

# Visualizing Data (in R)

An opinionated style guide

June Choe

Trueswell-Swingley-Gleitman Lab

2020-11-25

# Motivation

# Motivation

**Why bother improving the quality of your plot?**

# Motivation

**Why bother improving the quality of your plot?**

- Helps people **remember** your results better

# Motivation

**Why bother improving the quality of your plot?**

- Helps people **remember** your results better
- Makes your findings look more **credible**

# 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**

# 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

# 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**

# 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

# 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)

# 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

# Outline

**Four principles for explanatory data visualization**

# Outline

## Four principles for explanatory data visualization

- Make your **text** readable

# Outline

## Four principles for explanatory data visualization

- Make your **text** readable
- Be generous about **margins and spacing**

# Outline

## Four principles for explanatory data visualization

- Make your **text** readable
- Be generous about **margins and spacing**
- Provide a clear **legend**

# 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**

# 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**

# 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

# 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**

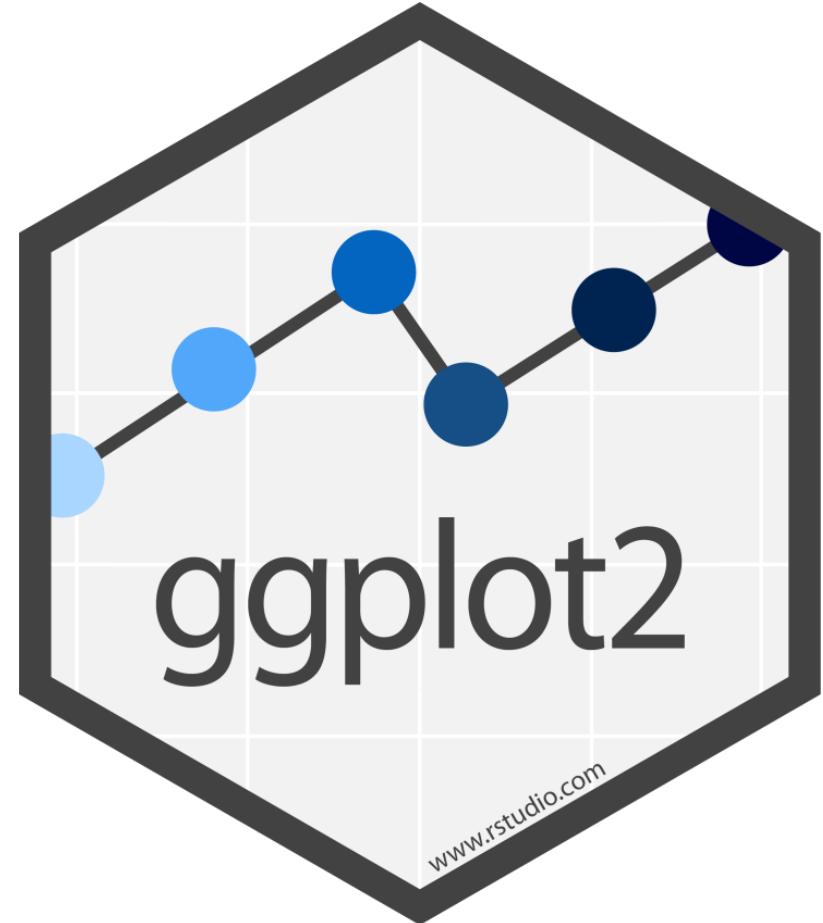
## Showcases

- 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

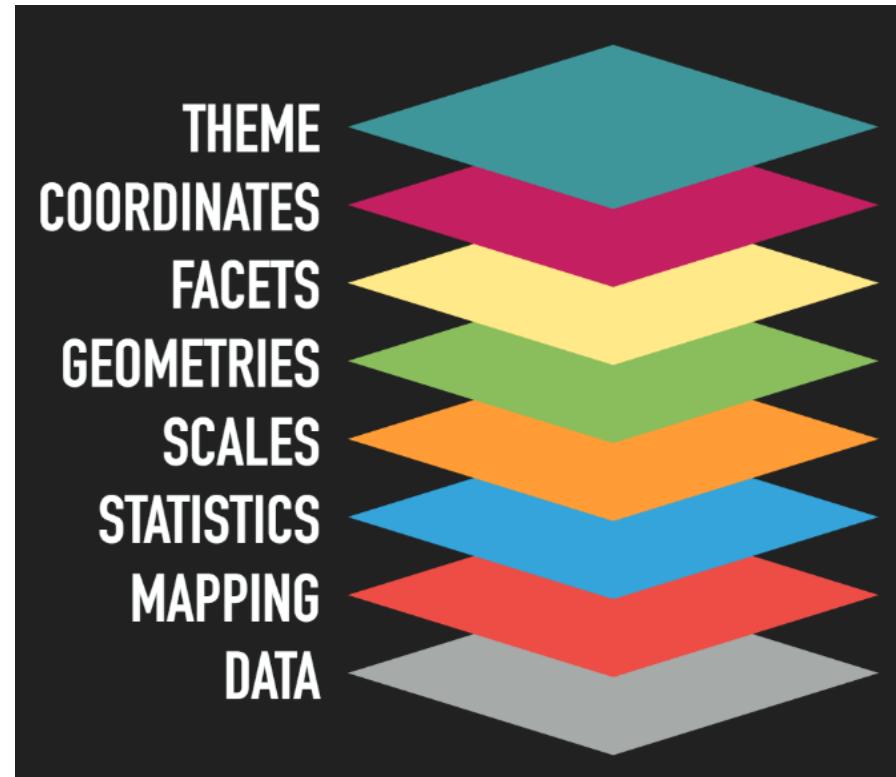
[Github stars ▾](#)

Sort

search name, author, descrip [Text Filter](#)



# The layered Grammar of Graphics



[Video Tutorial](#)

```
state_election_votes
```

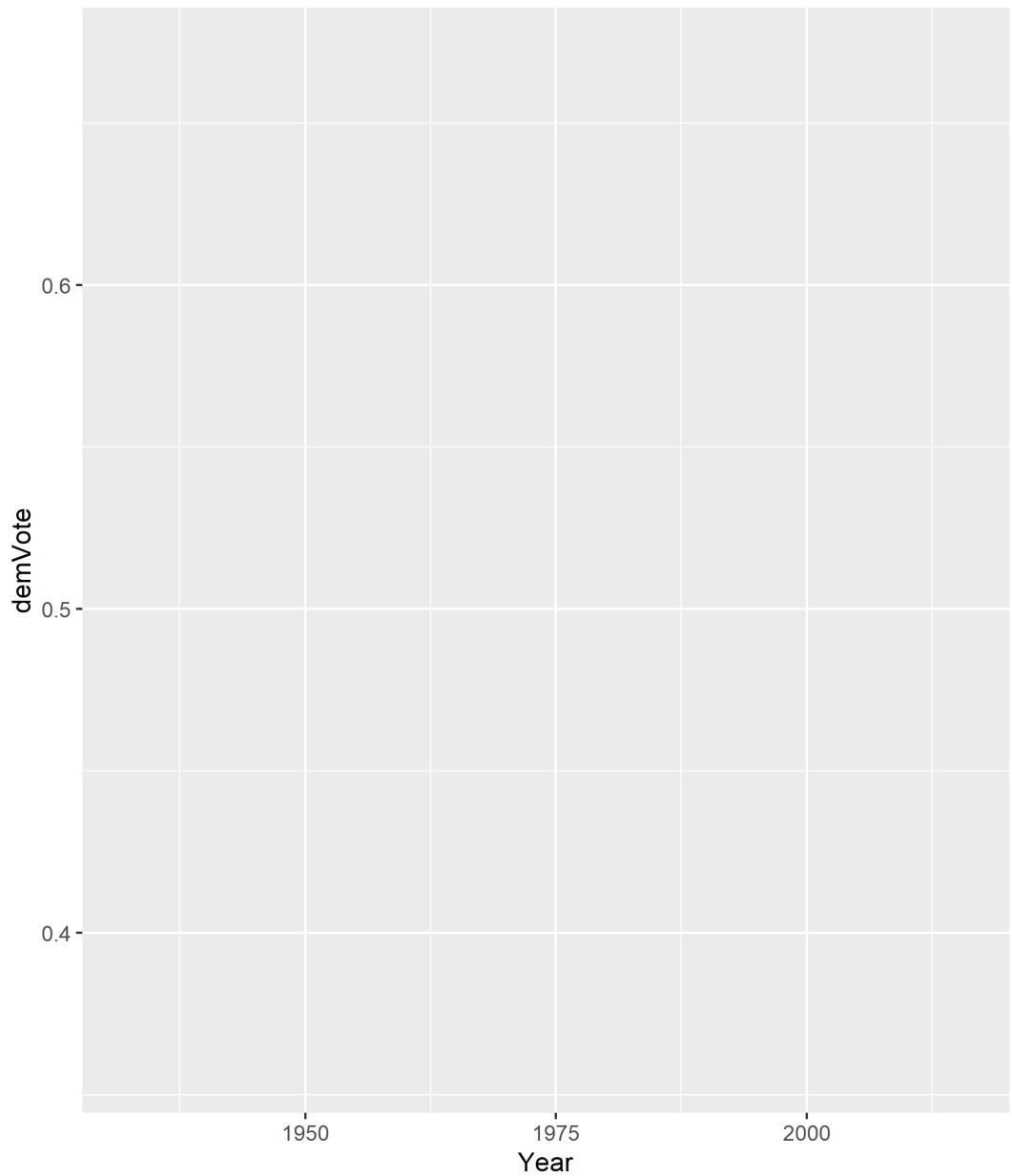
```
# A tibble: 1,097 x 3
  State      demVote Year
  <chr>        <dbl> <dbl>
1 Alabama     0.85   1932
2 Arizona     0.67   1932
3 Arkansas    0.86   1932
4 California  0.580  1932
5 Colorado    0.55   1932
6 Connecticut 0.47   1932
7 Delaware    0.48   1932
8 Florida     0.74   1932
9 Georgia     0.92   1932
10 Idaho      0.59   1932
# ... with 1,087 more rows
```

```
state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
}

# A tibble: 66 x 3
  State     demVote   Year
  <chr>       <dbl> <dbl>
1 California 0.580  1932
2 Illinois   0.55   1932
3 Pennsylvania 0.45   1932
4 California 0.67   1936
5 Illinois   0.580  1936
6 Pennsylvania 0.570  1936
7 California 0.570  1940
8 Illinois   0.51   1940
9 Pennsylvania 0.53   1940
10 California 0.56   1944
# ... with 56 more rows
```

```
state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
    "Illinois",
    "California"))
) %>%
ggplot()
```

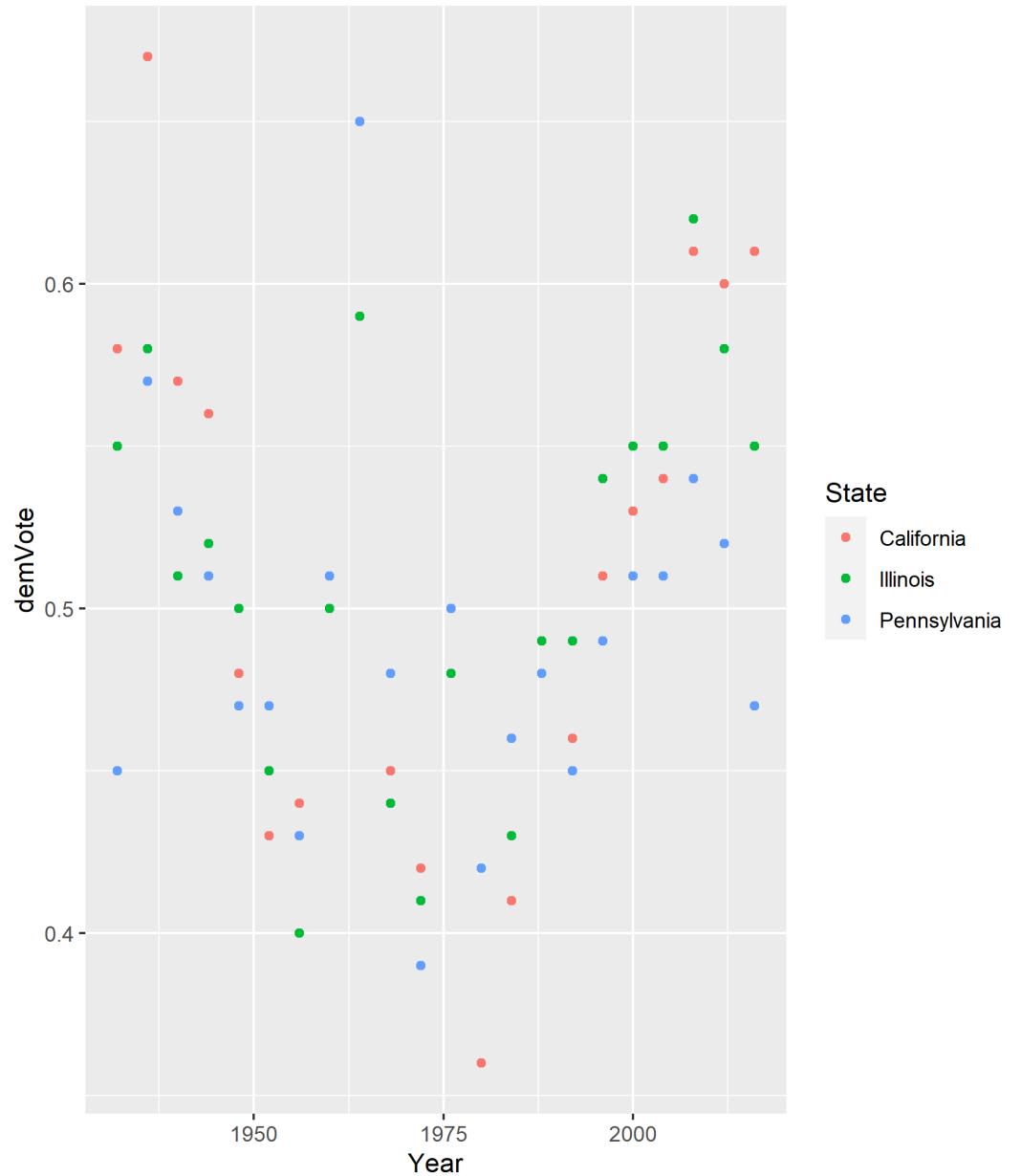
```
state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
)
```



```

state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
  geom_point()

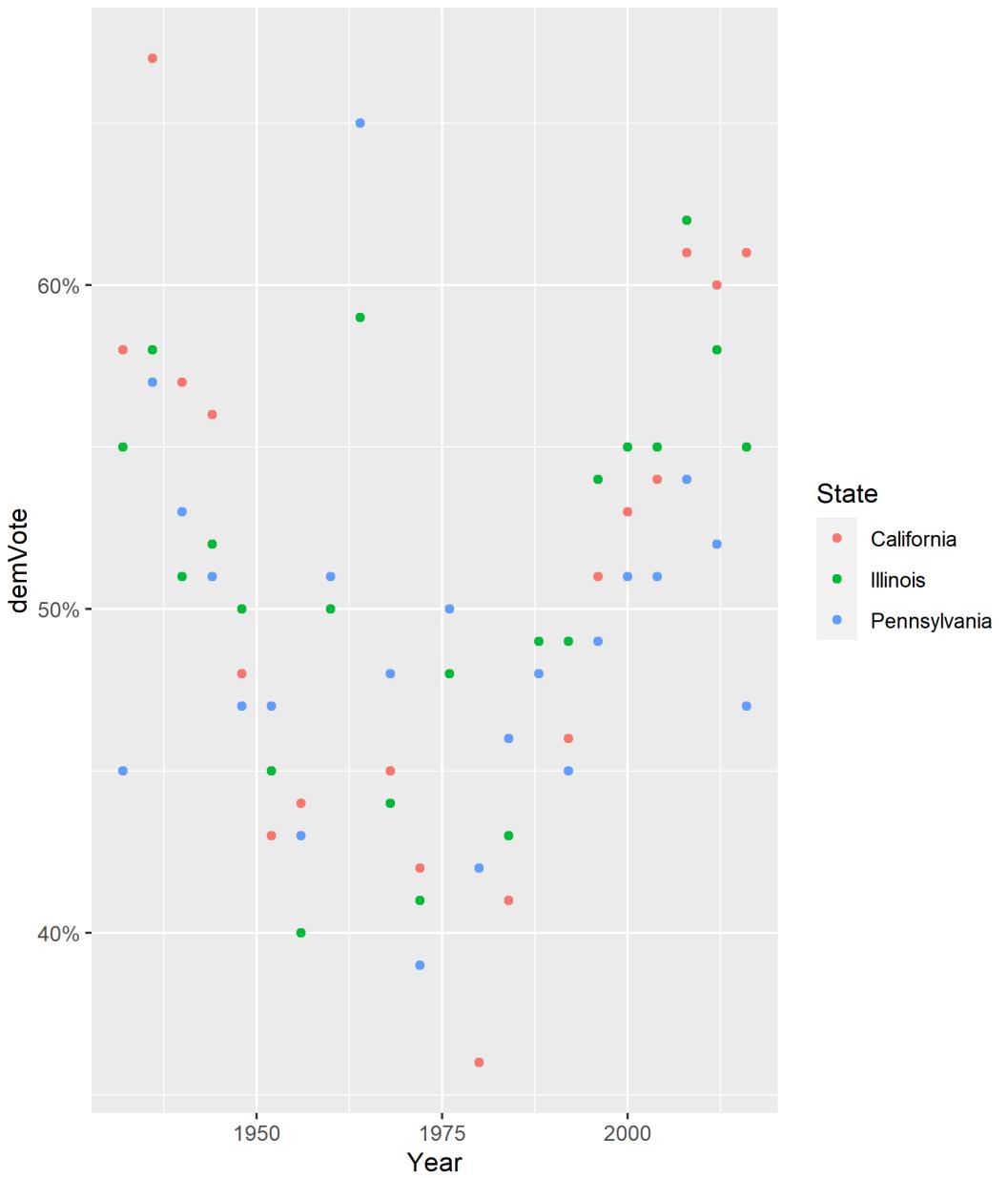
```



```

state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
geom_point() +
scale_y_continuous(
  labels = percent_format(accuracy = 1)
)

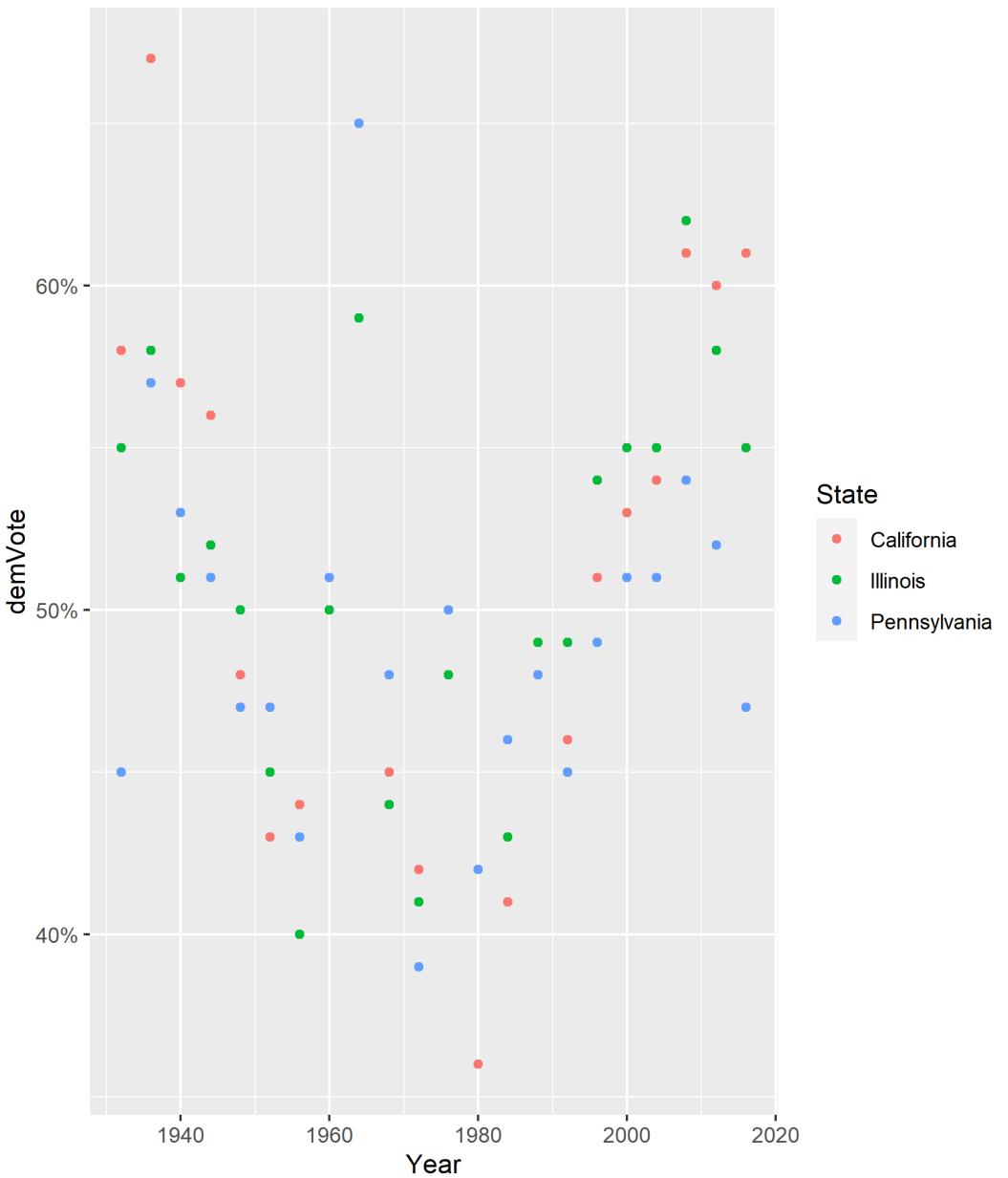
```



```

state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
geom_point() +
scale_y_continuous(
  labels = percent_format(accuracy = 1)
) +
scale_x_continuous(
  breaks = pretty_breaks(n= 5)
)

```

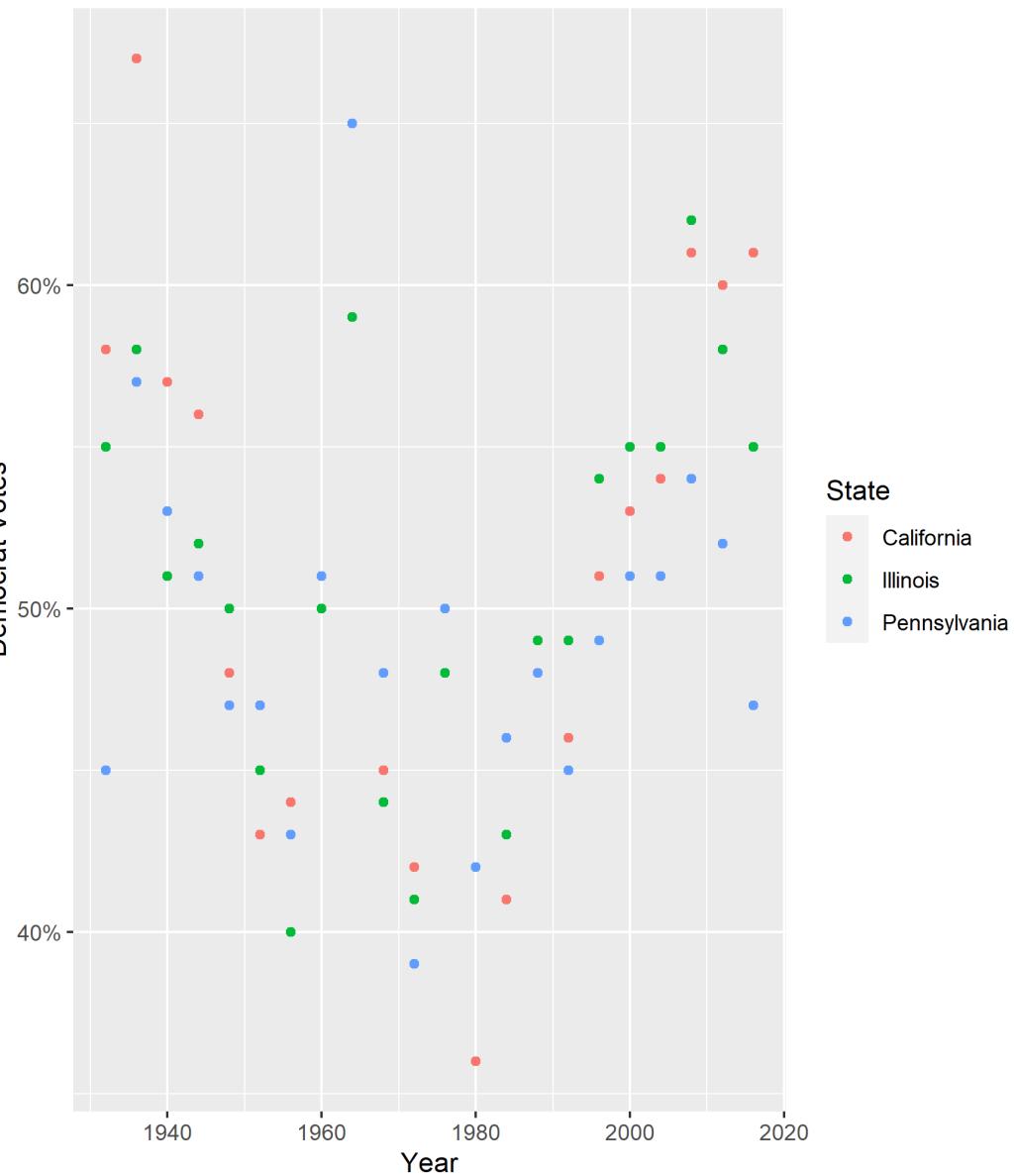


```

state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
geom_point() +
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!

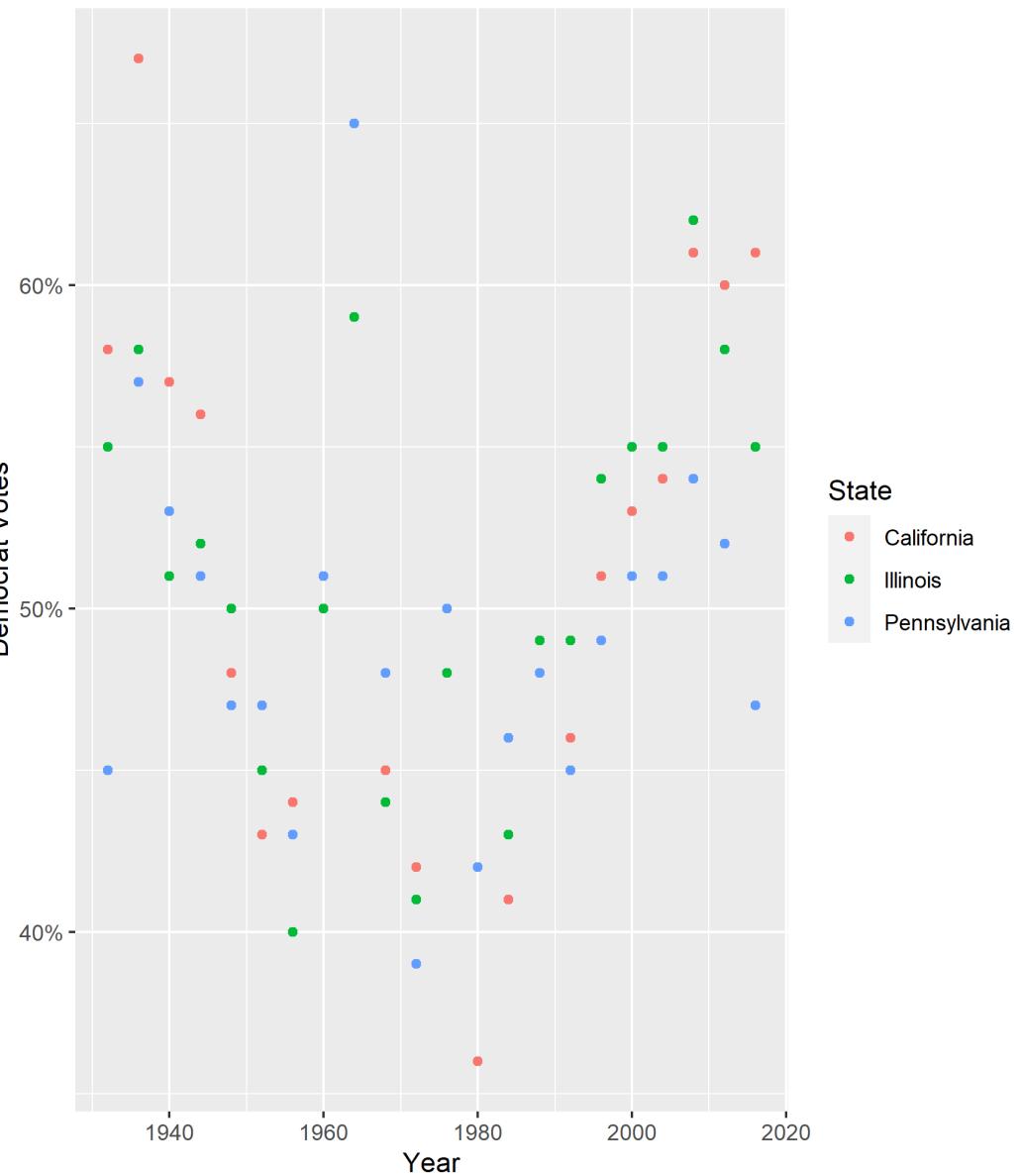


```

state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
    "Illinois",
    "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
geom_point() +
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!

