

Visualizing Data (in R)

An opinionated style guide

June Choe

Trueswell-Swingley-Gleitman Lab

2020-11-25

Motivation

Why bother improving the quality of your plot?

- Helps people **remember** your results better
- Makes your findings look more **credible**
- Increases the chance of your work being **shared**
- Shows **respect** for your audience

Goal of this presentation

- Share practical, language-agnostic tips for improving readability & accessibility
- Demonstrate implementations of these ideas (w/ R code)
- Convince you that the *design* is inseparable from the *content*

Outline

Four principles for explanatory data visualization

- Make your **text** readable
- Be generous about **margins and spacing**
- Provide a clear **legend**
- Maximize contrast in **color**
- Reduce Visual Noise

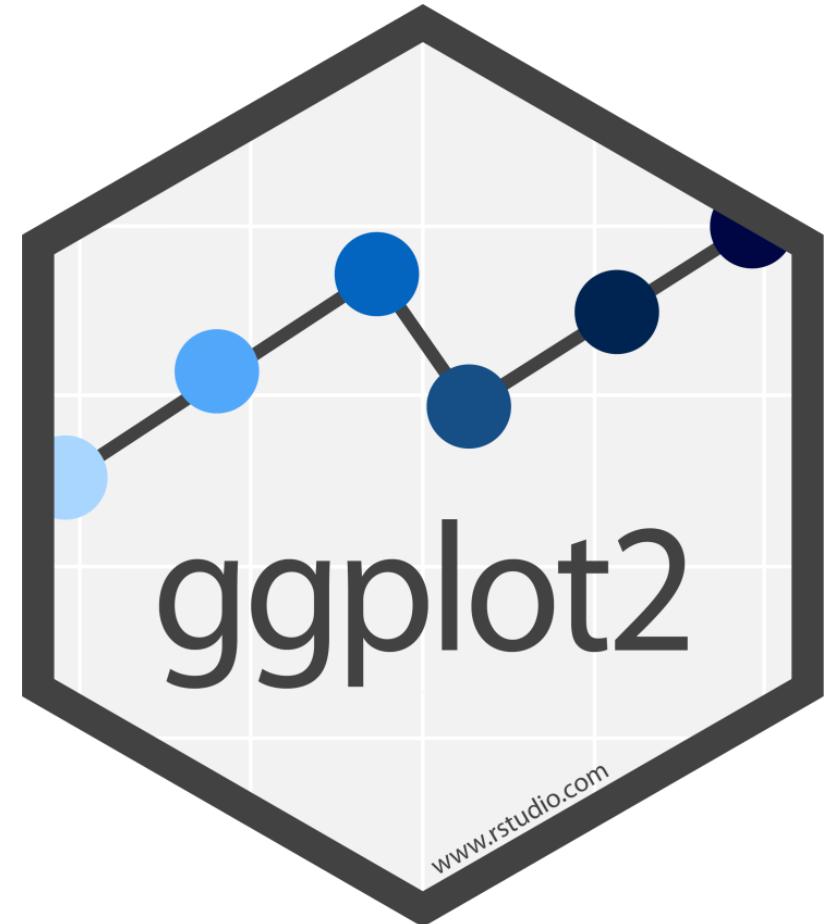
Showcases

- Vowel plots
- Bar plot of proportions
- Multiple categorical levels
- Animations

Preliminaries

The `{ggplot2}` ecosystem in R:

- Easy to use and highly customizable
- Tons of free resources for learning
- Widely used in academia
- Lots of extensions



Preliminaries

The `{ggplot2}` ecosystem in R:

- Easy to use and highly customizable
- Tons of free resources for learning
- Widely used in academia
- Lots of extensions

```
library(extrafont)
library(scales)
library(colorspace)
library(ggtext)
library(lemon)
library(gganimate)
# library(patchwork)
# library(gghighlight)
# library(ggforce)
# library(ggrepel)
```

ggplot2 extensions - gallery

- [Add Your Extension!](#)
- [exts.ggplot2.tidyverse.org](#)
- [Navbar Link](#)



81 registered extensions available to explore

▼ Github stars

- Name
- Author
- Github stars

[Github stars ▾](#)

Sort

Text Filter

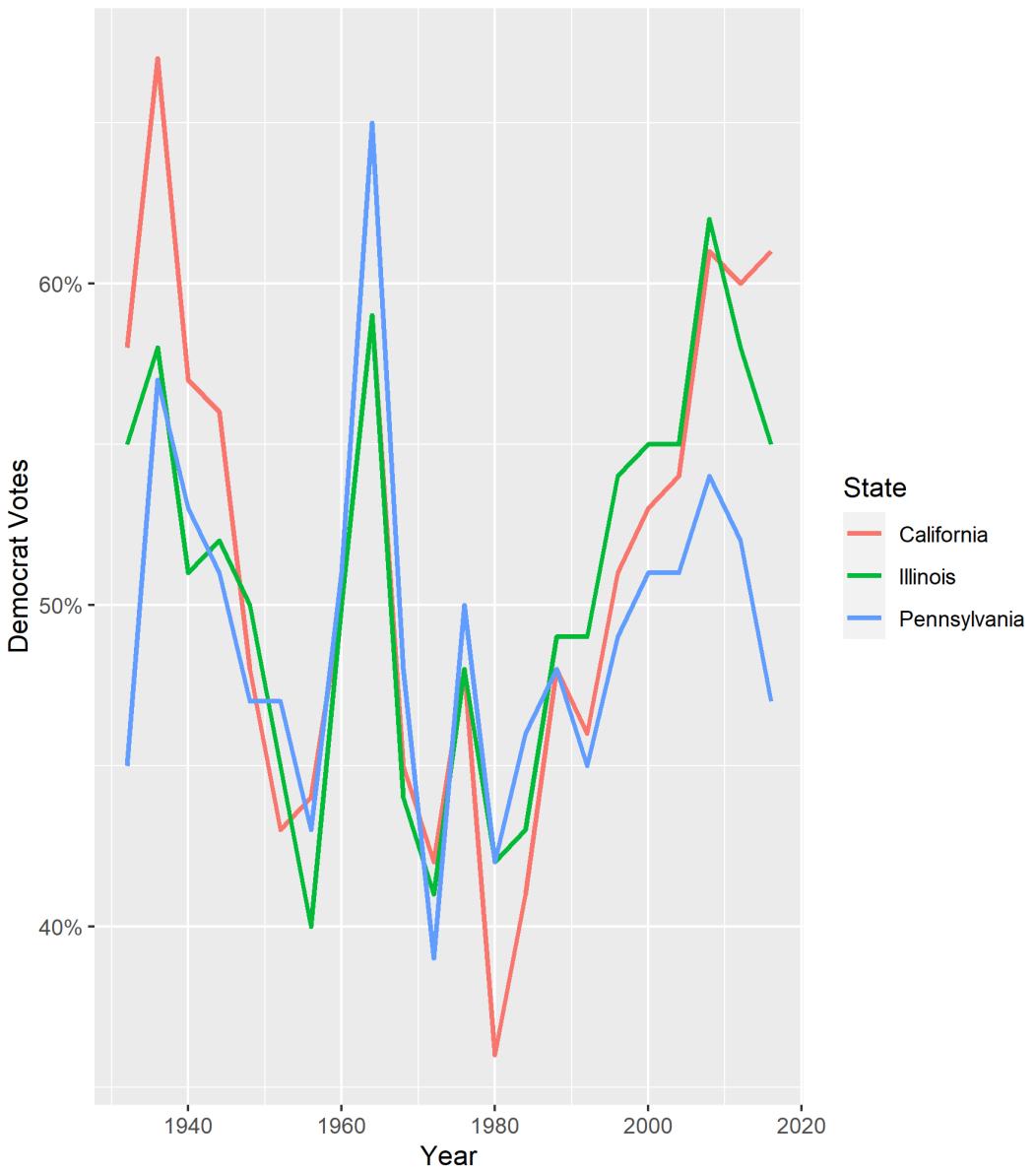


```

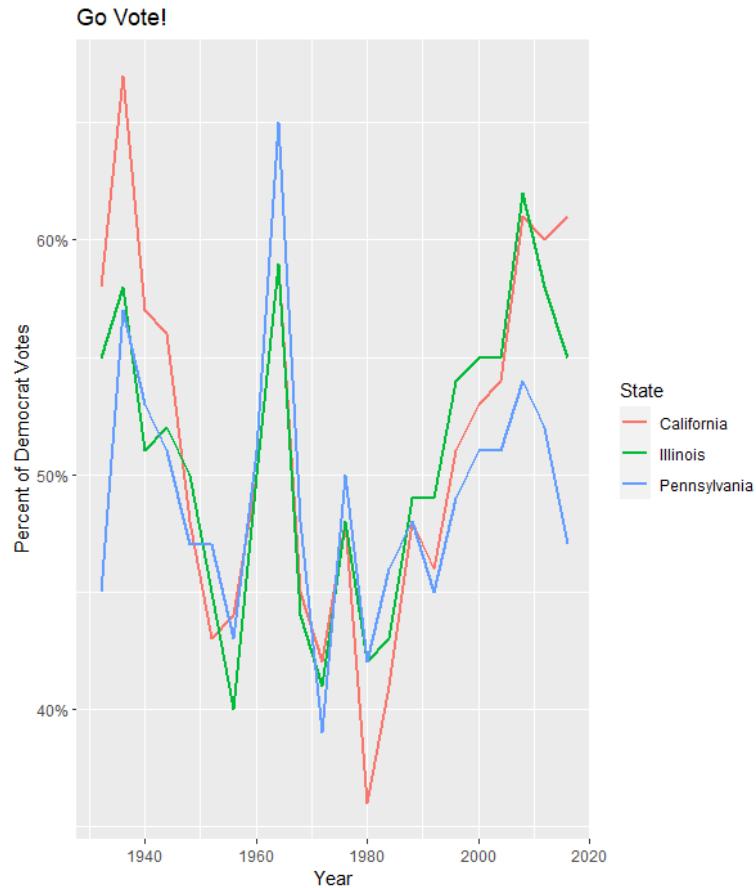
state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
  geom_line(size = 1) +
  scale_y_continuous(
    labels = percent_format(accuracy = 1)
) +
  scale_x_continuous(
    breaks = pretty_breaks(n= 5)
) +
  labs(
    y = "Democrat Votes",
    title = "Go Vote!"
)

```

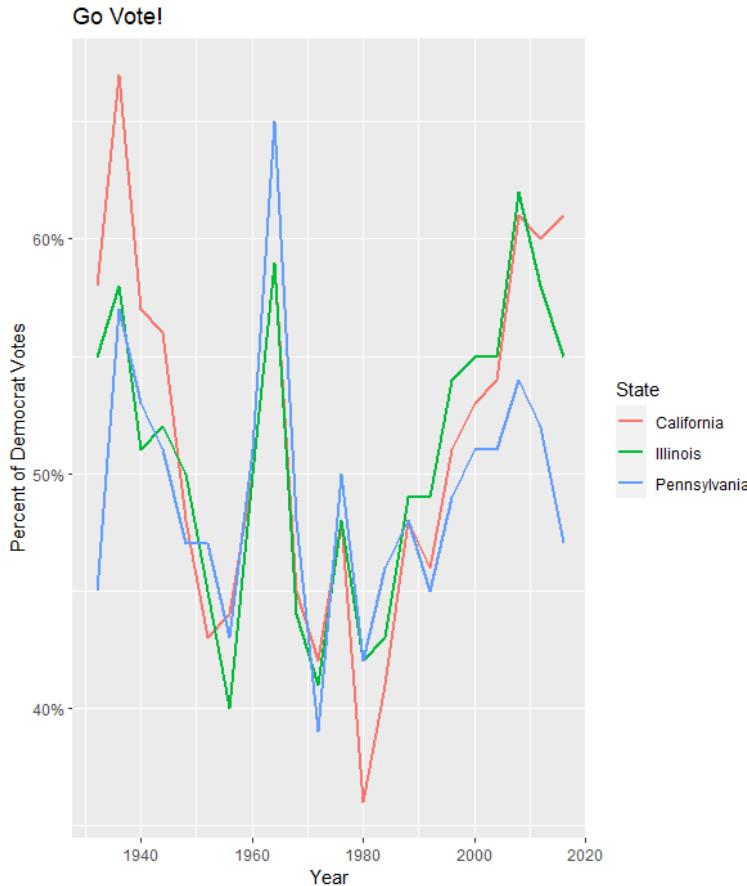
Go Vote!



Yay a plot!



Yay a plot... ?



A few problems

- Text is small and narrow
- Plot elements are squished together
- Color doesn't grab attention
- Legend is off to the side on its own

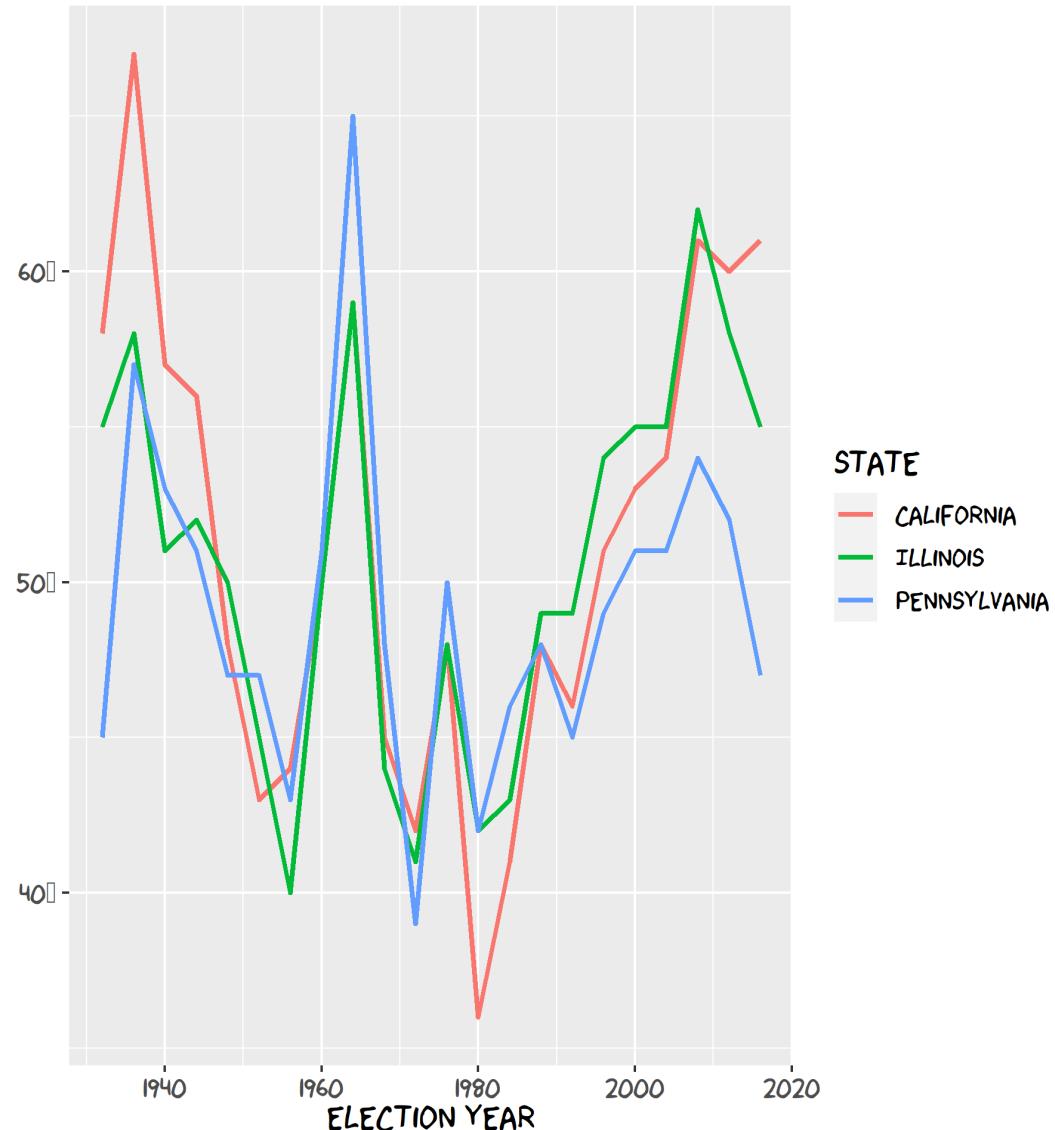
```

state_election_plot +
  theme(text = element_text(family = "xkcd")) +
  theme(text = element_text(size = 15)) +
  theme(plot.title = element_text(size = 20)) +
  theme(plot.title = element_text(family = "Roboto Slab")) +
  theme(plot.title.position = "plot") +
  labs(x = "Election Year") +
  labs(y = NULL) +
  labs(title = "Percent of democrat votes by state") +
  labs(subtitle = "We're a swing state! Go vote!") +
  theme(plot.subtitle = element_text(face = "italic"))

```

Percent of democrat votes by state

WE'RE A SWING STATE! GO VOTE!



