GVPT Data Science Camp

Data Visualization

Ryan Bookstein

August 25, 2026

University of Maryland

Learning Objectives

- Introduction to using `R` and `RStudio`

Learning Objectives

- Introduction to using `R` and `RStudio`
- Create your first plot in `R`

Learning Objectives

- Introduction to using `R` and `RStudio`
- Create your first plot in `R`
- Test your hypotheses using data visualizations

Lessons from Zinovyev (2010)

- Data visualization creates informative displays of data

Lessons from Zinovyev (2010)

- Data visualization creates informative displays of data
- We can display data in ways beyond what we will learn today

Lessons from Zinovyev (2010)

- Data visualization creates informative displays of data
- We can display data in ways beyond what we will learn today
- Data visualization is an important component of a research project

R Basics

- R evaluates the code you type — try some arithmetic:

```
1 + 2
```

- R evaluates the code you type — try some arithmetic:

```
1 + 2
```

- Functions take arguments inside parentheses:

```
sum(1, 2)
```

Exercise

- Find the sum of 412, 65, and 292

Exercise

- Find the sum of 412, 65, and 292
- What is 48 divided by 5?

Exercise

- Find the sum of `412`, `65`, and `292`
- What is `48` divided by `5`?
- What is the square root of `5232`? HINT: use the `sqrt()` function

Check Your Answers

- Two ways to get the sum:

```
sum(412, 65, 292)
#> [1] 769
412 + 65 + 292
#> [1] 769
```

Check Your Answers

- Two ways to get the sum:

```
sum(412, 65, 292)
#> [1] 769
412 + 65 + 292
#> [1] 769
```

- Division:

```
48 / 5
#> [1] 9.6
```

Check Your Answers

- Two ways to get the sum:

```
sum(412, 65, 292)
#> [1] 769
412 + 65 + 292
#> [1] 769
```

- Division:

```
48 / 5
#> [1] 9.6
```

- Square root:

```
sqrt(5232)
#> [1] 72.33257
```

Functions

- R has a large number of built-in functions that allow you to manipulate data

Functions

- R has a large number of built-in functions that allow you to manipulate data
- For example, the `seq()` function provides all numbers in a specified sequence:

```
seq(from = 1, to = 20, by = 3)
#> [1]  1  4  7 10 13 16 19
```

- R has a large number of built-in functions that allow you to manipulate data
- For example, the `seq()` function provides all numbers in a specified sequence:

```
seq(from = 1, to = 20, by = 3)
#> [1]  1  4  7 10 13 16 19
```

- Arguments are positional, so naming them is optional:

```
seq(1, 20, 3)
#> [1]  1  4  7 10 13 16 19
```

R Packages

- Packages are collections of R functions and data

R Packages

- Packages are collections of R functions and data
- Install the relevant package(s) once:

```
install.packages("tidyverse")
```

R Packages

- Packages are collections of R functions and data
- Install the relevant package(s) once:

```
install.packages("tidyverse")
```

- Load the packages in your current session:

```
library(tidyverse)
```

Exercise

- Using the RStudio console, install the `tidyverse` packages

```
install.packages("tidyverse")
```

Exercise

- Using the RStudio console, install the `tidyverse` packages

```
install.packages("tidyverse")
```

- Load these packages in your current session, along with `readxl` (installed with the tidyverse) for reading Excel files

```
library(tidyverse)
library(readxl)
```

- *For you and your co-author's sake*

- *For you and your co-author's sake*
- Keep everything:

- *For you and your co-author's sake*
- Keep everything:
  - Organized

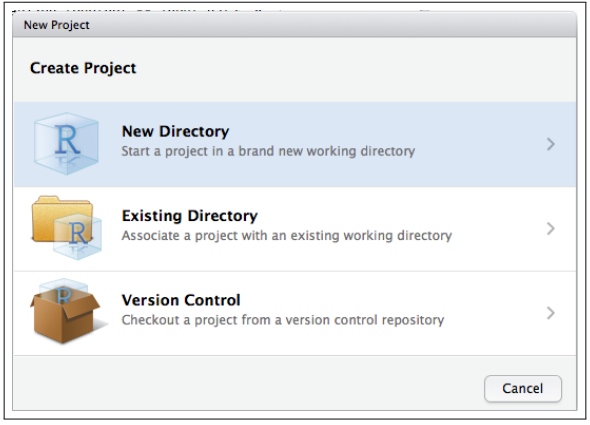
- *For you and your co-author's sake*
- Keep everything:
  - Organized
  - Reproducible