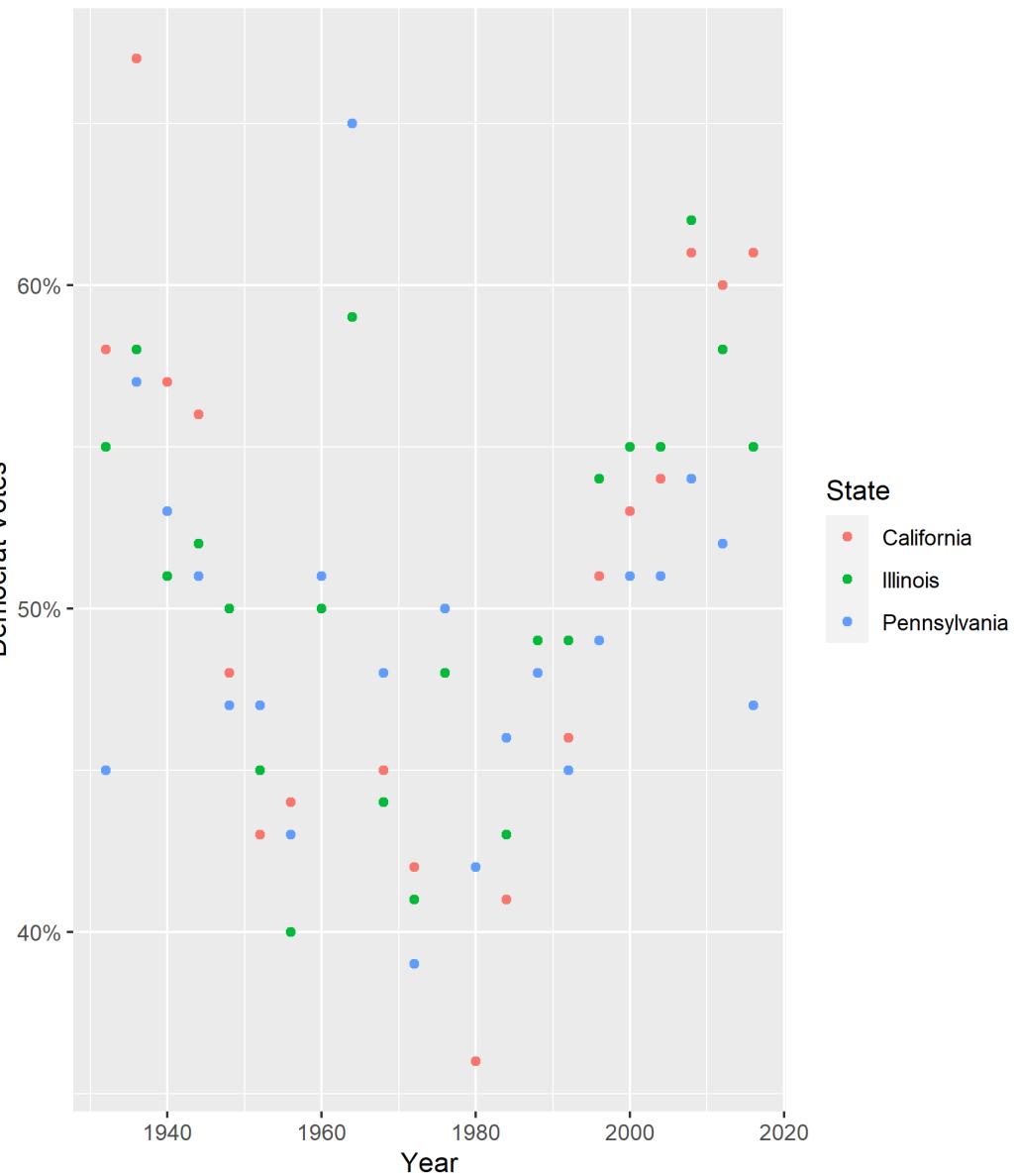


```

state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
geom_point() +
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!

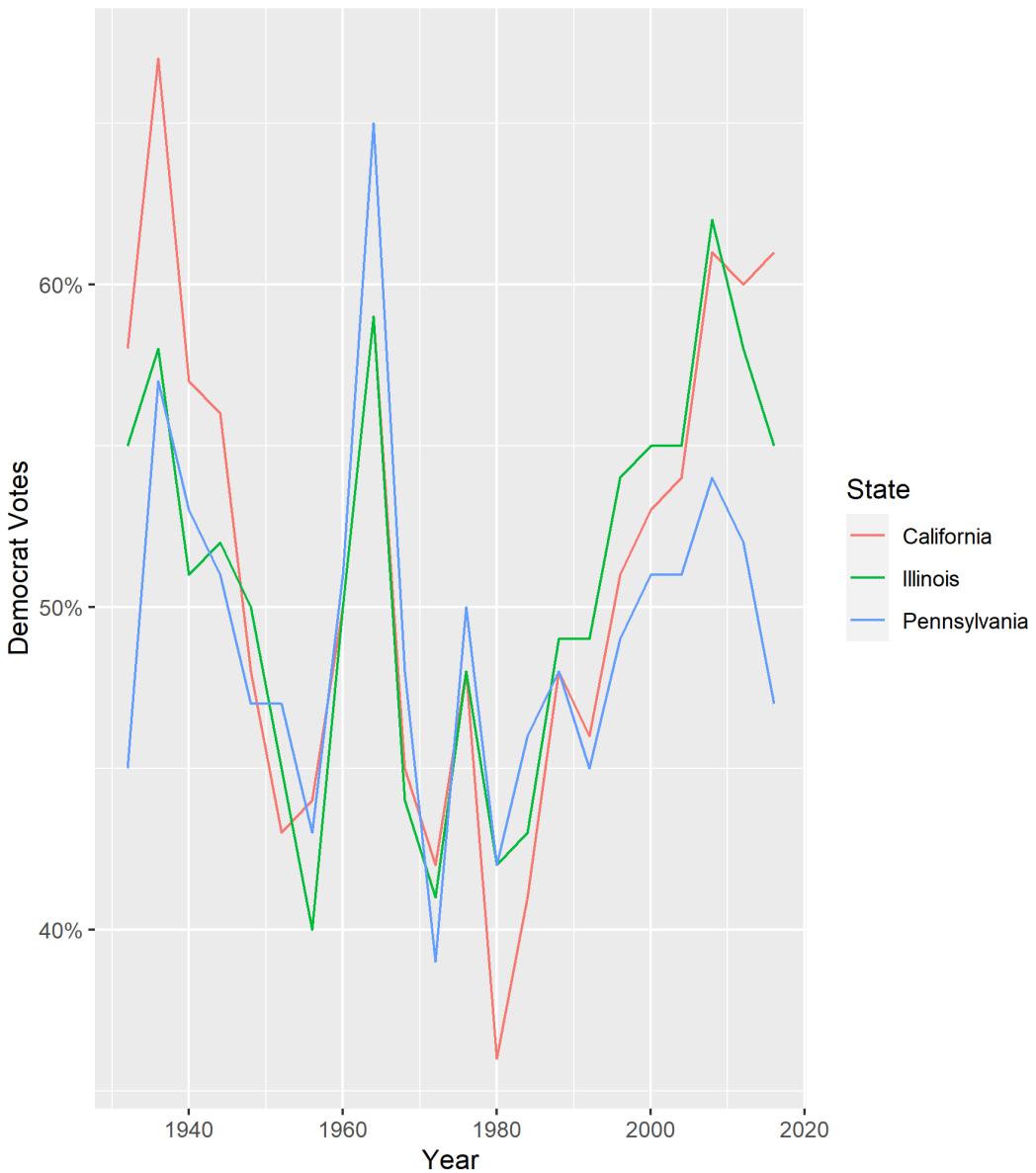


```

state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
geom_line() +
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!

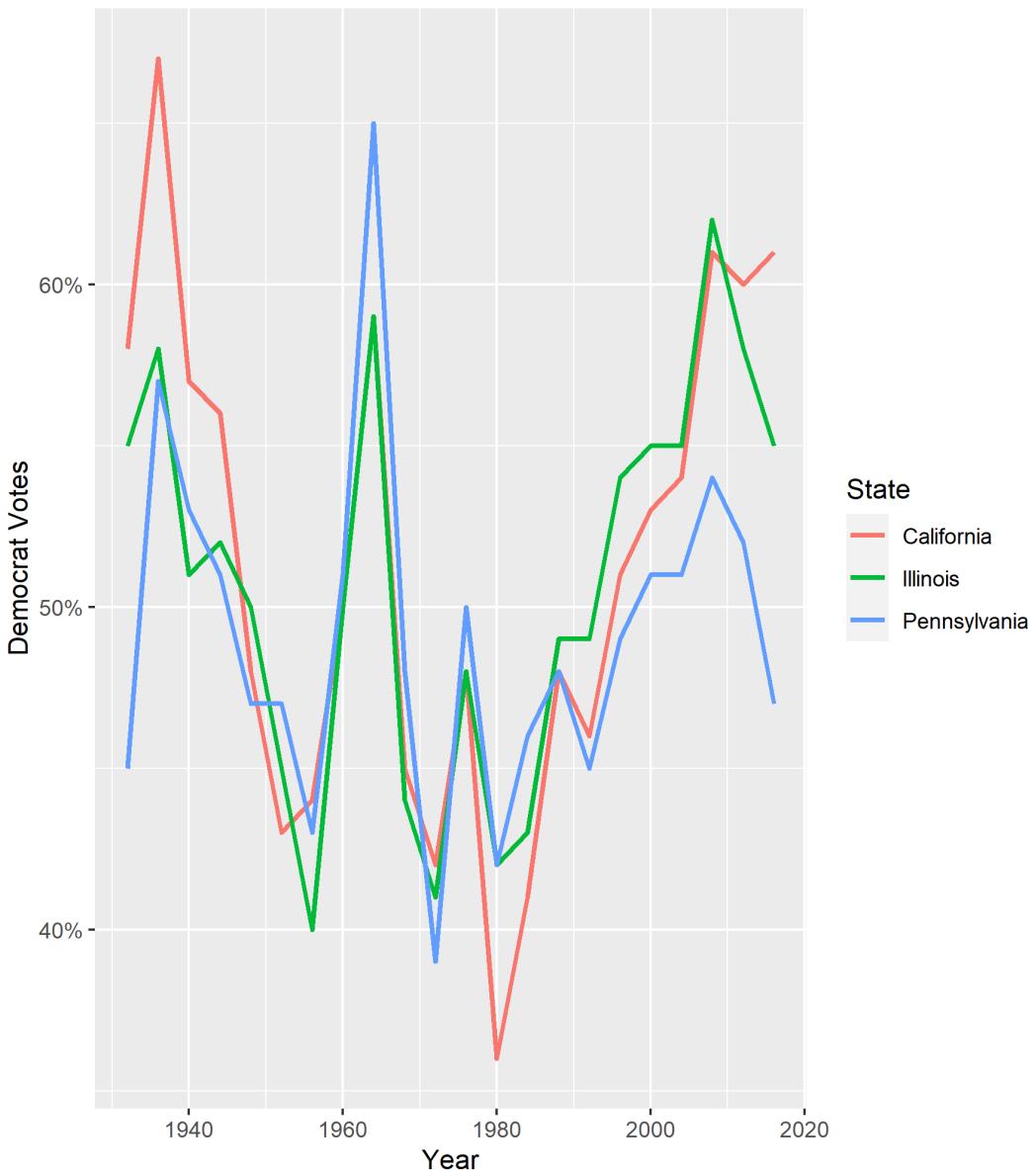


```

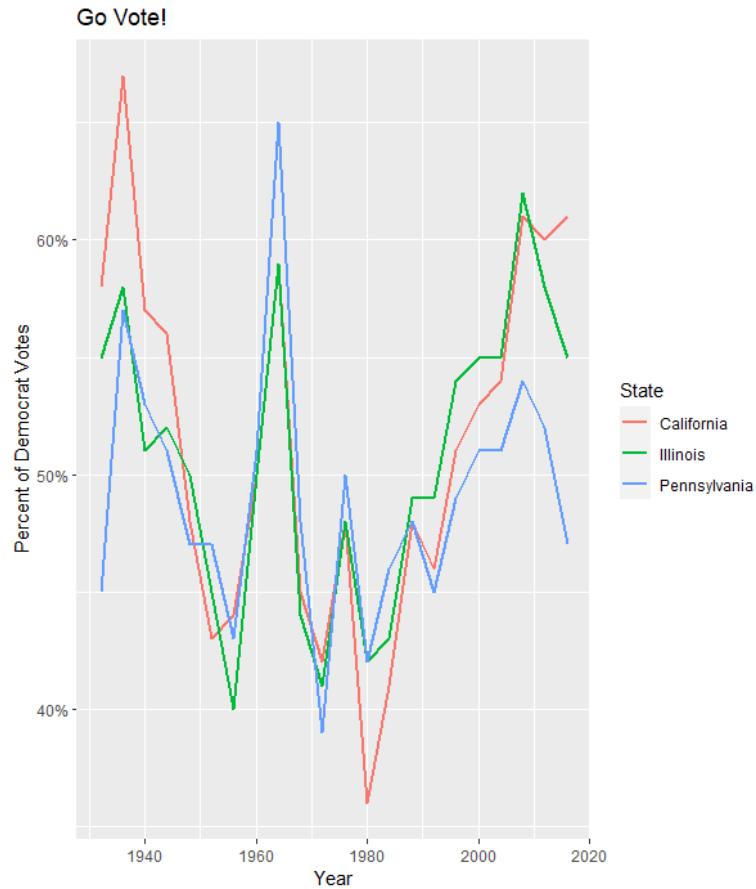
state_election_votes %>%
  filter(State %in%
    c("Pennsylvania",
      "Illinois",
      "California"))
) %>%
ggplot() +
aes(
  x = Year,
  y = demVote,
  color = State
) +
  geom_line(size = 1) +
  scale_y_continuous(
    labels = percent_format(accuracy = 1)
) +
  scale_x_continuous(
    breaks =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

```

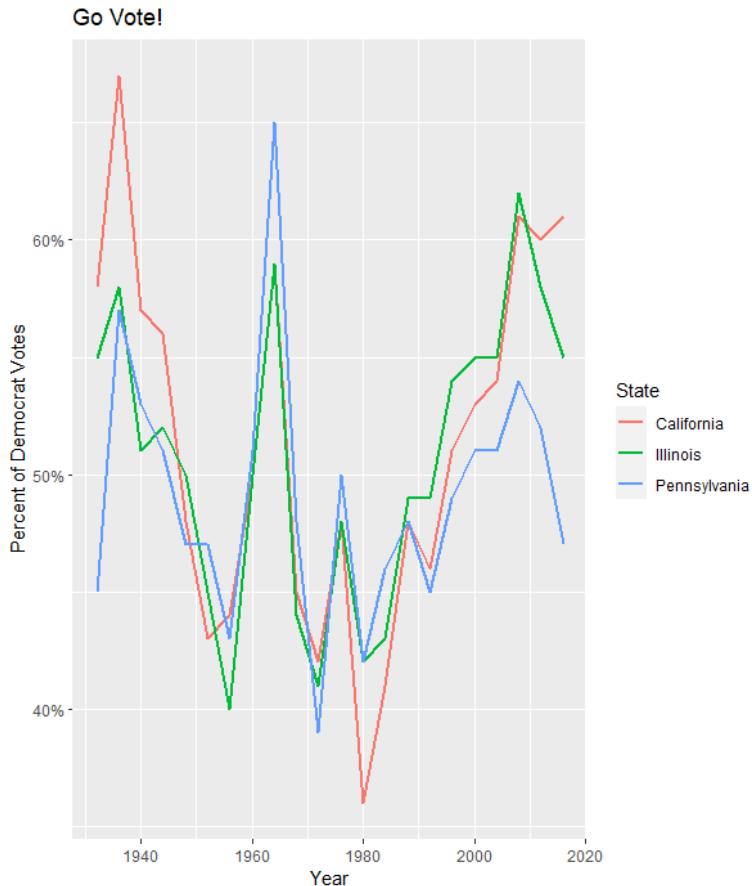
Go Vote!



# 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

# 1. Text

Many different ways to style text:

# 1. Text

Many different ways to style text:

- font **size**

# 1. Text

Many different ways to style text:

- font **size**
- Font face (**bold**, *italic*, SMALL CAPS)

# 1. Text

Many different ways to style text:

- font **size**
- Font face (**bold**, *italic*, **SMALL CAPS**)
- Font family ( Times New Roman, Calibri, Arial, ☞□☛☰■❑♦ )

# 1. Text

Many different ways to style text:

- font **size**
- Font face (**bold**, *italic*, **SMALL CAPS**)
- Font family ( Times New Roman, Calibri, Arial, ພ່າຍະນຸ້ມ້ວນ )
- Font **color**, letter spacing, <sup>angle</sup>, weight, etc.

# 1. Text

## Many different ways to style text:

- font **Size**
  - Font face (**bold**, *italic*, **SMALL CAPS**)
  - Font family ( Times New Roman, Calibri, Arial, ☰ የሃገና ሲሆን )
  - Font **color**, letter spacing, **angle**, weight, etc.

Unless you make your plots with html/css/js, you'll likely only have access to:

# 1. Text

Many different ways to style text:

- font **size**
- Font face (**bold**, *italic*, **SMALL CAPS**)
- Font family ( Times New Roman, Calibri, Arial, Ⓜ, ☰, ™, ■, ♦ )
- Font **color**, letter spacing, ⌈, weight, etc.

Unless you make your plots with html/css/js, you'll likely only have access to:

- Font size, font family, and limited set of font styles

# 1. Text

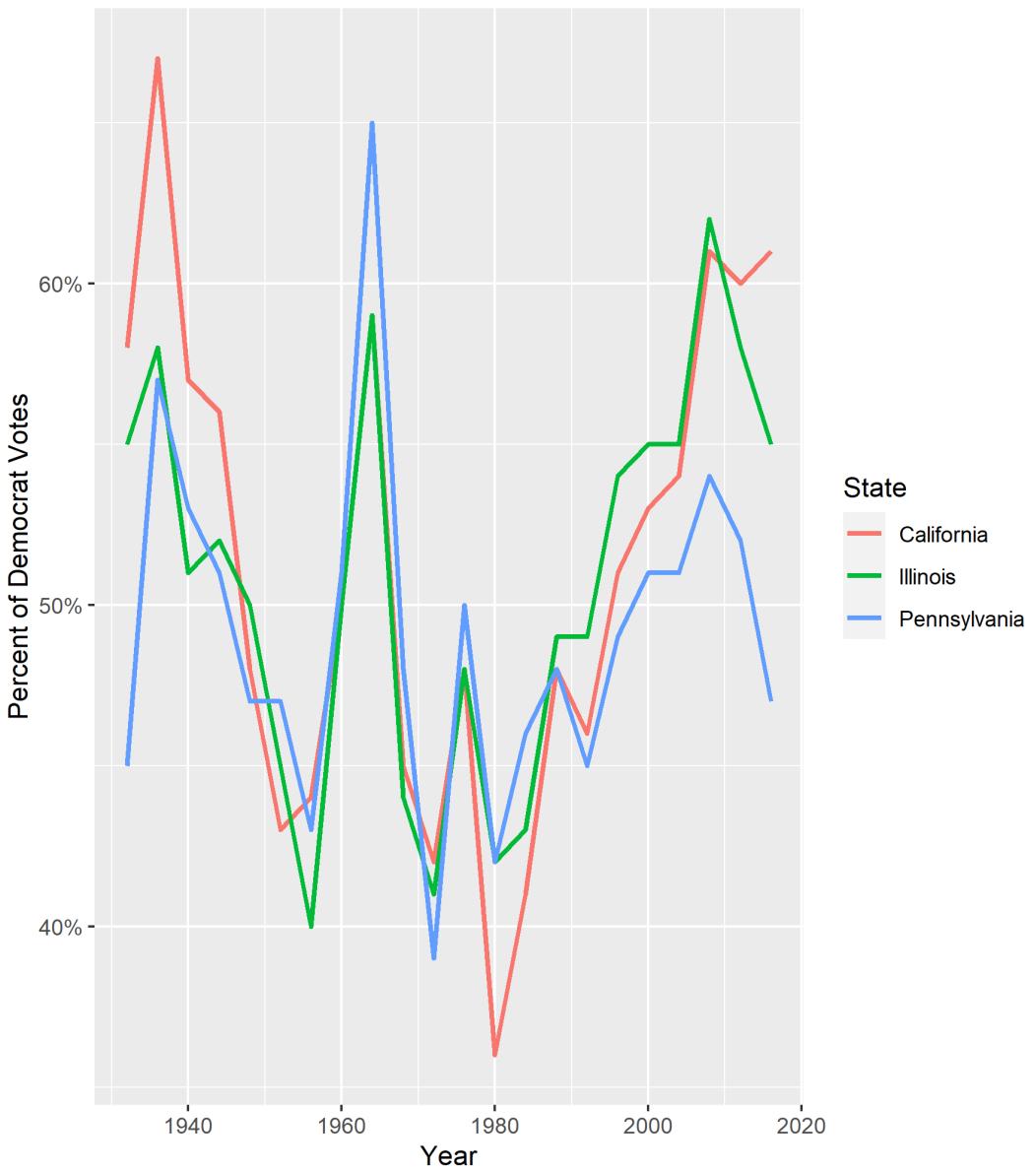
## Many different ways to style text:

- font **Size**
  - Font face (**bold**, *italic*, **SMALL CAPS**)
  - Font family ( Times New Roman, Calibri, Arial, ☰ የሃገና ሲሆን )
  - Font **color**, letter spacing, **angle**, weight, etc.

Unless you make your plots with html/css/js, you'll likely only have access to:

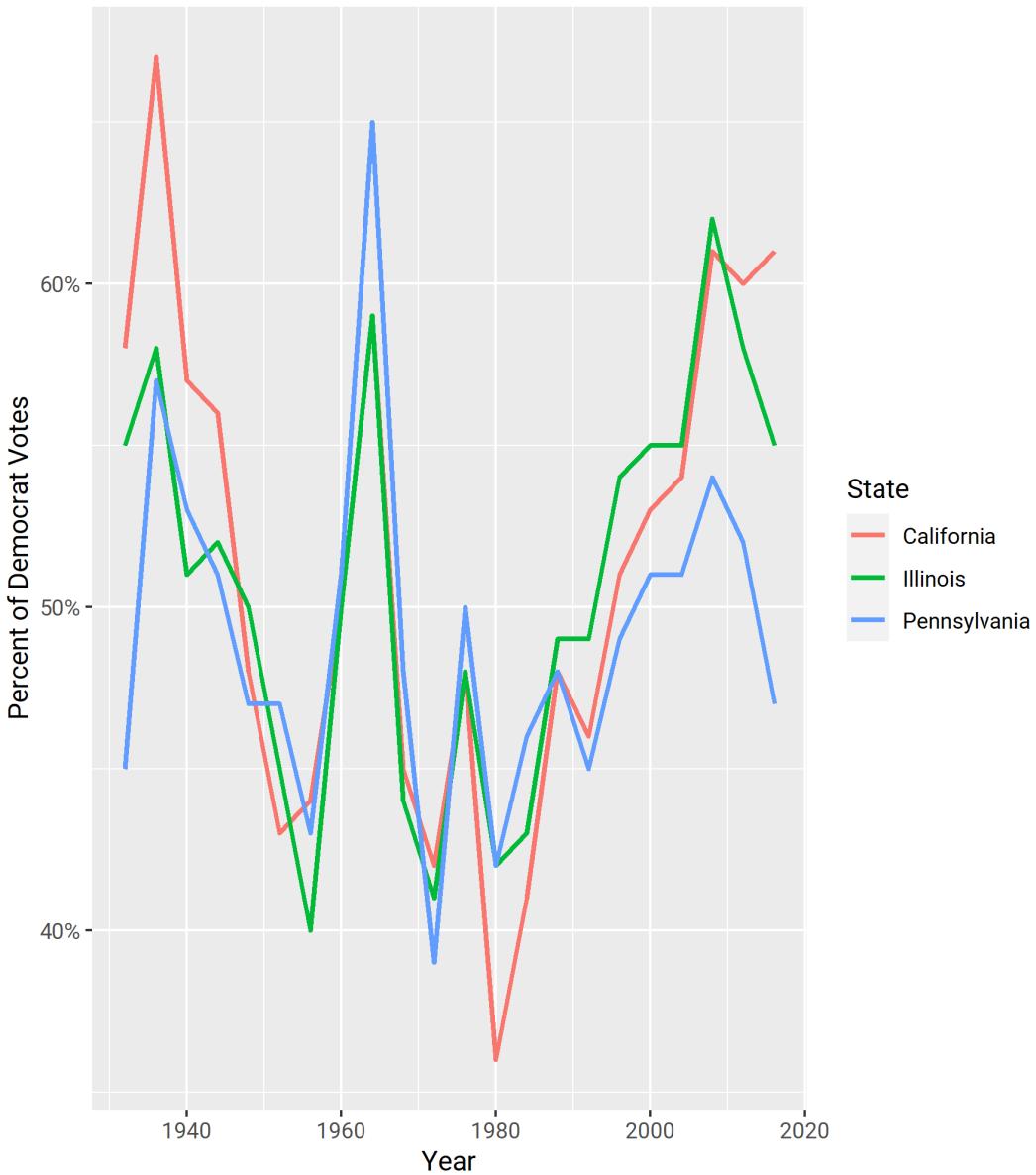
- Font size, font family, and limited set of font styles
  - But you can get pretty far with these!

## Go Vote!



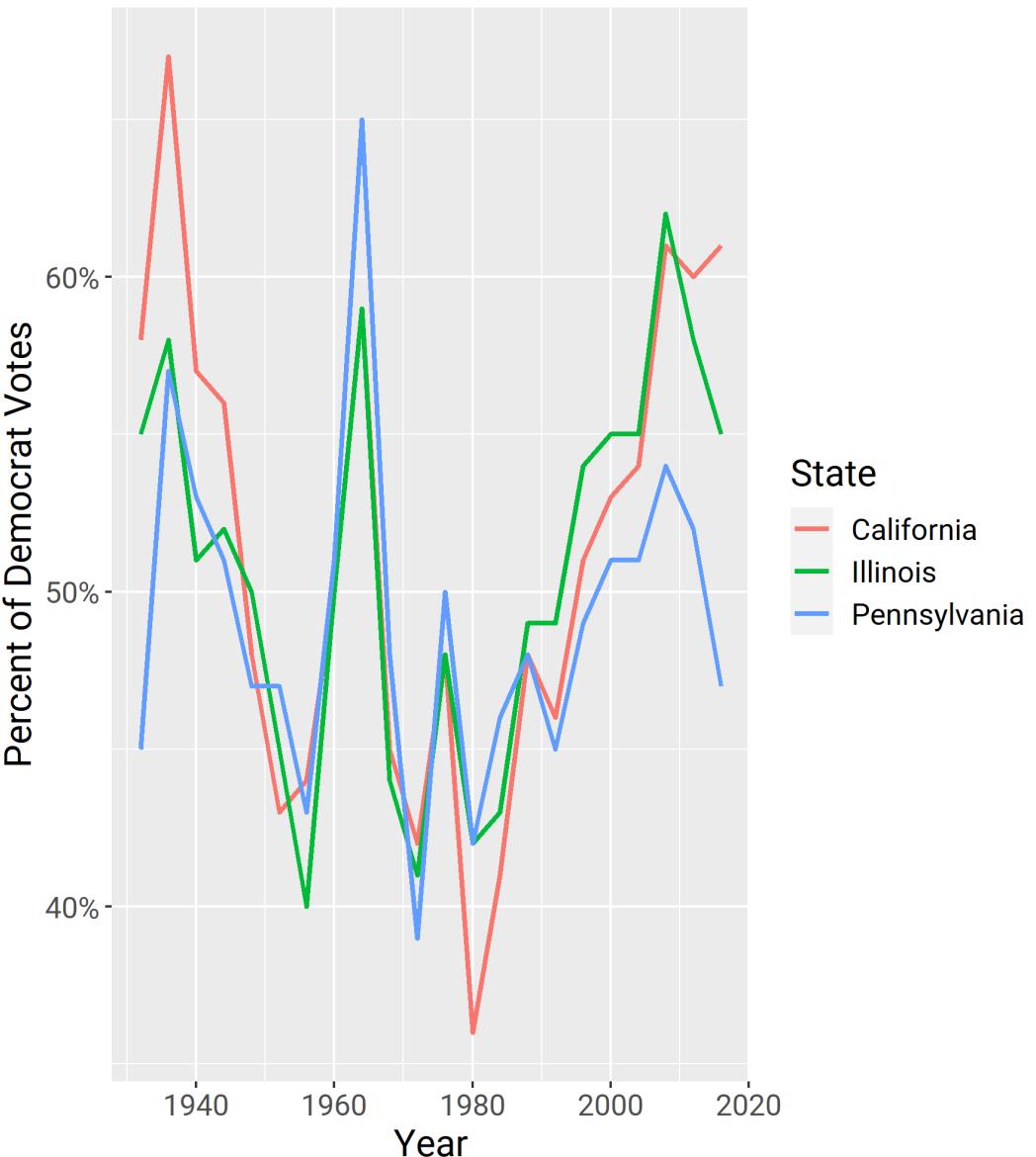
```
state_election_plot +  
  theme(text = element_text(family = "Roboto"))
```