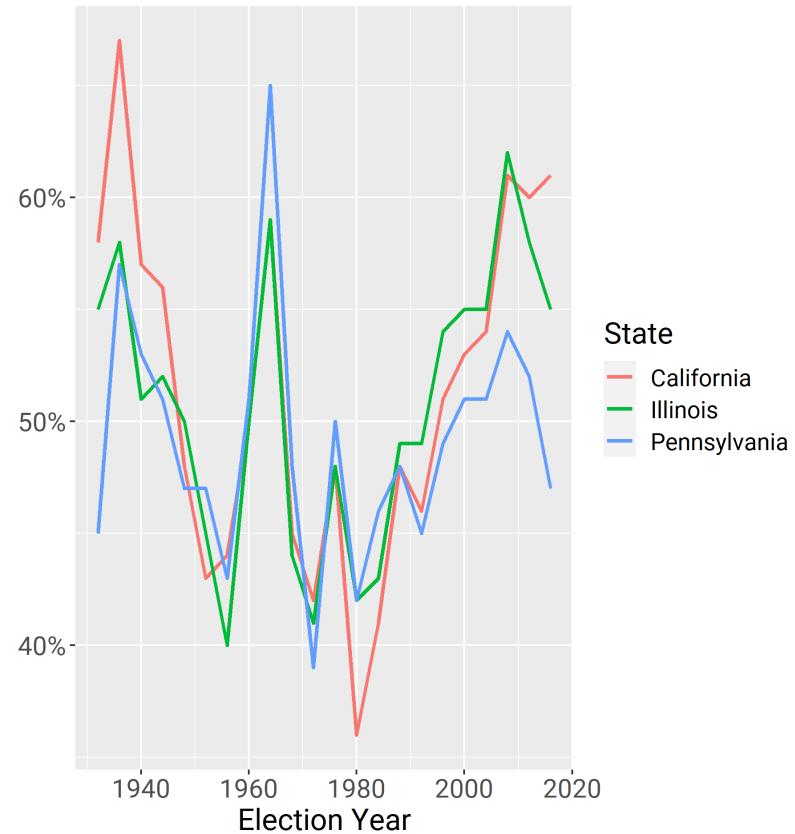
1. Text (End!)

```
state_election_plot +  
  theme(  
    text = element_text(size = 16, family = "Roboto"),  
    plot.title = element_text(size = 20, family = "Roboto"),  
    plot.title.position = "plot",  
    plot.subtitle = element_text(face = "italic"),  
    axis.text = element_text(size = 14)  
) +  
  labs(  
    x = "Election Year",  
    y = NULL,  
    title = "Percent of democrat votes by state",  
    subtitle = "We're a swing state! Go vote!"  
)
```

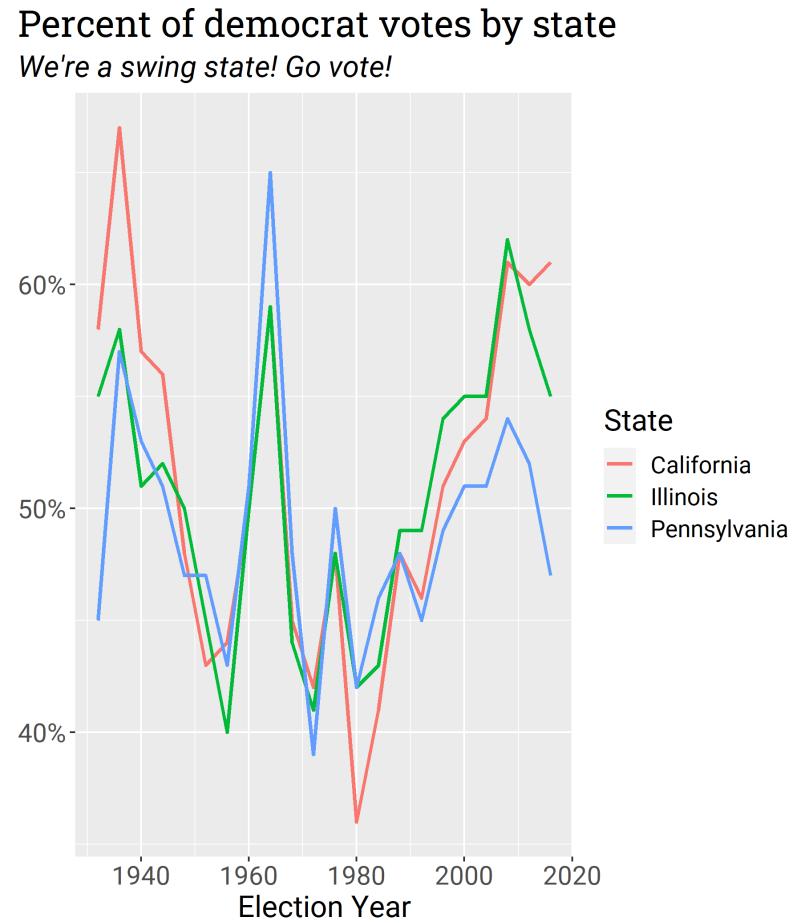
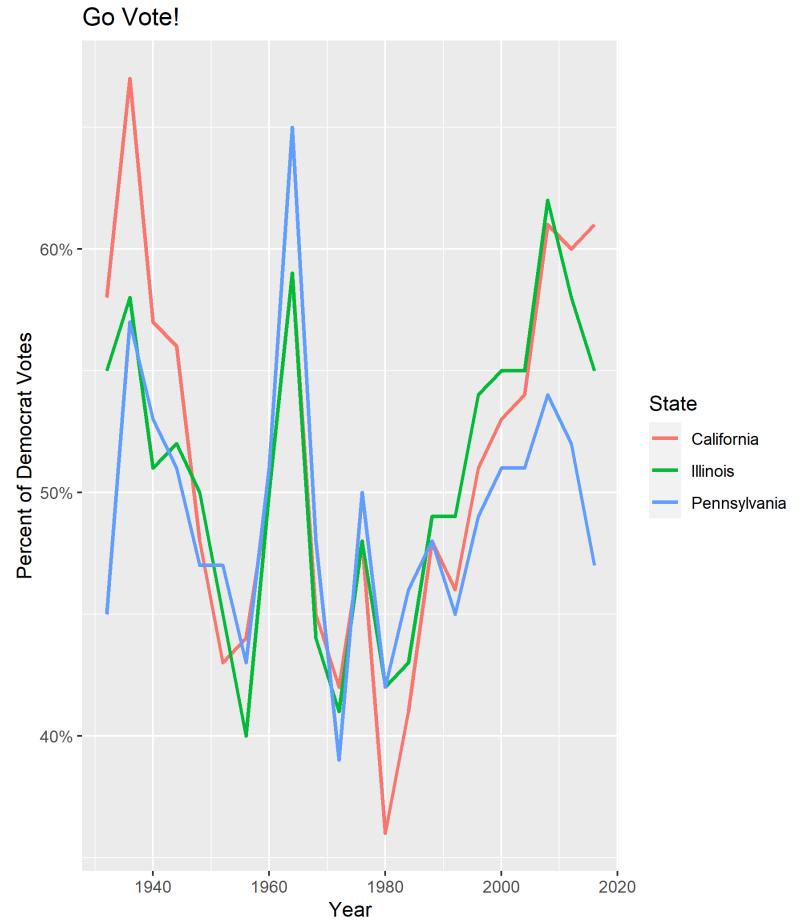
Save our progress!

```
state_election_plot_A
```

Percent of democrat votes by state
We're a swing state! Go vote!

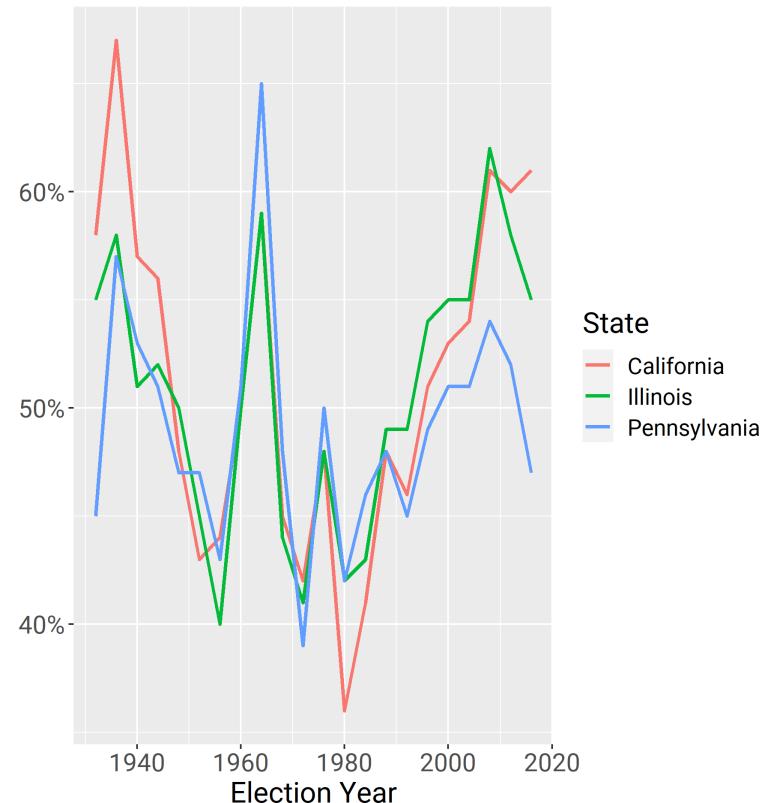


1. Text (Before-After)

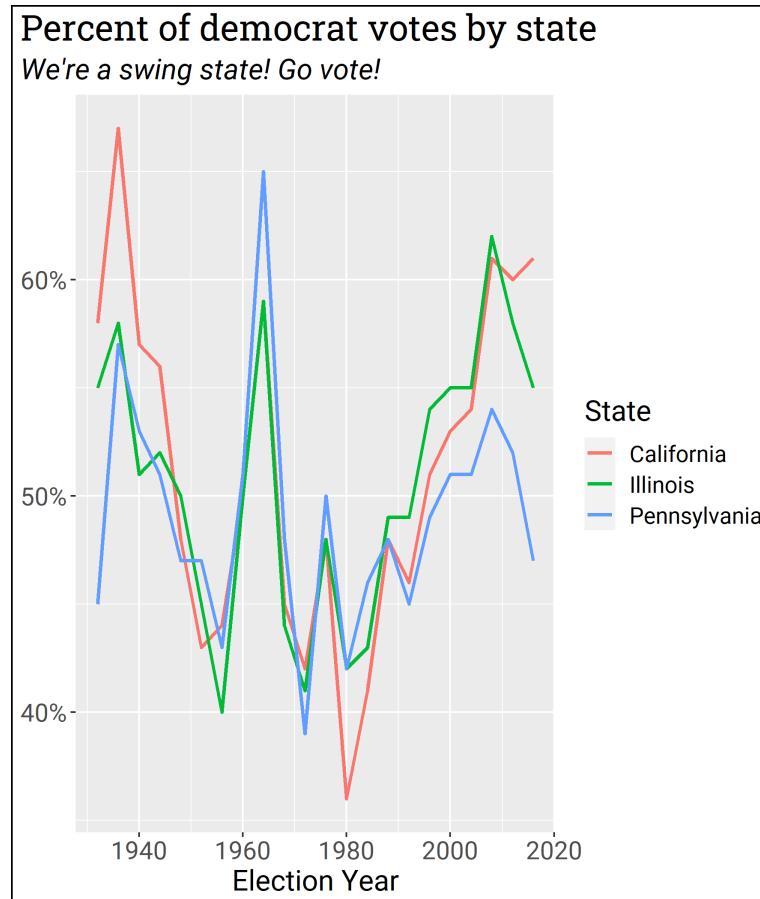


2. Margins & Spacing

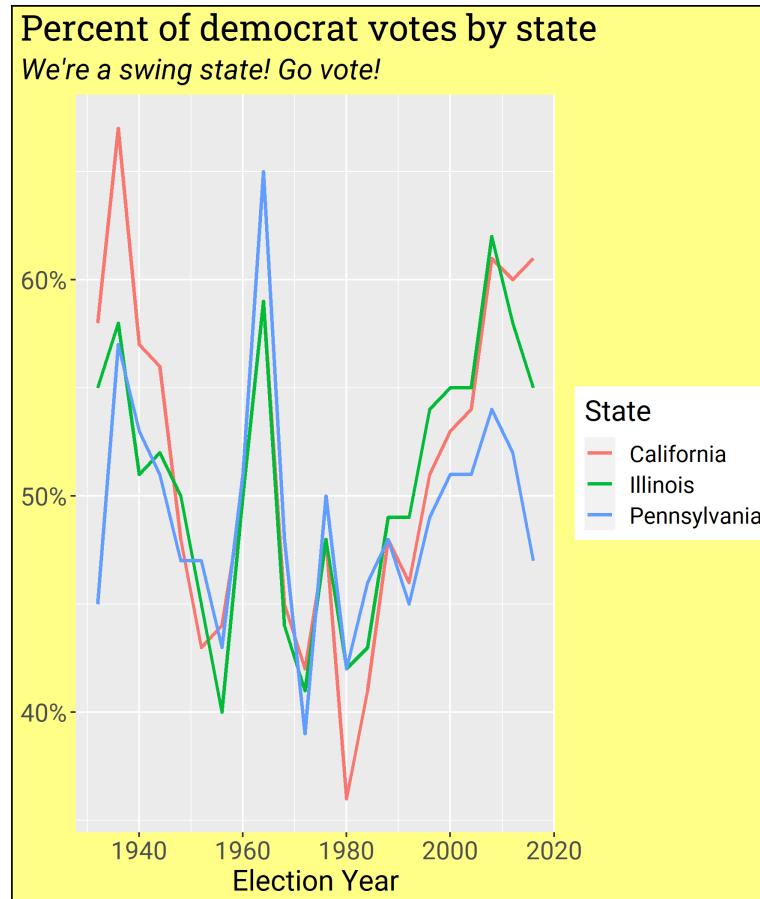
Percent of democrat votes by state
We're a swing state! Go vote!



2. Margins & Spacing



2. Margins & Spacing



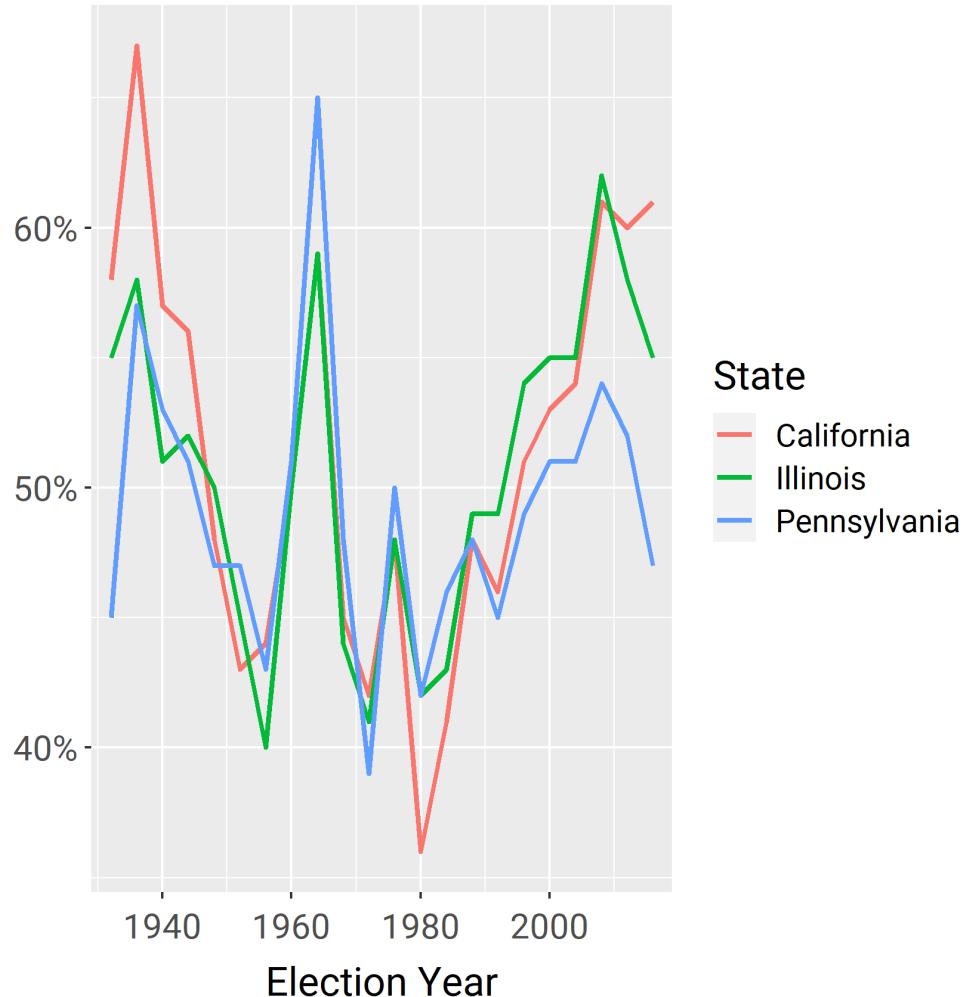
```

state_election_plot_A +
  theme(
    plot.background = element_rect(color = 'black')
  ) +
  theme(
    plot.margin = margin(1, .8, .8, .8, "cm")
  ) +
  theme(
    plot.title = element_text(margin = margin(b = .3, unit =
  ) +
  theme(
    plot.subtitle = element_text(margin = margin(b = .3, unit =
  ) +
  theme(
    axis.text.x = element_text(margin = margin(t = .2, unit =
  ) +
  theme(
    axis.text.y = element_text(margin = margin(r = .1, unit =
  ) +
  theme(
    axis.title.x = element_text(margin = margin(t = .3, unit =
  ) +
  scale_x_continuous(expand = expansion(mult = 0, add = 3),

```

Percent of democrat votes by state

We're a swing state! Go vote!