- *For you and your co-author's sake*
- Keep everything:
  - Organized
  - Reproducible
  - Annotated

Exercise

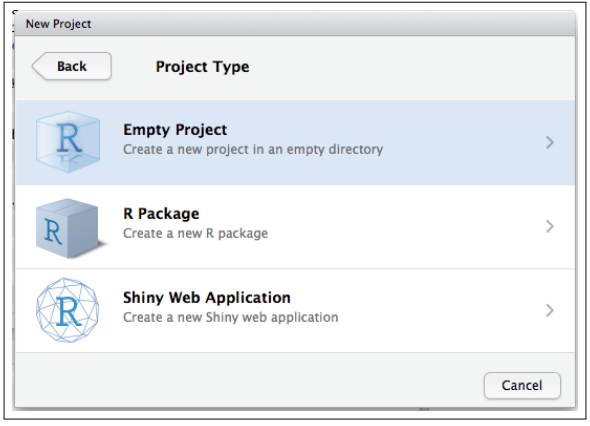
In the RStudio console, type `getwd()` to locate where you are in your computer

Create a new RStudio project for this Data Science Camp



Source: *R for Data Science* (R4DS)

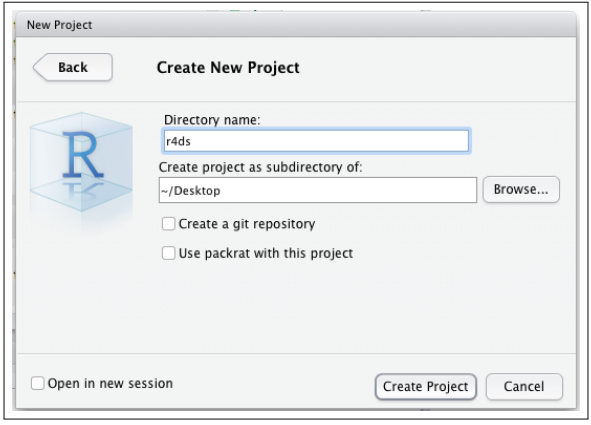
Create a new RStudio project for this Data Science Camp



Source: *R for Data Science* (R4DS)

Exercise

Create a new RStudio project for this Data Science Camp



Source: *R for Data Science* (R4DS)

Exercise

Re-run `getwd()` to locate where you are now in your computer

Introduction to Data Visualization

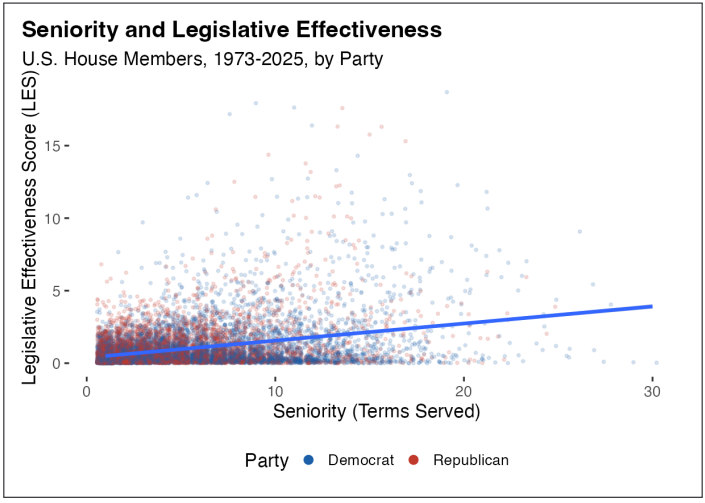
Data Visualization

- Data visualization is the graphical representation of information and data

Data Visualization

- Data visualization is the graphical representation of information and data
- From the Center for Effective Lawmaking (CEL) data: Do more senior members of Congress pass more legislation than newer members?

Skipping to the End



R code:

```
ggplot(data = cel, mapping = aes(x = seniority, y = les)) +
  geom_point(mapping = aes(color = party), alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0)) +
  scale_color_manual(name = "Party", values = party_colors) +
  geom_smooth(method = "lm") +
  theme(legend.position = "bottom",
        panel.grid = element_blank(),
        panel.background = element_blank(),
        plot.title.position = "plot",
        plot.title = element_text(face = "bold")) +
  labs(title = "Seniority and Legislative Effectiveness",
       subtitle = "U.S. House Members, 1973-2025, by Party",
       x = "Seniority (Terms Served)",
       y = "Legislative Effectiveness Score (LES)")
```