## Go Vote!



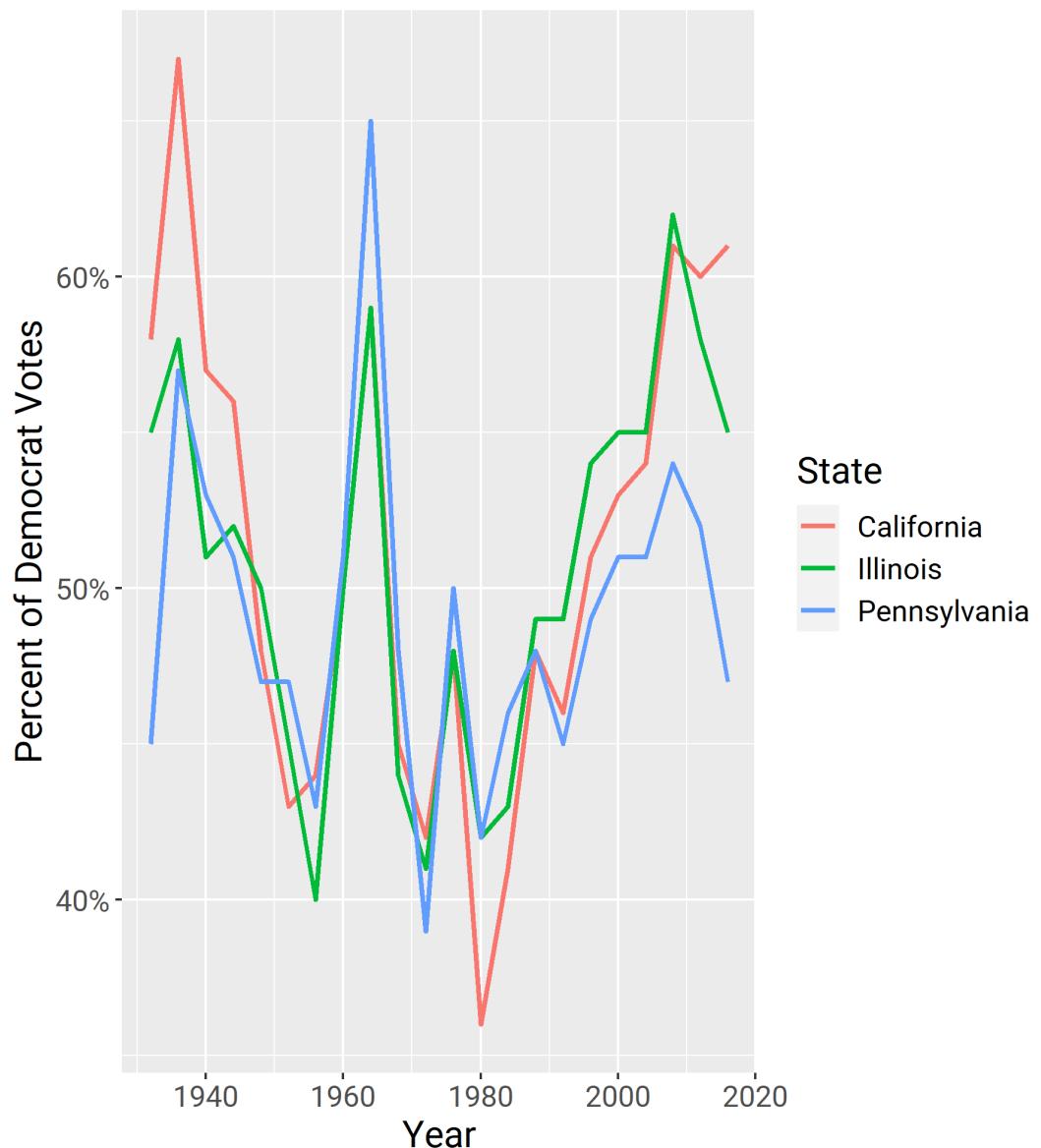
```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  theme(text = element_text(size = 15))
```

## Go Vote!



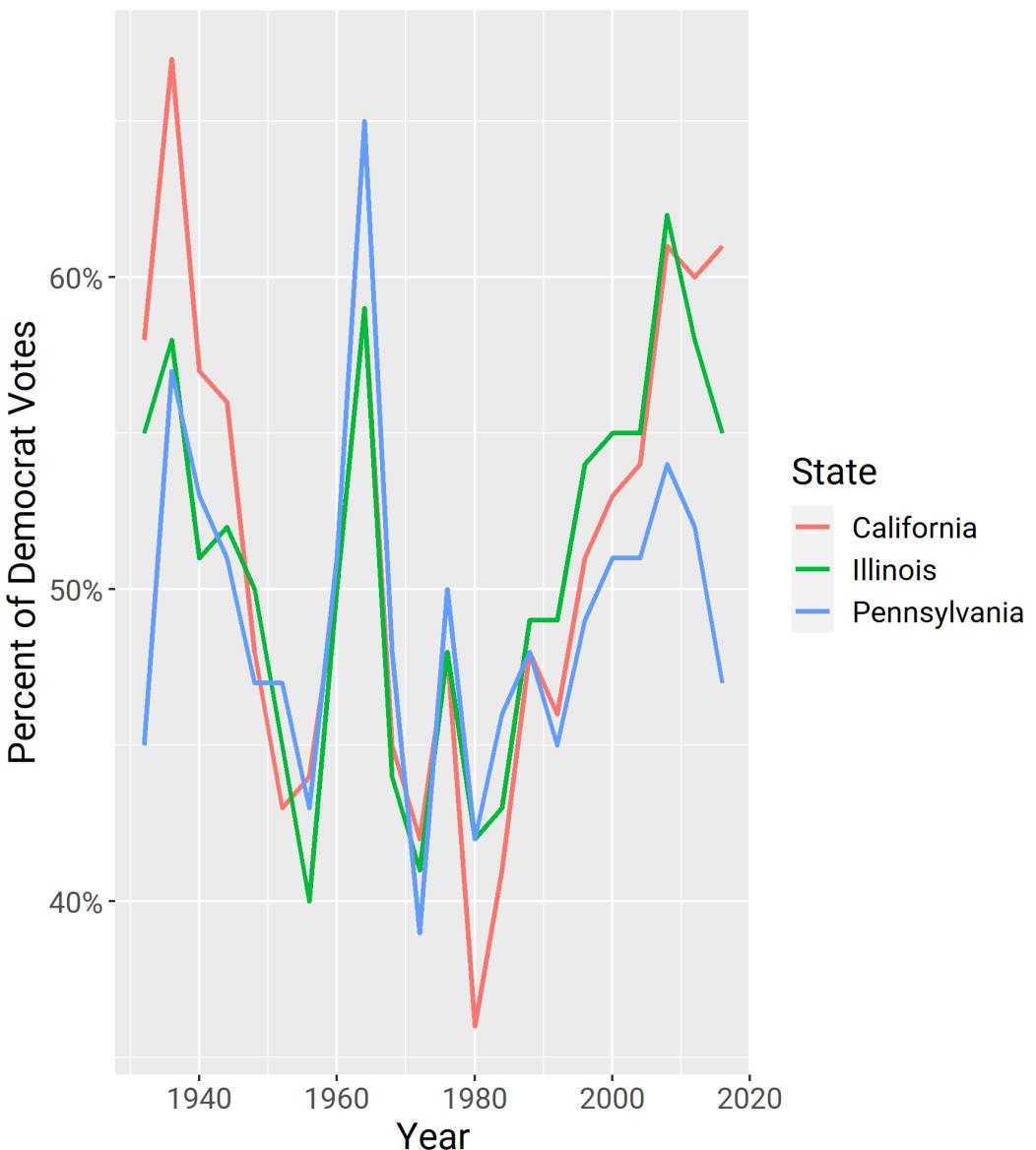
```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  theme(text = element_text(size = 15)) +  
  theme(plot.title = element_text(size = 20))
```

## Go Vote!



```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  theme(text = element_text(size = 15)) +  
  theme(plot.title = element_text(size = 20)) +  
  theme(plot.title = element_text(family = "Roboto Slab"))
```

# Go Vote!



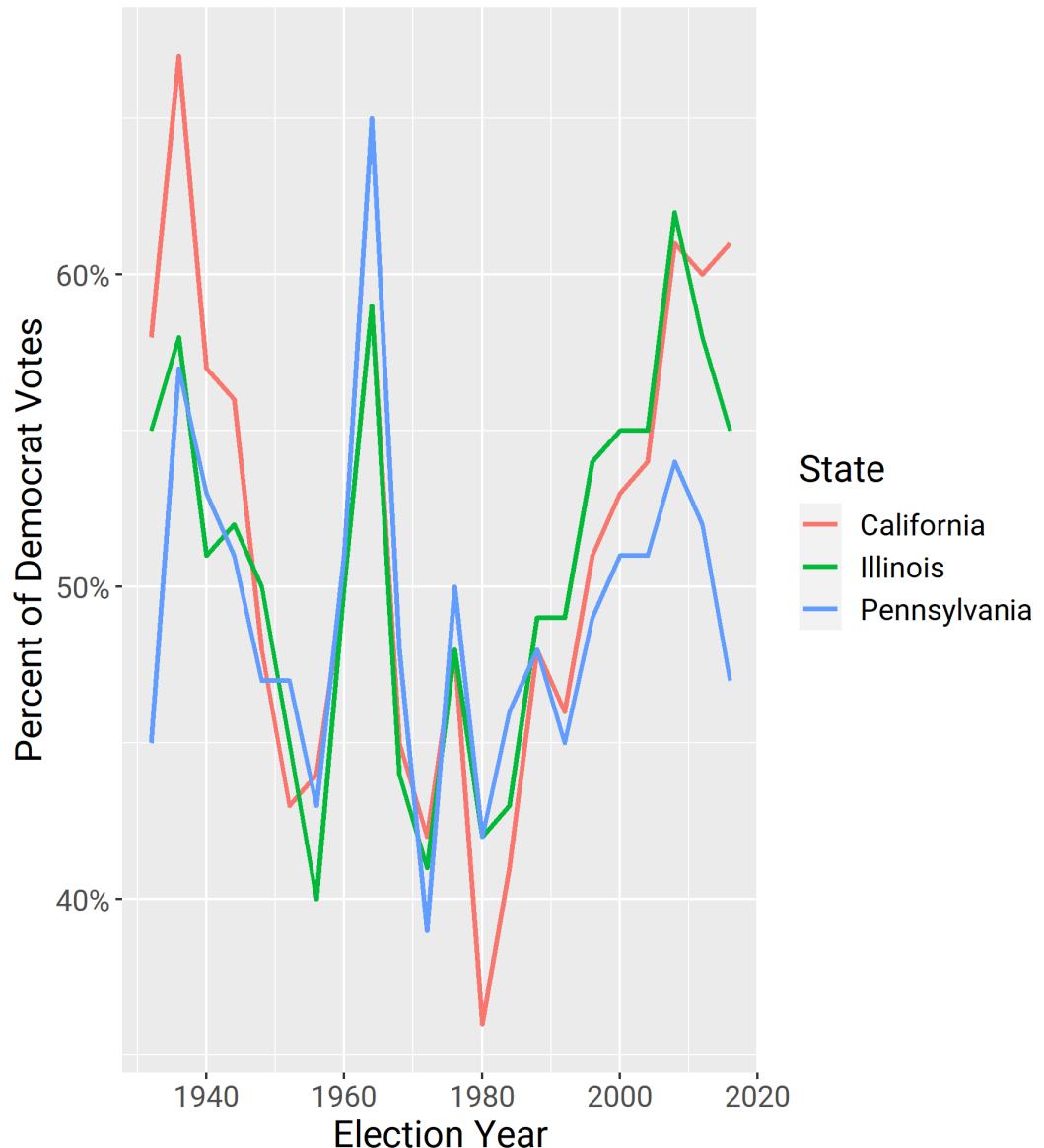
```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  theme(text = element_text(size = 15)) +  
  theme(plot.title = element_text(size = 20)) +  
  theme(plot.title = element_text(family = "Roboto Slab")) +  
  theme(plot.title.position = "plot")
```

# Go Vote!



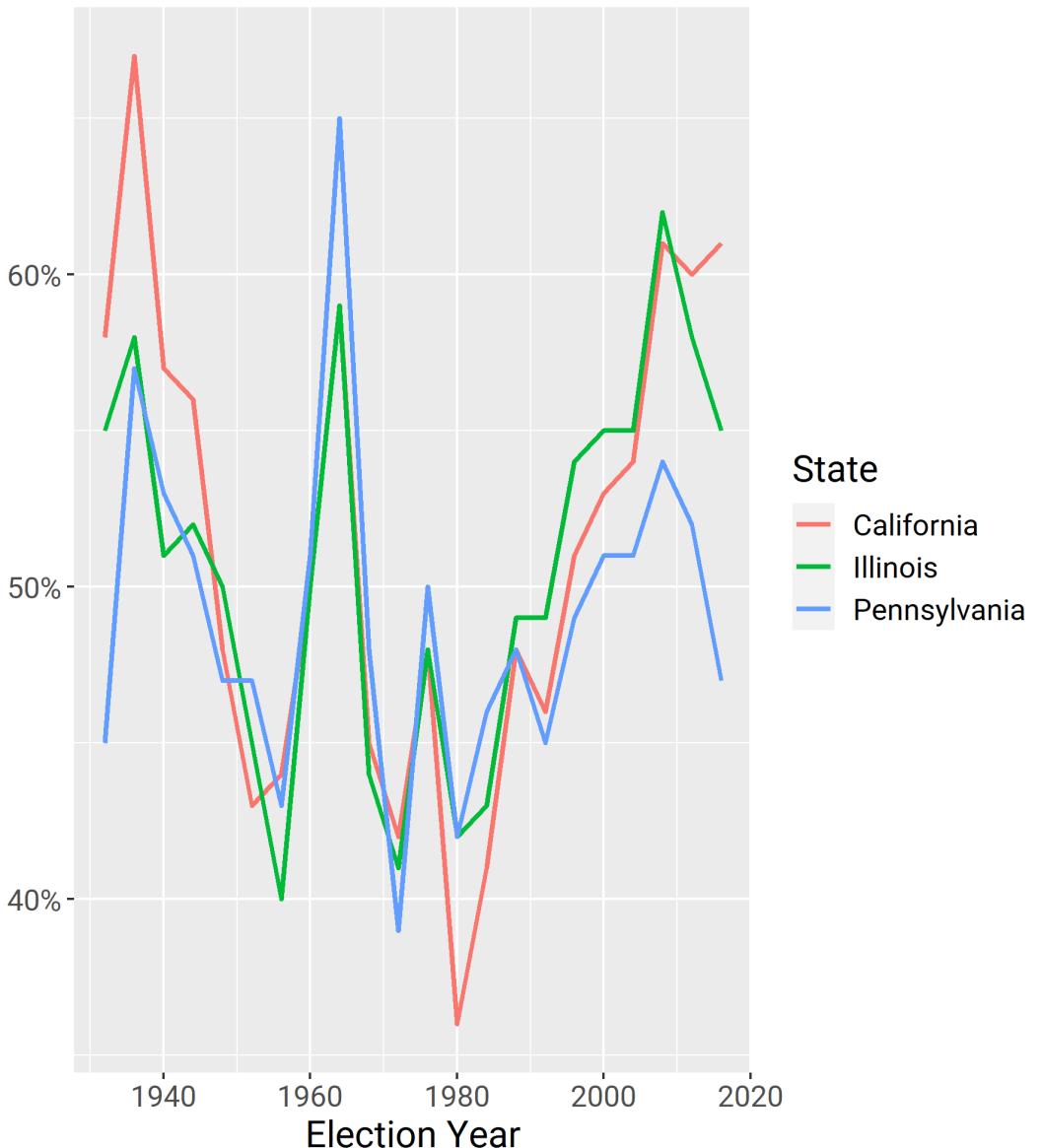
```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  theme(text = element_text(size = 15)) +  
  theme(plot.title = element_text(size = 20)) +  
  theme(plot.title = element_text(family = "Roboto Slab")) +  
  theme(plot.title.position = "plot") +  
  labs(x = "Election Year")
```

# Go Vote!



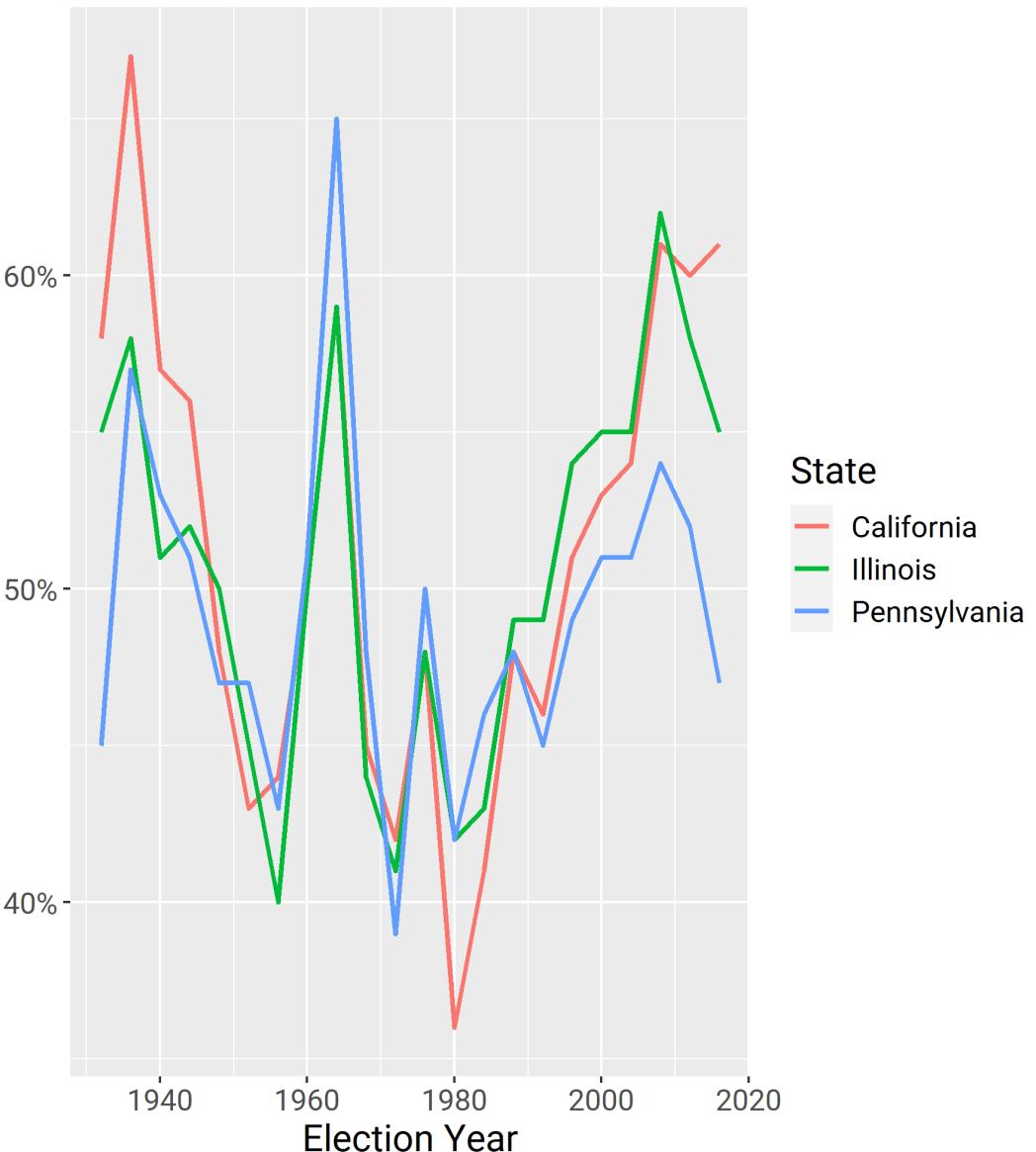
```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  theme(text = element_text(size = 15)) +  
  theme(plot.title = element_text(size = 20)) +  
  theme(plot.title = element_text(family = "Roboto Slab")) +  
  theme(plot.title.position = "plot") +  
  labs(x = "Election Year") +  
  labs(y = NULL)
```

# Go Vote!



```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  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")
```

## Percent of democrat votes by state



```
state_election_plot +  
  theme(text = element_text(family = "Roboto")) +  
  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!")
```

# Percent of democrat votes by state

## We're a swing state! Go vote!



```

state_election_plot +
  theme(text = element_text(family = "Roboto")) +
  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!*



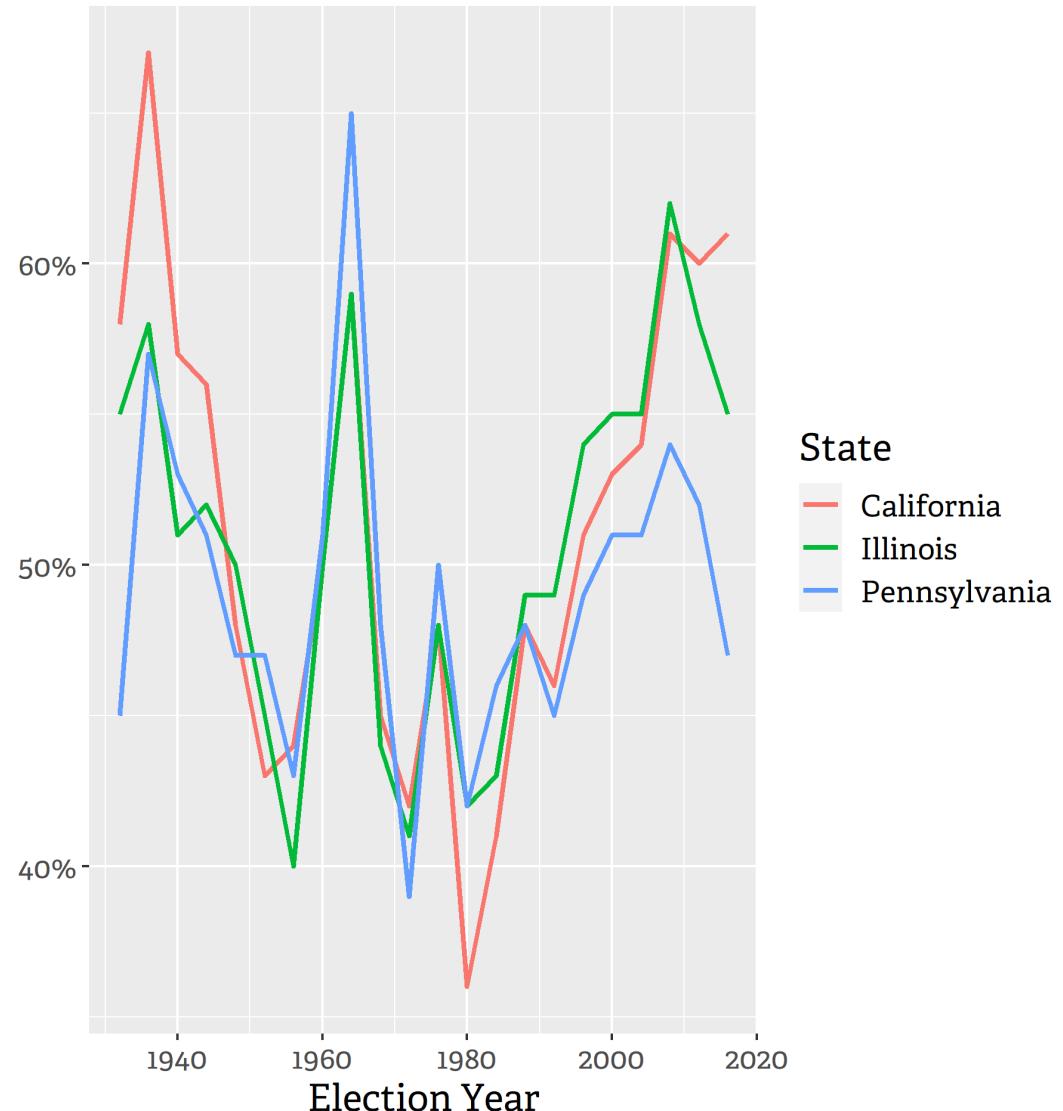
```

state_election_plot +
  theme(text = element_text(family = "Adelle")) +
  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!*



```

state_election_plot +
  theme(text = element_text(family = "Bitter")) +
  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!*



State

- California
- Illinois
- Pennsylvania

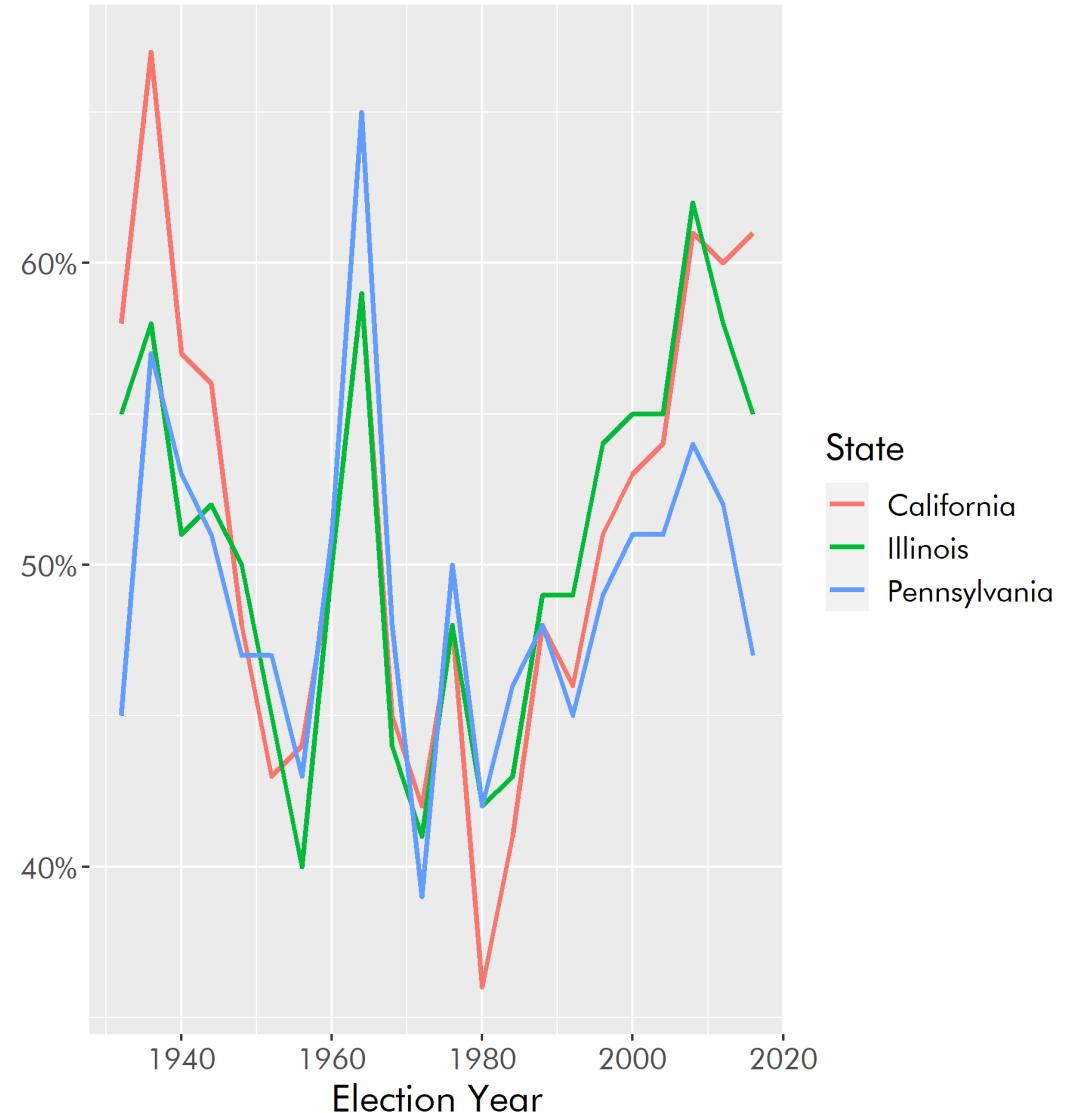
```

state_election_plot +
  theme(text = element_text(family = "Futura Bk BT")) +
  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!