State

- California
- Illinois
- Pennsylvania

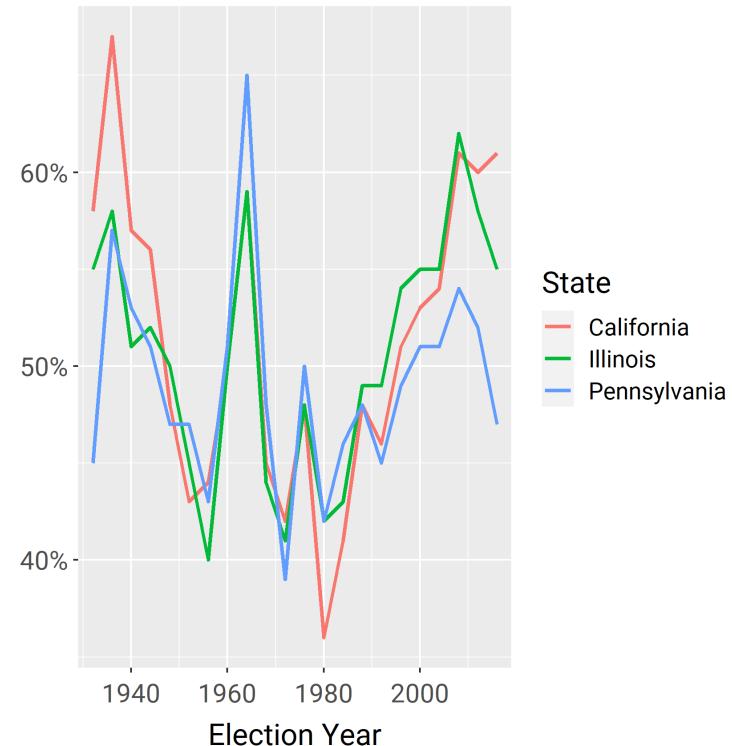
2. Margins & Spacing (End!)

```
state_election_plot_A +  
  theme(  
    plot.margin = margin(1, .8, .8, .8, "cm"),  
    plot.title = element_text(margin = margin(b = .3, t = .3, l = .3, r = .3)),  
    plot.subtitle = element_text(margin = margin(b = .3, t = .3, l = .3, r = .3)),  
    axis.text.x = element_text(margin = margin(t = .2, b = .1, l = .1, r = .1)),  
    axis.text.y = element_text(margin = margin(r = .1, b = .1, l = .1, t = .1)),  
    axis.title.x = element_text(margin = margin(t = .3, b = .1, l = .1, r = .1)),  
  ) +  
  scale_x_continuous(  
    expand = expansion(mult = 0, add = 3),  
    breaks = pretty_breaks(5)  
  )
```

Save our progress!

```
state_election_plot_B
```

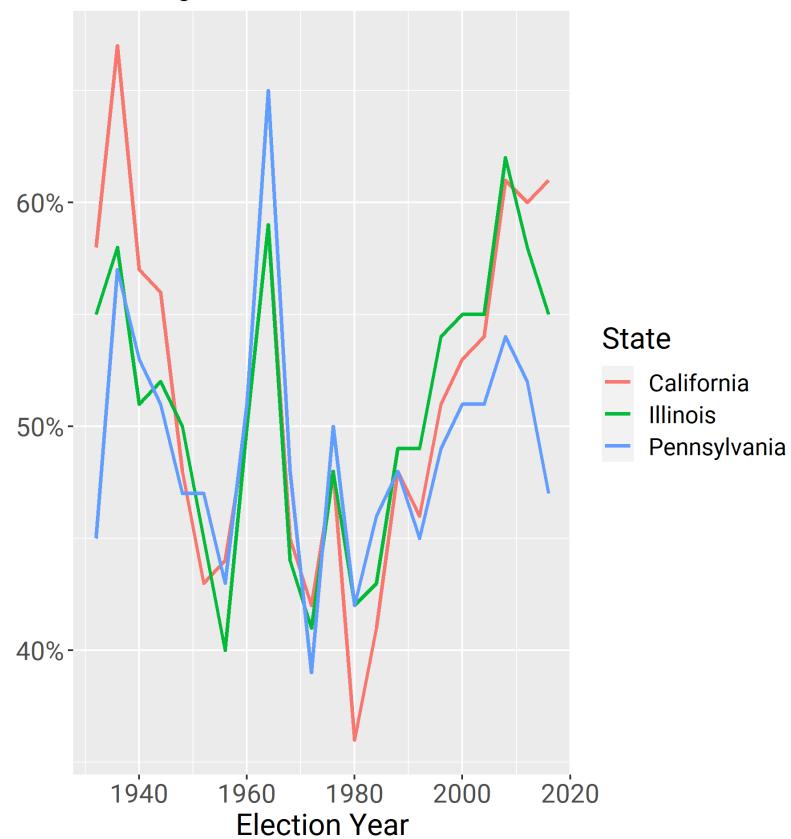
Percent of democrat votes by state
We're a swing state! Go vote!



2. Margins & Spacing (Before-After)

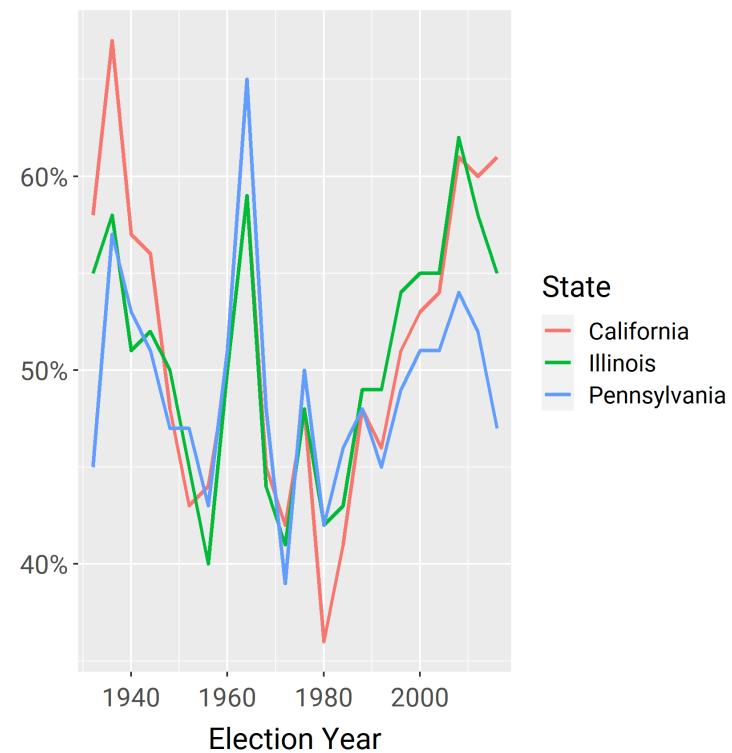
Percent of democrat votes by state

We're a swing state! Go vote!



Percent of democrat votes by state

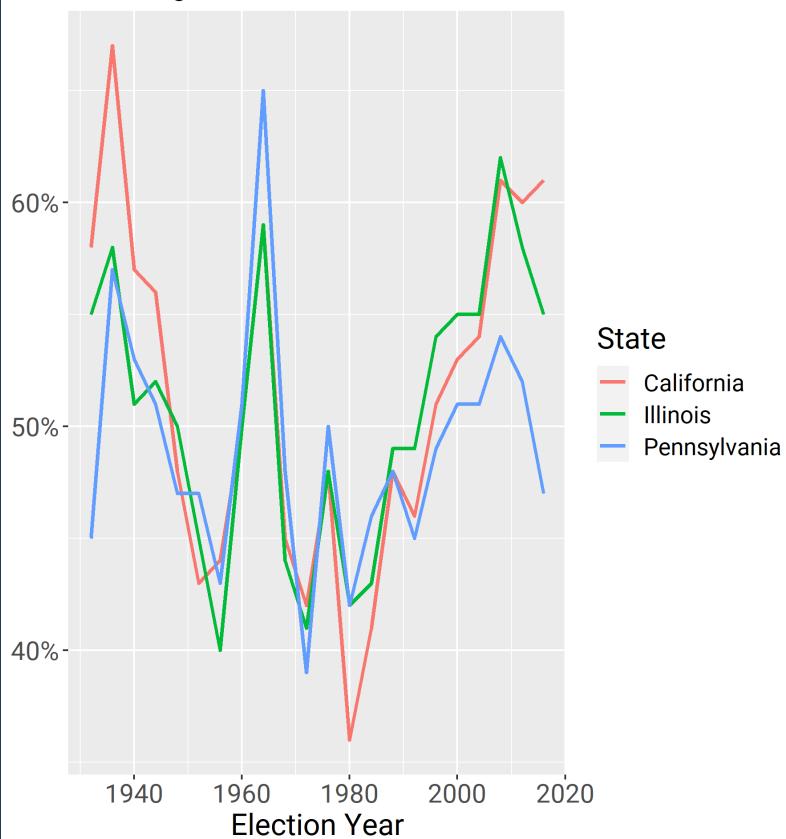
We're a swing state! Go vote!



2. Margins & Spacing (Before-After)

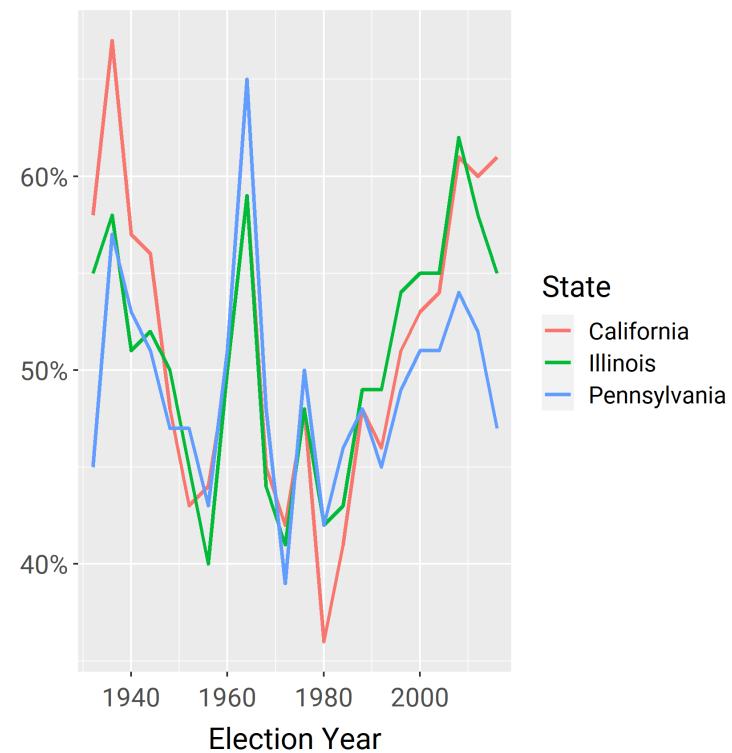
Percent of democrat votes by state

We're a swing state! Go vote!



Percent of democrat votes by state

We're a swing state! Go vote!



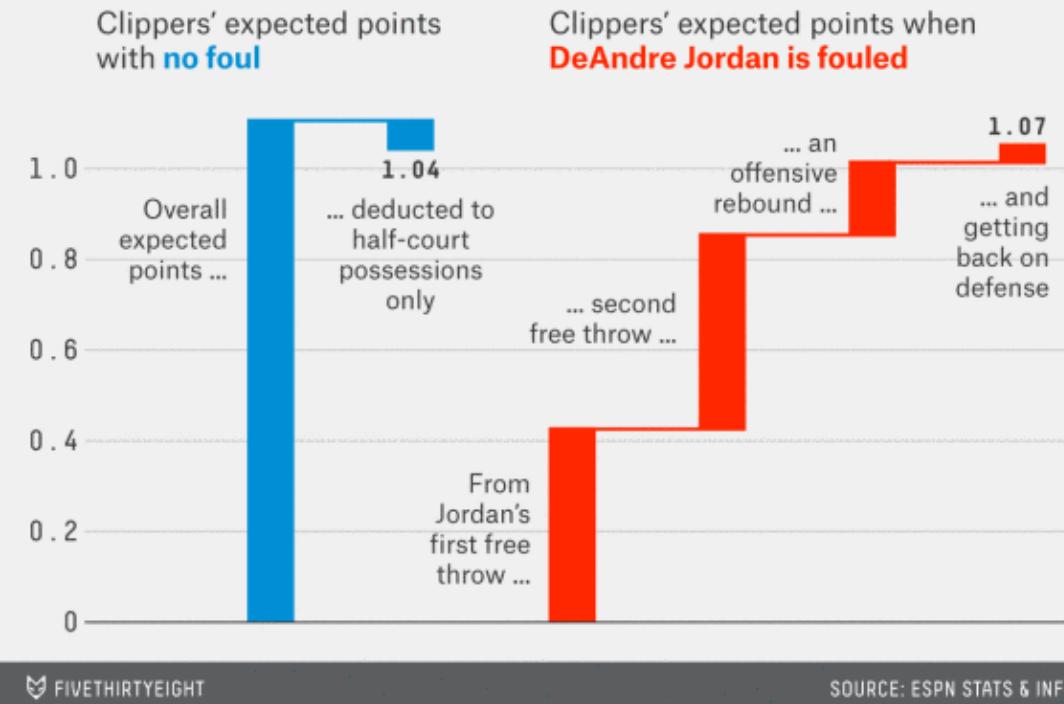
3. Legends

Legends are really hard:

- They contain important info, but difficult to make them not look out of place
- It's better not to have a legend, if you can get away with it
- But sometimes we don't have the luxury of doing so

Alternative: consider *labeling the data directly*

To Hack or Not to Hack?



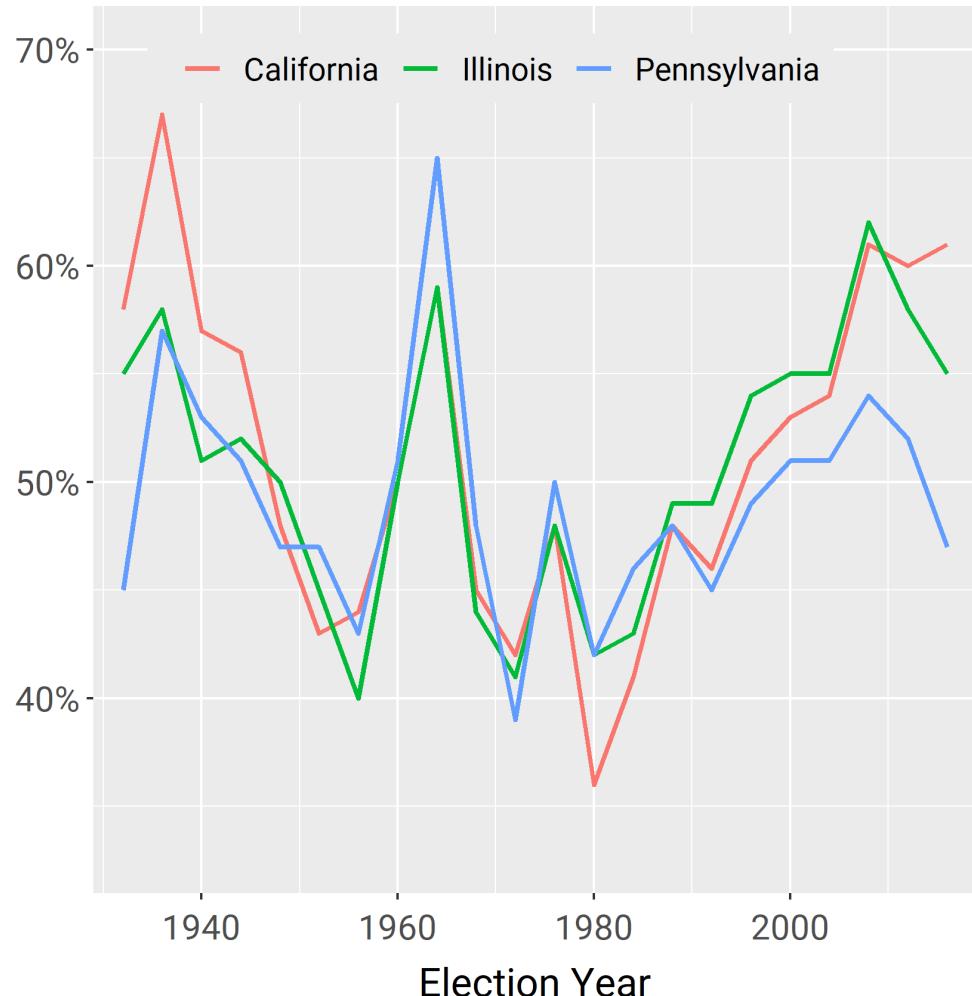
```

state_election_plot_B +
  theme(
    legend.key = element_rect(fill = NA)
  ) +
  theme(
    legend.position = c(.45, .93)
  ) +
  theme(
    legend.direction = "horizontal"
  ) +
  theme(
    legend.background = element_rect(fill = "grey92")
  ) +
  scale_y_continuous(
    expand = expansion(0, .05),
    labels = percent_format(accuracy = 1)
  ) +
  labs(color = NULL)

```

Percent of democrat votes by state

We're a swing state! Go vote!



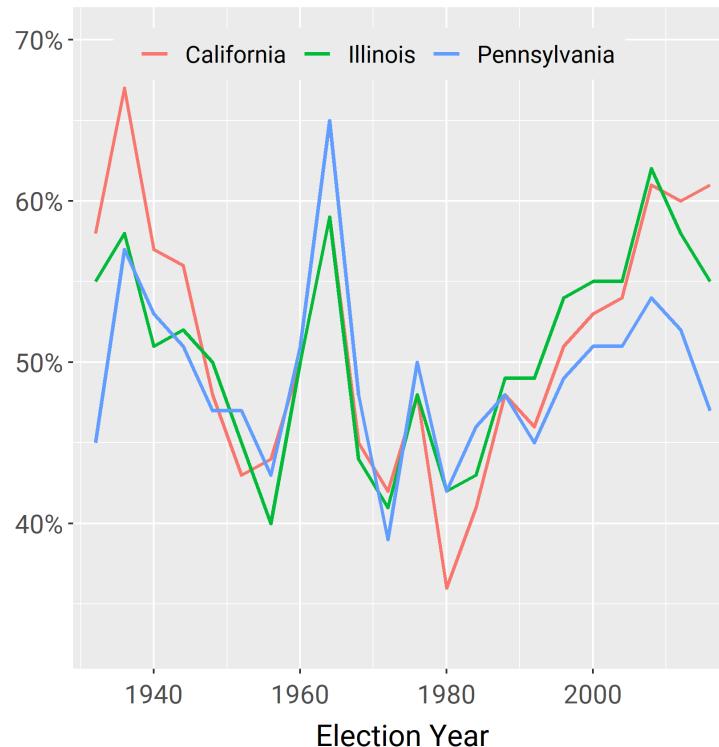
3. Legends (End!)

```
state_election_plot_B +  
  theme(  
    legend.key = element_rect(fill = NA),  
    legend.position = c(.45, .93),  
    legend.direction = "horizontal",  
    legend.background = element_rect(fill = "grey92"))  
  ) +  
  scale_y_continuous(  
    expand = expansion(0, .05),  
    labels = percent_format(accuracy = 1)  
  ) +  
  labs(color = NULL)
```

Save our progress!

```
state_election_plot_C
```

Percent of democrat votes by state
We're a swing state! Go vote!