Setting Up the CEL Data

Load Relevant Packages and Data

- Load the relevant packages:

```
library(tidyverse)
library(readxl)
```

Load Relevant Packages and Data

- Load the relevant packages:

```
library(tidyverse)
library(readxl)
```

- Download the Center for Effective Lawmaking House dataset (`CELHouse93to118`) from thelawmakers.org

Load Relevant Packages and Data

- Load the relevant packages:

```
library(tidyverse)
library(readxl)
```

- Download the Center for Effective Lawmaking House dataset (`CELHouse93to118`) from thelawmakers.org
- Save it in your project folder as `CELHouse93to118.xlsx`

Load the CEL Data

- Read the Excel file into an object named `cel`:

```
cel <- read_excel("CELHouse93to118.xlsx")
```

Load the CEL Data

- Read the Excel file into an object named `cel`:

```
cel <- read_excel("CELHouse93to118.xlsx")
```

- One row per member per Congress: every member of the U.S. House from the 93rd Congress (1973) through the 118th (2025)

Rename the Variables

The columns arrive with long descriptive names — give them short ones. Copy this chunk as is; we will unpack how it works tomorrow:

```
cel <- rename(cel,
  member     = "Legislator name, as given in THOMAS",
  congress   = "Congress number",
  year       = "Year at start of Congress",
  state      = "Two-letter state code",
  dem        = "1 = Democrat",
  female     = "1 = female",
  votepct    = "Percent vote received to enter this Congress",
  ideology   = "First-dimension DW-NOMINATE score",
  seniority = "Seniority, number of terms served counting current",
  born       = "Year born",
  les        = "LES 1.0",
  bills_passed = "Total number of standalone bills passing House",
  benchmark = "1 = Below, 2 = Meets, 3 = Above, Based on Classic LES")
```

Keep and Create Variables

Keep only the renamed columns, and add readable labels for party and for the CEL benchmark categories:

```
cel <- select(cel, member, congress, year, state, dem, female,
              votepct, ideology, seniority, born, les,
              bills_passed, benchmark)

cel <- mutate(cel,
  party = if_else(dem == 1, "Democrat", "Republican"),
  benchmark = factor(benchmark,
                     labels = c("Below", "Meets", "Above")))
```

Exercise

- Learn more about this dataset by browsing the CEL codebook at thelawmakers.org

Exercise

- Learn more about this dataset by browsing the CEL codebook at thelawmakers.org
- Get a quick overview of every variable in your console:

```
glimpse(cel)
```

- Open the data in a spreadsheet-style viewer:

```
view(cel)
```

- Open the data in a spreadsheet-style viewer:

```
view(cel)
```

- A couple of useful variables — `seniority`: number of terms served, counting the current one; `les`: Legislative Effectiveness Score, a summary of how successful a member is at moving their bills forward

Exercise

- How many rows and columns are in `cel`?

```
nrow(cel)
ncol(cel)
```

Exercise

- How many rows and columns are in `cel`?

```
nrow(cel)
ncol(cel)
```

- What does the `votepct` variable describe?

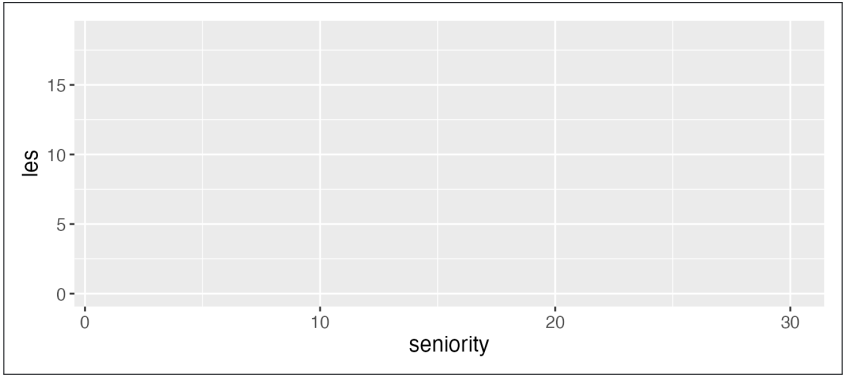
Building the Plot

Set Up Your Plot

```
ggplot(data = cel)
```

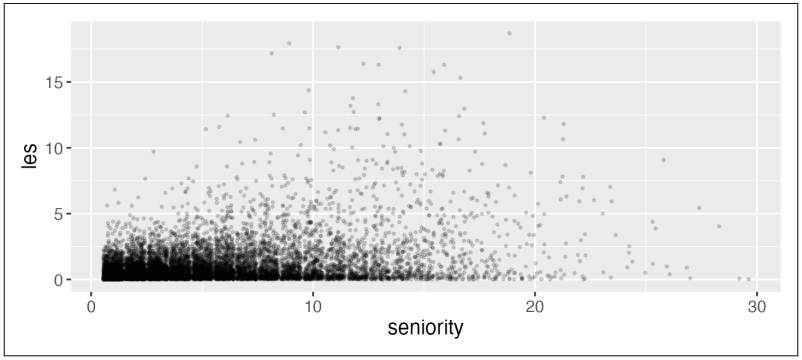
Map Your Aesthetics

```
ggplot(data = cel, mapping = aes(x = seniority, y = les))
```