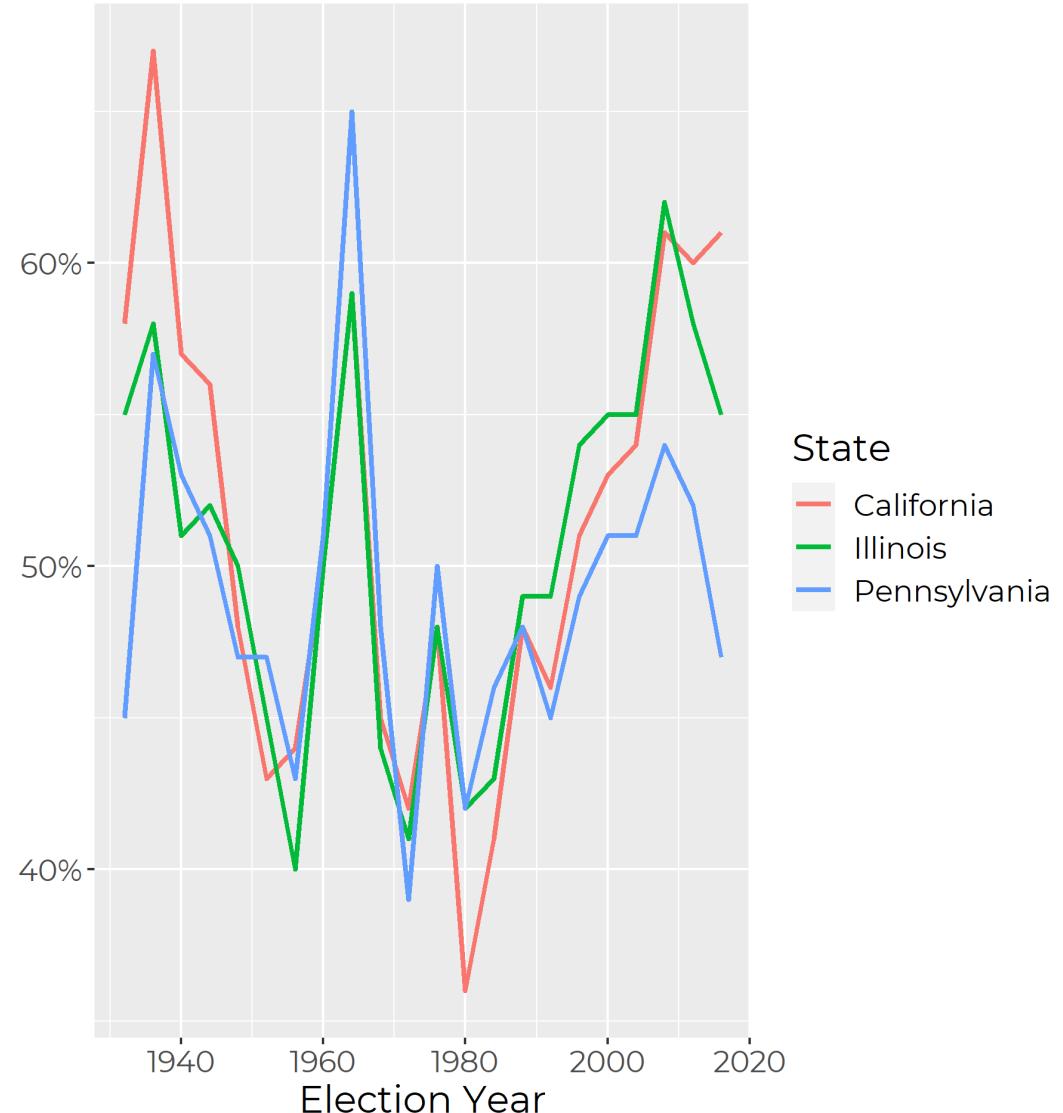
```

state_election_plot +
  theme(text = element_text(family = "Montserrat")) +
  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!*



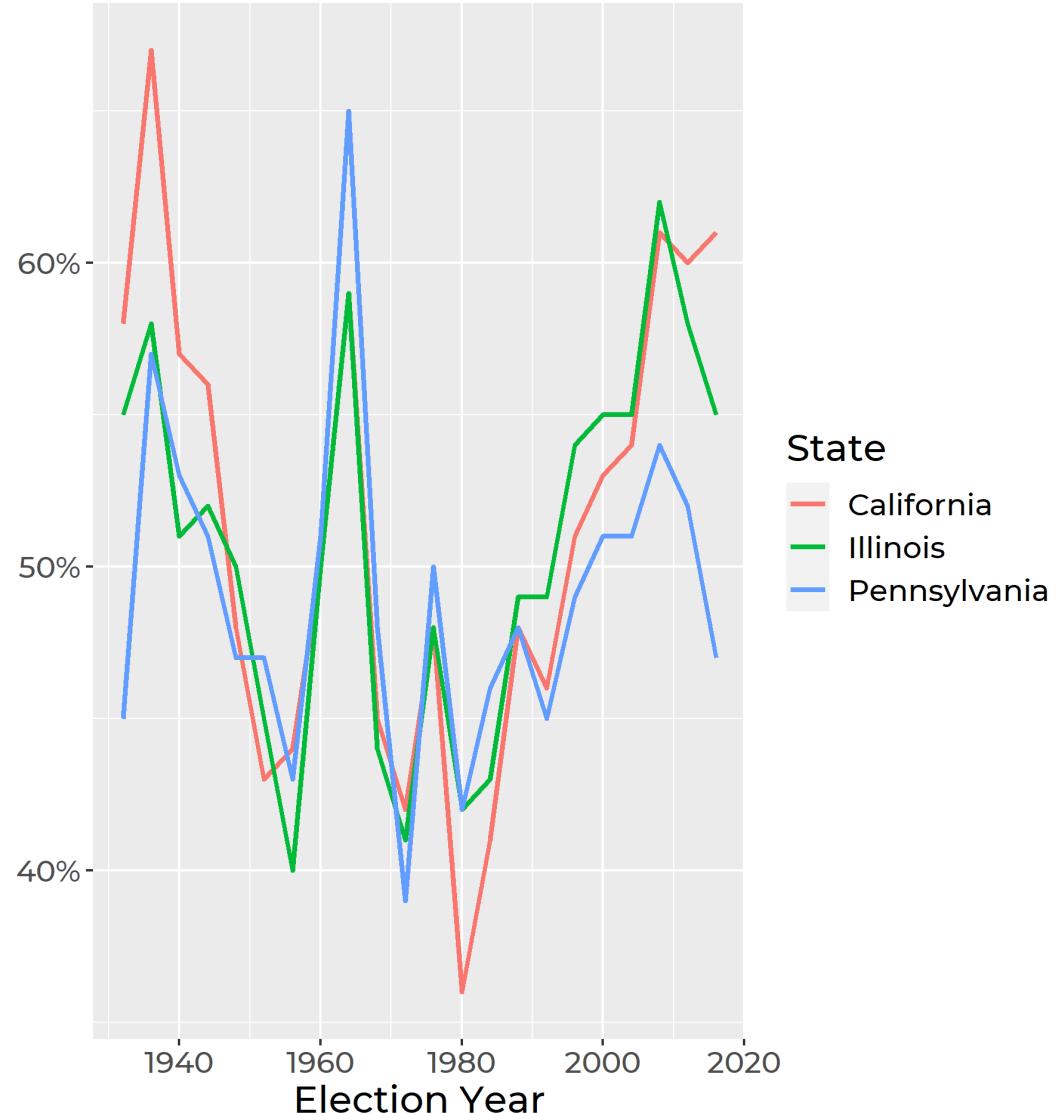
```

state_election_plot +
  theme(text = element_text(family = "Montserrat Medium")) +
  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!*



```
state_election_plot +  
  theme(text = element_text(family = "Arial")) +  
  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!*



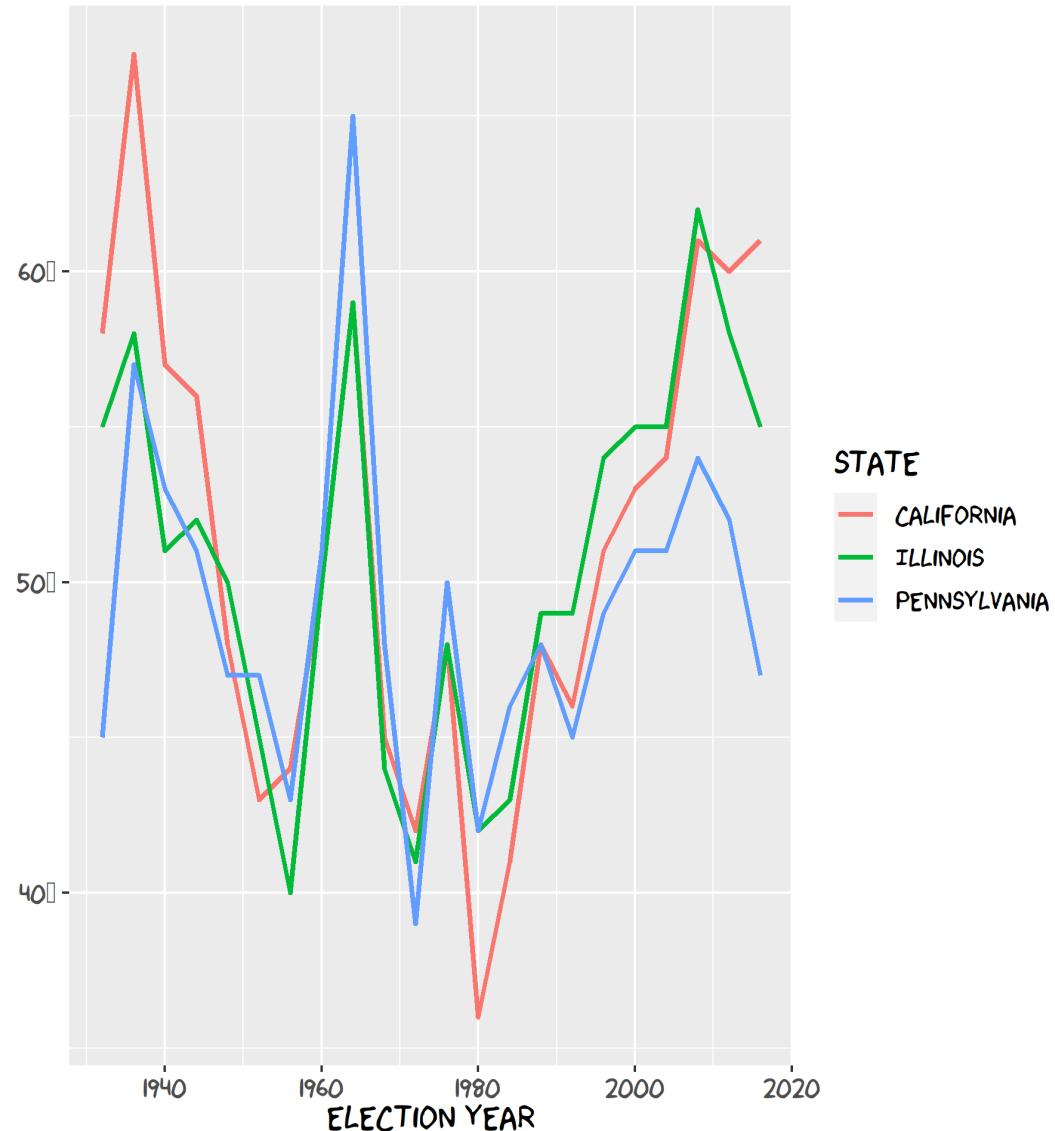
```

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!*



## STATE

- CALIFORNIA
- ILLINOIS
- PENNSYLVANIA

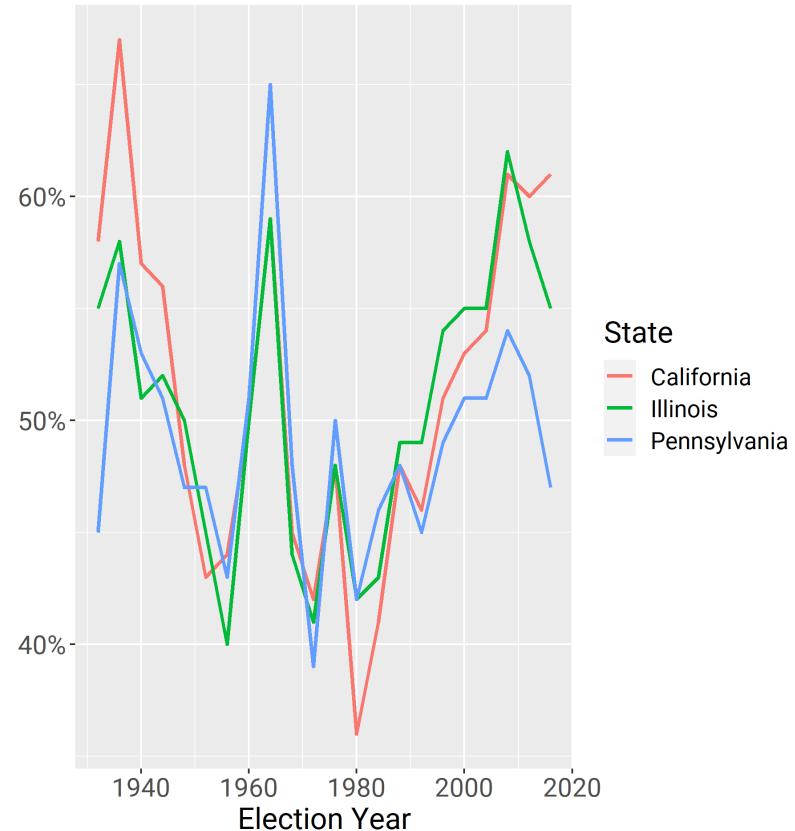
# 1. Text (End!)

```
state_election_plot +  
  theme(  
    text = element_text(size = 16, family = "Roboto"),  
    plot.title = element_text(size = 20, family = "Robo  
    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!"  
)
```

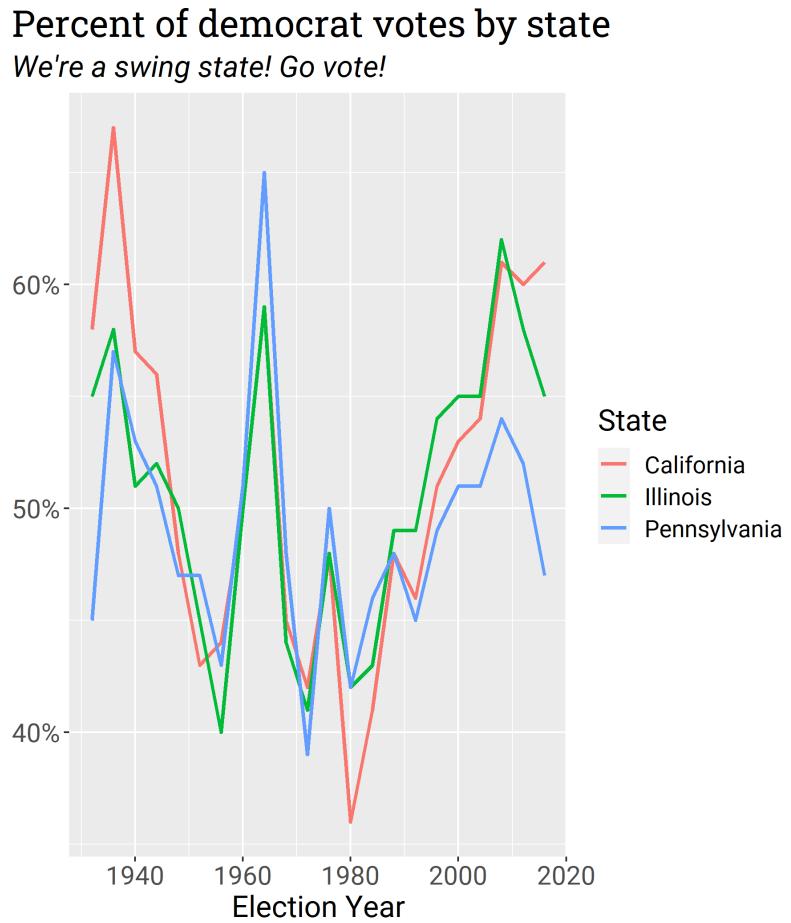
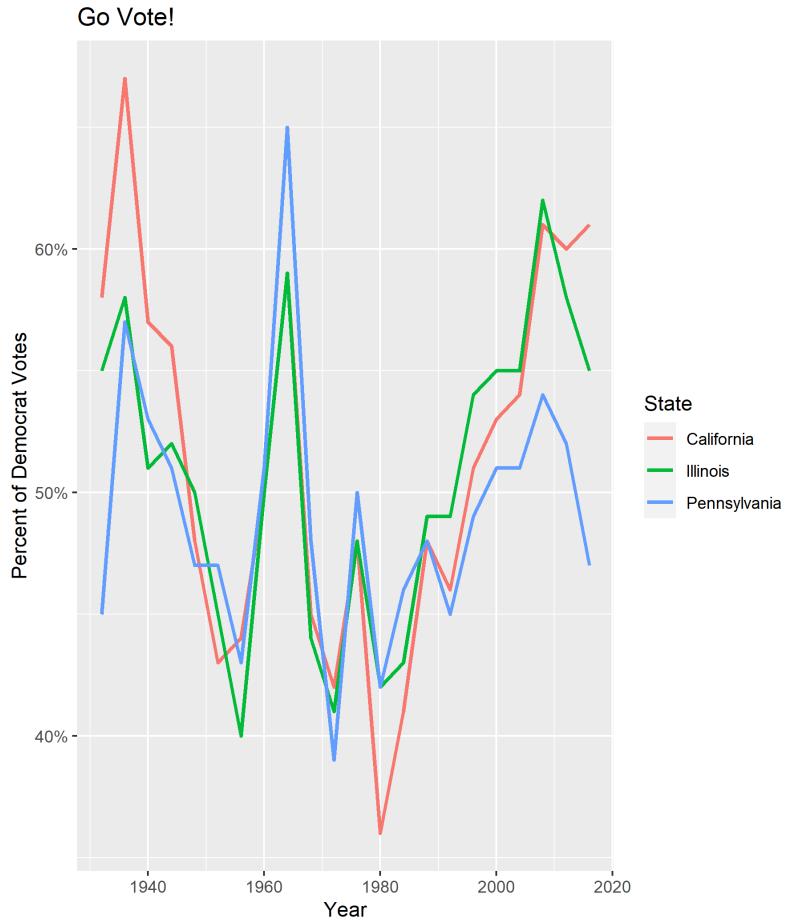
Save our progress!

```
state_election_plot_A
```

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

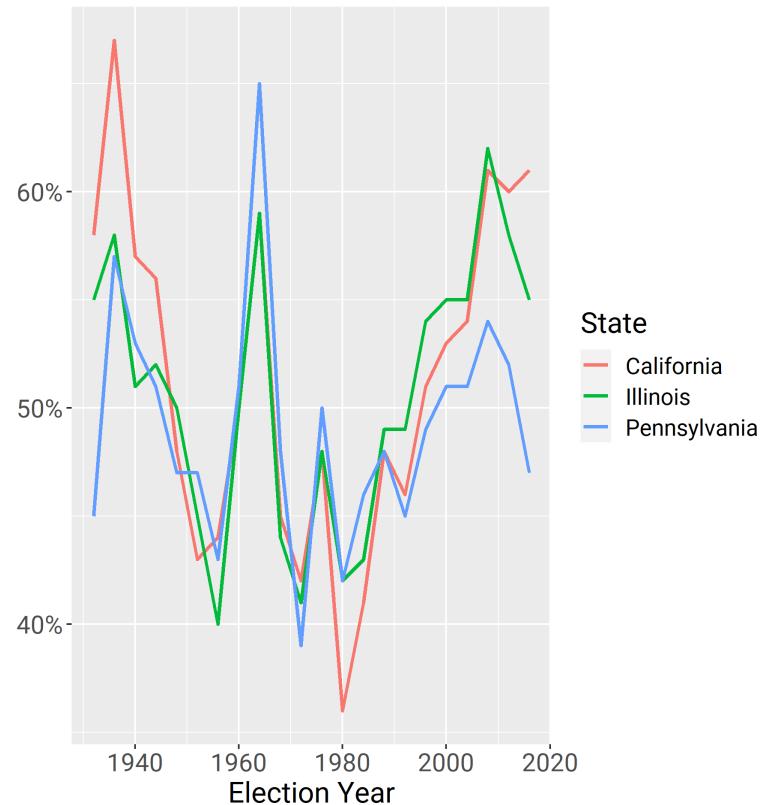


# 1. Text (Before-After)

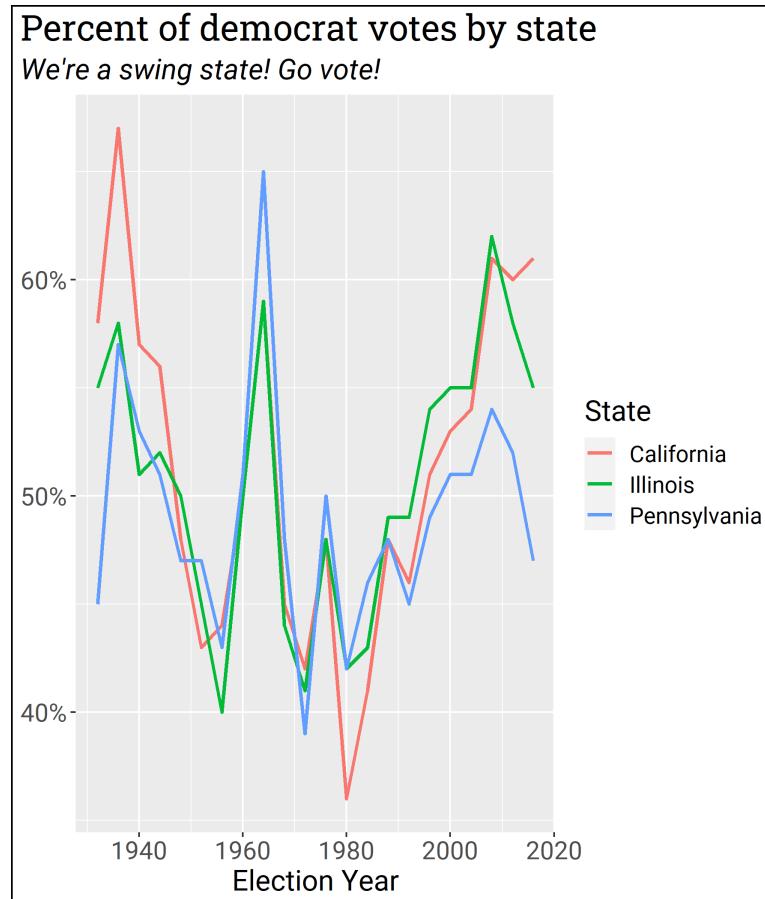


# 2. Margins & Spacing

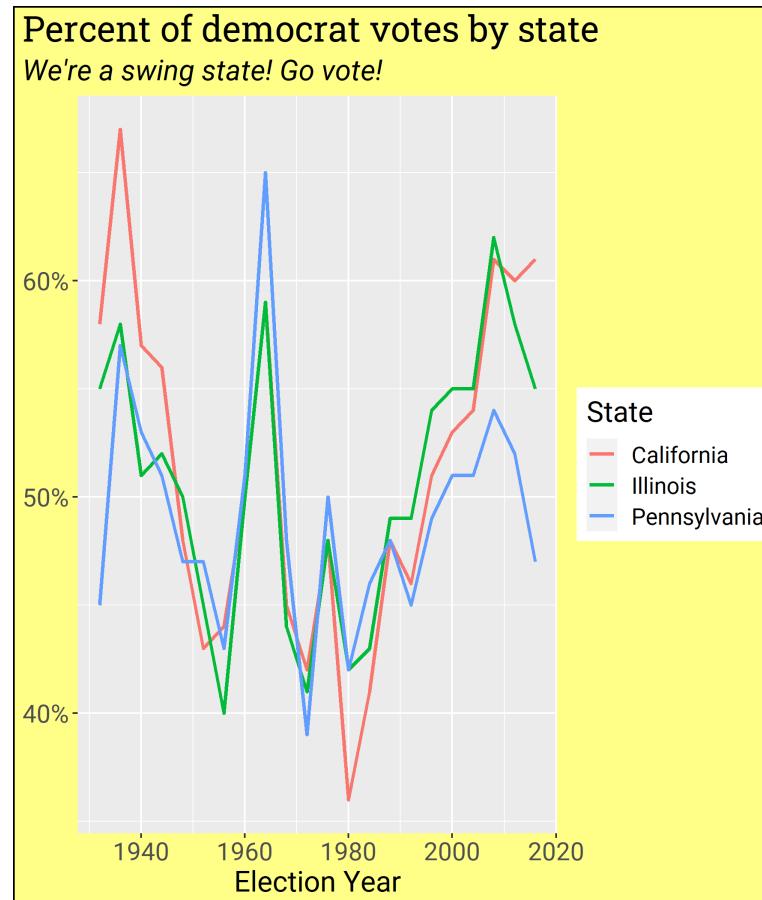
Percent of democrat votes by state  
*We're a swing state! Go vote!*



## 2. Margins & Spacing



## 2. Margins & Spacing



## 2. Margins & Spacing

Base `{ggplot2}` themes don't have *great* margin defaults for:

# 2. Margins & Spacing

Base `{ggplot2}` themes don't have *great* margin defaults for:

- Margins around plot

## 2. Margins & Spacing

Base `{ggplot2}` themes don't have *great* margin defaults for:

- Margins around plot
- Margins between plot title and panel

## 2. Margins & Spacing

Base `{ggplot2}` themes don't have *great* margin defaults for:

- Margins around plot
- Margins between plot title and panel
- Margins between axis titles and panel

# 2. Margins & Spacing

Base `{ggplot2}` themes don't have *great* margin defaults for:

- Margins around plot
- Margins between plot title and panel
- Margins between axis titles and panel
- Margins between axis texts and axis title

# 2. Margins & Spacing

Base {ggplot2} themes don't have *great* margin defaults for:

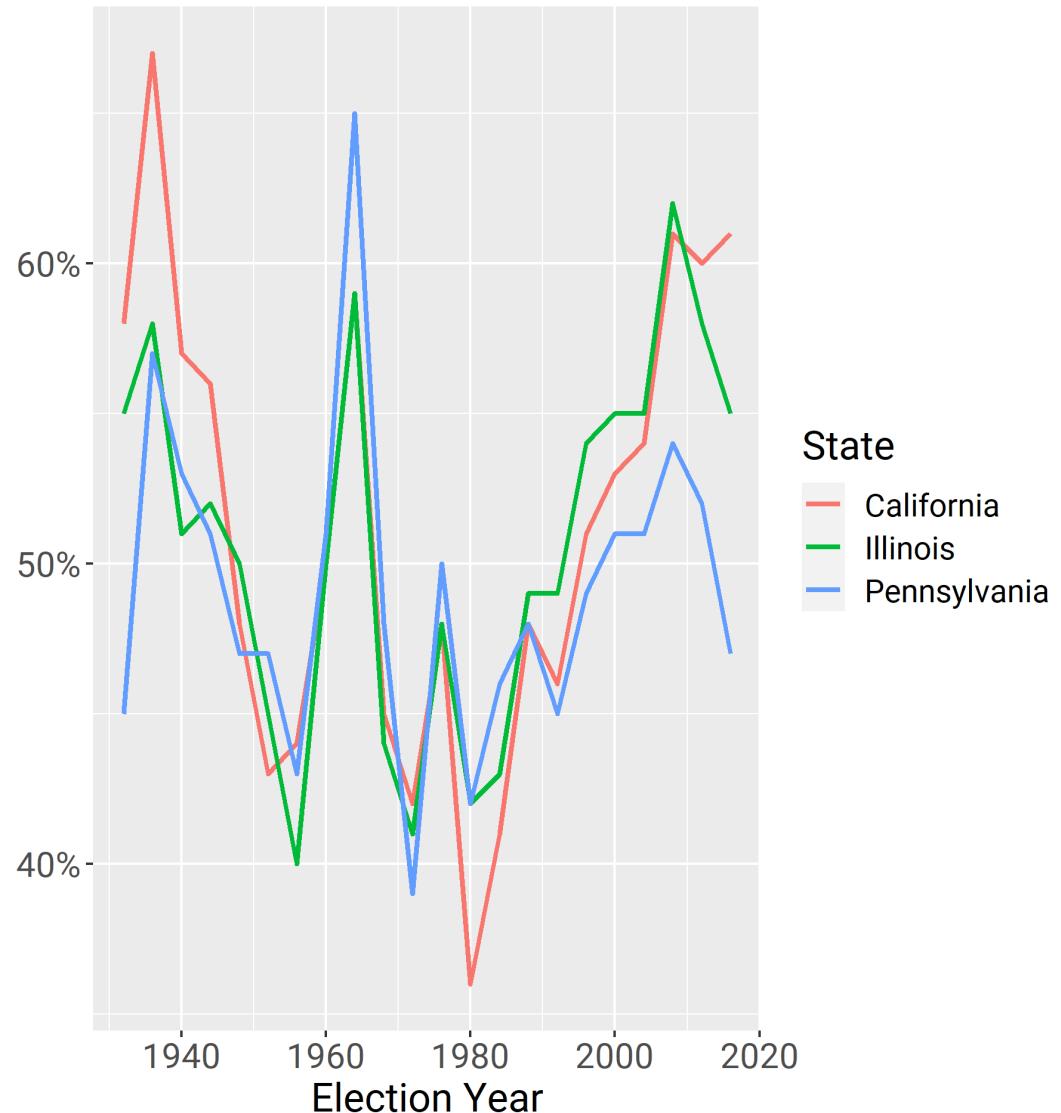
- Margins around plot
- Margins between plot title and panel
- Margins between axis titles and panel
- Margins between axis texts and axis title

Know your margin/spacing elements!

