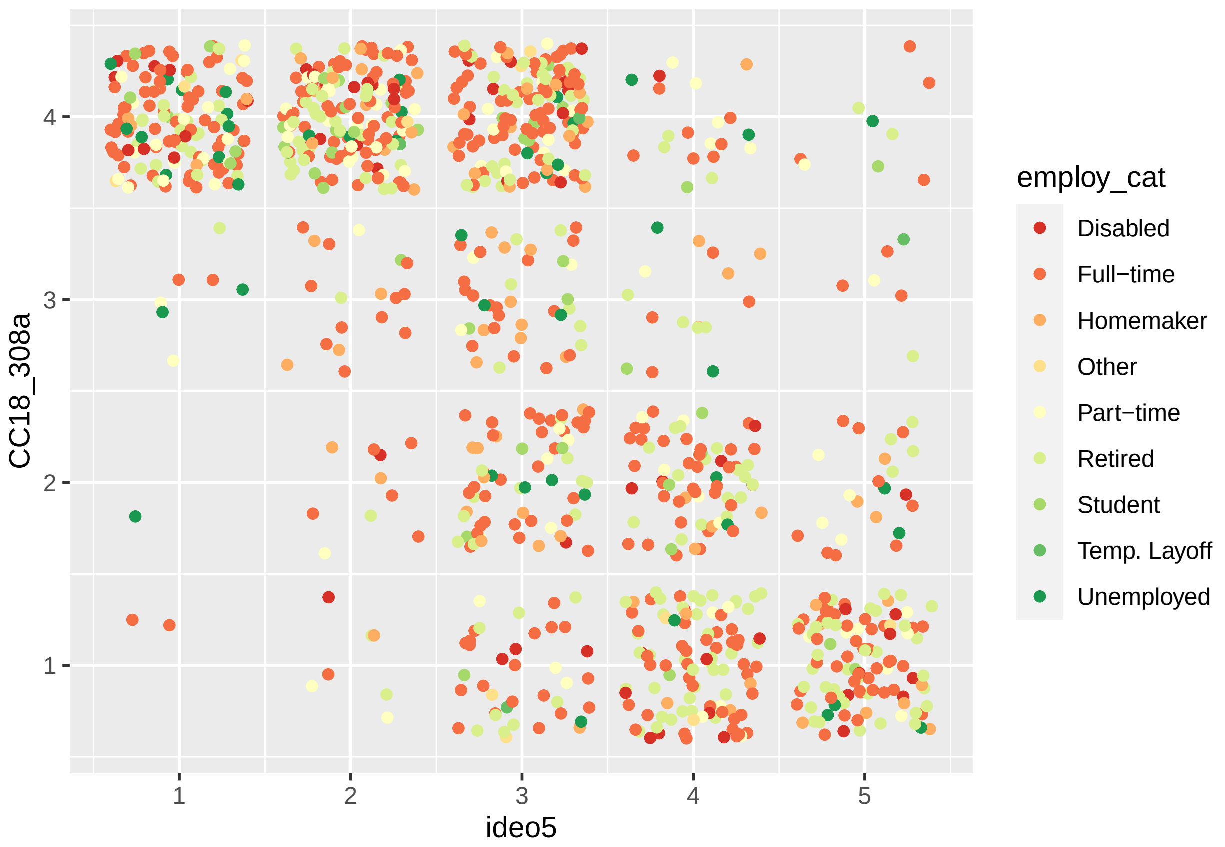
```
ggplot(plot_data, aes(y = CC18_308a, x = ideo5, color = educ, size = faminc_new))+
  geom_jitter()+
  # use scale_color_gradient here to show continuous change in a numeric variable
  scale_color_gradient(low = 'gray', high = 'purple')
```



Scatter Plot with Color Brewer

```
# Use employment as categorical variable
plot_data$employ_cat <- recode(plot_data$employ, '1' = "Full-time", '2' = "Part-time", '3' = "Temp. Lay

# Instead of scale color gradient, use scale color brewer for the discrete variable you have created
ggplot(plot_data, aes(y = CC18_308a, x = ideo5, color = employ_cat))+
  geom_jitter()+
  scale_color_brewer(palette = "RdYlGn")
```