Add in Your Legislators

```
ggplot(data = cel, mapping = aes(x = seniority, y = les)) +
  geom_point()
```



Exercise

- Inspect the `les` and `votepct` variables and describe them

Exercise

- Inspect the `les` and `votepct` variables and describe them
- Make a scatter plot of them

Exercise

- Why does the following code give an error, and how would you fix it?

```
ggplot(data = cel) +
  geom_point()
```

Let's Look at Groups in the Data

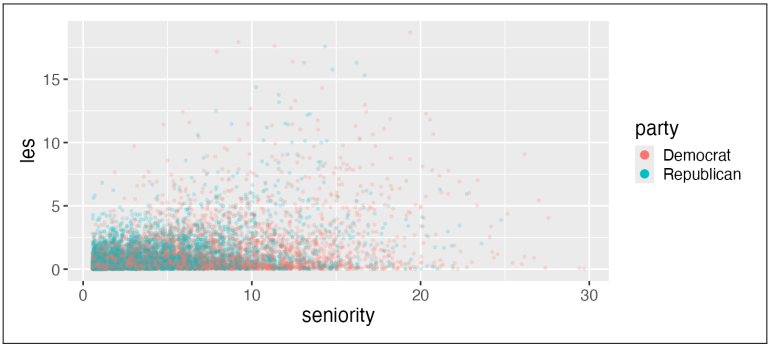
- You can use visual elements or aesthetics (`aes`) to communicate your data

Let's Look at Groups in the Data

- You can use visual elements or aesthetics (`aes`) to communicate your data
- Let's look at a categorical variable: the party of the member (Democrat or Republican)

Let's Plot Groups in the Data

```
ggplot(data = cel,
       mapping = aes(x = seniority, y = les, color = party)) +
  geom_point(alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0))
```



Two Problems With That Plot

- `cel` has 11,606 rows — one per member per Congress — so the points sit on top of each other. `alpha` and `position_jitter()` let us see through the pile

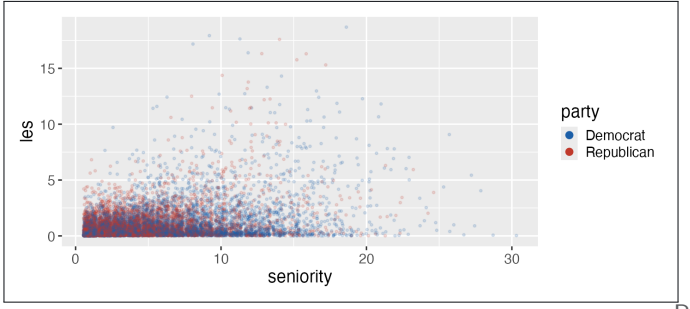
Two Problems With That Plot

- `cel` has 11,606 rows — one per member per Congress — so the points sit on top of each other. `alpha` and `position_jitter()` let us see through the pile
- R chose the colors alphabetically, so Democrats came out red and Republicans blue — we should choose them ourselves

Choosing Your Own Colors

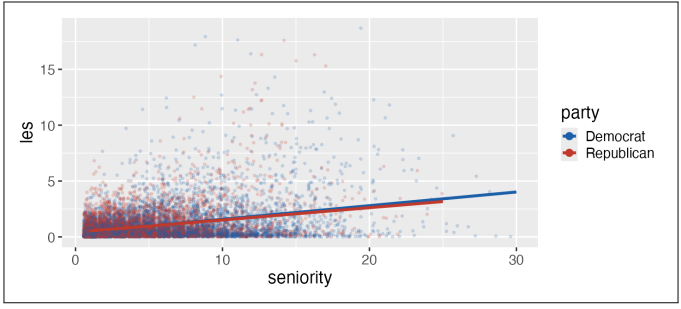
```
party_colors <- c(Democrat = "#1B5FA8", Republican = "#C0392B")

ggplot(data = cel,
       mapping = aes(x = seniority, y = les, color = party)) +
  geom_point(alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0)) +
  scale_color_manual(values = party_colors)
```