- `margin(t = 0, r = 0, b = 0, l = 0, unit = c("pt", "mm", "cm", "in"))`
- `hjust`, `vjust`, and `lineheight` arguments in `element_text()`
- `expand` argument with `expansion()` in `scale_*`() and `coord_*`() layers

# Percent of democrat votes by state

*We're a swing state! Go vote!*



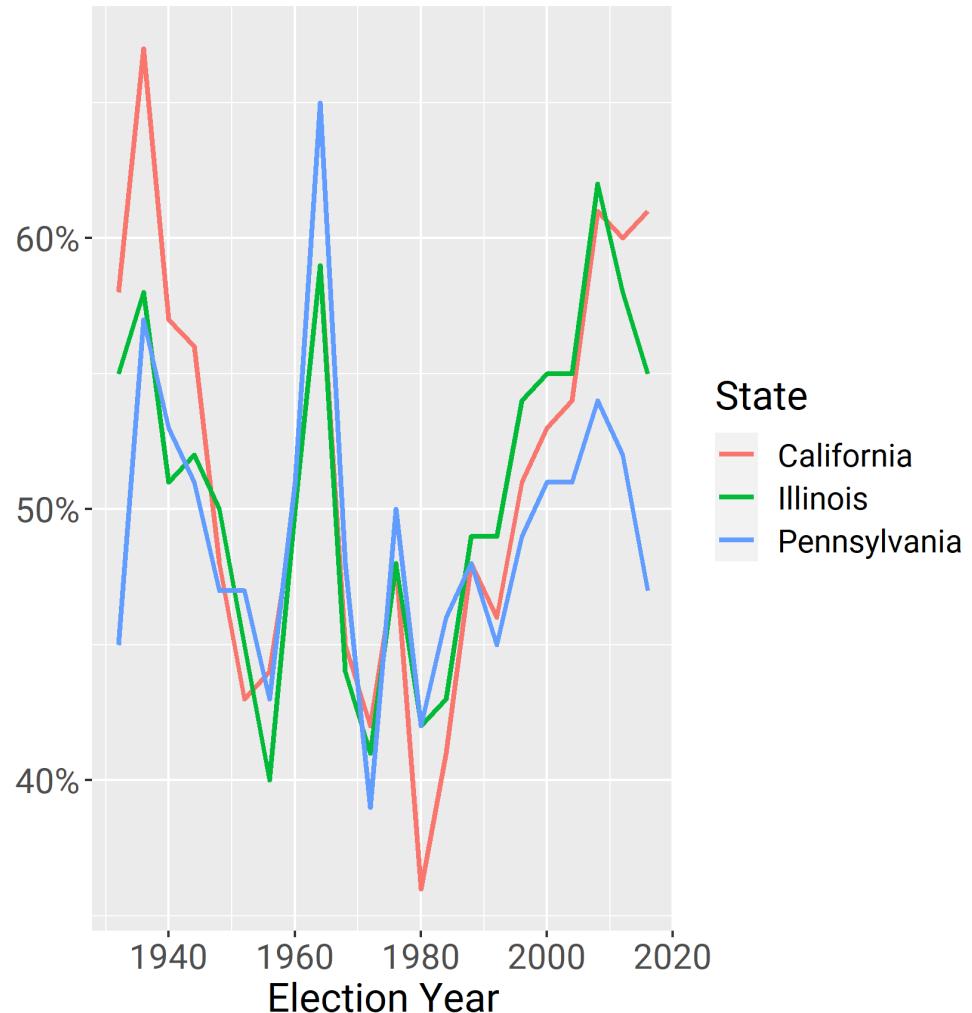
```
state_election_plot_A +  
  theme(  
    plot.background = element_rect(color = 'black')  
)
```

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



```
state_election_plot_A +  
  theme(  
    plot.background = element_rect(color = 'black')  
  ) +  
  theme(  
    plot.margin = margin(1, .8, .8, .8, "cm")  
  )
```

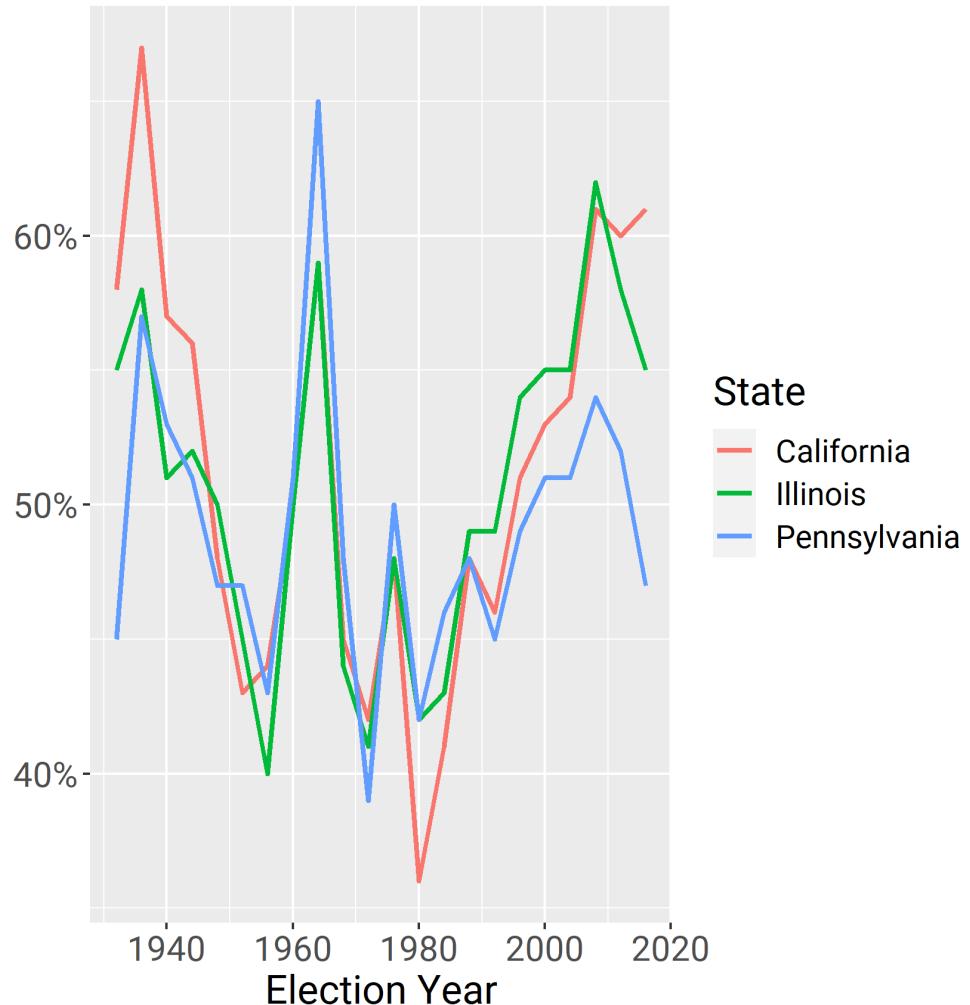
## Percent of democrat votes by state *We're a swing state! Go vote!*



```
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 =
```

## Percent of democrat votes by state

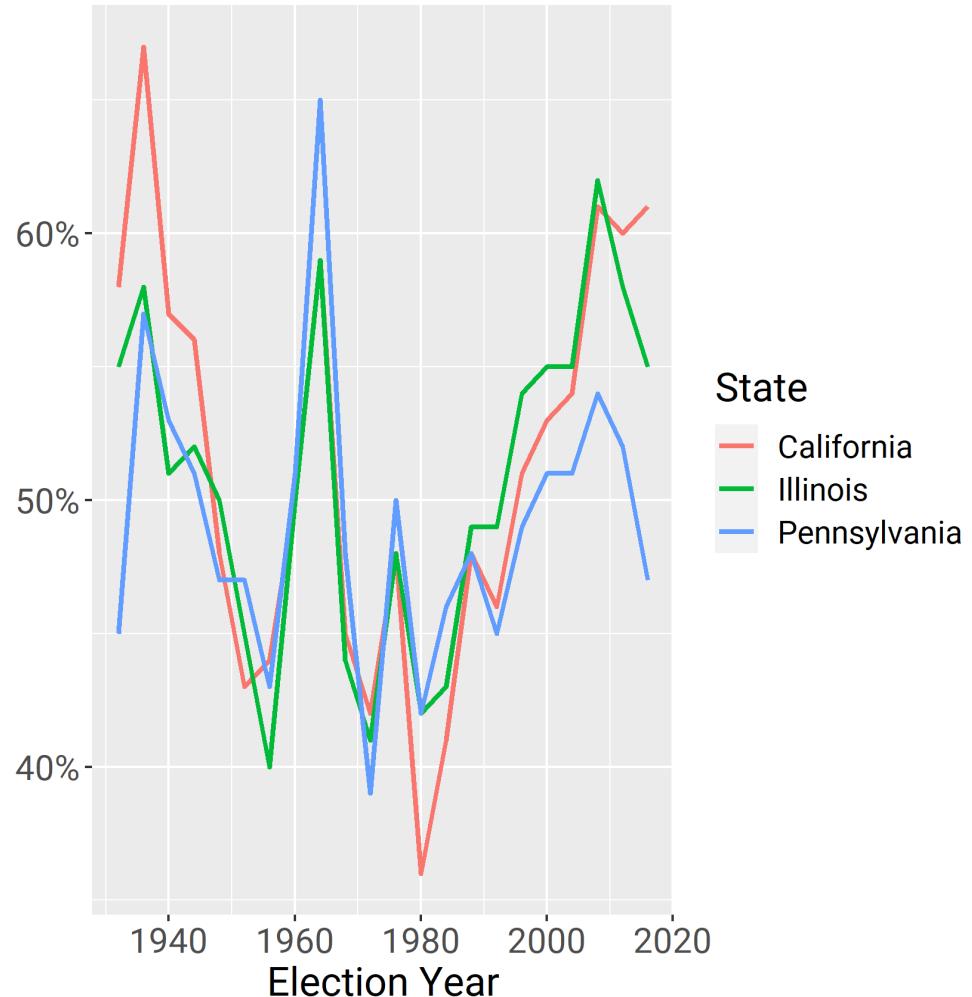
*We're a swing state! Go vote!*



```
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, uni  
)
```

## Percent of democrat votes by state

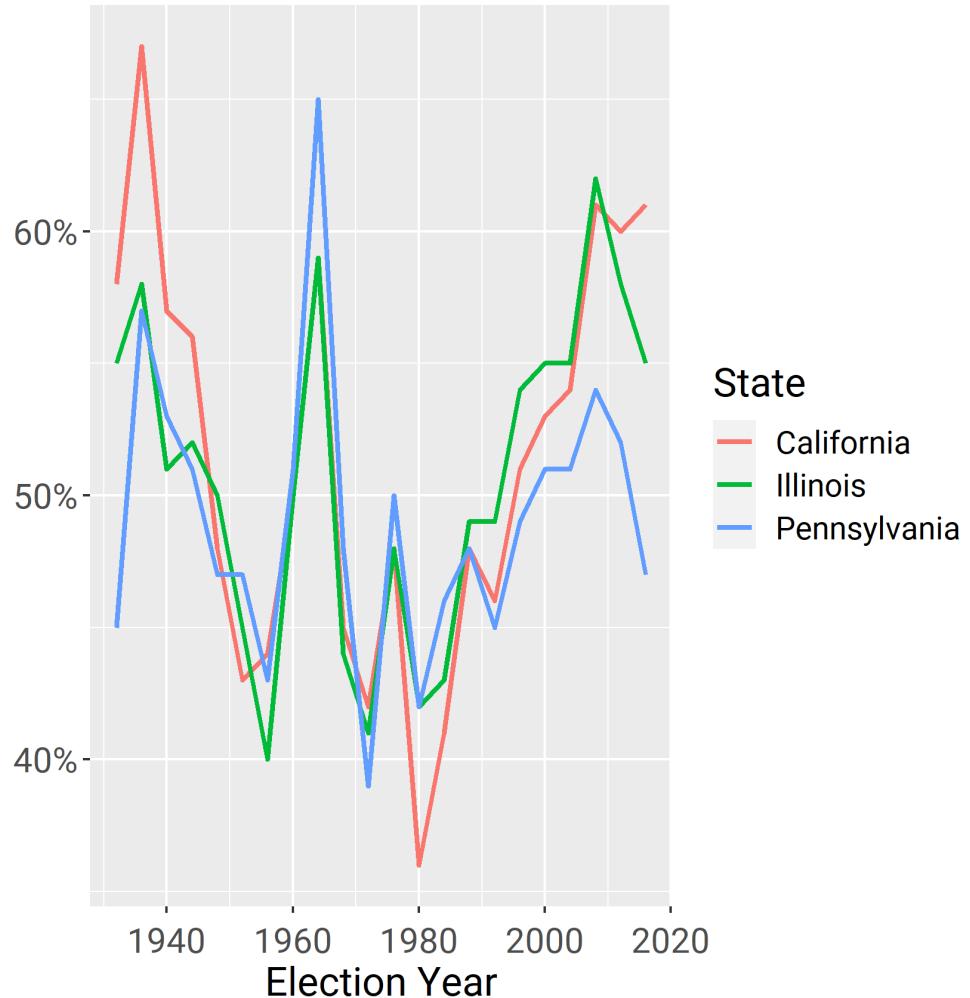
*We're a swing state! Go vote!*



```
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 =  
)
```

# Percent of democrat votes by state

*We're a swing state! Go vote!*



## State

- California
- Illinois
- Pennsylvania

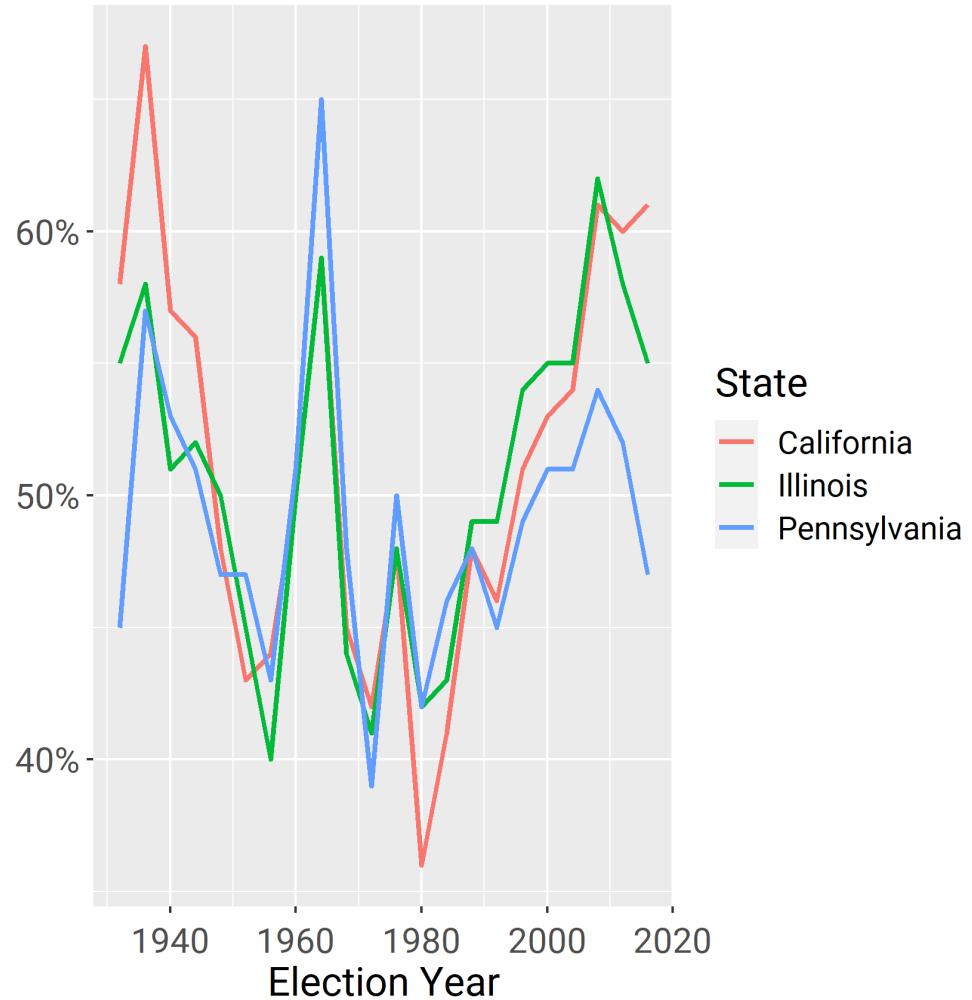
```

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 =
  )

```

# Percent of democrat votes by state

*We're a swing state! Go vote!*



## State

- California
- Illinois
- Pennsylvania

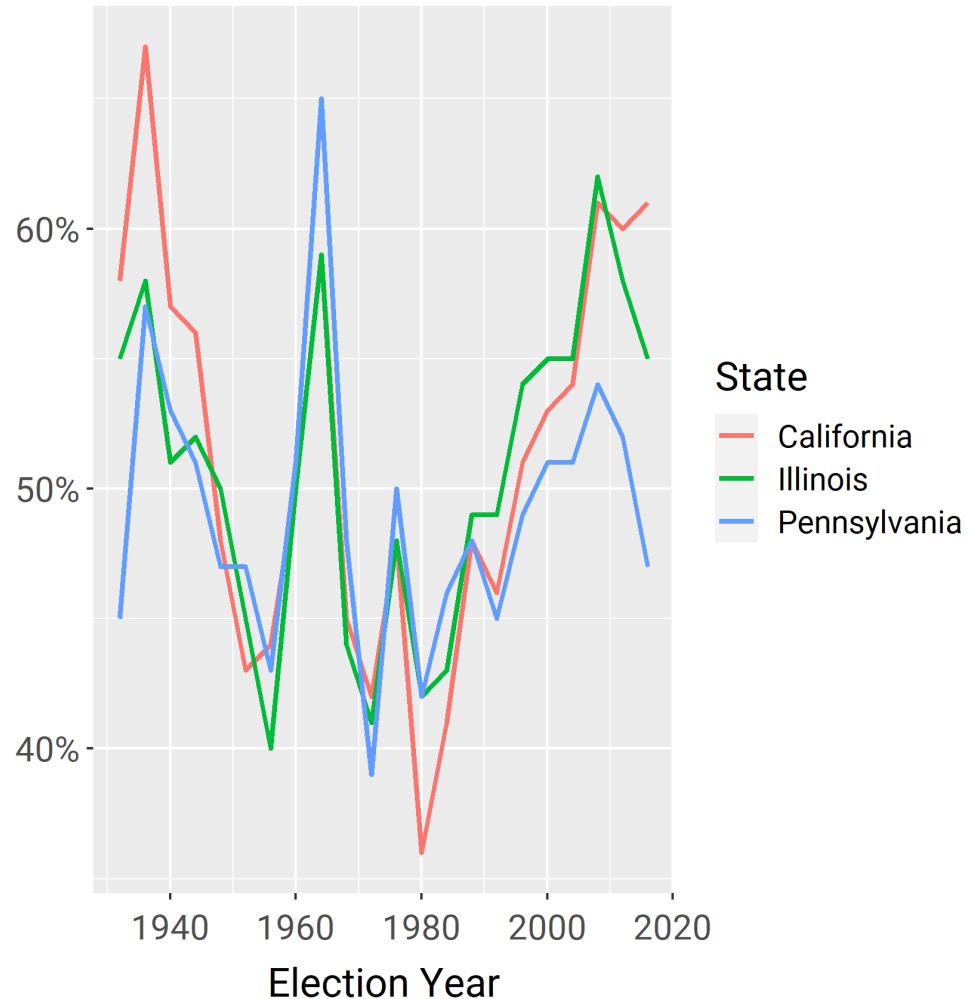
```

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 =
  )

```

# Percent of democrat votes by state

*We're a swing state! Go vote!*



State

- California
- Illinois
- Pennsylvania

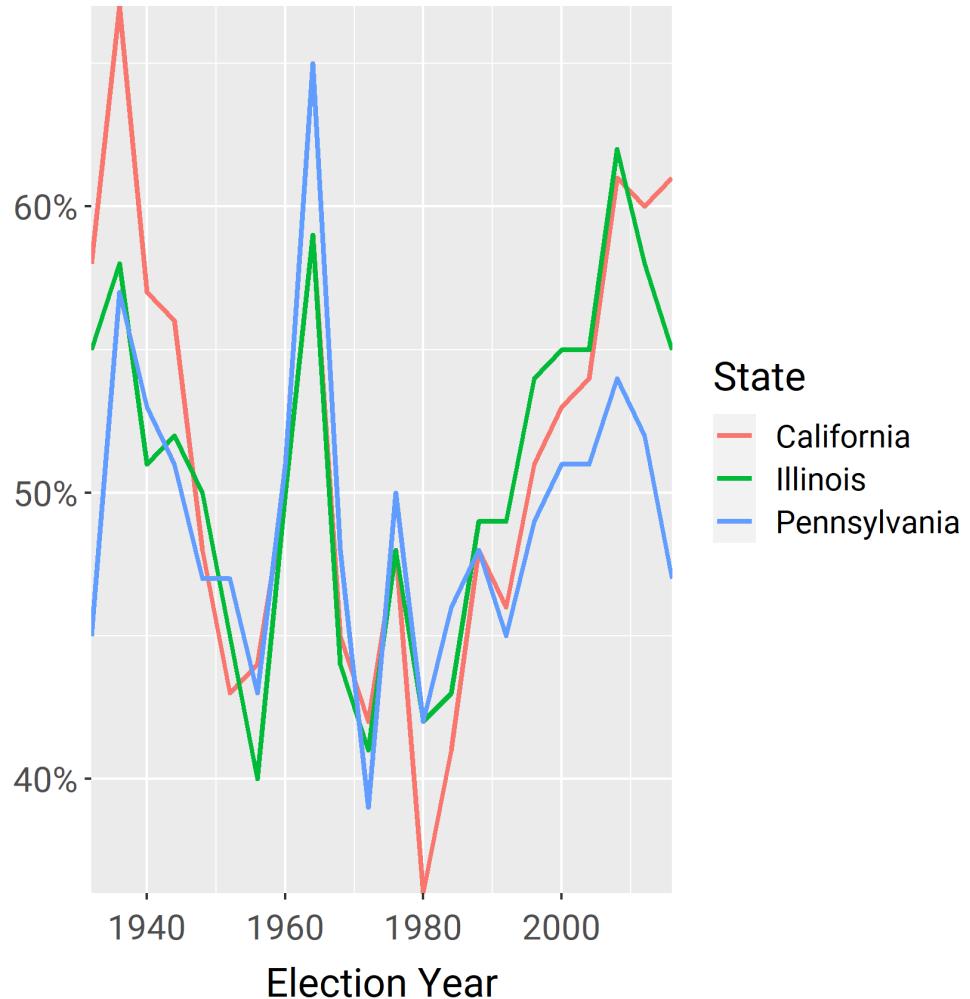
```

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 =
  ) +
  coord_cartesian(expand = FALSE)

```

# Percent of democrat votes by state

*We're a swing state! Go vote!*



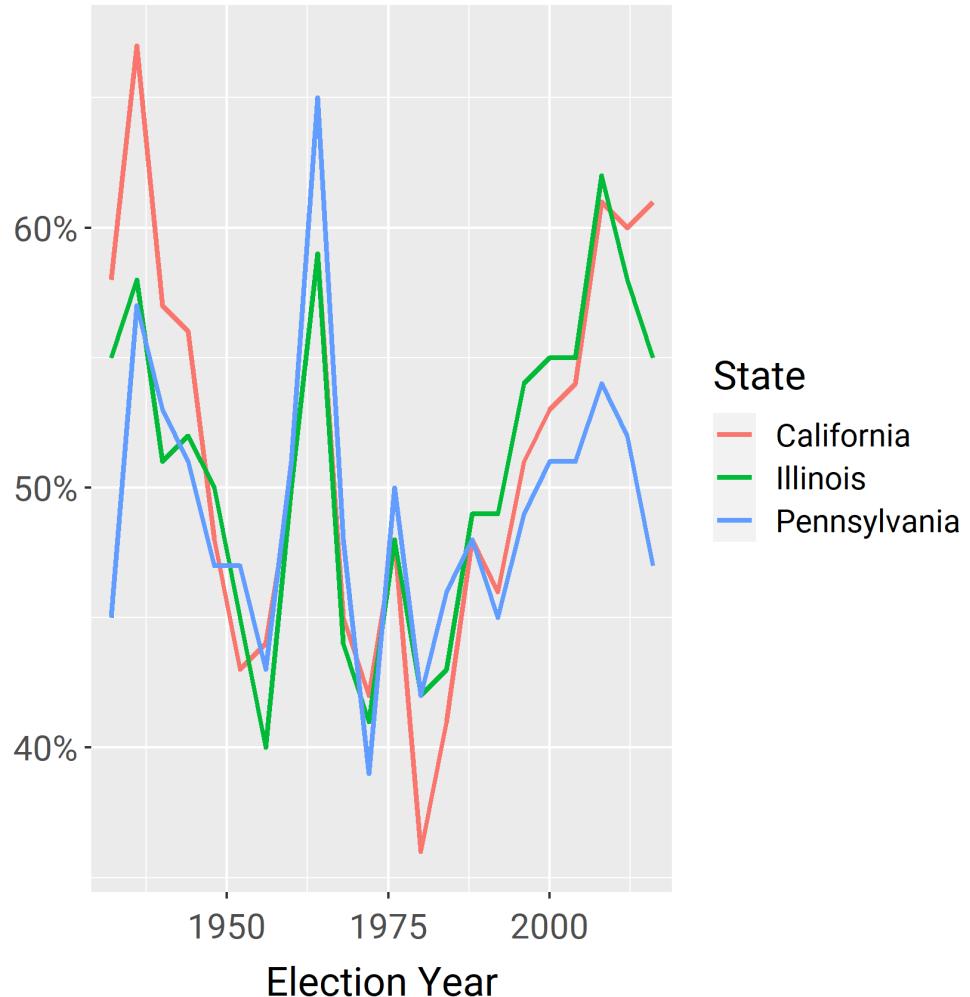
```

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!*



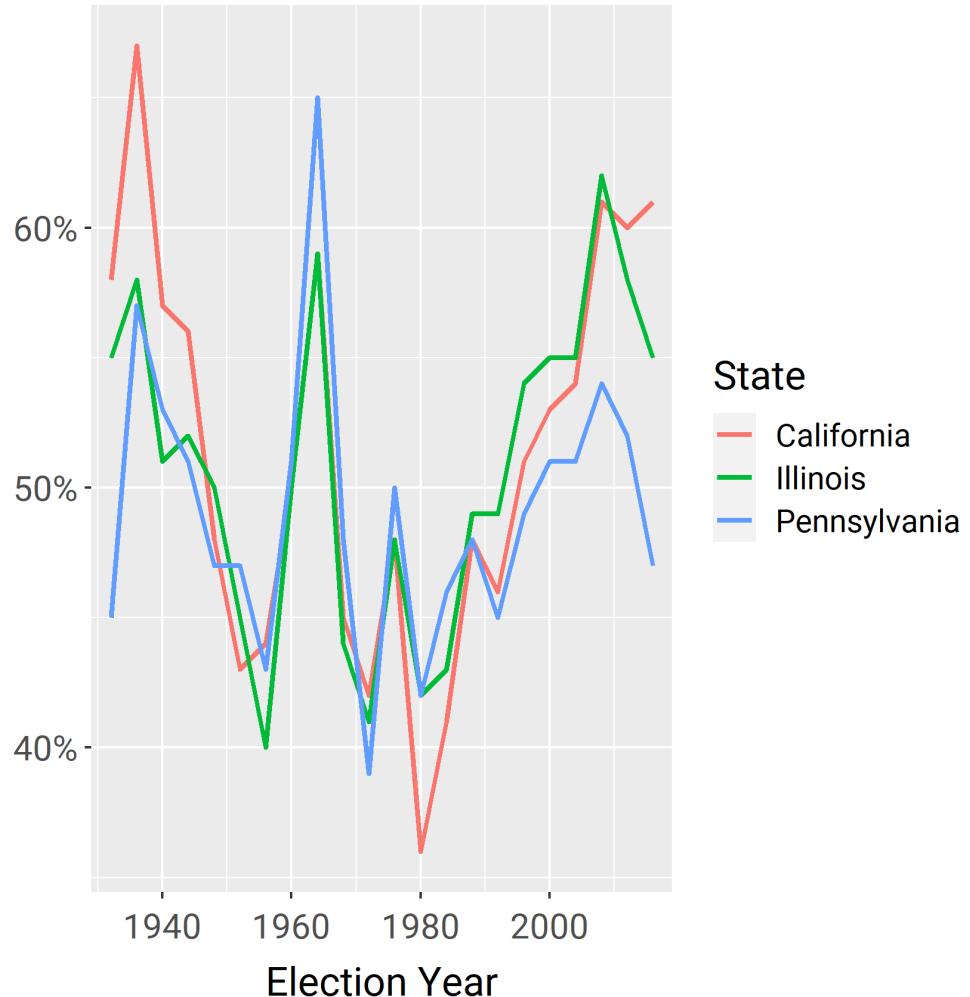
```