3. Legends (Before-After)

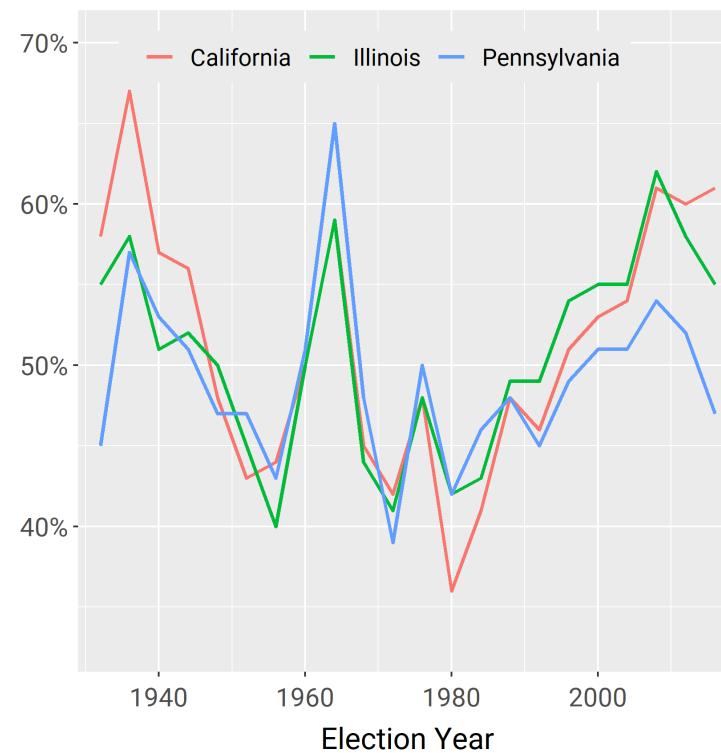
Percent of democrat votes by state

We're a swing state! Go vote!



Percent of democrat votes by state

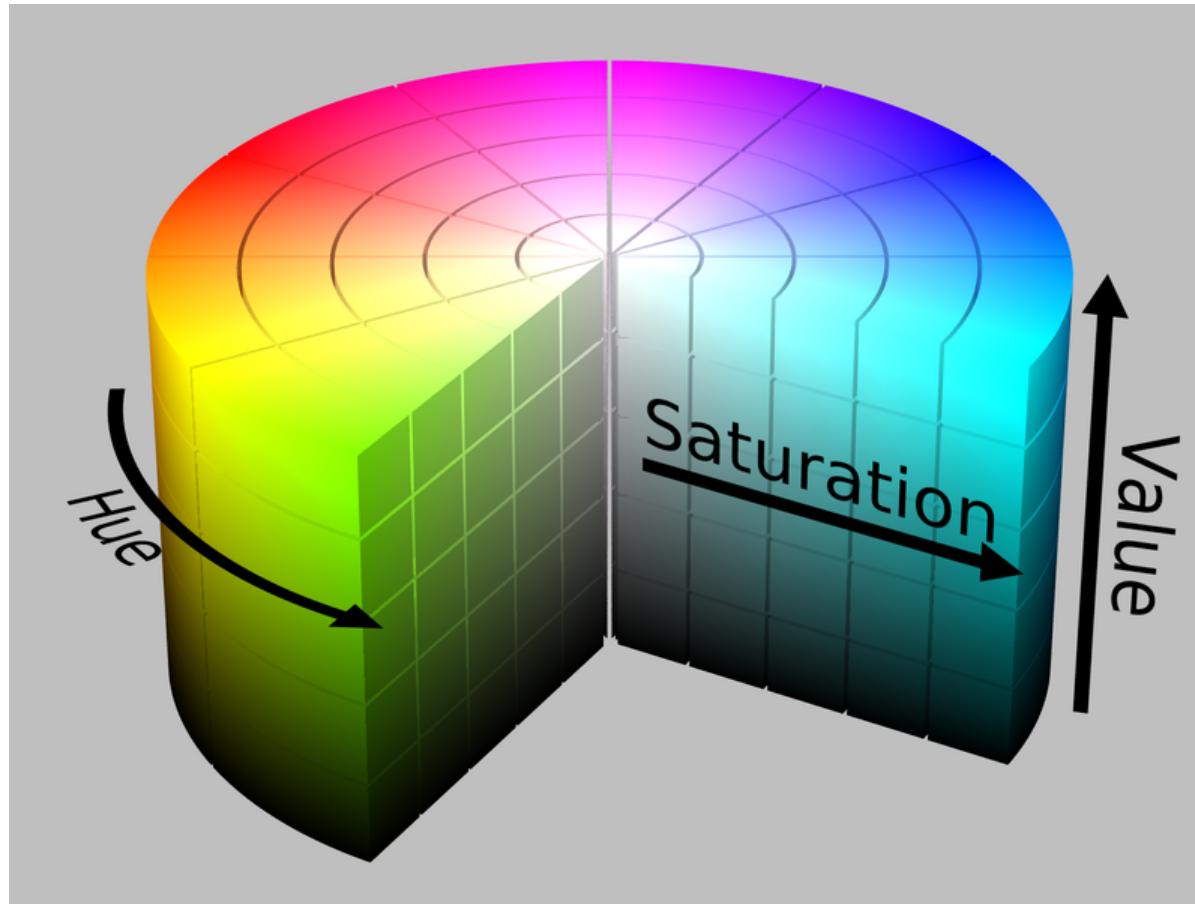
We're a swing state! Go vote!



4. Color

Colors are a double-edged sword

- Perception can vary widely depending on reader, medium, culture, etc.
- There are multiple, complex representations (RGB, wavelength, hex, [HSV](#))



4. Color

Colors are a double-edged sword

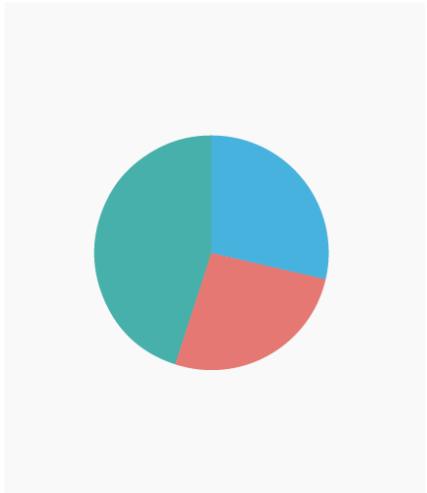
- Perception can vary widely depending on reader, medium, culture, etc.
- There are multiple, complex representations (RGB, wavelength, hex, [HSV](#))

If you must, DO:

- Avoid pure colors (no random sampling from the rainbow!)
- Contrast colors in more than one dimension



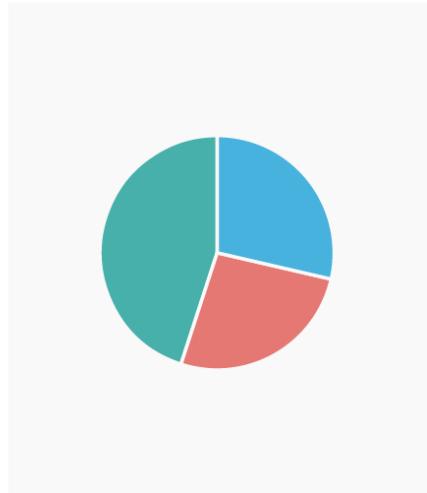
source: <https://blog.datawrapper.de/beautifulcolors/>



NOT IDEAL



BETTER



BETTER

source: <https://blog.datawrapper.de/beautifulcolors/>

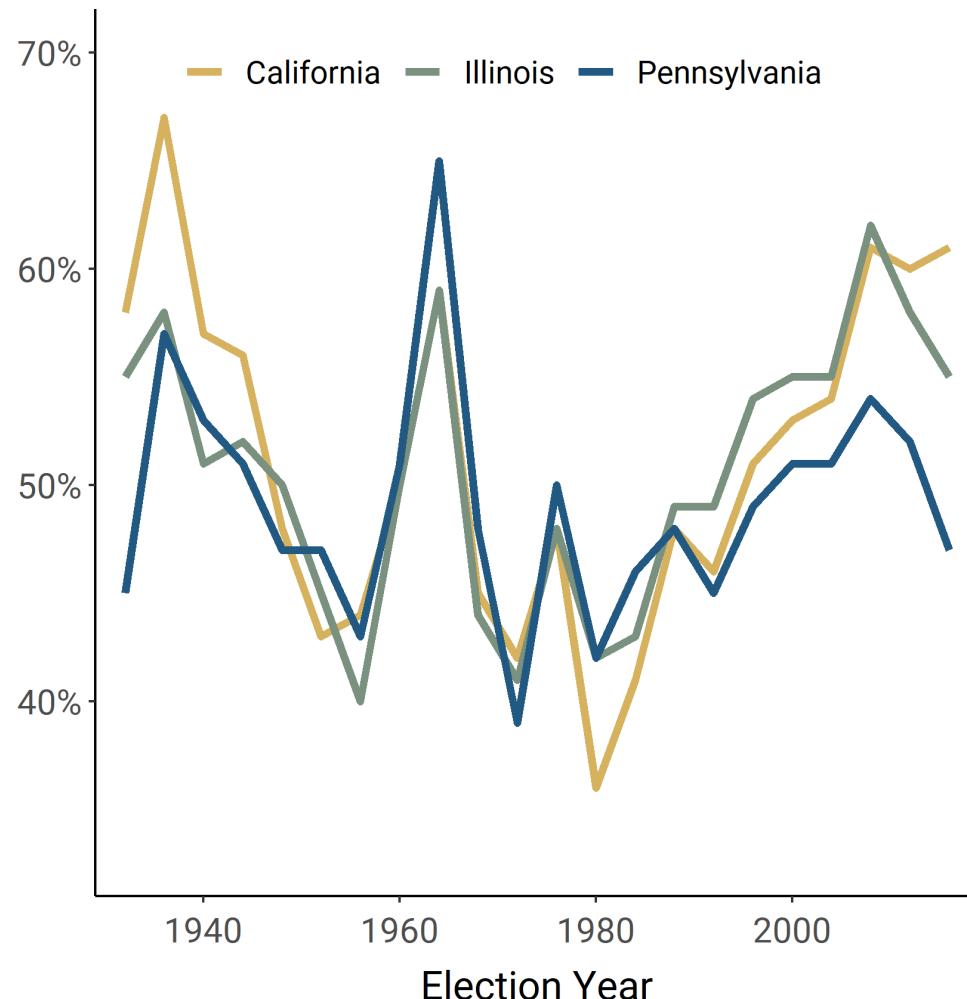
```

state_election_plot_C +
  theme(panel.background = element_rect(fill = NA)) +
  theme(legend.background = element_rect(fill = NA)) +
  theme(axis.line = element_line()) +
  scale_color_manual(
    values = c("#F8766D", "#00BA38", "#619cff")
  ) +
  scale_color_manual(
    values = c(
      hex(HSV( 4, 0.56, 0.97)),
      hex(HSV(138, 1, 0.73)),
      hex(HSV(218, 0.62, 1))
    )
  ) +
  scale_color_manual(
    values = c(
      hex(HSV( 0, 1, 1)),
      hex(HSV(120, 1, 1)),
      hex(HSV(210, 1, 1))
    )
  ) +
  scale_color_manual(
    values = c(
      hex(HSV( 42, 0.56, 0.84)),
      hex(HSV(133, 0.17, 0.57)),
      hex(HSV(205, 0.75, 0.51))
    )
  ) +
  scale_color_manual(
    values = c("#D6B25E", "#79917E", "#215982")
  ) +
  geom_line(size = 1.5)

```

Percent of democrat votes by state

We're a swing state! Go vote!



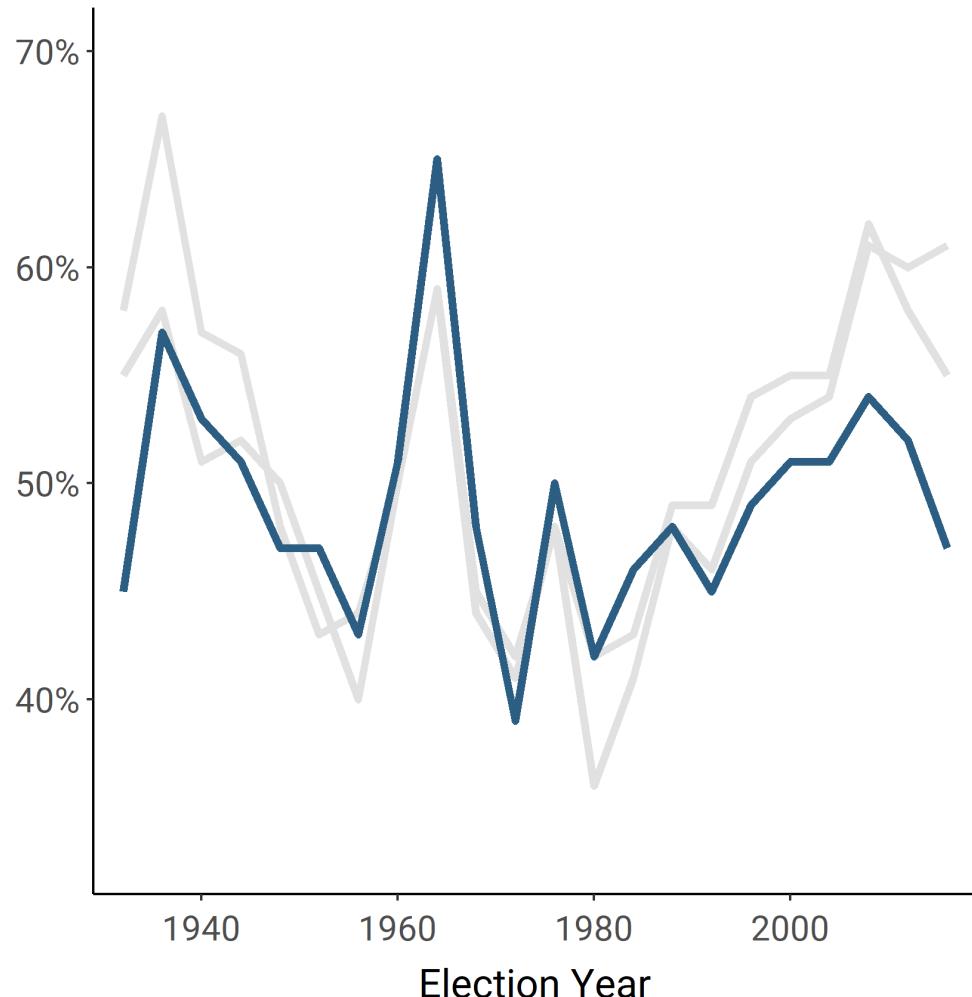
```

state_election_plot_C +
  theme(
    panel.background = element_rect(fill = NA)
  ) +
  theme(
    legend.background = element_rect(fill = NA)
  ) +
  theme(
    axis.line = element_line()
  ) +
  scale_color_manual(
    values = c("#e1e1e1", "#e1e1e1", "#2b5e82")
  ) +
  guides(
    color = guide_none()
  ) +
  labs(subtitle = "<strong style='color:#0E4369'>Pennsylvania</strong> is a swing state! Go vote!")
  theme(
    plot.subtitle = ggtext::element_markdown()
  ) +
  geom_line(size = 1.5)

```

Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!



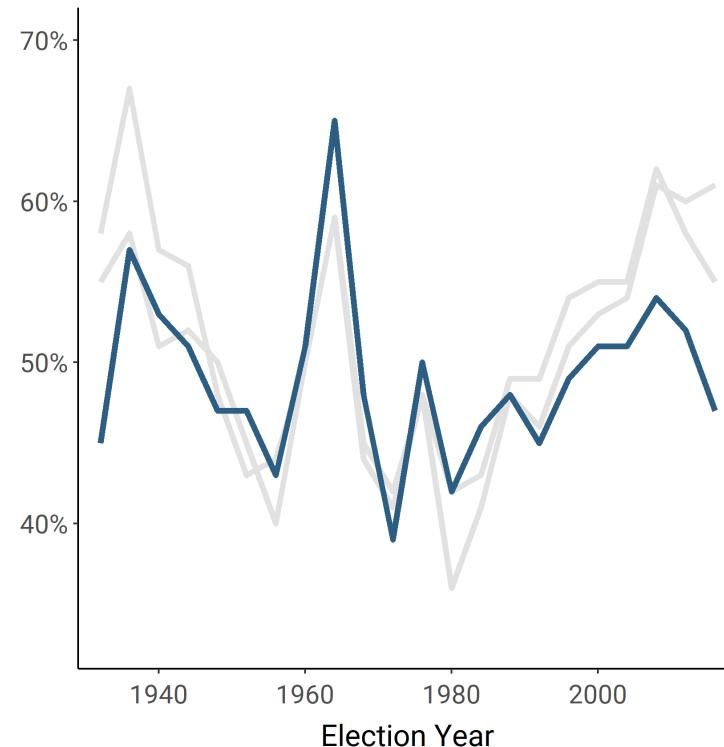
4. Colors (End!)

```
state_election_plot_C +  
  geom_line(size = 1.5) +  
  theme(  
    panel.background = element_rect(fill = NA),  
    legend.background = element_rect(fill = NA),  
    axis.line = element_line(),  
    plot.subtitle = element_markdown()  
) +  
  guides(color = guide_none()) +  
  scale_color_manual(values = c("#e1e1e1", "#e1e1e1", '  
  labs(  
    subtitle = "<strong style='color:#0E4369'>Pennsylvania  
      is a swing state! Go vote!"  
)
```

Save our progress!

```
state_election_plot_D
```

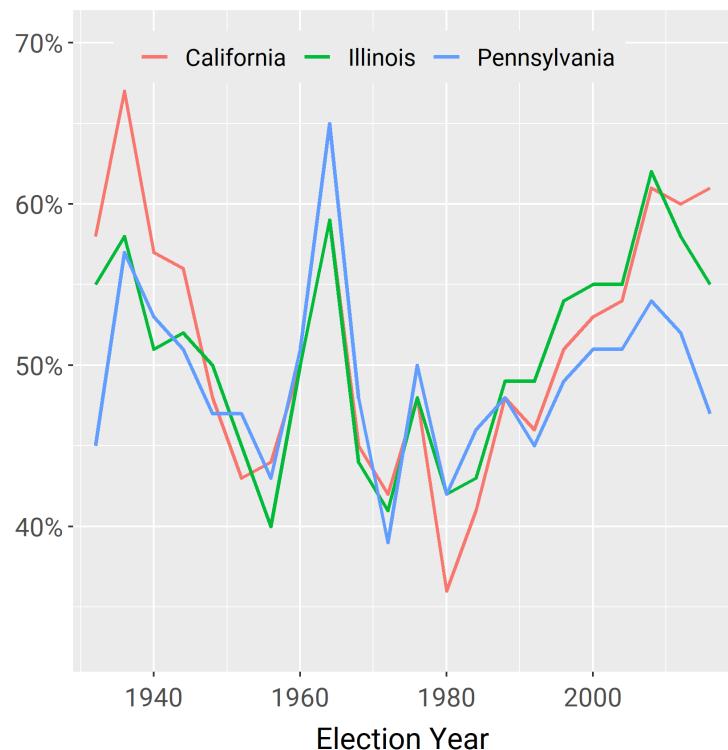
Percent of democrat votes by state
Pennsylvania is a swing state! Go vote!