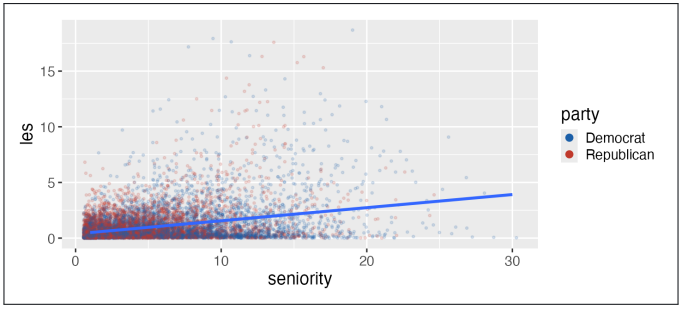
Add More Information

```
ggplot(data = cel,
       mapping = aes(x = seniority, y = les, color = party)) +
  geom_point(alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0)) +
  scale_color_manual(values = party_colors) +
  geom_smooth(method = "lm", se = FALSE)
```



Look at the Relationship Among All Members

```
ggplot(data = cel, mapping = aes(x = seniority, y = les)) +
  geom_point(mapping = aes(color = party), alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0)) +
  scale_color_manual(values = party_colors) +
  geom_smooth(method = "lm", se = FALSE)
```

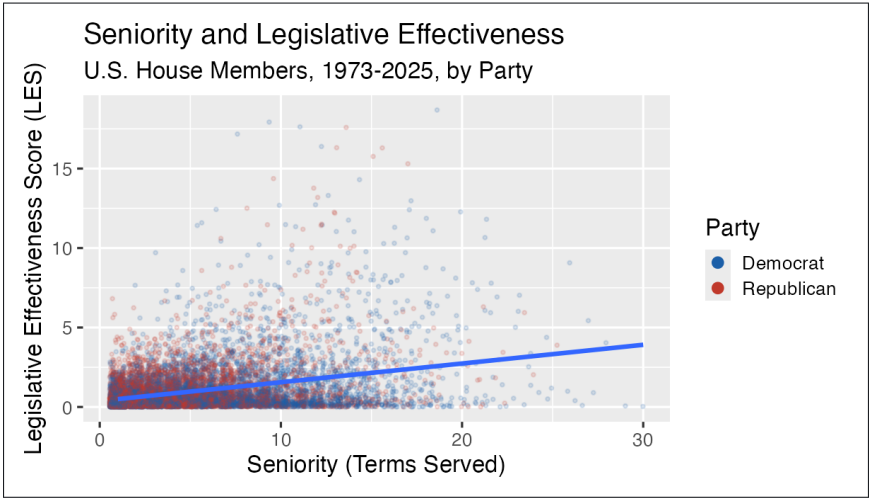


Polishing the Plot

Add Titles and Labels

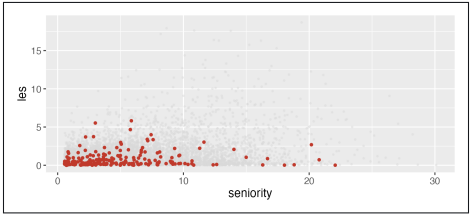
```
ggplot(data = cel, mapping = aes(x = seniority, y = les)) +
  geom_point(mapping = aes(color = party), alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0)) +
  scale_color_manual(name = "Party", values = party_colors) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Seniority and Legislative Effectiveness",
    subtitle = "U.S. House Members, 1973-2025, by Party",
    x = "Seniority (Terms Served)",
    y = "Legislative Effectiveness Score (LES)"
  )
```

Output with Titles and Labels

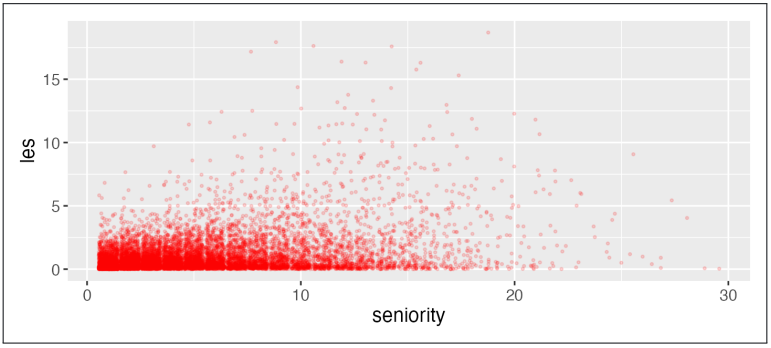


```
ggplot(cel, aes(x = seniority, y = les)) +
  geom_point(data = filter(cel, state != "MD"),
             color = "gray85", alpha = 0.4) +
  geom_point(data = filter(cel, state == "MD"),
             color = "#C0392B")
```