state_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!*



# 2. Margins & Spacing (End!)

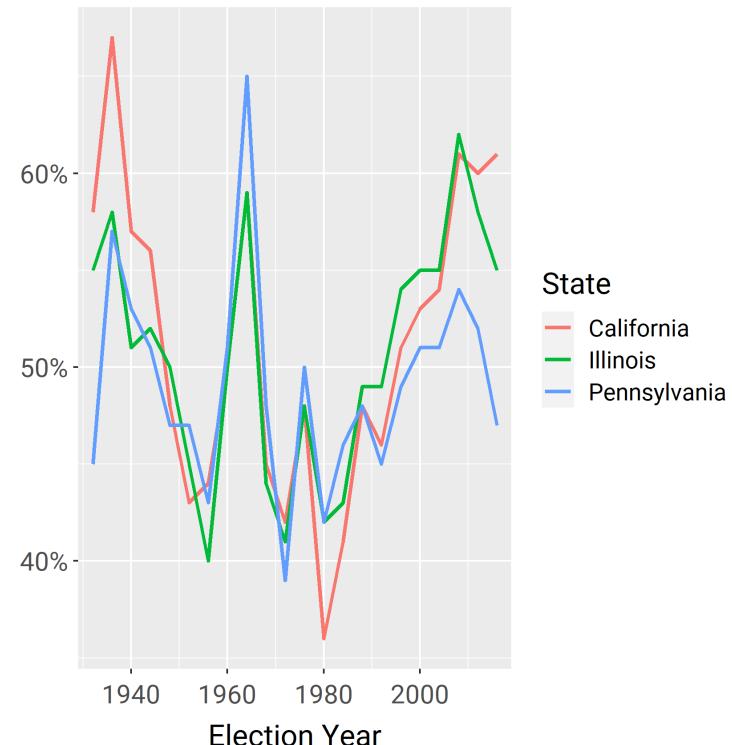
```
state_election_plot_A +  
  theme(  
    plot.margin = margin(1, .8, .8, .8, "cm"),  
    plot.title = element_text(margin = margin(b = .3, u  
    plot.subtitle = element_text(margin = margin(b = .3  
    axis.text.x = element_text(margin = margin(t = .2,  
    axis.text.y = element_text(margin = margin(r = .1,  
    axis.title.x = element_text(margin = margin(t = .3,  
) +  
  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

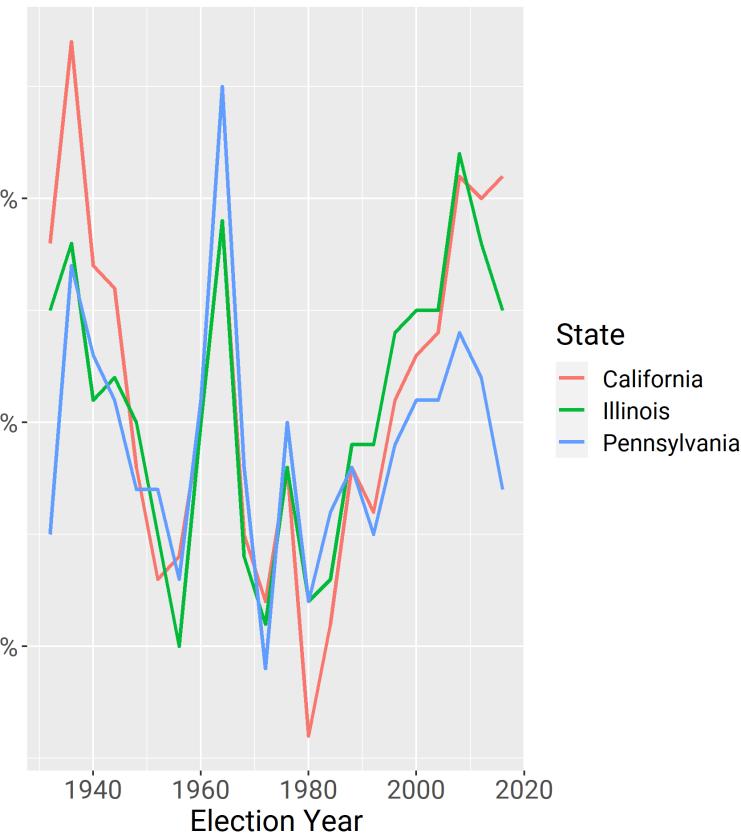
We're a swing state! Go vote!



## 2. Margins & Spacing (Before-After)

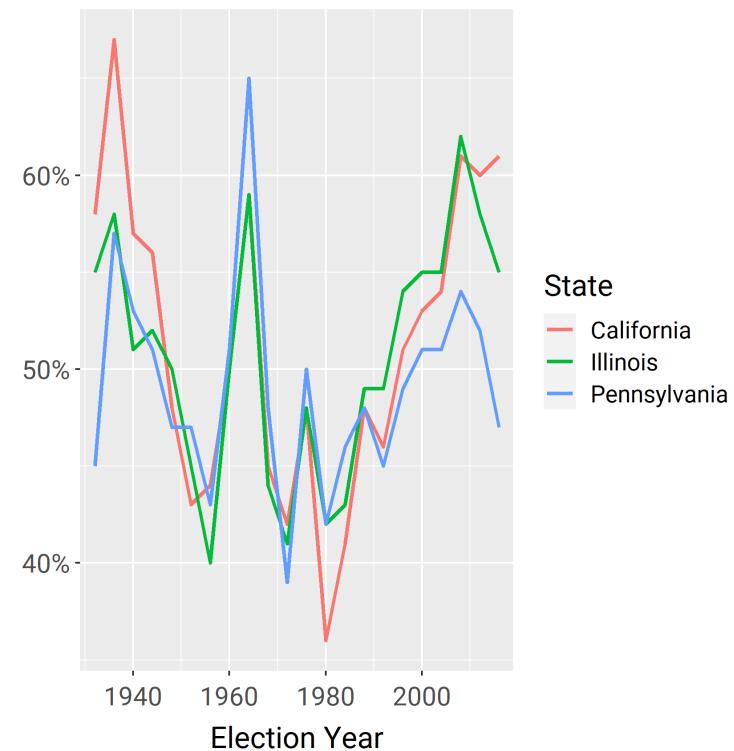
Percent of democrat votes by state

*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)

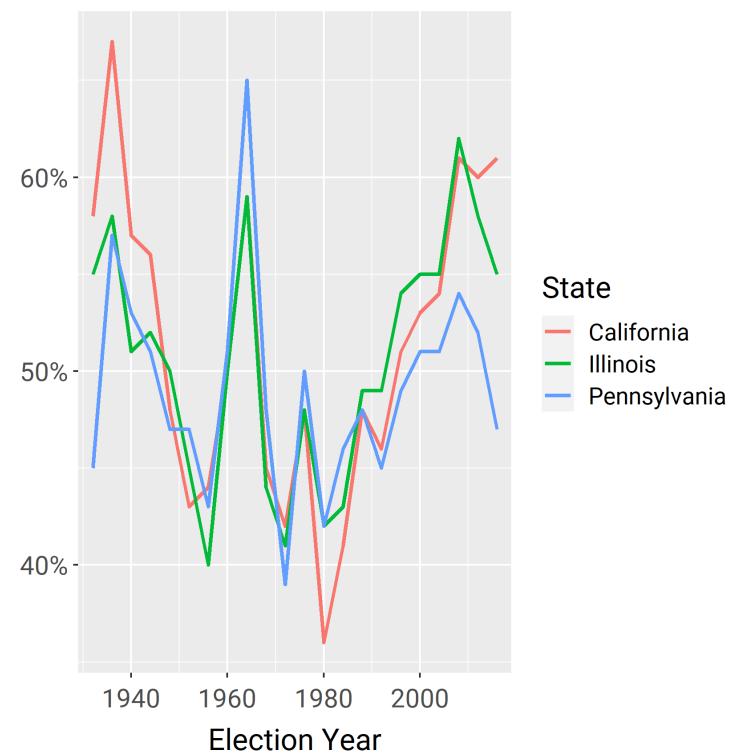
Percent of democrat votes by state

*We're a swing state! Go vote!*



Percent of democrat votes by state

*We're a swing state! Go vote!*



# 3. Legends

# 3. Legends

Legends are really hard:

# 3. Legends

Legends are really hard:

- They contain important info, but difficult to make them not look out of place

# 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

# 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

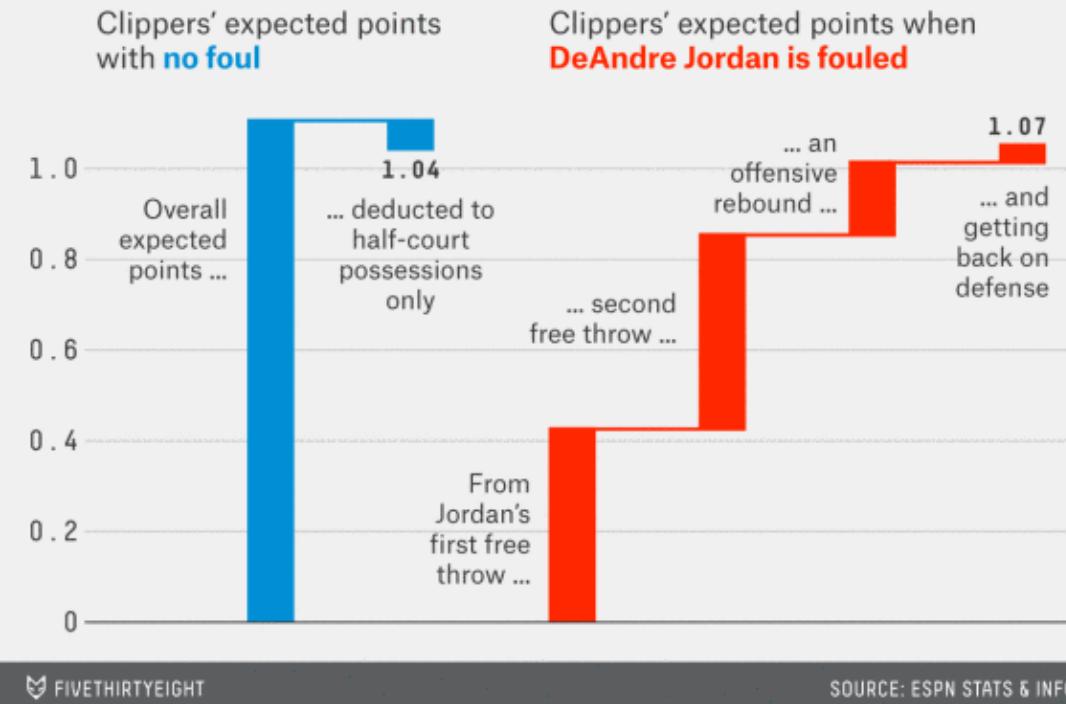
# 3. Legends

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?



# 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*

# 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*

But if you must have a legend, make it so that the reader is **naturally guided** to it

# 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*

But if you must have a legend, make it so that the reader is **naturally guided** to it

- Positioned at the top-left or top-center of the plot

# 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

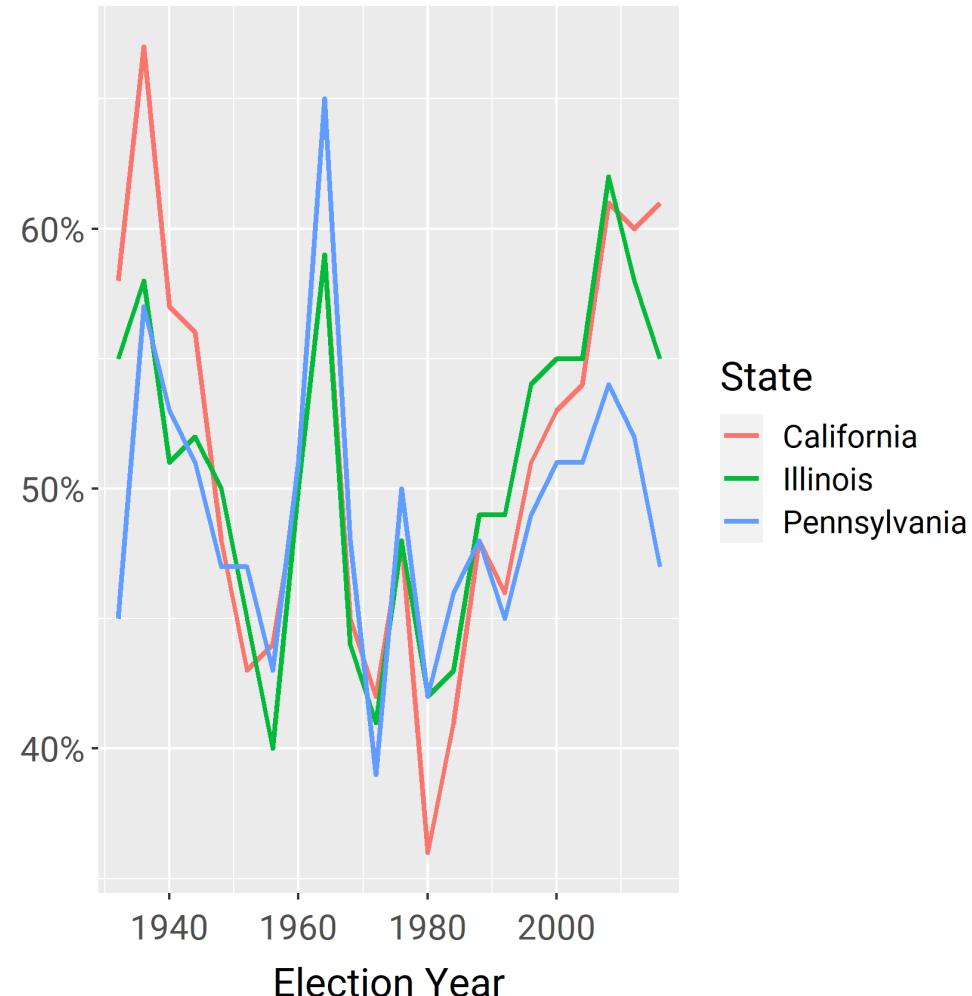
Alternative: consider *labeling the data directly*

But if you must have a legend, make it so that the reader is **naturally guided** to it

- Positioned at the top-left or top-center of the plot
- Blend smoothly into the rest of the plot (NOT make them stand out!)

# Percent of democrat votes by state

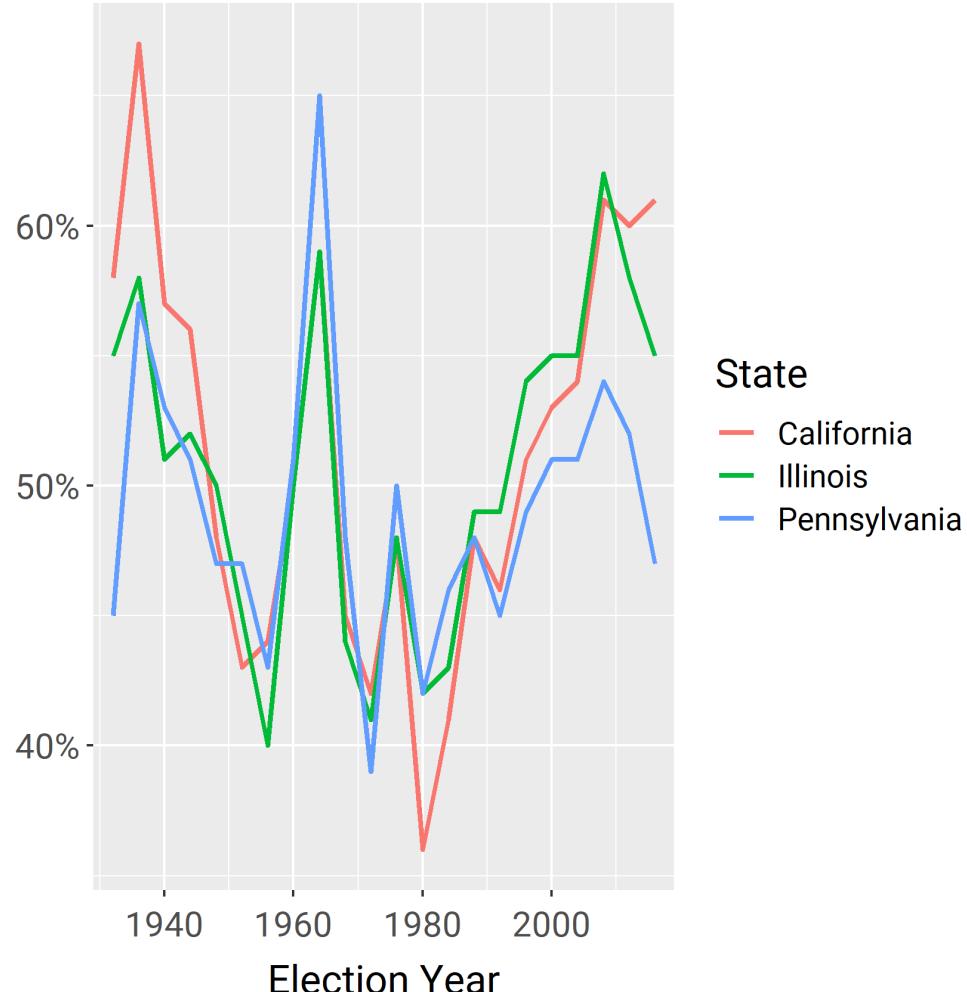
*We're a swing state! Go vote!*



```
state_election_plot_B +  
  theme(  
    legend.key = element_rect(fill = NA)  
)
```

## Percent of democrat votes by state

*We're a swing state! Go vote!*

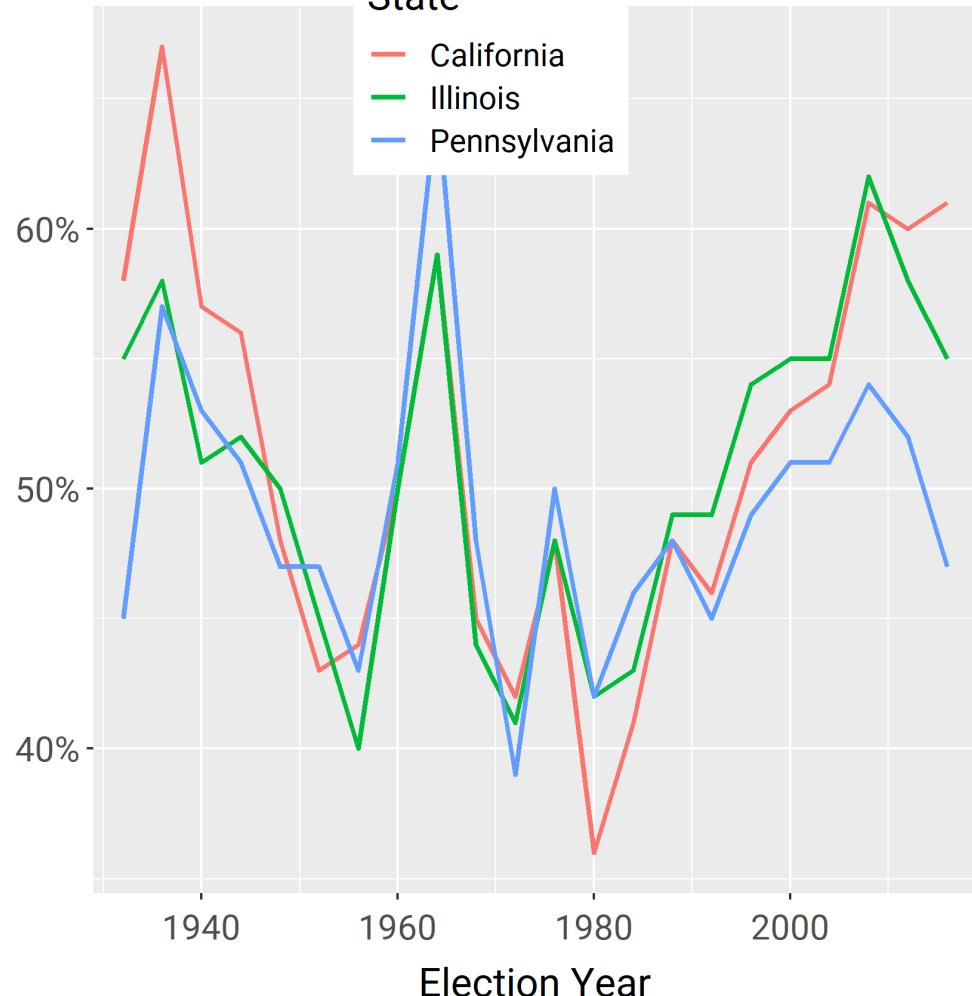


```
state_election_plot_B +  
  theme(  
    legend.key = element_rect(fill = NA)  
) +  
  theme(  
    legend.position = c(.45, .93)  
)
```

## Percent of democrat votes by state

We're a swing state! Go vote!

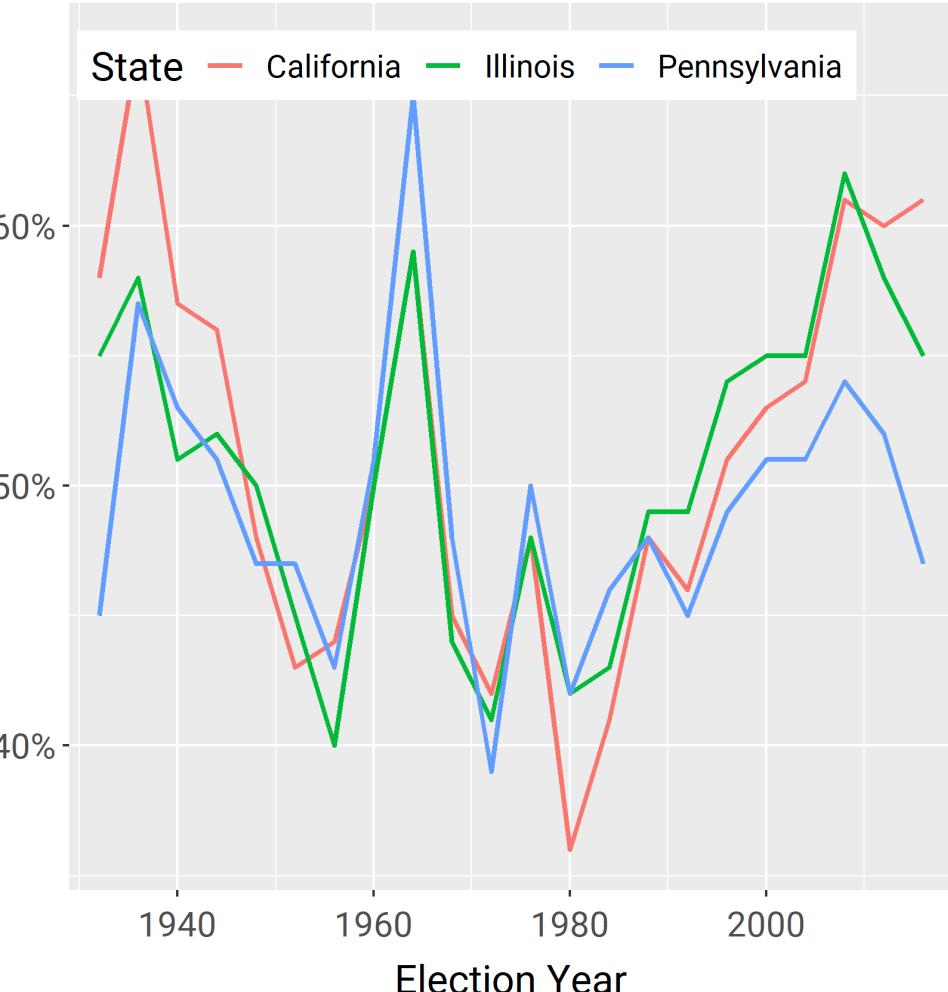
State



```
state_election_plot_B +  
  theme(  
    legend.key = element_rect(fill = NA)  
) +  
  theme(  
    legend.position = c(.45, .93)  
) +  
  theme(  
    legend.direction = "horizontal"  
)
```

## Percent of democrat votes by state

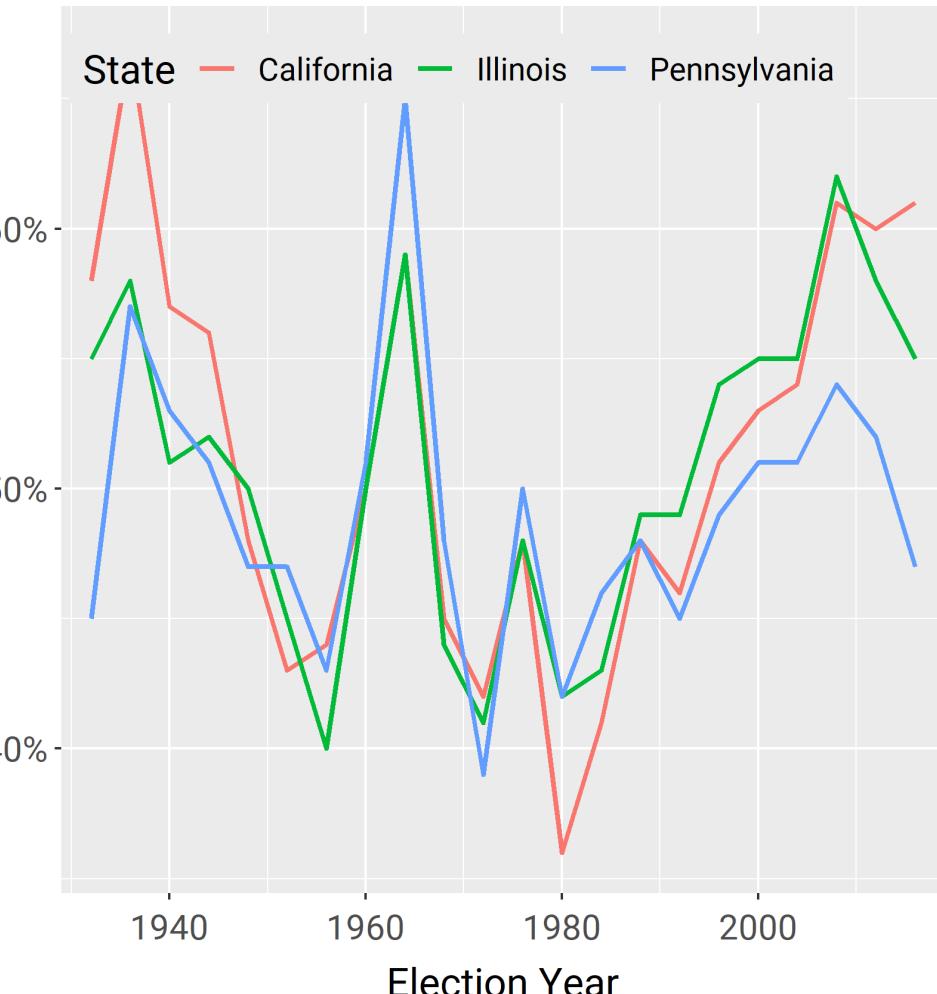
*We're a swing state! Go vote!*



```
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")  
)
```

## Percent of democrat votes by state

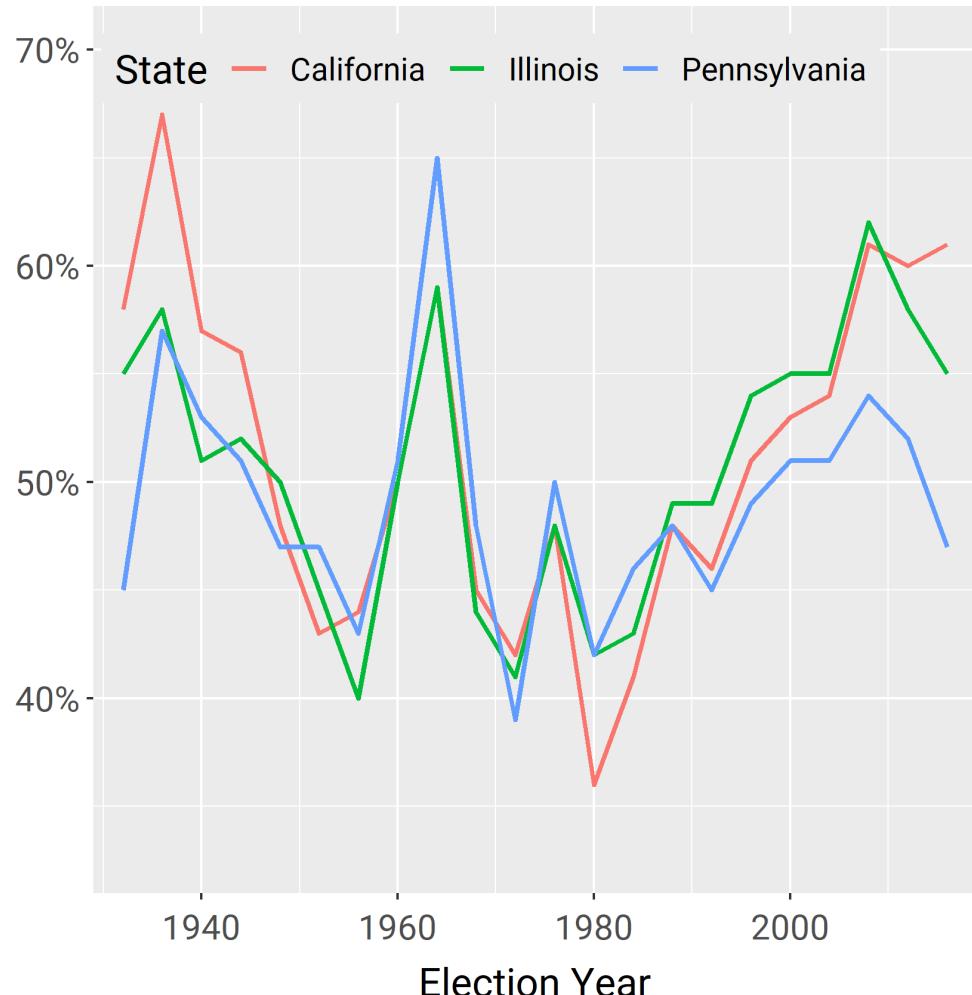
*We're a swing state! Go vote!*



```
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)  
)
```

## Percent of democrat votes by state

*We're a swing state! Go vote!*



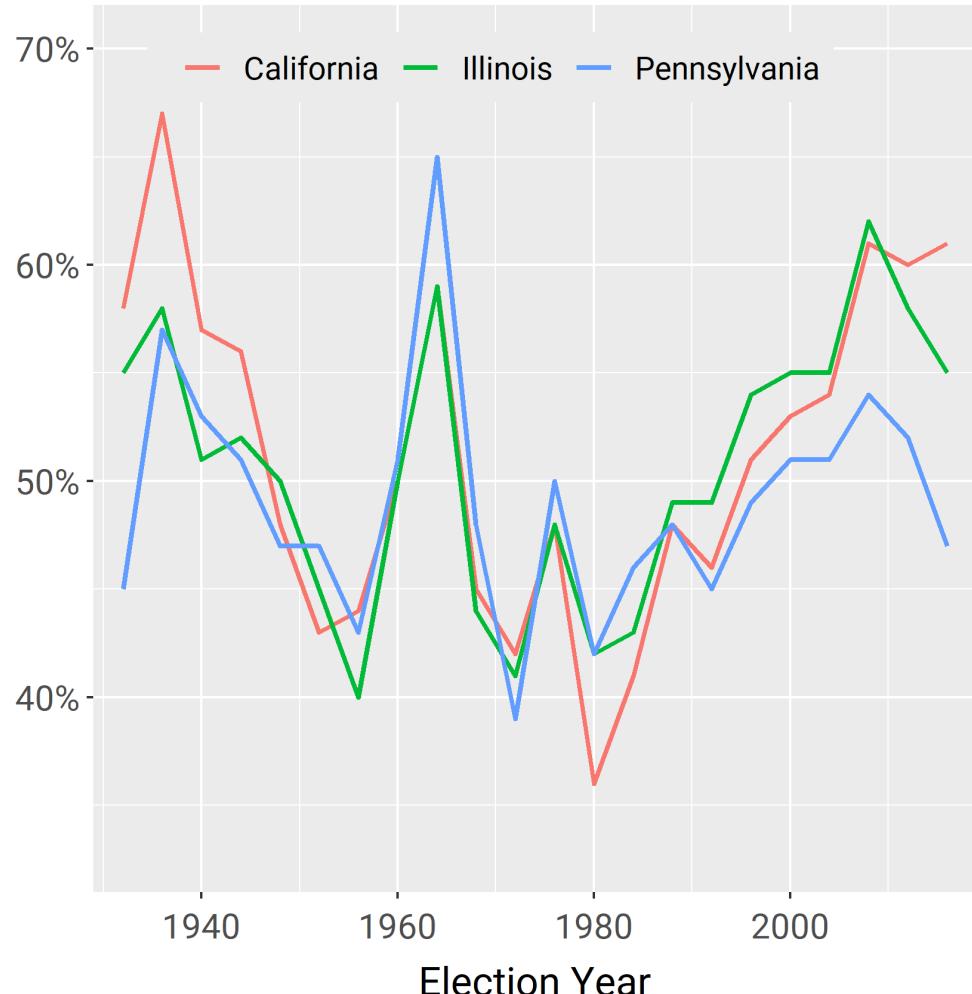
```

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!)