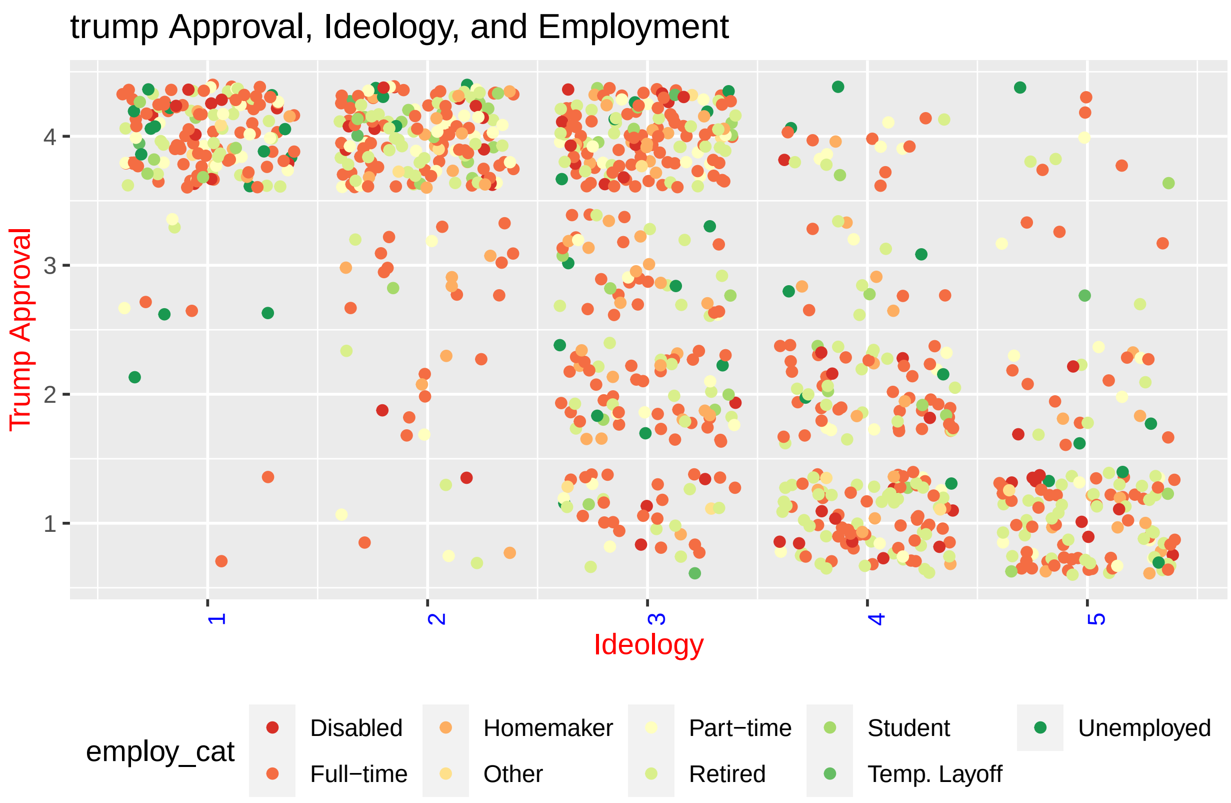


- There are many palletes – https://www.r-graph-gallery.com/38-rcolorbrewers-palettes

Using Themes

- Theme function is useful to modify elements of the figure (plot) that are not tied directly to values of the data.
- In the previous code we show you how to change the color or size of points based on the data itself. Now, we will use theme to modify elements that are consistent regardless of what the underlying data are.
- Note: **theme** is a complicated function since it has many arguments, so make sure to look at the documentation for theme in the tidyverse.

```
ggplot(plot_data, aes(x = ideo5, y = CC18_308a, color = employ_cat))+
  geom_jitter()+
  scale_color_brewer(name = "employ_cat", palette = "RdYlGn")+
  # look at all the control you have using the theme() function
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90, hjust = 1, color = "blue"),
        axis.title = element_text(color = "red"))+
  labs(x = "Ideology",y = "Trump Approval",
       title = "trump Approval, Ideology, and Employment", caption = "A caption might have a quick note
```



A caption might have a quick note about the figure or a sourcing