4. Colors (Before-After)

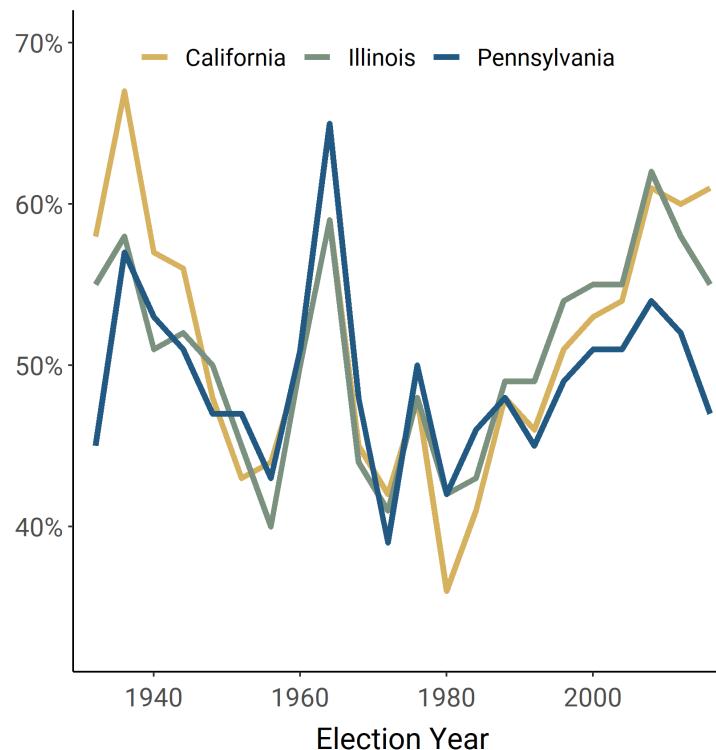
Percent of democrat votes by state

We're a swing state! Go vote!



Percent of democrat votes by state

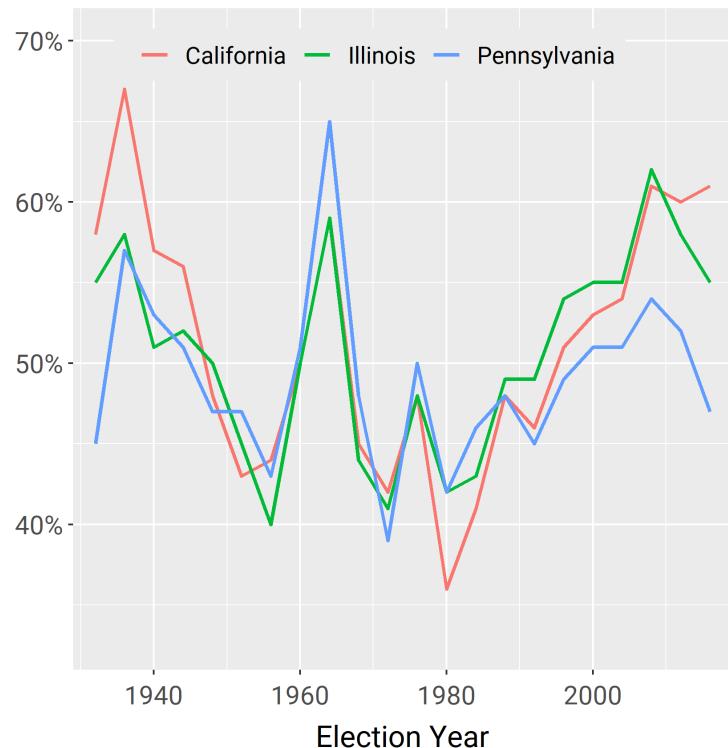
We're a swing state! Go vote!



4. Colors (Before-After)

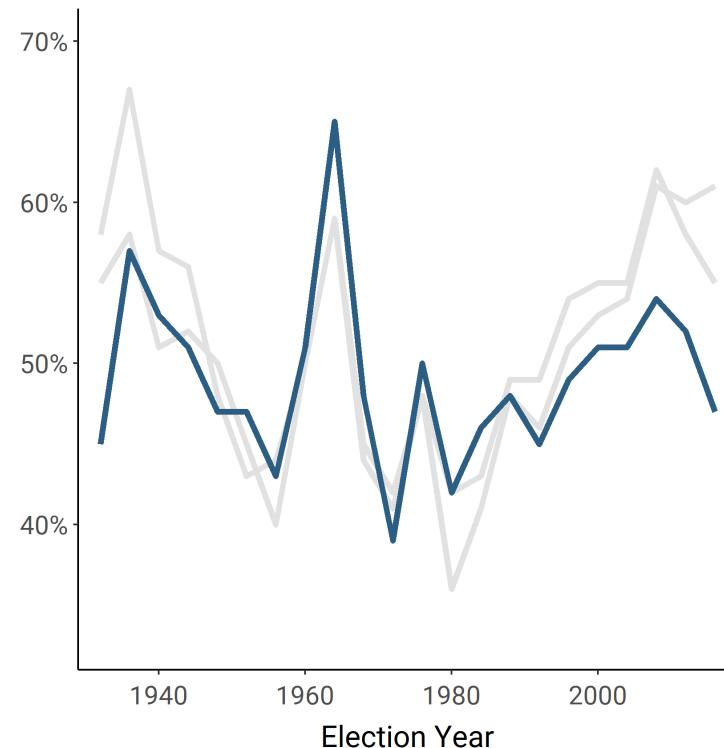
Percent of democrat votes by state

We're a swing state! Go vote!



Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!



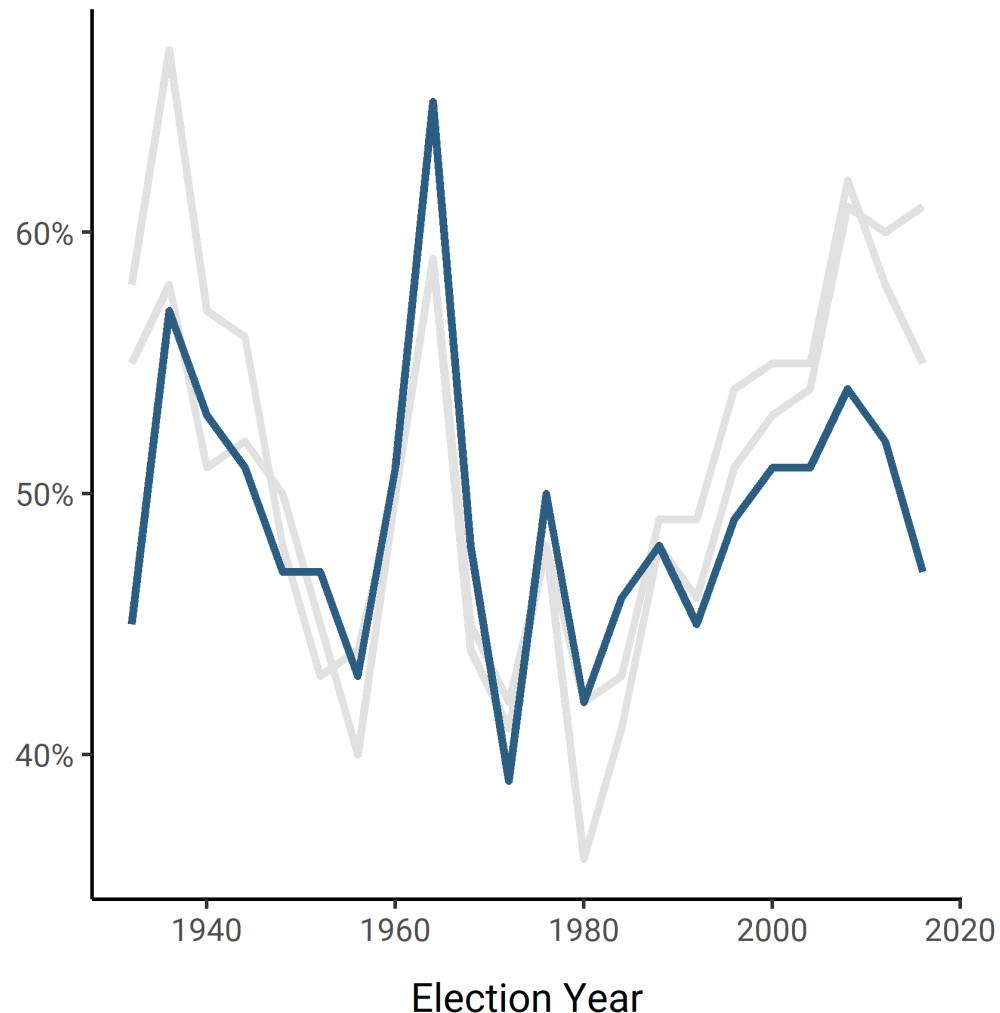
```

state_election_plot +
  theme_classic(
    base_family = "Roboto",
    base_size = 16
  ) +
  guides(color = guide_none()) +
  geom_line(size = 1.5) +
  scale_color_manual(
    values = c("#e1e1e1", "#e1e1e1", "#2b5e82")
  ) +
  labs(
    y = NULL,
    x = "Election Year",
    title = "Percent of democrat votes by state",
    subtitle = "<strong style='color:#0E4369'>Pennsylvania</strong>"
  ) +
  theme(
    plot.margin = margin(.8, 1, .7, .8, "cm"),
    plot.title = element_text(
      family = "Roboto Slab",
      size = 24,
      margin = margin(b = .3, unit = "cm")
    ),
    plot.title.position = "plot",
    plot.subtitle = element_markdown(
      margin = margin(b = .3, unit = "cm")
    ),
    axis.title.x = element_text(
      margin = margin(t = .5, unit = "cm")
    )
  )
)

```

Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!



5(?). Plot quality: Resolution

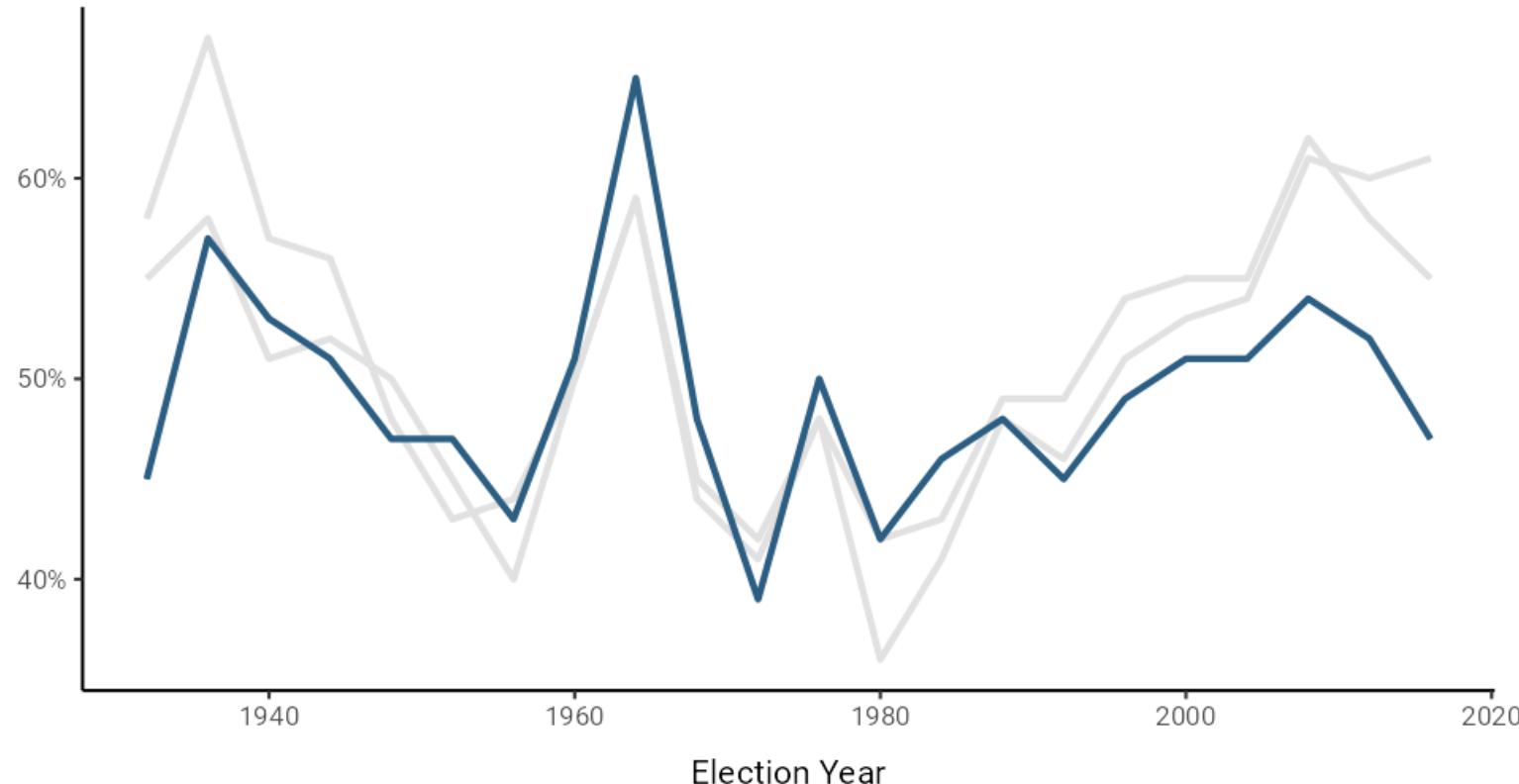
The screenshot shows the RStudio interface with several tabs open:

- Code tab:** Displays R code for a plot. The code includes setting up the plot with a manual color scale, defining a theme with a Roboto Slab font, and adding a subtitle about Pennsylvania being a swing state. It also includes a chunk reveal command and a final section for plot quality.
- Environment tab:** Shows the global environment with two objects: `state_election_plot` (a ggplot2 object) and `state_election_votes` (a data frame with 1097 observations and 3 variables).
- Plots tab:** Displays a single plot titled "Pennsylvania". The plot shows a map of Pennsylvania with various regions colored according to the democrat vote percentage. A legend on the right side of the plot indicates the color mapping for different vote percentages.
- Console tab:** Shows the R session history, including the execution of the code and the creation of the plot object.
- Terminal tab:** Shows the command used to run the R script.
- R Markdown tab:** Shows the R Markdown file structure.
- Jobs tab:** Shows the current jobs running.

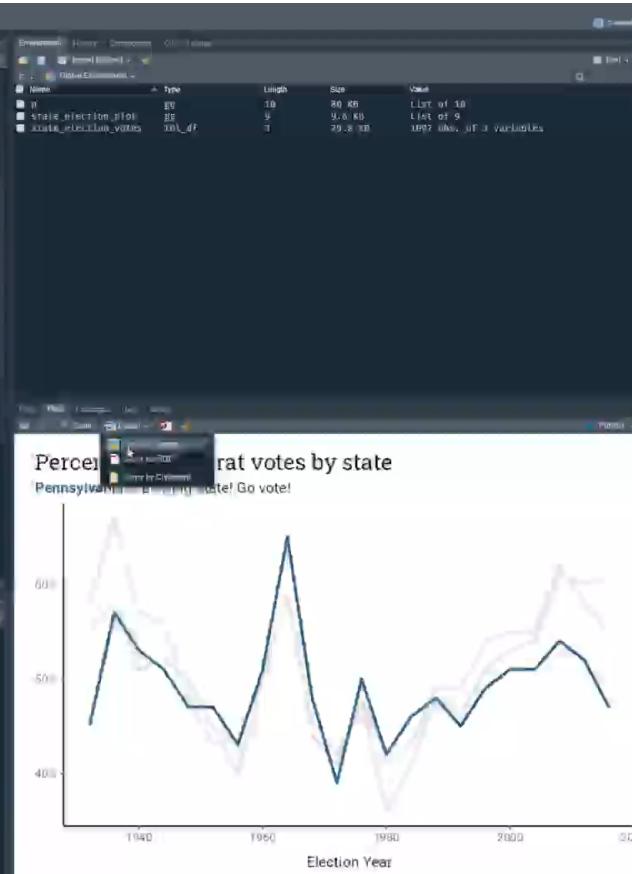
Point-and-click method in RStudio

Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!