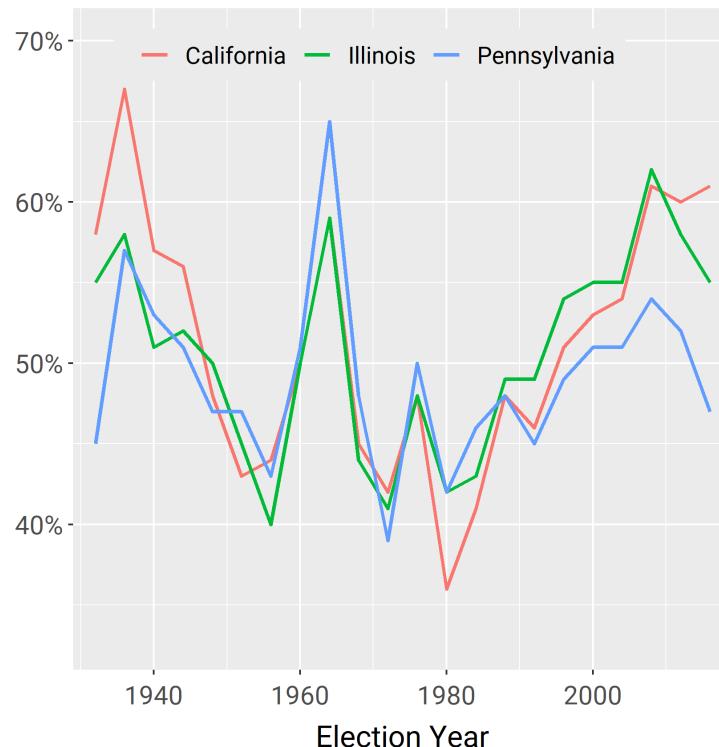
```
state_election_plot_B +  
  theme(  
    legend.key = element_rect(fill = NA),  
    legend.position = c(.45, .93),  
    legend.direction = "horizontal",  
    legend.background = element_rect(fill = "grey92")  
  ) +  
  scale_y_continuous(  
    expand = expansion(0, .05),  
    labels = percent_format(accuracy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

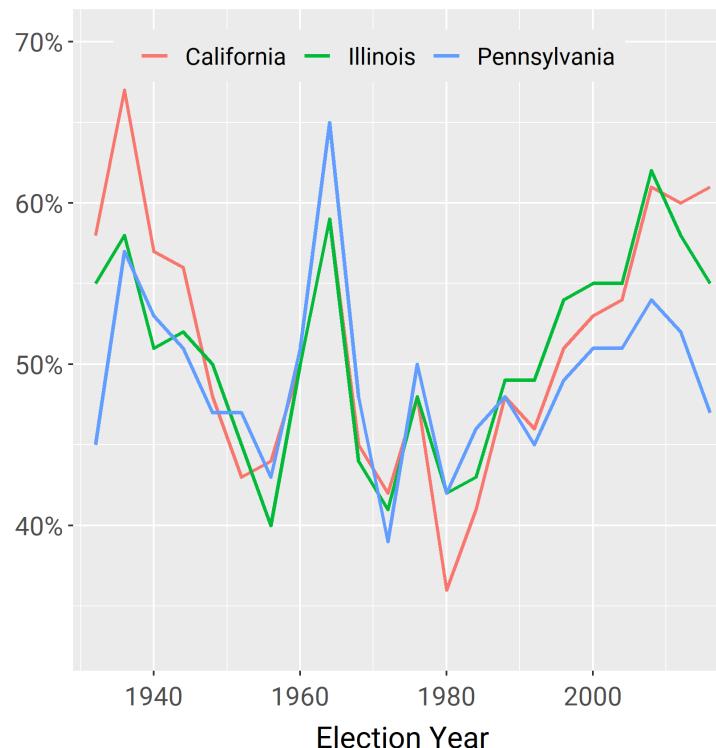
Percent of democrat votes by state

*We're a swing state! Go vote!*



Percent of democrat votes by state

*We're a swing state! Go vote!*



# 4. Color

# 4. Color

Colors are a double-edged sword

# 4. Color

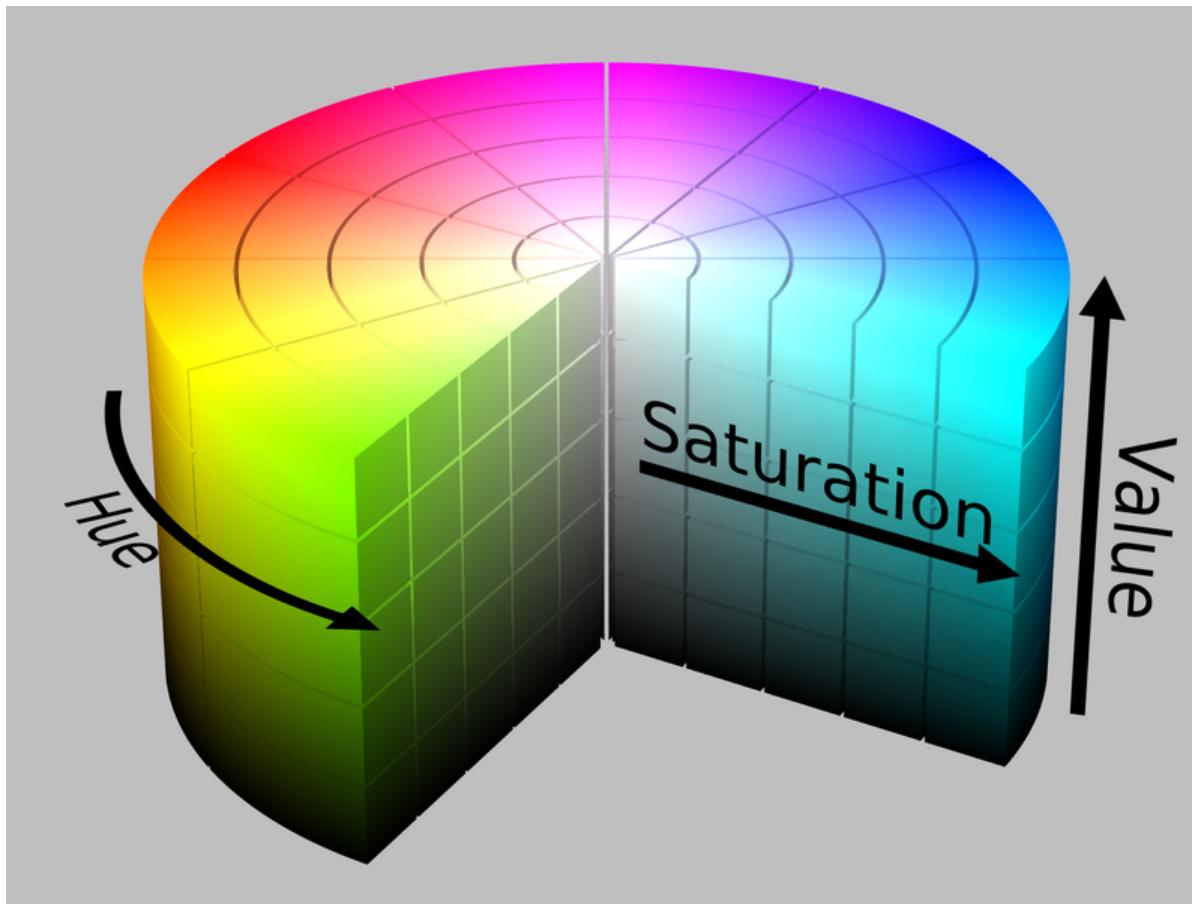
Colors are a double-edged sword

- Perception can vary widely depending on reader, medium, culture, etc.

# 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)

# 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:

# 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!)

# 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:

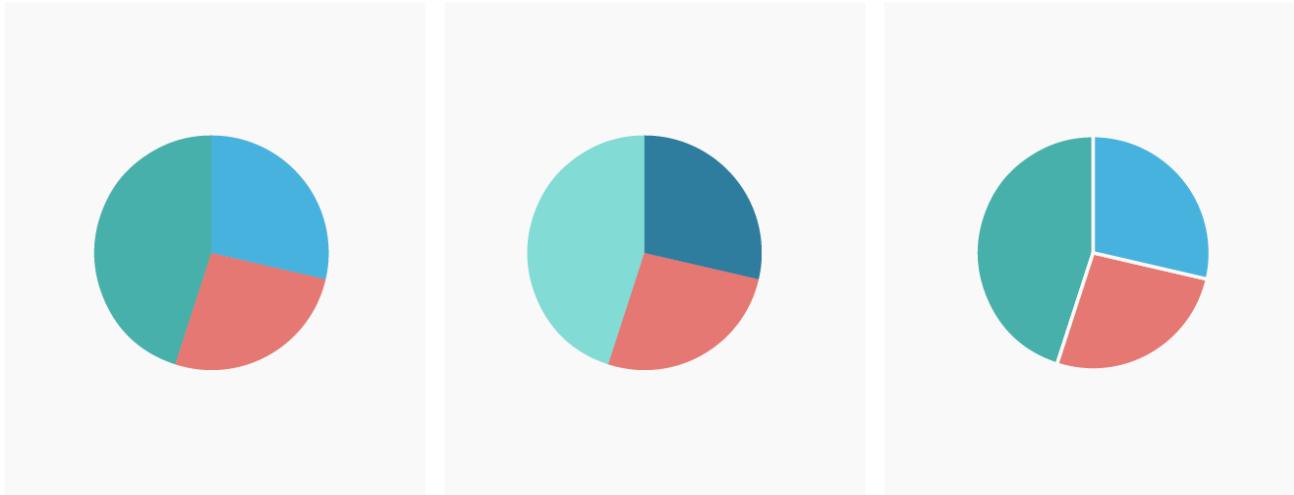
- Avoid pure colors (no random sampling from the rainbow!)
- Contrast colors in more than one dimension



NOT IDEAL

BETTER

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



NOT IDEAL

BETTER

BETTER

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

# 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](#))

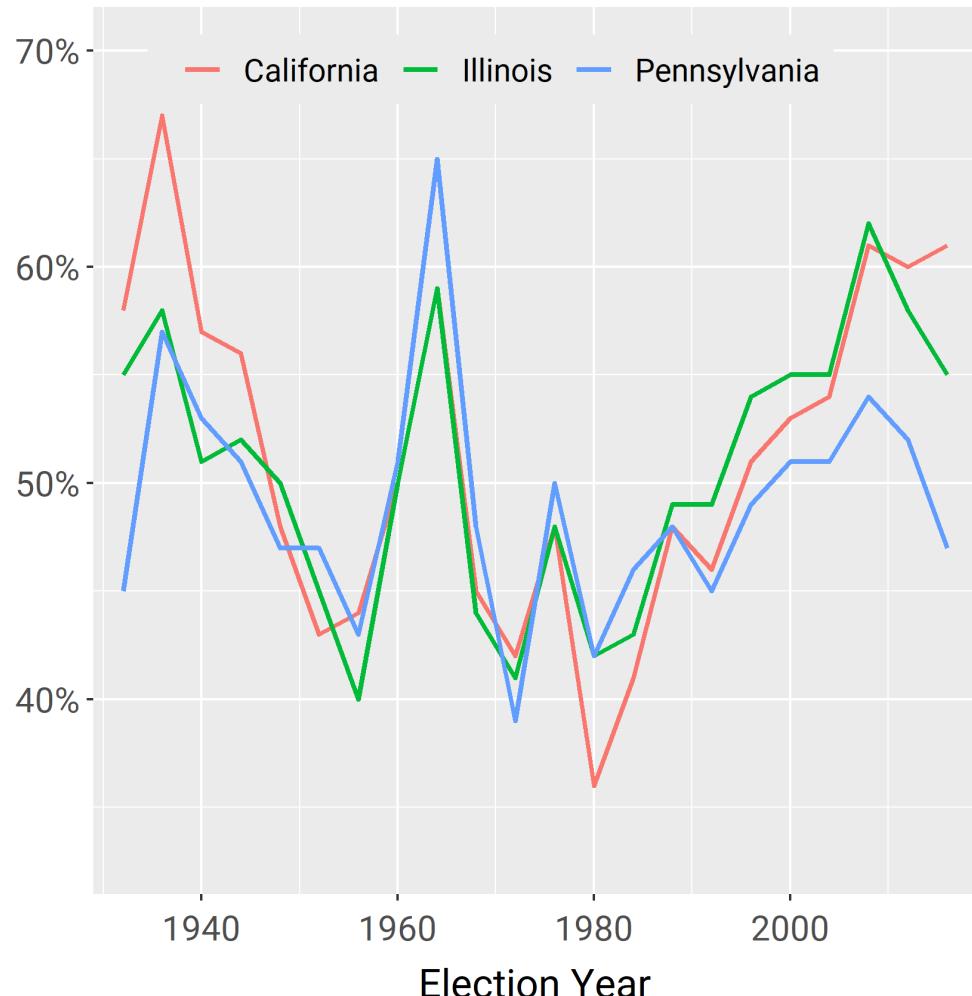
If you must, DO:

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

If you aren't sure, use [pre-made palettes](#) or play around with [online color tools](#)

## Percent of democrat votes by state

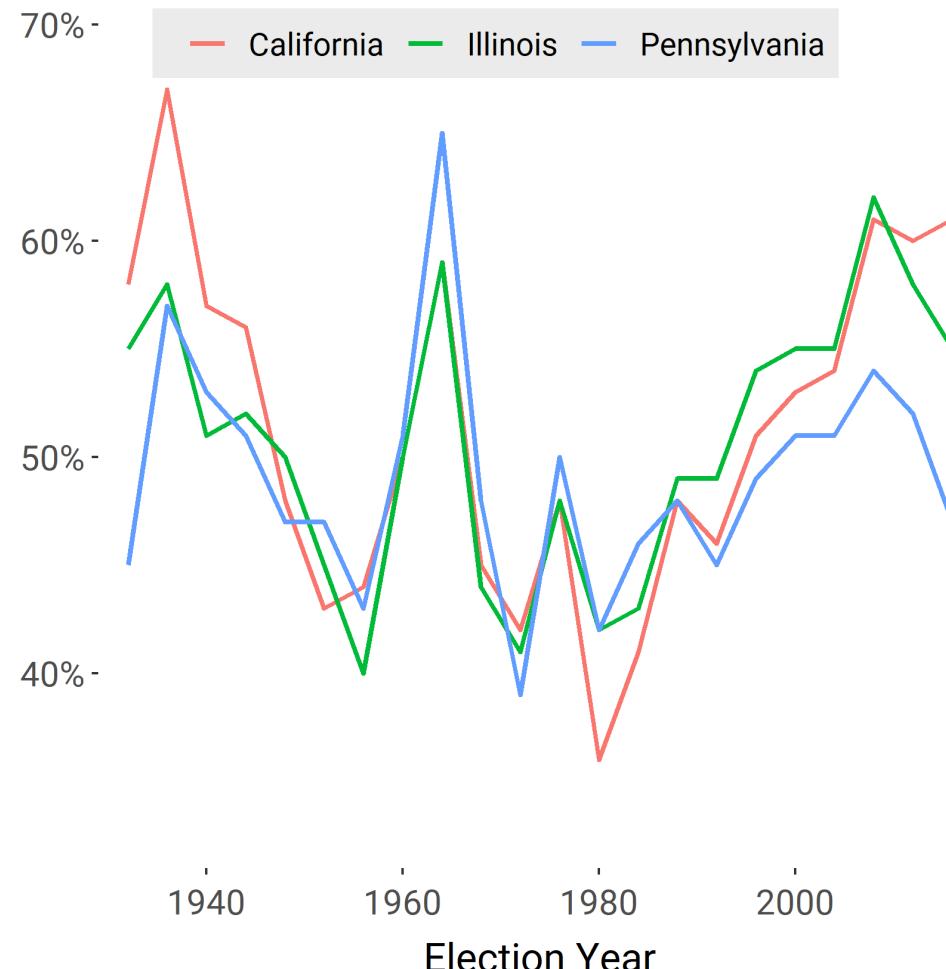
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA))
```

## Percent of democrat votes by state

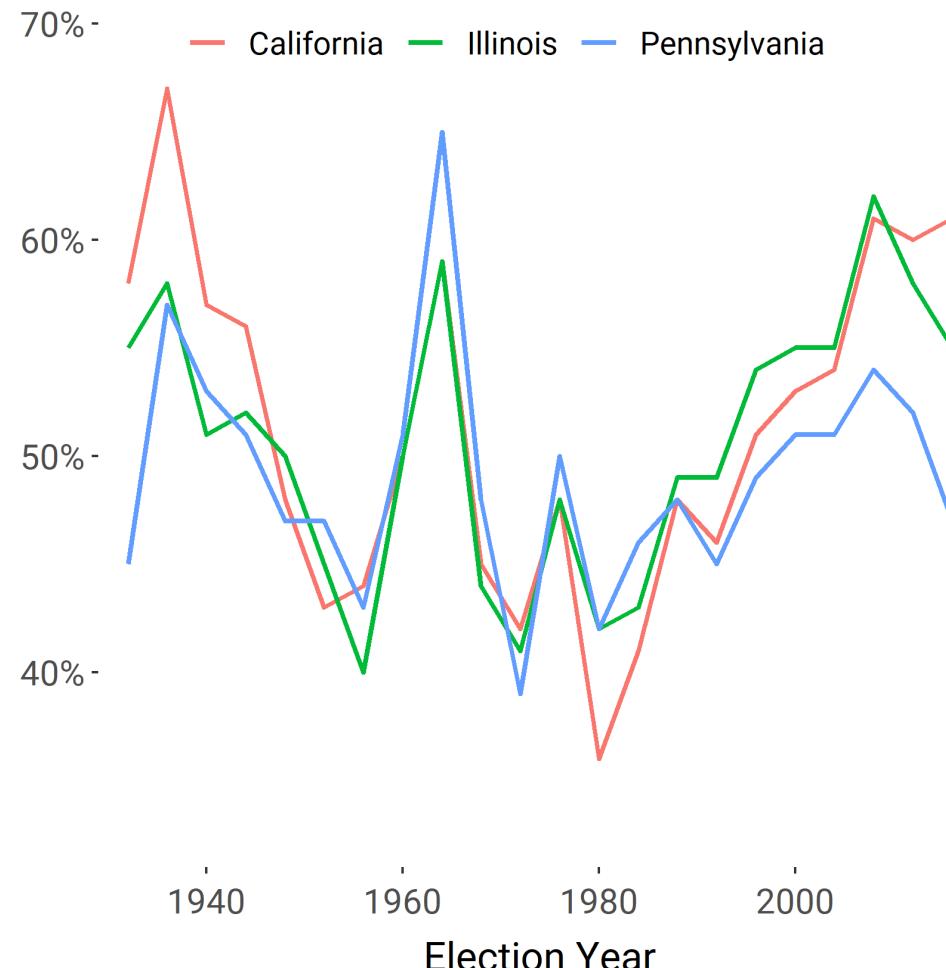
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA))
```

## Percent of democrat votes by state

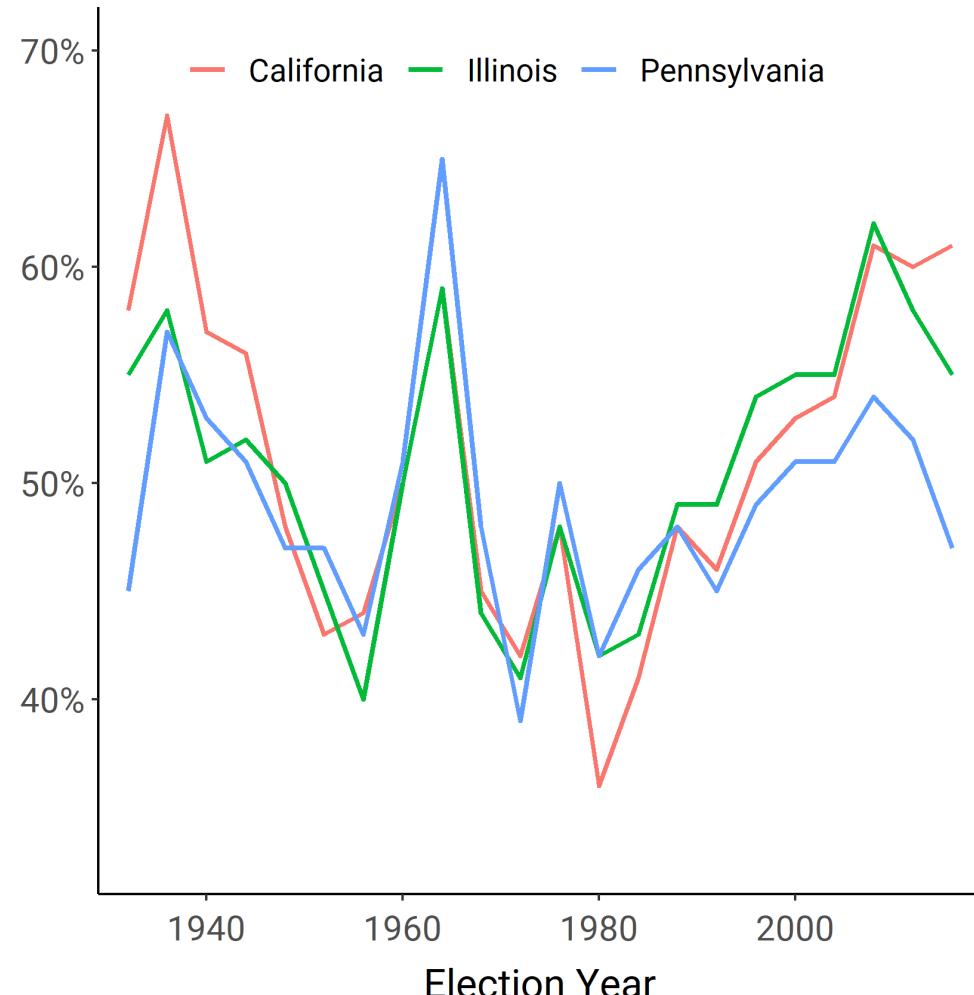
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line())
```

## Percent of democrat votes by state

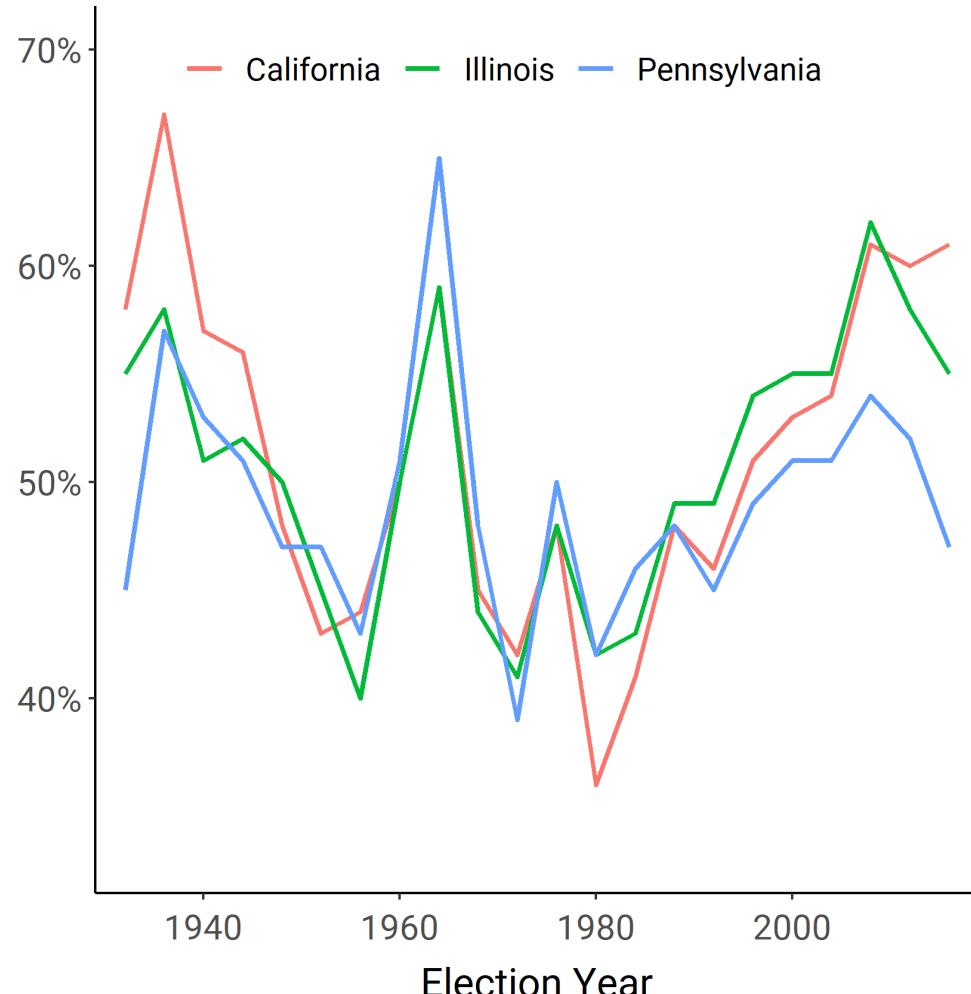
*We're a swing state! Go vote!*



```
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("#F8766D", "#00BA38", "#619cff"))  
)
```

## Percent of democrat votes by state

We're a swing state! Go vote!



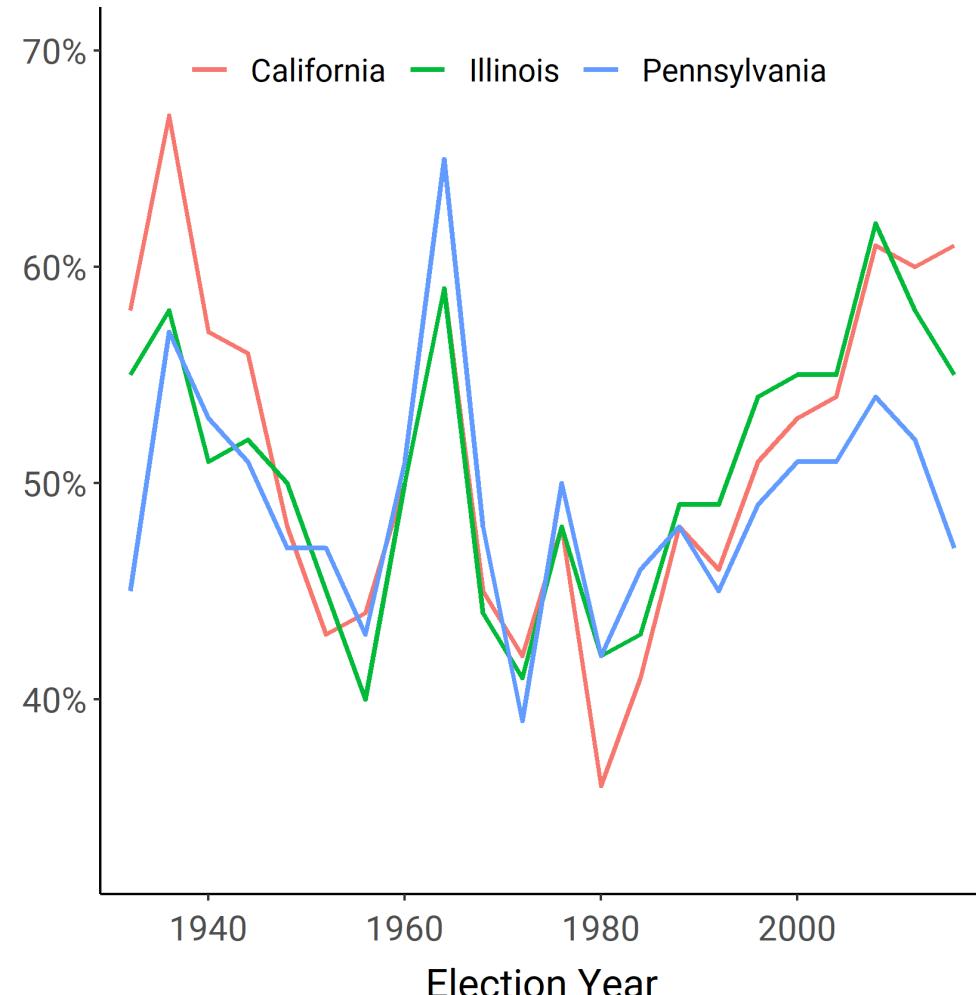
```

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("#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))
    )
  )
)

```

## Percent of democrat votes by state

*We're a swing state! Go vote!*



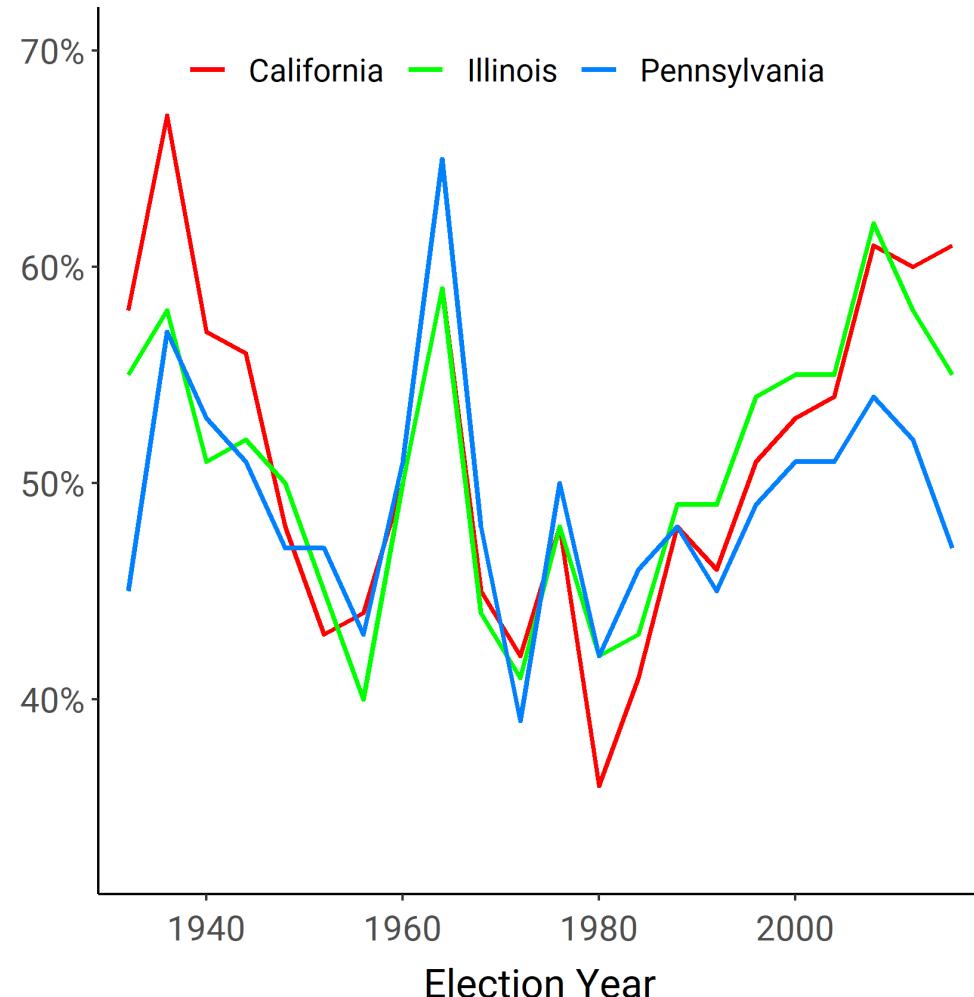
```

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("#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))
    )
  )
)