- Only 211 of the 11,606 rows are Maryland members, so we draw everyone else in gray first and put Maryland on top



```
ggplot(data = cel) +
  geom_point(mapping = aes(x = seniority, y = les), color = "red")
```



Exercise

- What is wrong with this code?

```
ggplot(data = cel) +
  geom_point(mapping = aes(x = seniority, y = les, color = "blue"))
```

Exercise

- Name a categorical and continuous variable in `cel`

Exercise

- Name a categorical and continuous variable in `cel`
- Map a continuous variable (try `votepct`) to color. How do the aesthetics behave differently for categorical and continuous variables?

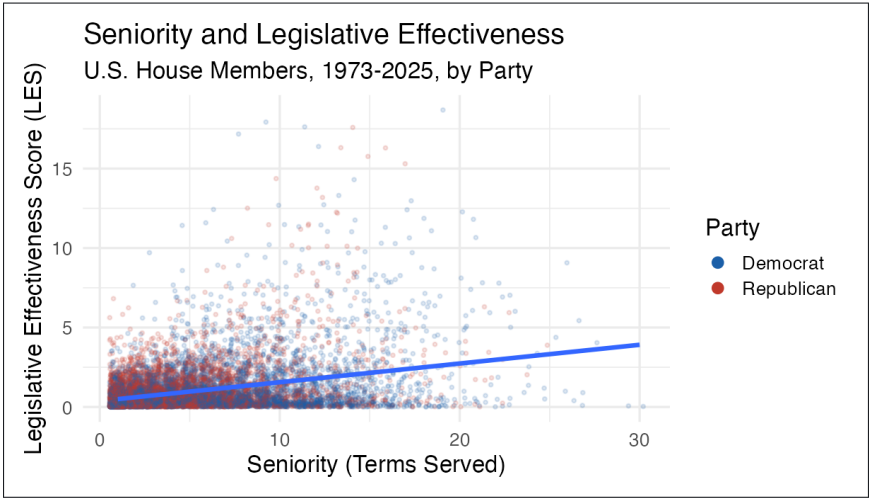
Exercise

- Name a categorical and continuous variable in `cel`
- Map a continuous variable (try `votepct`) to color. How do the aesthetics behave differently for categorical and continuous variables?
- Map `state` to the shape aesthetic. What does the warning say?

Cleaning Our Graph

```
ggplot(data = cel, mapping = aes(x = seniority, y = les)) +
  geom_point(mapping = aes(color = party), alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0)) +
  scale_color_manual(name = "Party", values = party_colors) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_minimal() +
  labs(
    title = "Seniority and Legislative Effectiveness",
    subtitle = "U.S. House Members, 1973-2025, by Party",
    x = "Seniority (Terms Served)",
    y = "Legislative Effectiveness Score (LES)"
  )
```

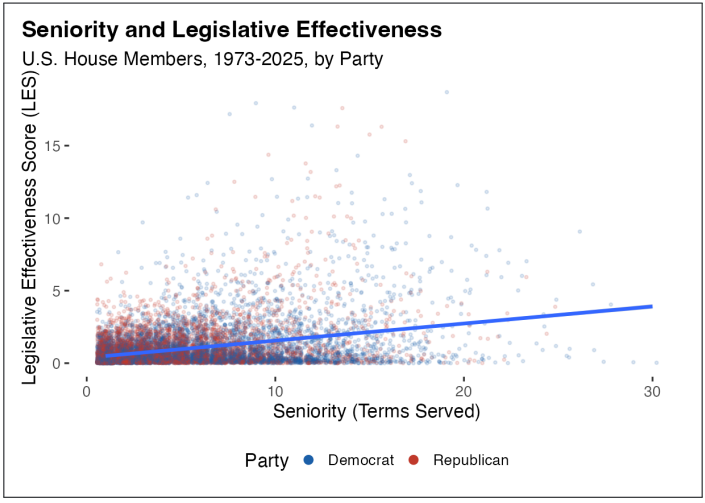
Cleaning Our Graph



Creating a Unique Theme

```
ggplot(data = cel, mapping = aes(x = seniority, y = les)) +
  geom_point(mapping = aes(color = party), alpha = 0.15,
             position = position_jitter(width = 0.45, height = 0)) +
  scale_color_manual(name = "Party", values = party_colors) +
  geom_smooth(method = "lm") +
  theme(legend.position = "bottom",
        panel.grid = element_blank(),
        panel.background = element_blank(),
        plot.title.position = "plot",
        plot.title = element_text(face = "bold")) +
  labs(title = "Seniority and Legislative Effectiveness",
       subtitle = "U.S. House Members, 1973-2025, by Party",
       x = "Seniority (Terms Served)",
       y = "Legislative Effectiveness Score (LES)")
```