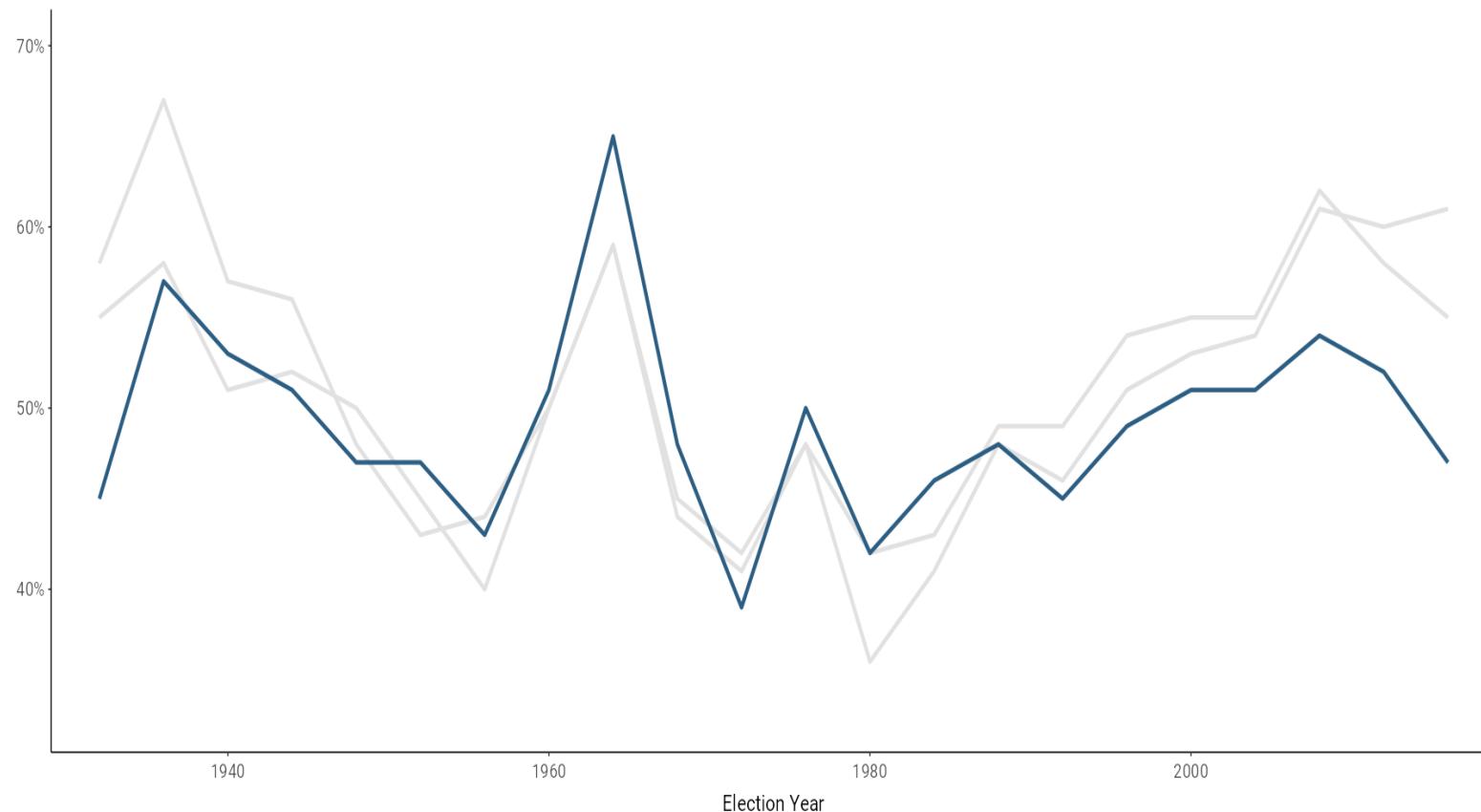
5(?). Plot quality: Scaling



Stretching in RStudio

Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!



5(?). Plot quality: Solution

Use the `ggsave()` function (only works for ggplot2 plots):

- Automatically configures sensible defaults
- Makes your figures fully reproducible!

```
ggsave("highres.png", p, width = 10, height = 6.2, units = "in", dpi = 300) # png format  
ggsave("highres.pdf", p, width = 10, height = 6.2, units = "in", device = cairo_pdf) # pdf format w/ Cairo
```

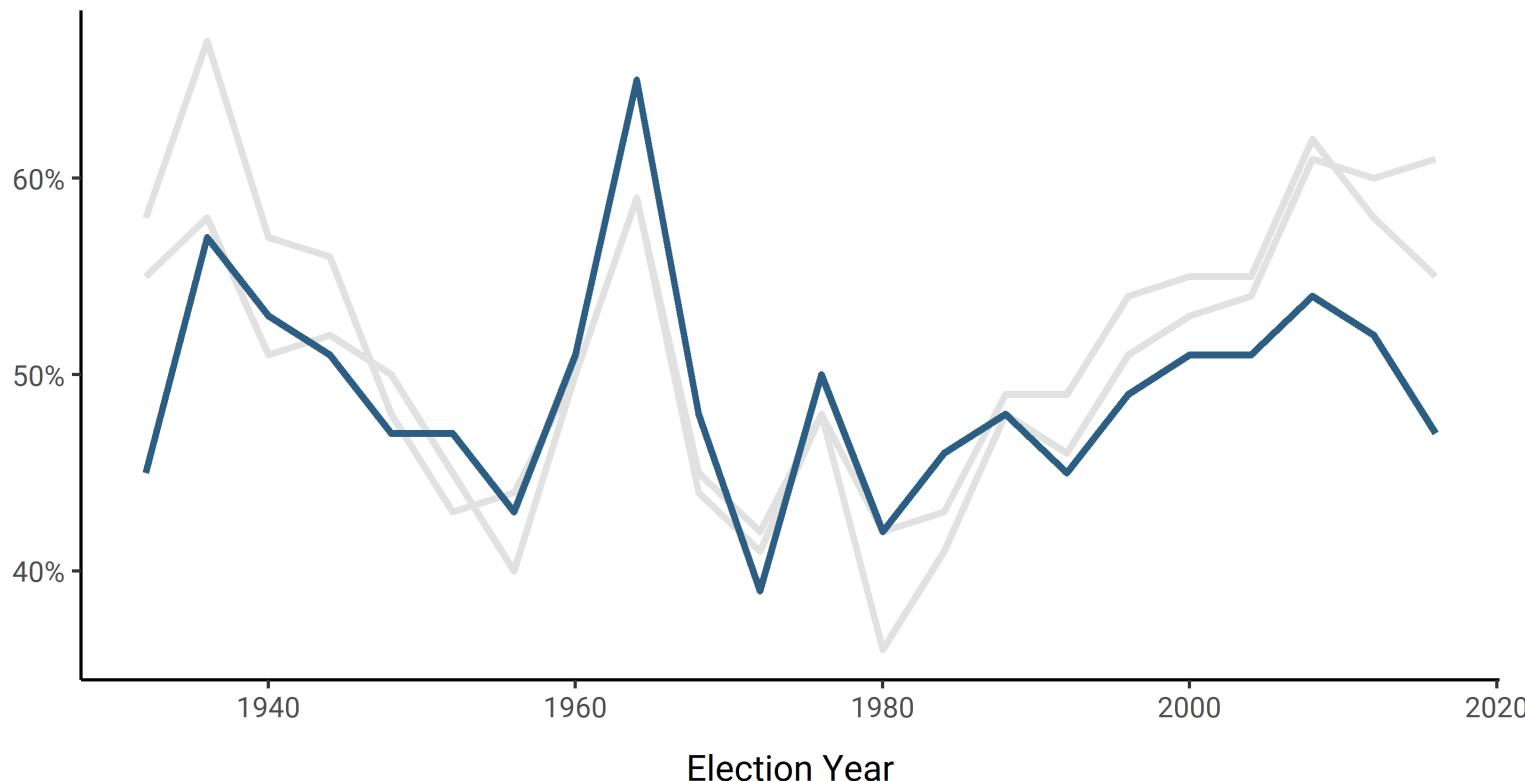
Use better-quality, OS-independent graphic devices (works for any figure):

```
library(ragg) # R interface to the AGG device - https://github.com/r-lib/ragg  
  
# Step 1: initialize device  
agg_png("highres_agg.png", width = 10, height = 6.2, units = "in", res = 300)  
  
# Step 2: plot  
plot(p)  
  
# Step 3: close device  
invisible(dev.off())
```

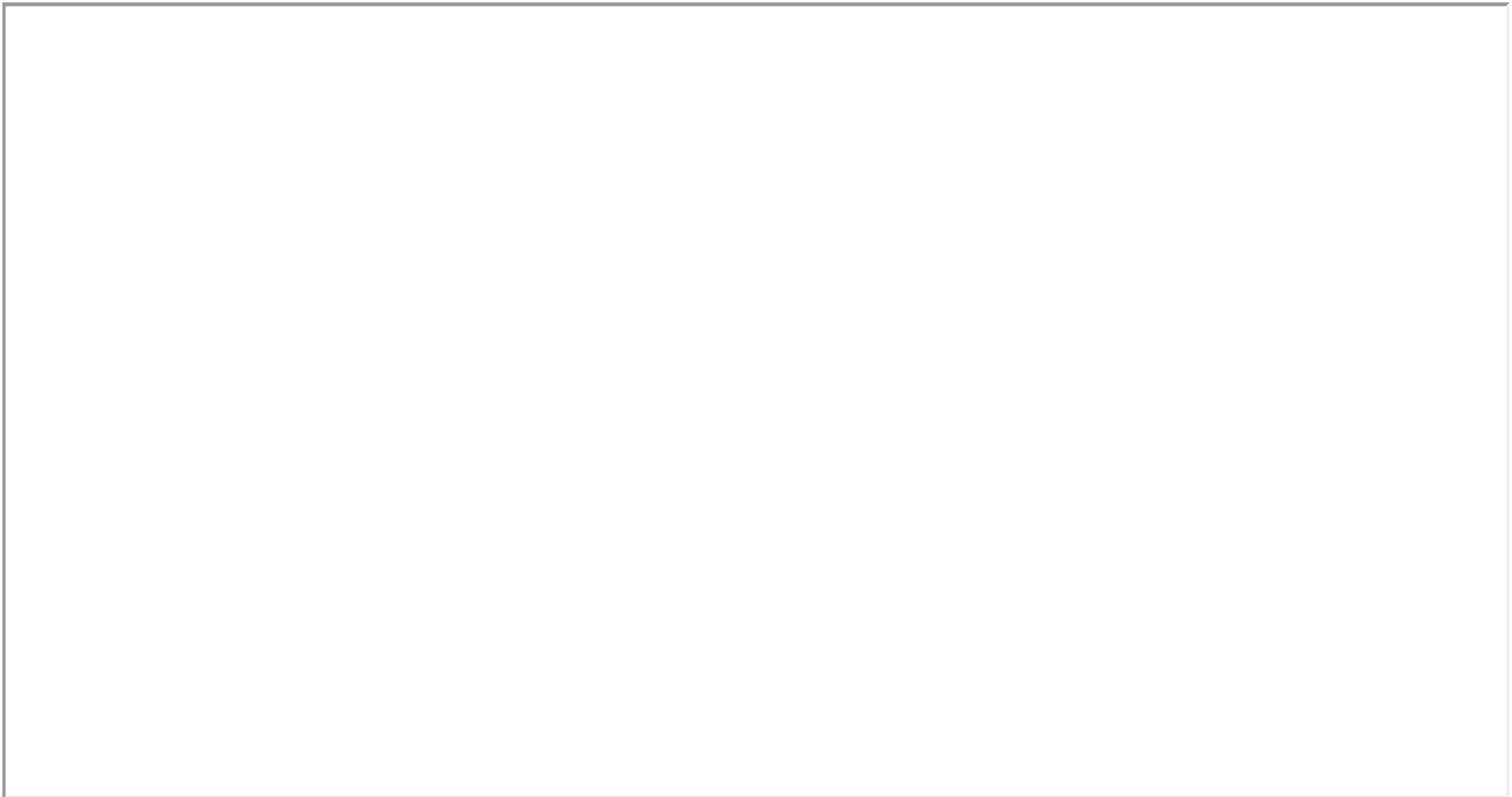
ggsave() png

Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!

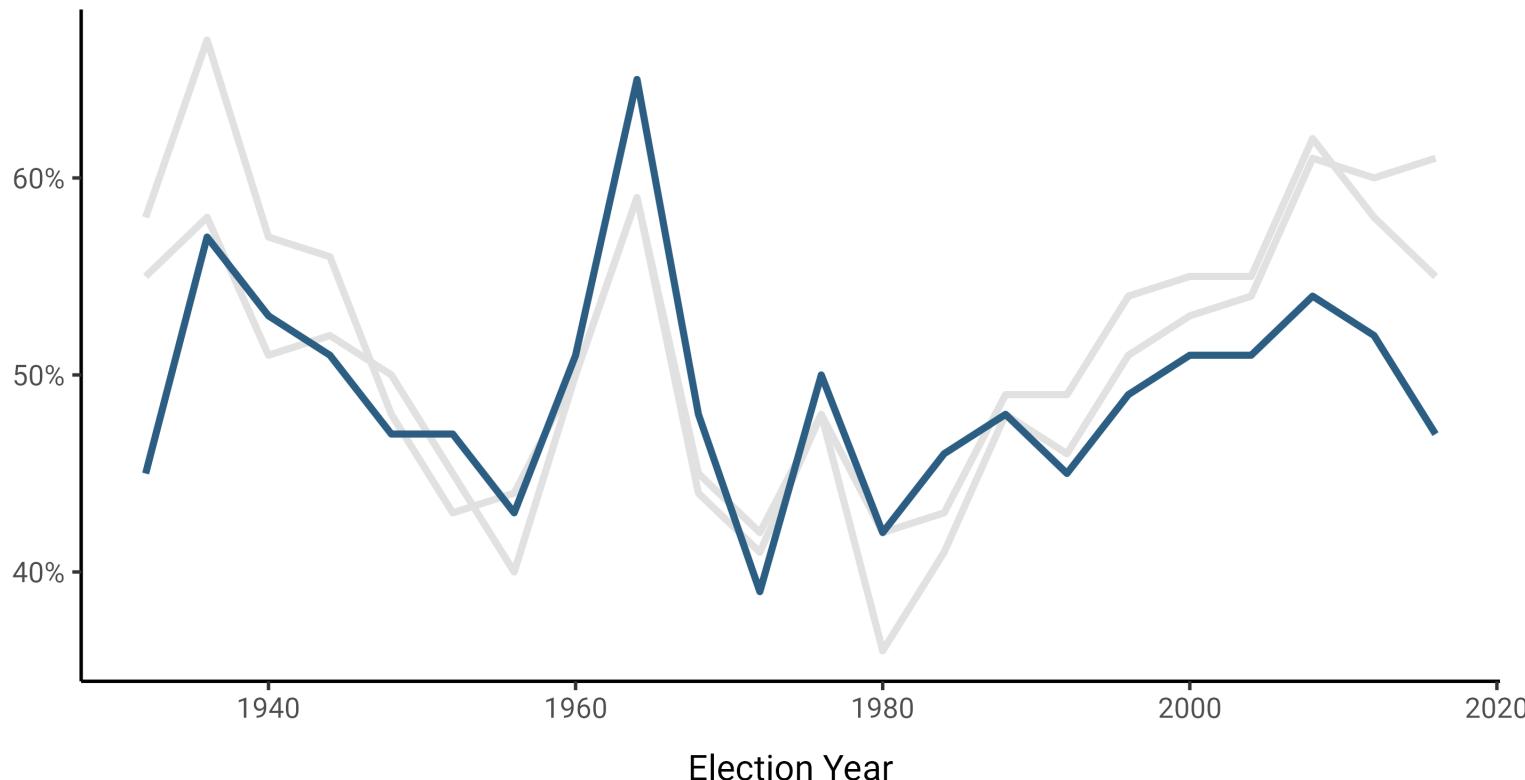


`ggsave()` pdf



Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!



Showcasing

First, save our theme!