Creating a Unique Theme



Exercise

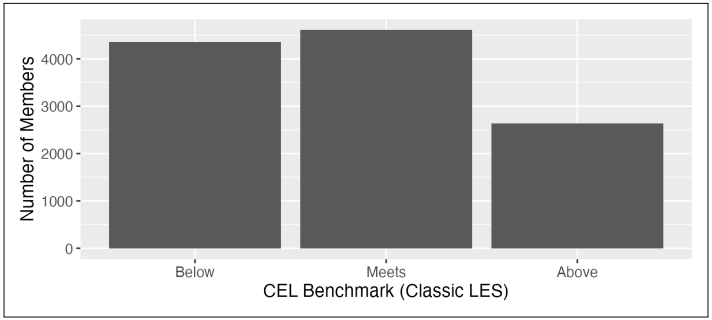
- Customize the scatter plot you created using the `theme()` argument

Other Chart Types

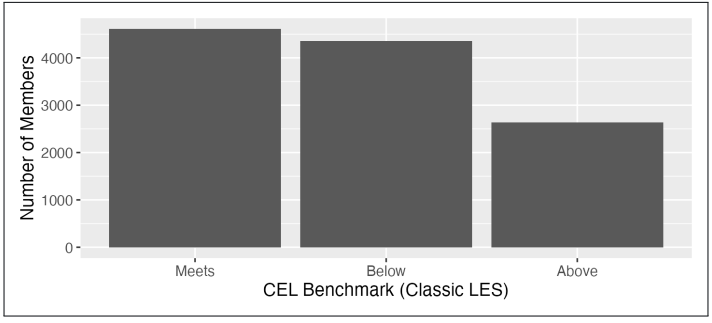
- When visualizing and reporting categorical data, we must use different tools

```
select(cel, member, state, party)
```

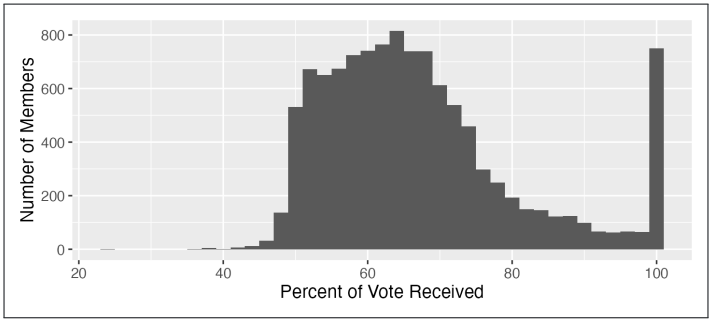
```
ggplot(cel, aes(x = benchmark)) +
  geom_bar() +
  labs(x = "CEL Benchmark (Classic LES)", y = "Number of Members")
```



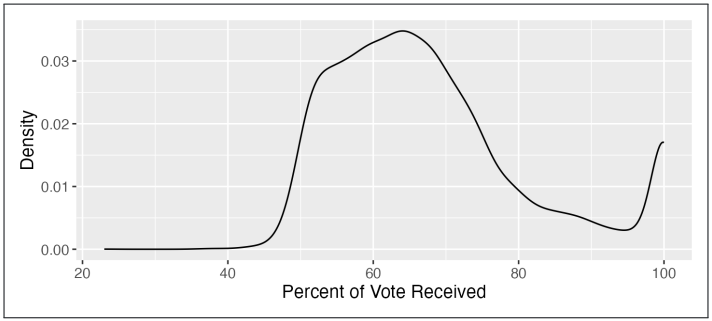
```
ggplot(cel, aes(x = fct_infreq(benchmark))) +
  geom_bar() +
  labs(x = "CEL Benchmark (Classic LES)", y = "Number of Members")
```