You can set global theme with `theme_set()` and `theme_update()`:

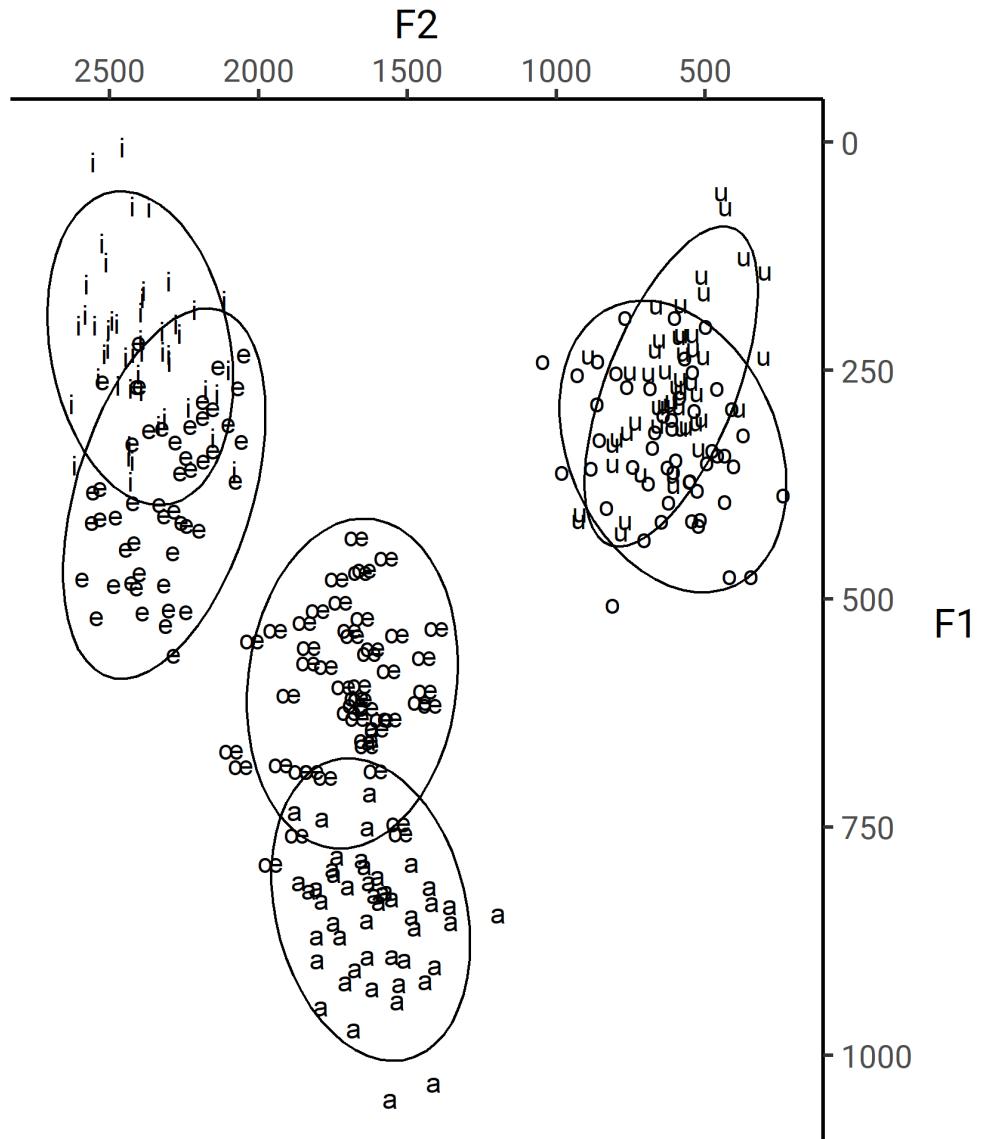
- `theme_set()` takes a custom theme as a argument (e.g., `theme_bw()`, `theme_classic()`, etc.)
- `theme_update()` takes individual theme elements as arguments
- `theme_get()` returns the current theme that's being used

```
theme_set(  
  theme_classic(  
    base_family = "Roboto",  
    base_size = 16  
  )  
)  
  
theme_update(  
  plot.margin = margin(.8, 1, .7, .8, "cm"),  
  plot.title = element_text(  
    family = "Roboto Slab",  
    size = 24,  
    margin = margin(b = .5, unit = "cm")  
  ),  
  plot.title.position = "plot",  
  axis.title.x = element_text(  
    margin = margin(t = .5, unit = "cm")  
  )  
)  
  
# theme_get()
```

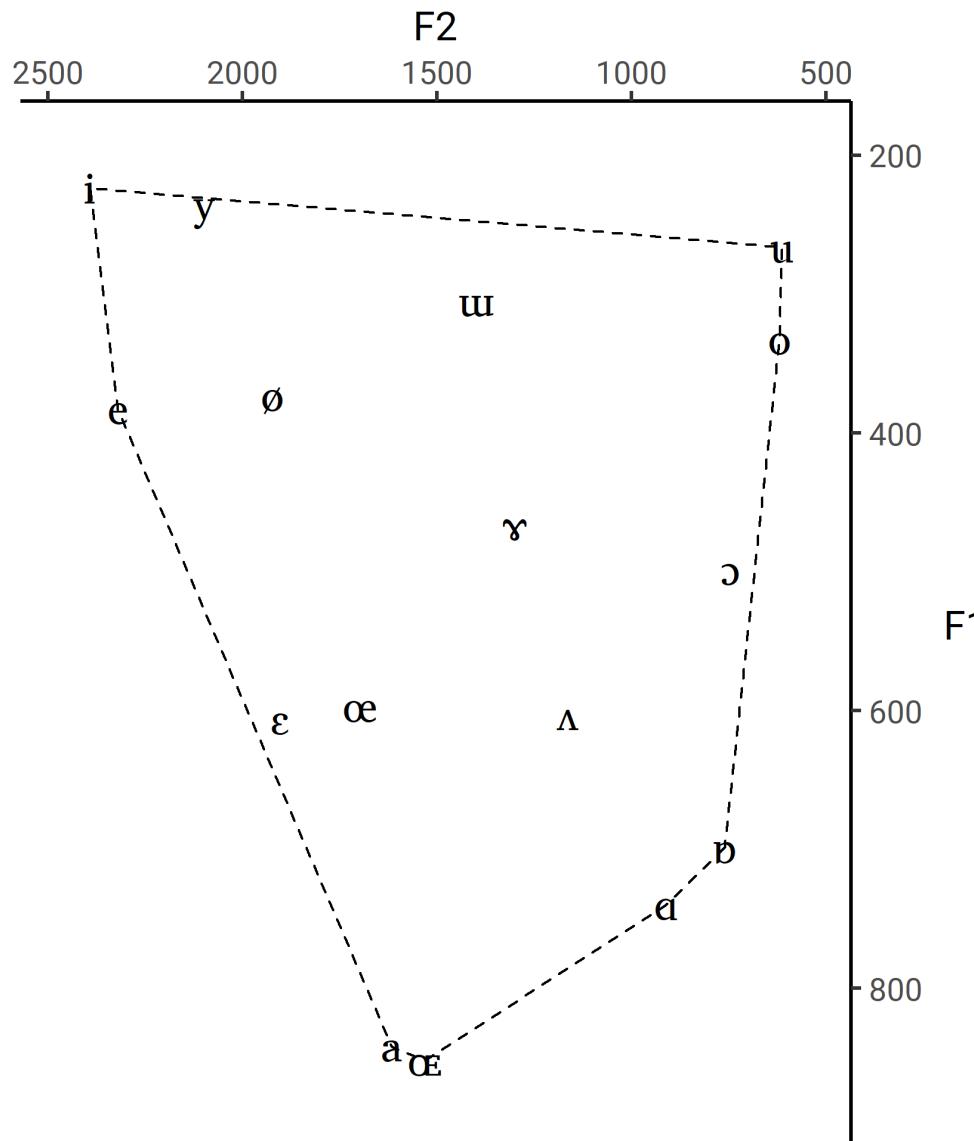
```

sim_vowel_data %>%
  ggplot(aes(x = F2, y = F1)) +
  geom_text(aes(label = Vowel)) +
  scale_x_reverse(
    position = "top",
    breaks = pretty_breaks(5)
  ) +
  scale_y_reverse(position = "right") +
  theme(
    plot.margin = margin(.5,.5, 1, 1, unit = "cm"),
    axis.title.x.top = element_text(margin = margin(b = .2,
    axis.title.y.right = element_text(
      angle = 0, vjust = 0.5, margin = margin(l = .3, unit =
    )
  ) +
  stat_ellipse(aes(group = Vowel))

```



```
sim_vowel_data_all %>%
  group_by(Vowel) %>%
  summarize(across(c(F1, F2), mean), .groups = 'drop') %>%
  ggplot(aes(x = F2, y = F1)) +
  geom_text(aes(label = Vowel), size = 6, family = "Charis S")
  scale_x_reverse(
    position = "top",
    breaks = pretty_breaks(5),
    expand = expansion(.1)
  ) +
  scale_y_reverse(
    position = "right",
    expand = expansion(.1)
  ) +
  theme(
    plot.margin = margin(.5,.5, 1, 1, unit = "cm"),
    axis.title.x.top = element_text(margin = margin(b = .2,
    axis.title.y.right = element_text(
      angle = 0, vjust = 0.5, margin = margin(l = .3, unit =
    )
  ) +
  stat_chull(
    fill = NA,
    color = "black",
    linetype = 2
  )
)
```



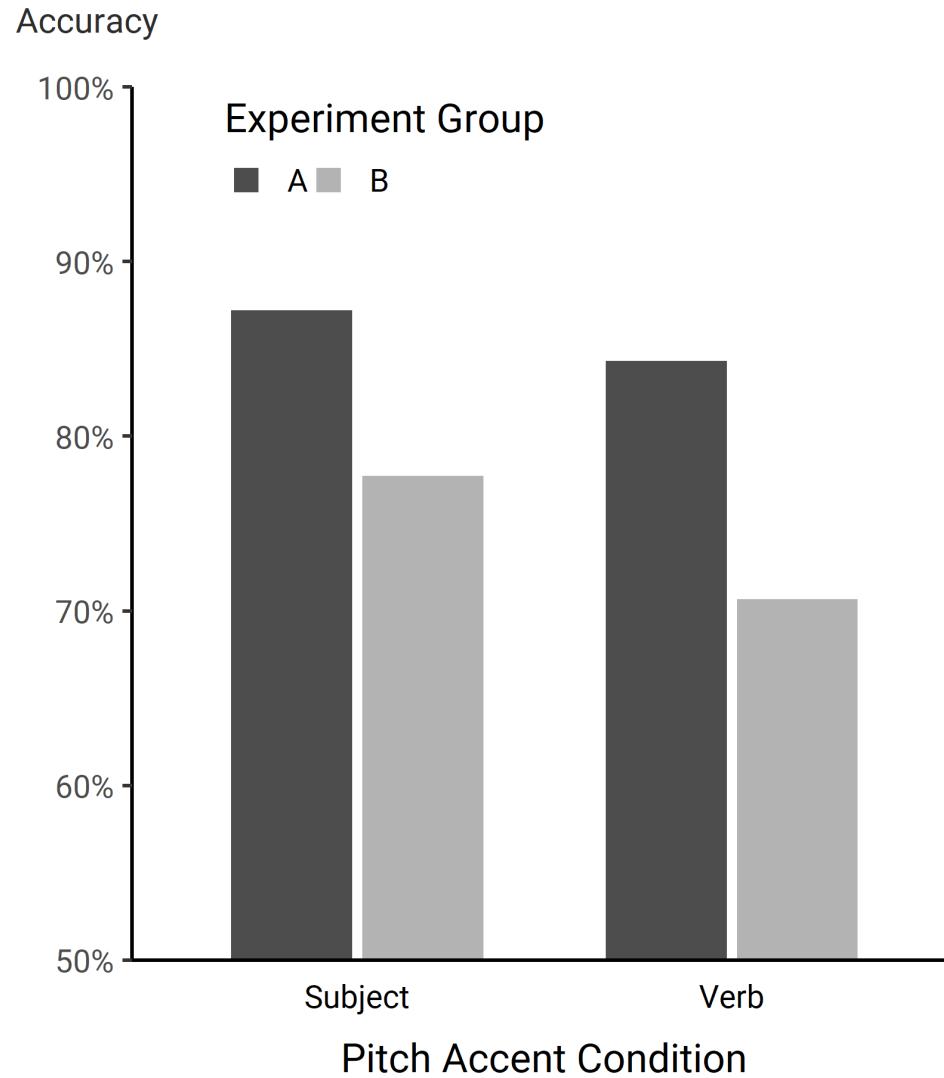
```

p <- read_csv("https://raw.githubusercontent.com/yjunechoe/9-Experiment-Group-Accent-Condition/main/accuracy.csv")
filter(Type == "Critical") %>%
group_by(Cond, Group) %>%
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups = "drop")
ggplot(aes(x = Cond, y = Accuracy, fill = Group)) +
geom_col(position = "dodge", color = "white", width = .7,
scale_fill_manual(values = c("grey30", "grey70")) +
labs(
  title = "Comprehension Task",
  x = "Pitch Accent Condition", y = NULL,
  fill = "Experiment Group"
) +
coord_cartesian(ylim = c(0.5, 1)) +
guides(fill = guide_legend(direction = "horizontal", title = "Experiment Group"))
theme(
  axis.ticks.x = element_blank(),
  axis.text.x = element_text(color = "black", margin = margin(0, 0, 0, 0)),
  legend.position = c(.2, .93),
  plot.title = element_text(
    margin = margin(b = 1.5, unit = "cm"),
    hjust = .5
),
  plot.margin = margin(.8, 1, .7, 1.2, "cm")
) +
scale_y_continuous(
  expand = expansion(0, 0),
  labels = percent_format(accuracy = 1)
)

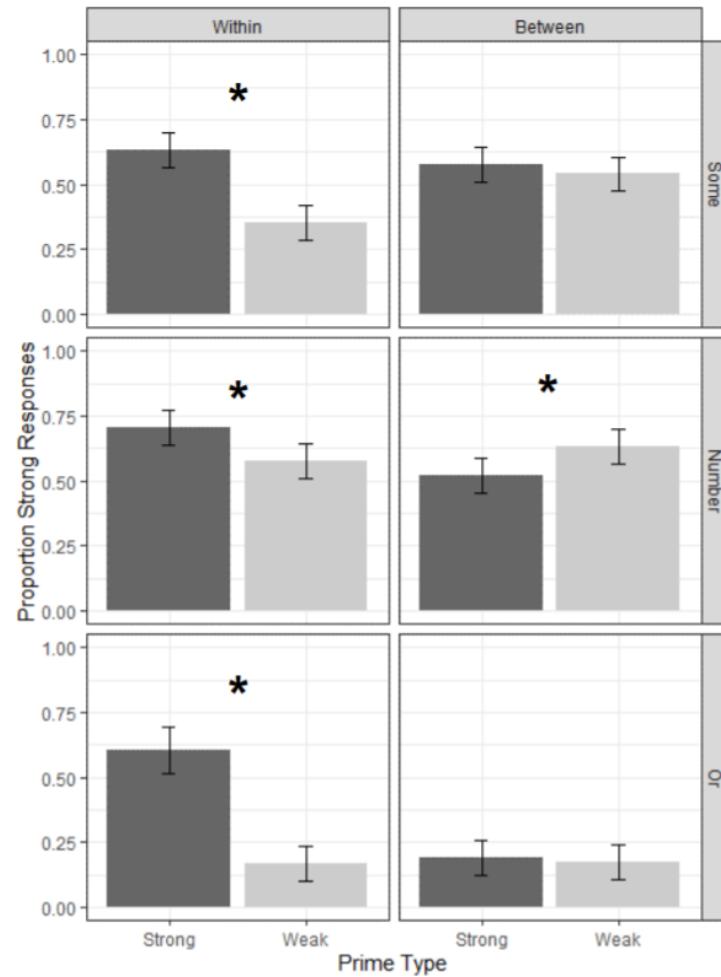
library(cowplot)
ggdraw(p) +
  draw_label("Accuracy", x = .125, y = .88, fontfamily = "Rockwell")

```

Comprehension Task



Multiple Categorical Levels



Source: Husband & Patson (2020).