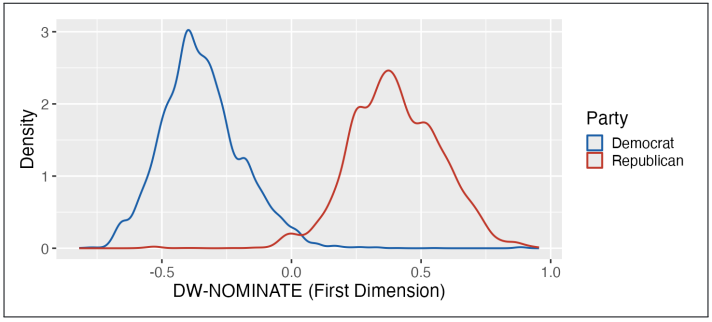
```
ggplot(cel, aes(x = votepct)) +
  geom_histogram(binwidth = 2) +
  labs(x = "Percent of Vote Received", y = "Number of Members")
```



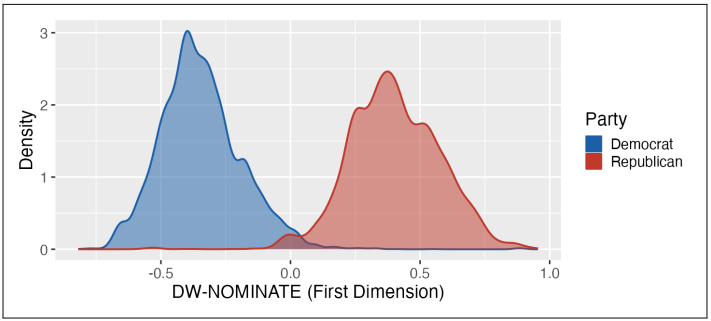
```
ggplot(cel, aes(x = votepct)) +
  geom_density() +
  labs(x = "Percent of Vote Received", y = "Density")
```



```
ggplot(cel, aes(x = ideology, color = party)) +
  geom_density() +
  scale_color_manual(name = "Party", values = party_colors) +
  labs(x = "DW-NOMINATE (First Dimension)", y = "Density")
```

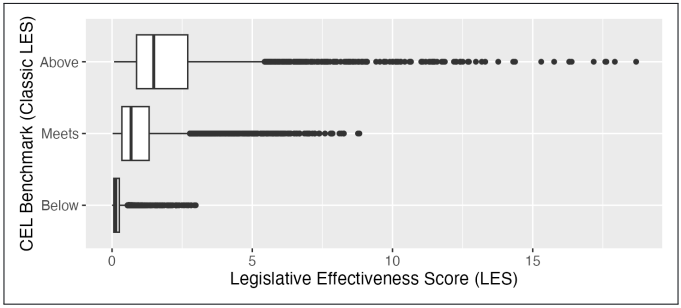


```
ggplot(cel, aes(x = ideology, color = party, fill = party)) +
  geom_density(alpha = 0.5) +
  scale_color_manual(name = "Party", values = party_colors) +
  scale_fill_manual(name = "Party", values = party_colors) +
  labs(x = "DW-NOMINATE (First Dimension)", y = "Density")
```



Comparing a Continuous Variable Across Groups

```
ggplot(cel, aes(x = les, y = benchmark)) +
  geom_boxplot() +
  labs(x = "Legislative Effectiveness Score (LES)",
       y = "CEL Benchmark (Classic LES)")
```



Summary

- Set up data science tools

Summary

- Set up data science tools
- Plotted data in multiple ways

Summary

- Set up data science tools
- Plotted data in multiple ways
- Discovered relationships in the data

Summary

- Set up data science tools
- Plotted data in multiple ways
- Discovered relationships in the data
- Connected these relationships to expectations (hypotheses)

- On Thursday, you will apply the skills learned in this course to an idea that interests you. In doing so, you will need to find a dataset that:

Homework

- On Thursday, you will apply the skills learned in this course to an idea that interests you. In doing so, you will need to find a dataset that:
  - Is relevant to your research interests

- On Thursday, you will apply the skills learned in this course to an idea that interests you. In doing so, you will need to find a dataset that:
  - Is relevant to your research interests
  - Contains continuous and categorical variables

Homework

- On Thursday, you will apply the skills learned in this course to an idea that interests you. In doing so, you will need to find a dataset that:
  - Is relevant to your research interests
  - Contains continuous and categorical variables
- If needed, reach out to me for dataset recommendations

- Zinovyev, Andrei. 2010. "Data Visualization in Political and Social Sciences." In *International Encyclopedia of Political Science*, ed. Bertrand Badie, Dirk Berg-Schlosser, and Leonardo Morlino. Thousand Oaks, CA: SAGE.

Questions?