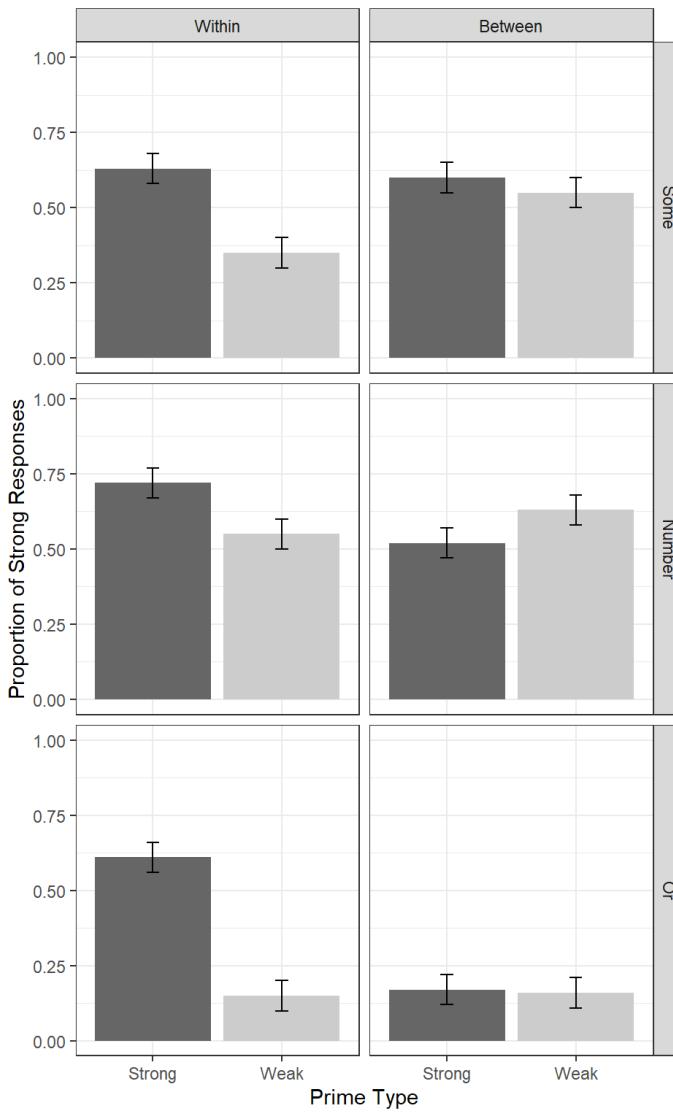
```

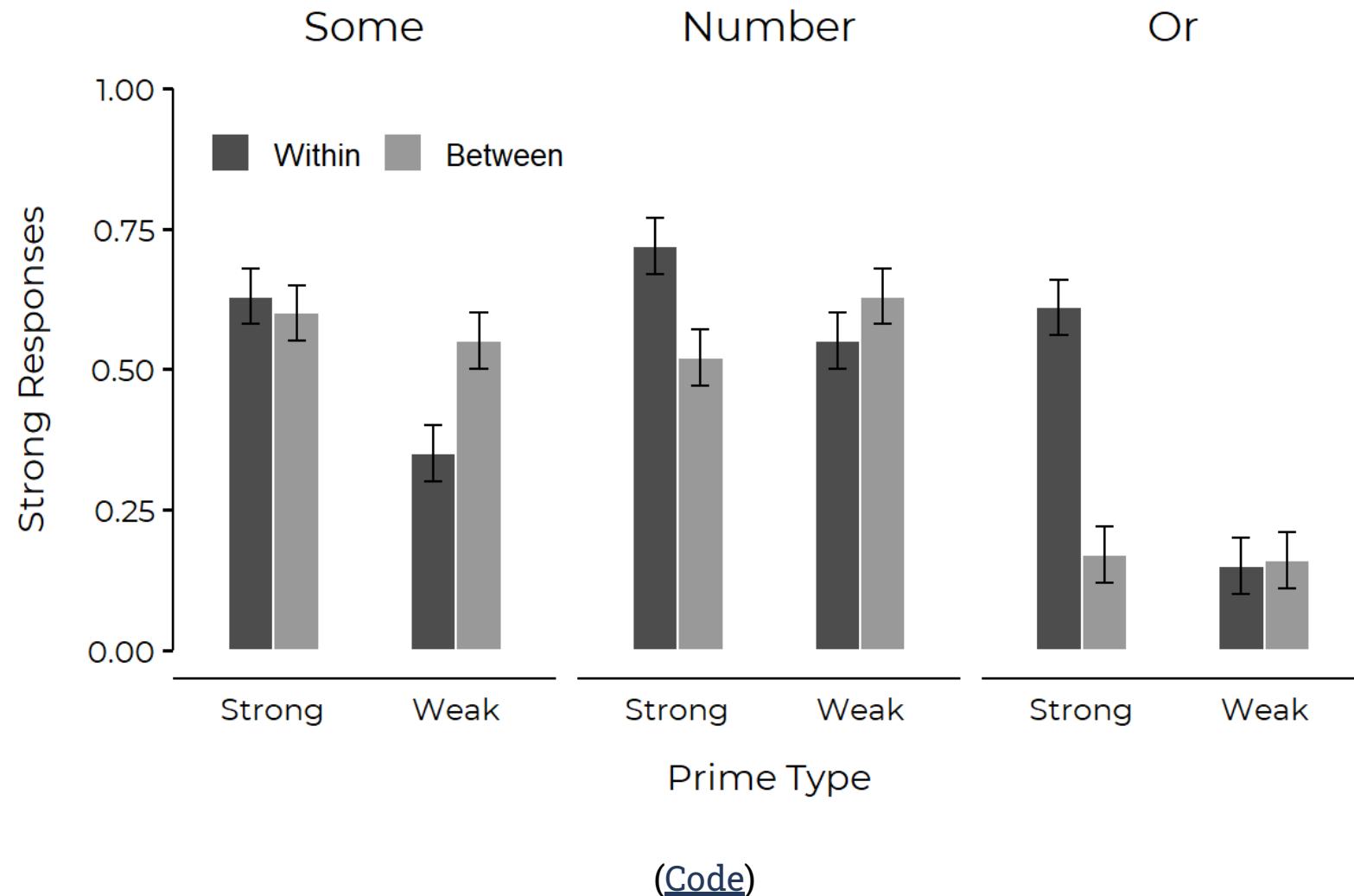
df <- crossing(level_1 = fct_inorder(c("Within", "Between"),
                                      level_2 = fct_inorder(c("Some", "Number")),
                                      level_3 = factor(c("Strong", "Weak"))))

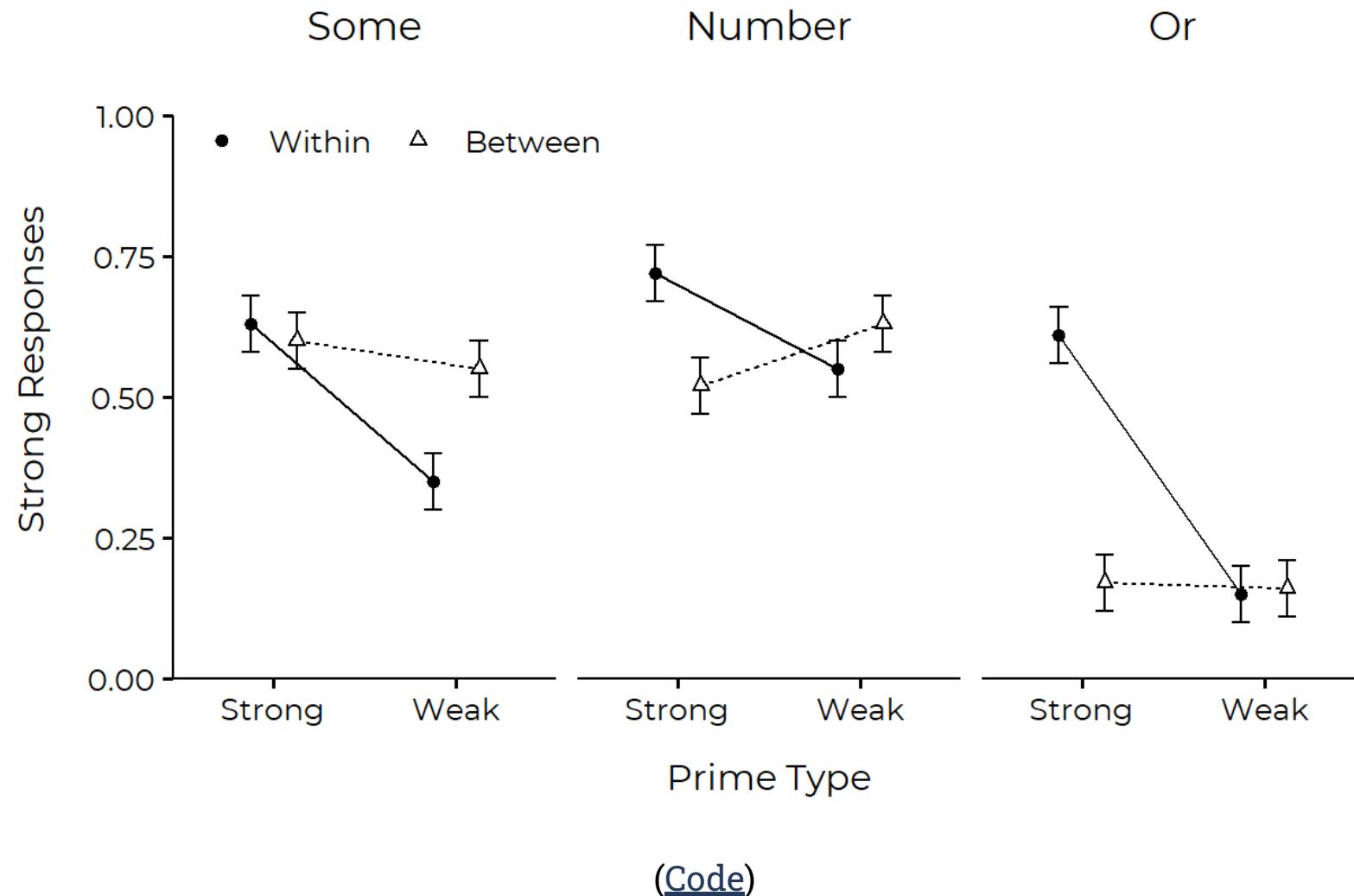
df$barheight <- c(.63, .35, .72, .55, .61, .15, .60, .52)

df %>%
  ggplot(aes(level_3, barheight)) +
  geom_col(
    aes(fill = level_3),
    show.legend = FALSE
  ) +
  geom_errorbar(
    aes(ymin = barheight - .05, ymax = barheight + .05),
    width = .1
  ) +
  facet_grid(level_2 ~ level_1) +
  theme_bw() +
  scale_fill_manual(values = c('grey40', 'grey80')) +
  ylim(0, 1) +
  labs(
    y = "Proportion of Strong Responses",
    x = "Prime Type" +
  theme_bw()

```







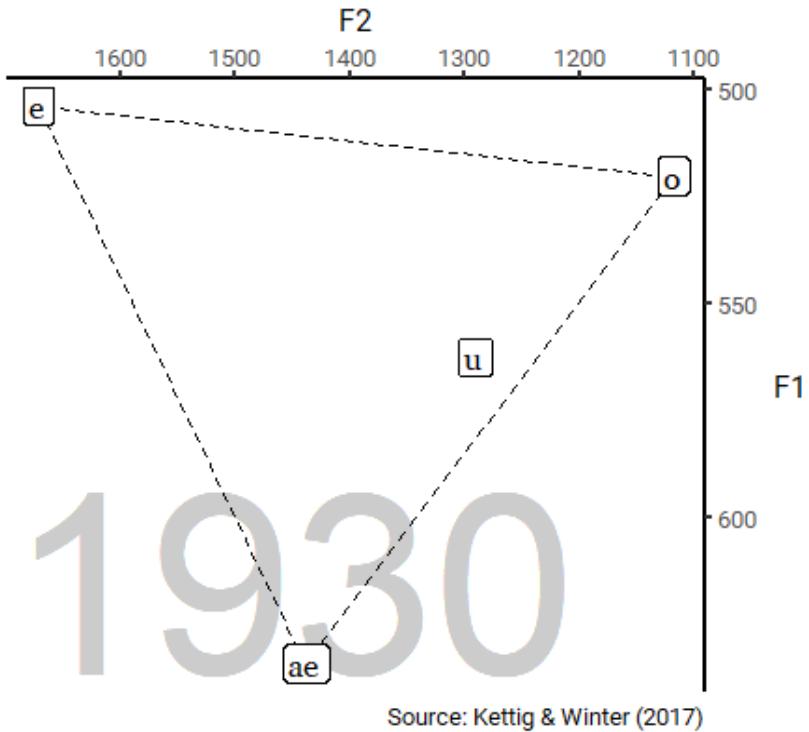
Animations: Vowel shift

```

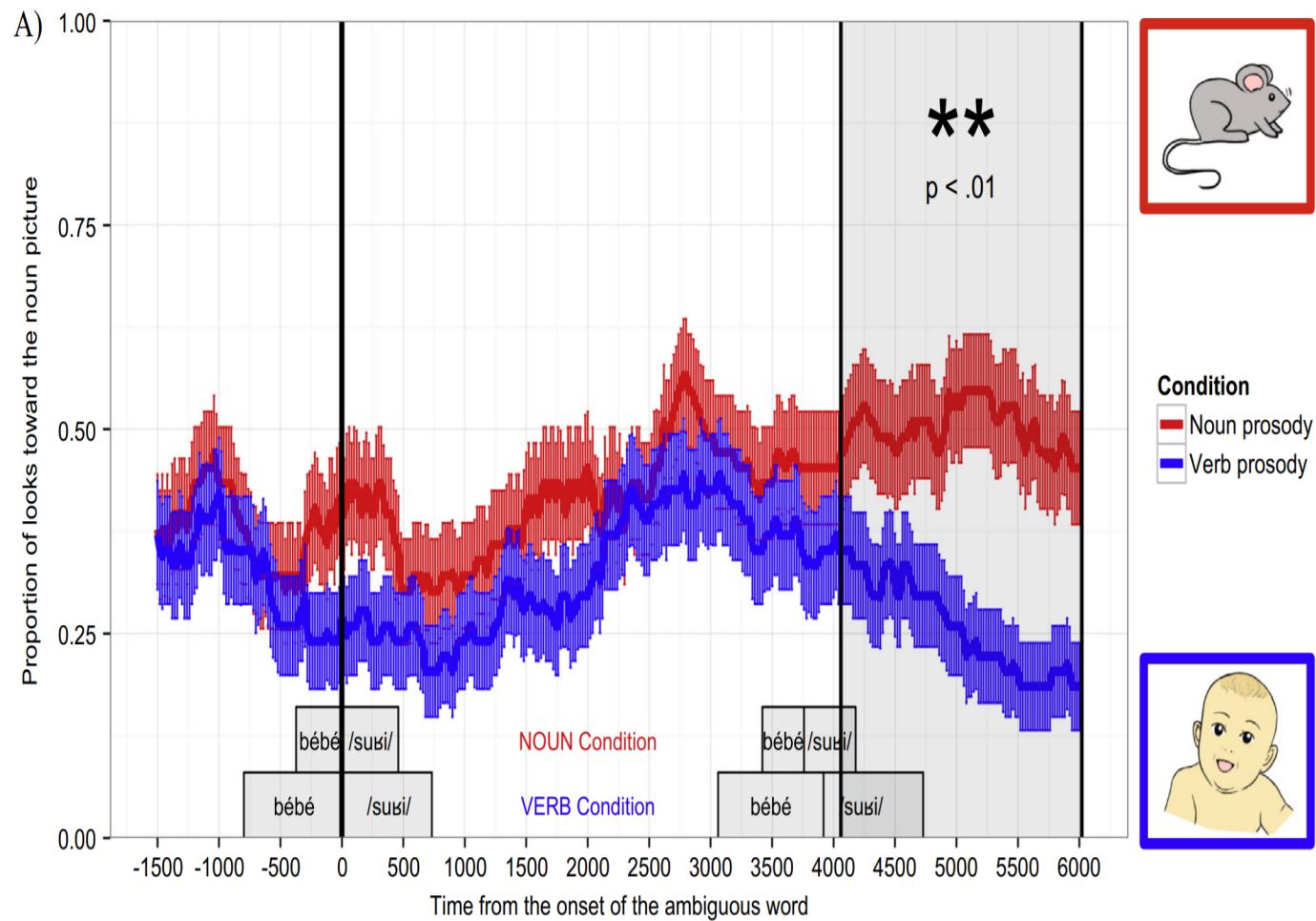
read_csv("https://raw.githubusercontent.com/bodowinter/
group_by(Gender, Vowel, BirthDecade = 10 * BirthYear
summarize(across(F1:F2, mean), .groups = 'drop') %>%
filter(Gender == "F") %>%
ggplot(aes(F2, F1)) +
geom_text(aes(1450, 615, label = as.character(BirthDe
          color = "gray80", size = 48) +
stat_chull(fill = NA, color = "black", linetype = 2)
geom_label(aes(label = Vowel), size = 6, family = "C
scale_x_reverse(position = "top") +
scale_y_reverse(position = "right") +
theme(
  plot.margin = margin(.5,.8, 1.2, 1, unit = "cm"),
  axis.title.x.top = element_text(margin = margin(b =
  axis.title.y.right = element_text(angle = 0, vjust
) +
coord_cartesian(clip = 'off') +
labs(title = "Canadian Vowel Shift", caption = "Sourc
transition_states(BirthDecade) +
shadow_mark(
  alpha = .1,
  color = "grey",
  exclude_layer = c(1, 2)
)

```

Canadian Vowel Shift

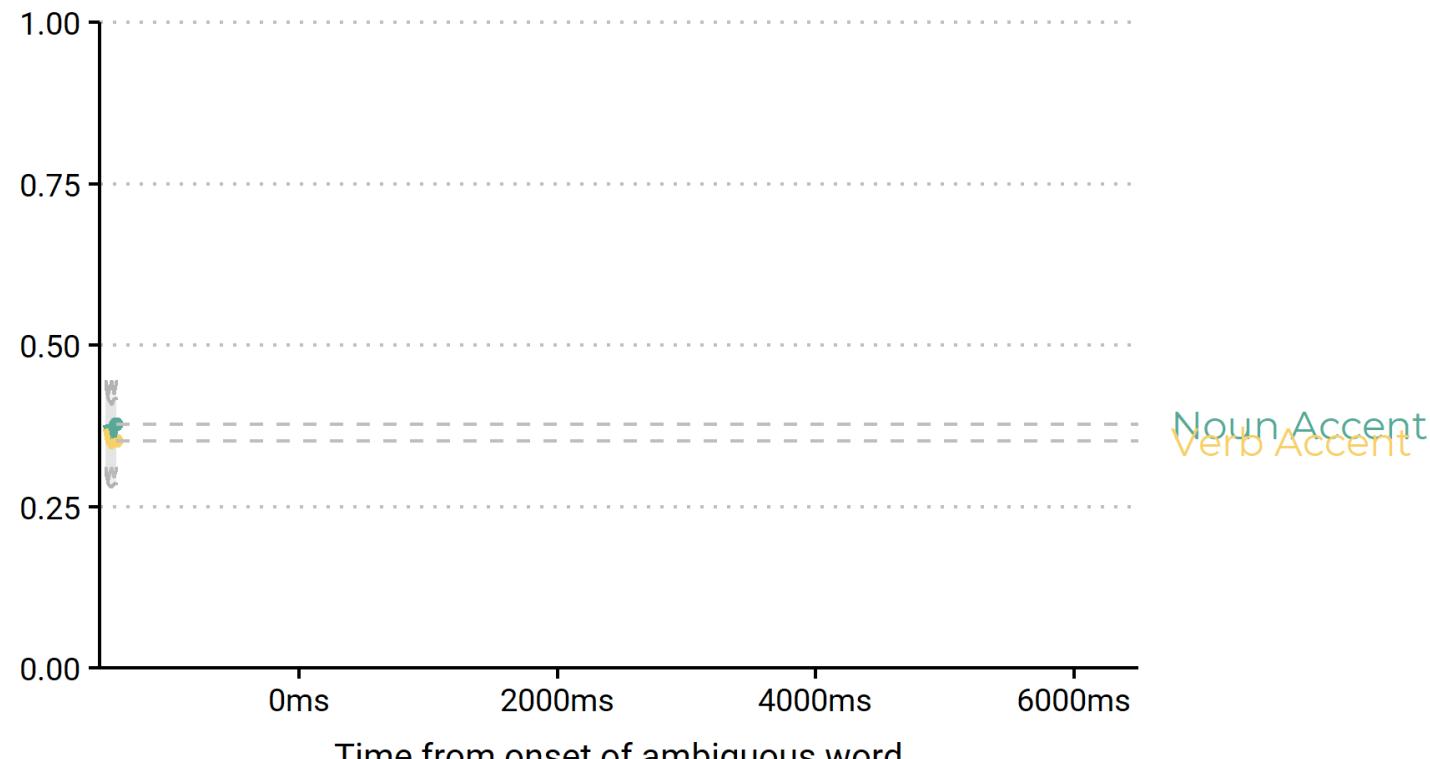


Animations: Eye-tracking



de Carvalho et al. (2017)

Proportion of looks towards NOUN picture



(Code)

Resources

LIGHT

- [The Glamour of Graphics](#) (20-min video)
- [Gallery of plot types](#)
- [What to consider when choosing colors for data viz](#)
- [Palettes \(in R\)](#)

HEAVY

- [Theme elements in ggplot2](#) (documentation)
- [Font Recommendations](#) (book chapter)
- [ggplot2 tutorial](#) (4-hour video, 2 parts)
- [ggplot2 book](#)