```

## Percent of democrat votes by state

*We're a swing state! Go vote!*



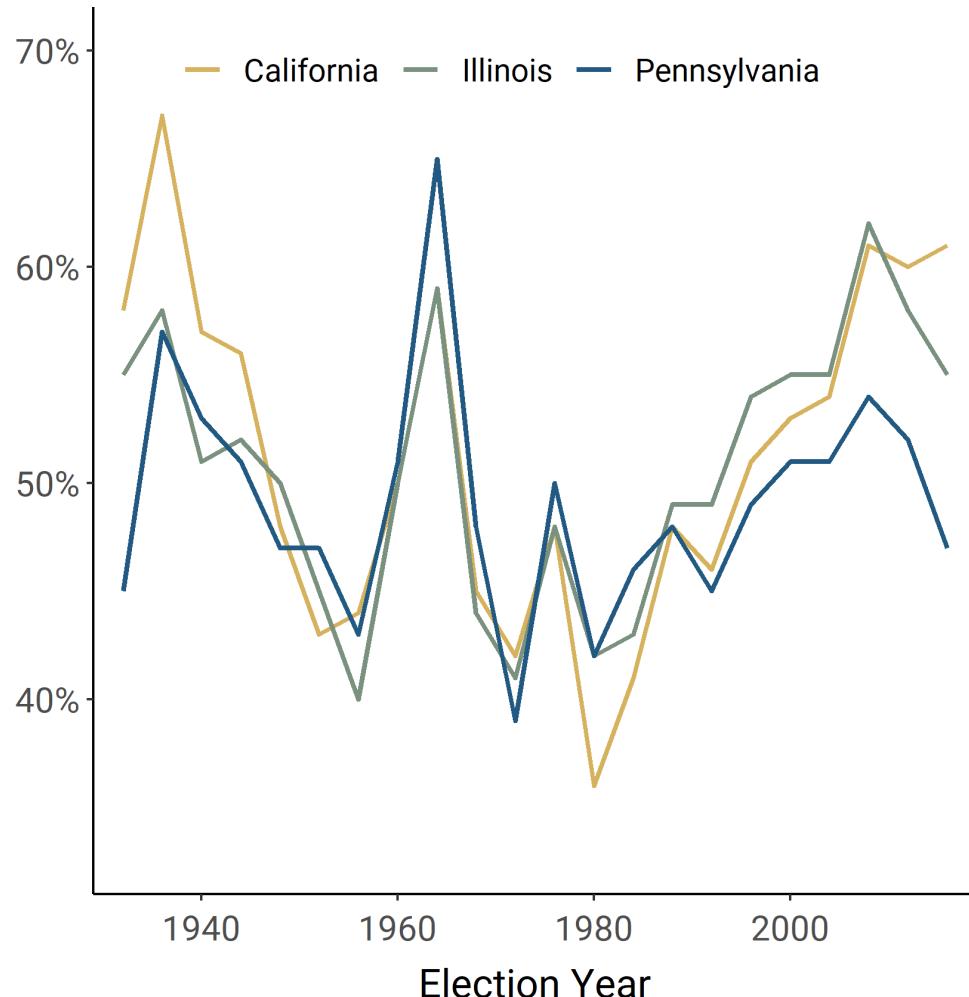
```

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("#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))
    )
  )
)

```

## Percent of democrat votes by state

*We're a swing state! Go vote!*



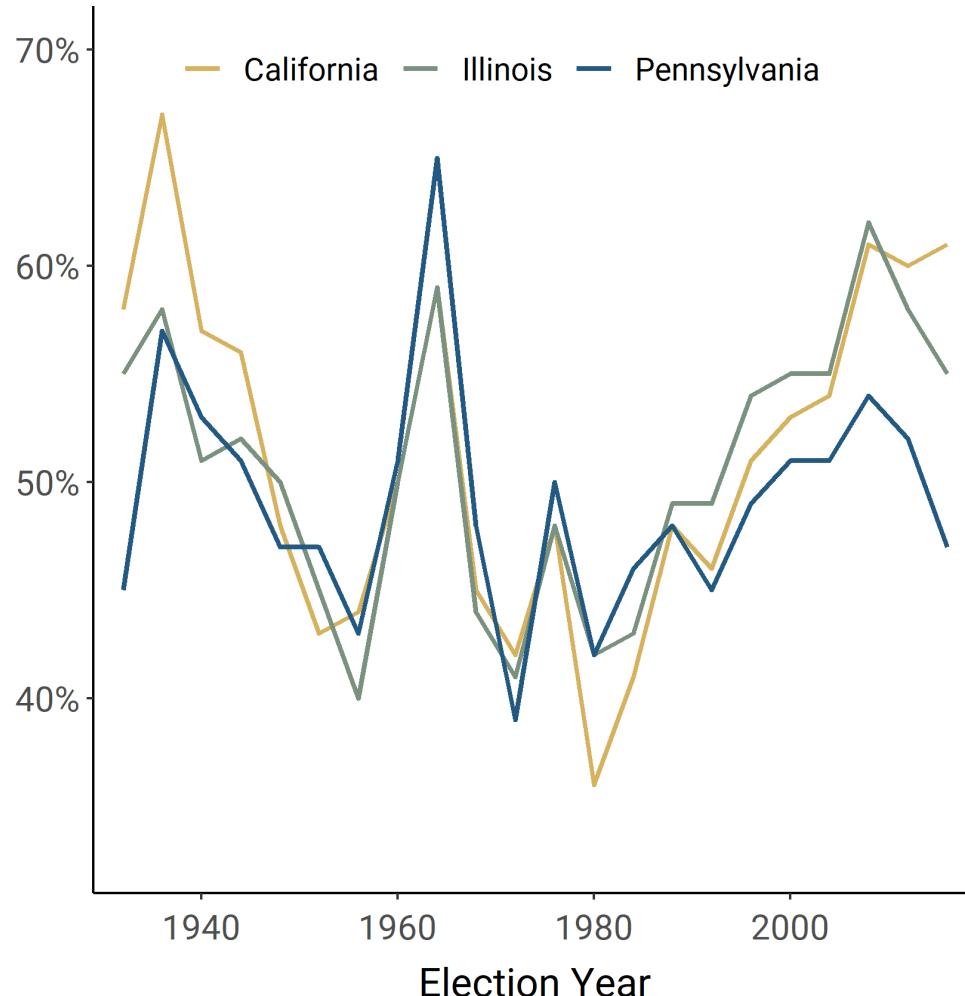
```

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("#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")
  )
)

```

## Percent of democrat votes by state

*We're a swing state! Go vote!*



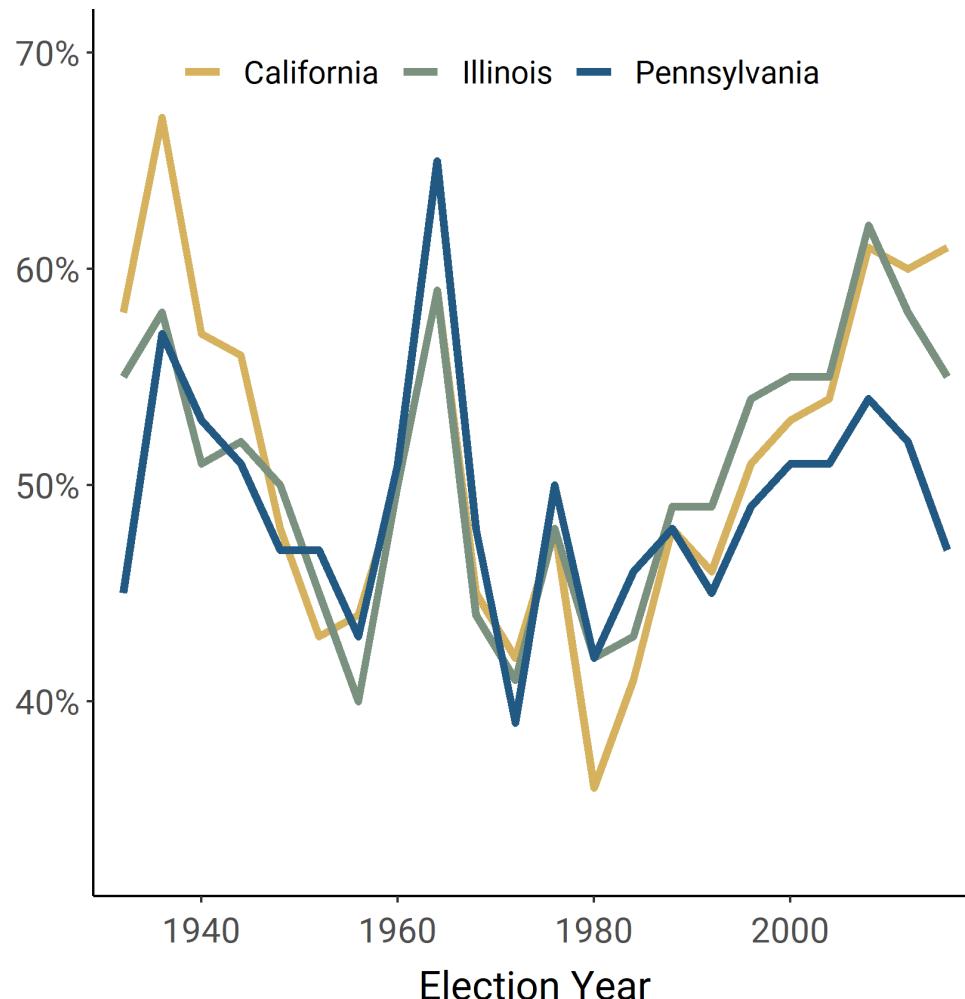
```

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("#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

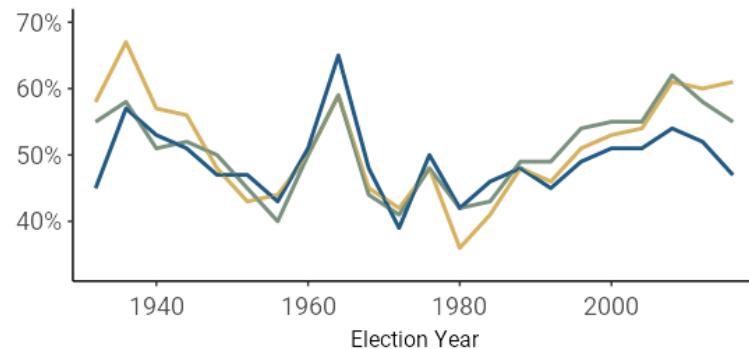
*We're a swing state! Go vote!*



```
library(colorBlindness)
```

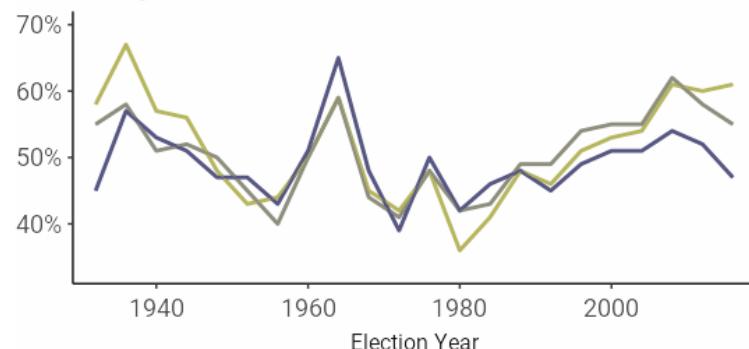
### normal vision

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



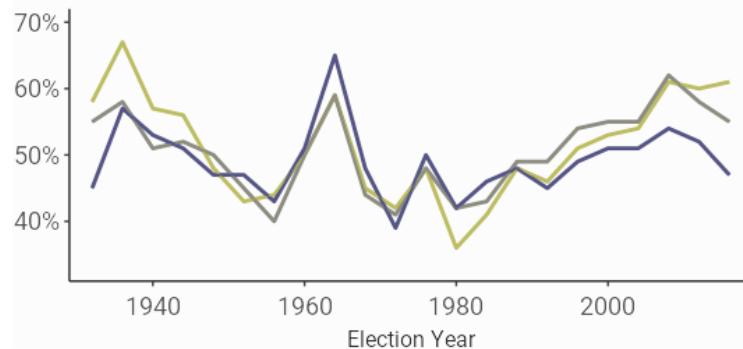
### protanopia (2%)

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



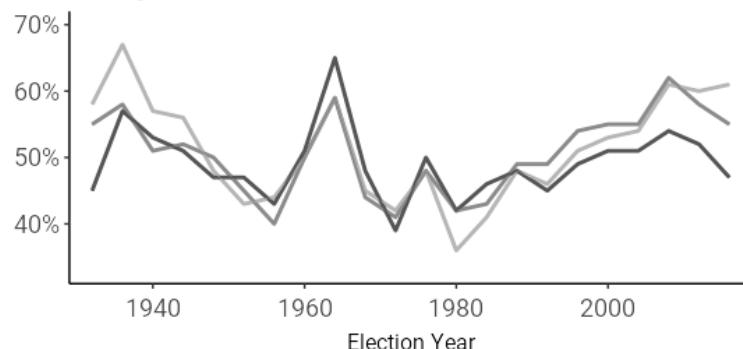
### deuteranopia (6%)

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



### desaturated (BW)

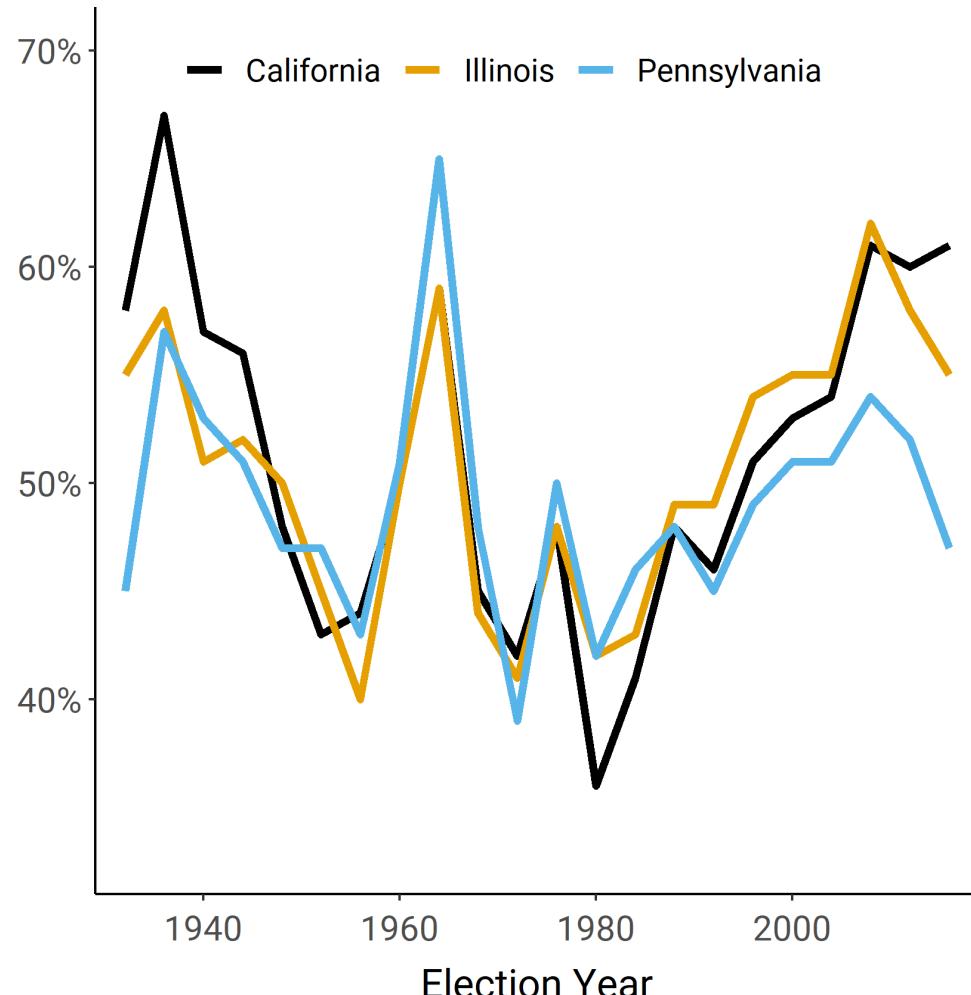
Percent of democrat votes by state  
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  ggthemes::scale_color_colorblind()
```

## Percent of democrat votes by state

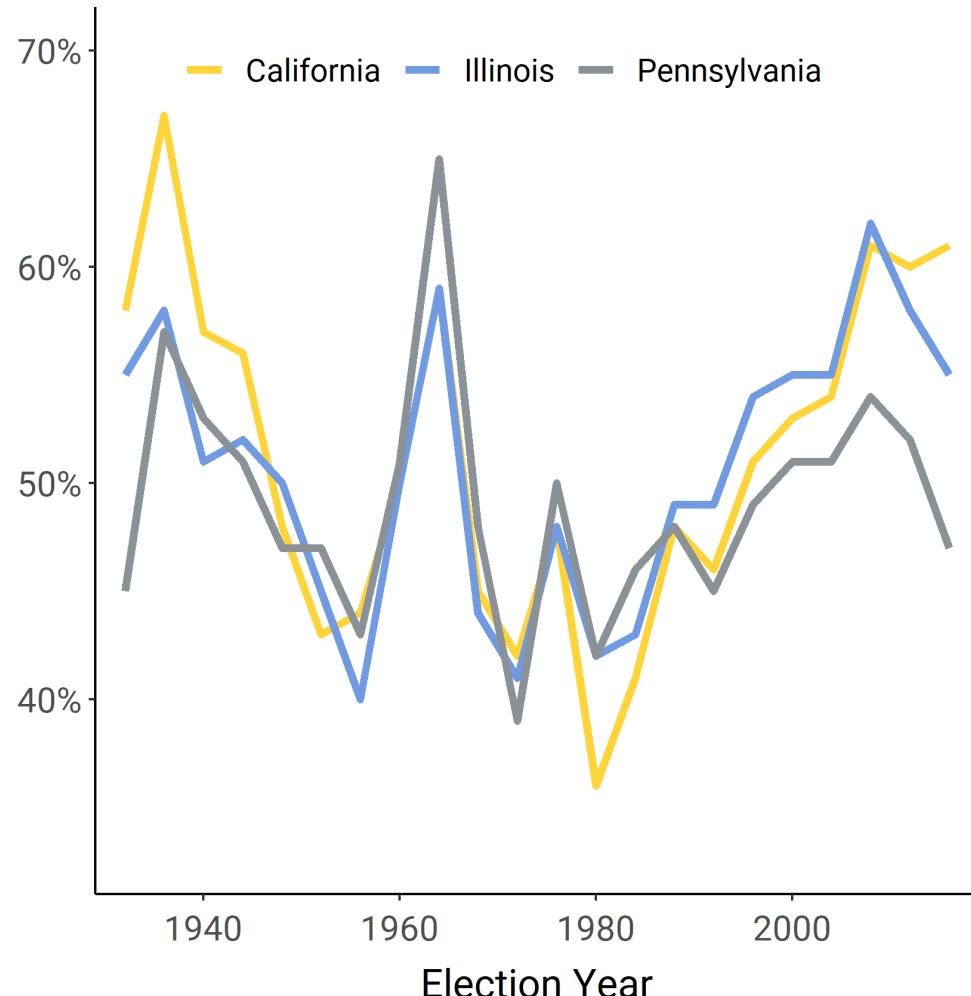
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  ggsci::scale_color_simpsons()
```

## Percent of democrat votes by state

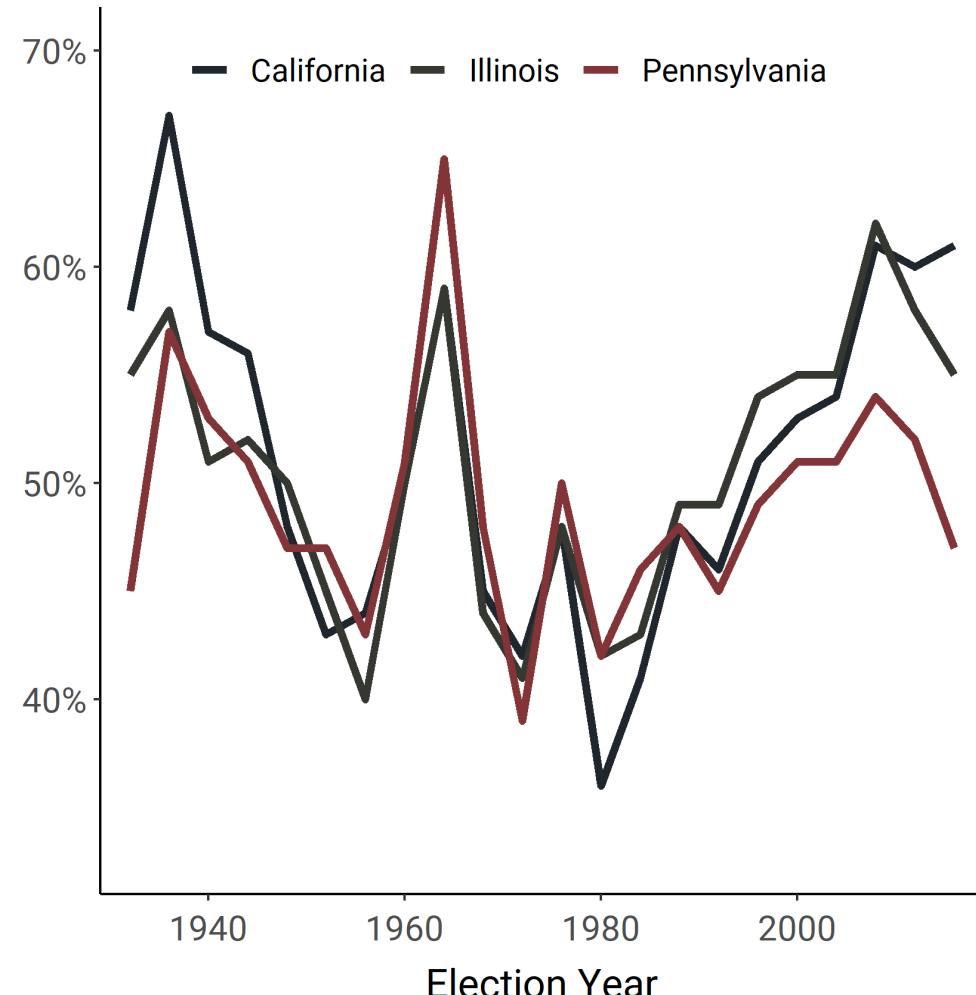
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  ghibli::scale_color_ghibli_d("SpiritedMedium")
```

## Percent of democrat votes by state

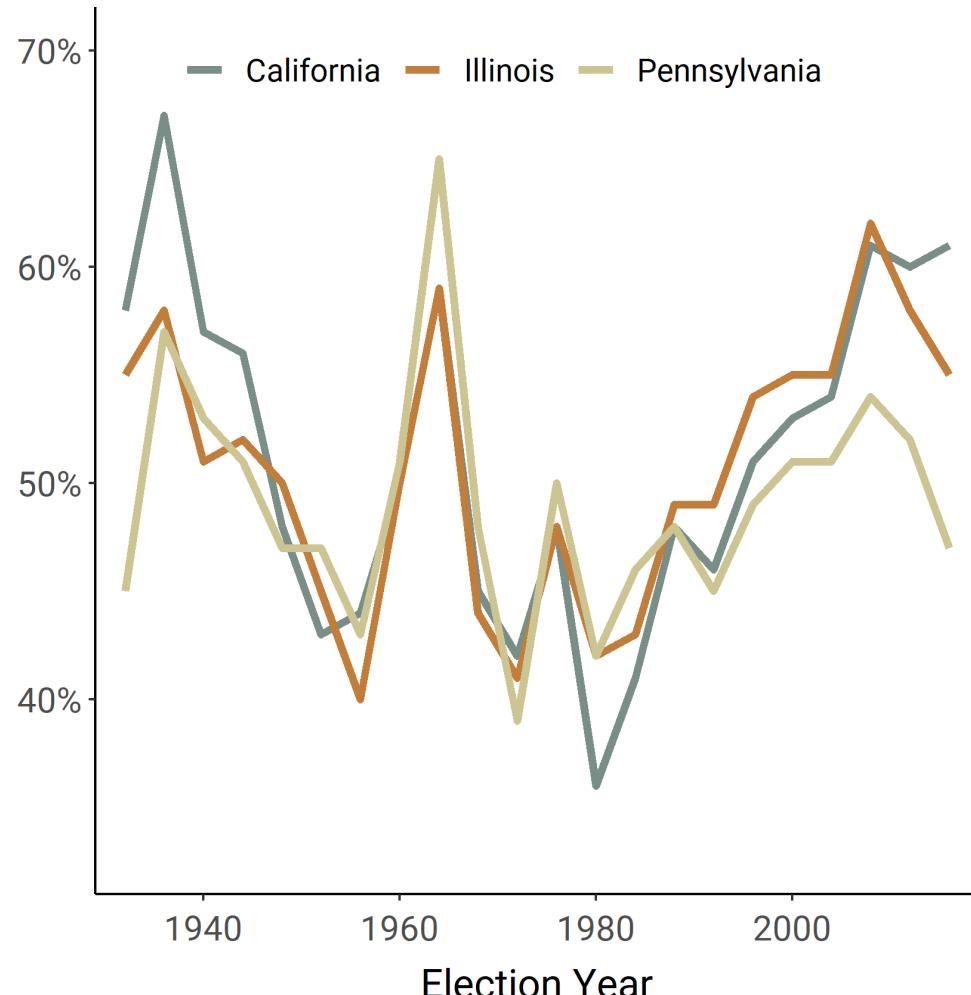
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  scale_color_manual(values = wesanderson::wes_palette("Moorish"))
```

## Percent of democrat votes by state

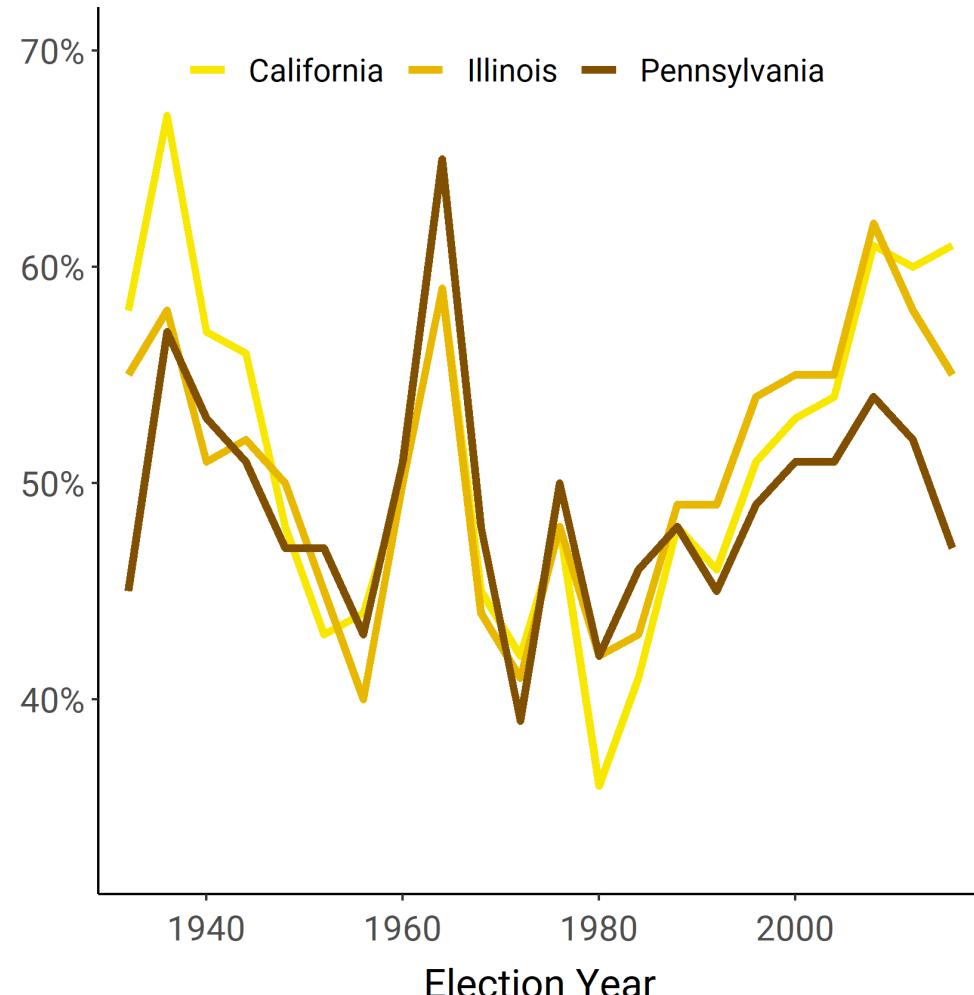
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  palettetown::scale_color_pokemon(pokemon = "pikachu")
```

## Percent of democrat votes by state

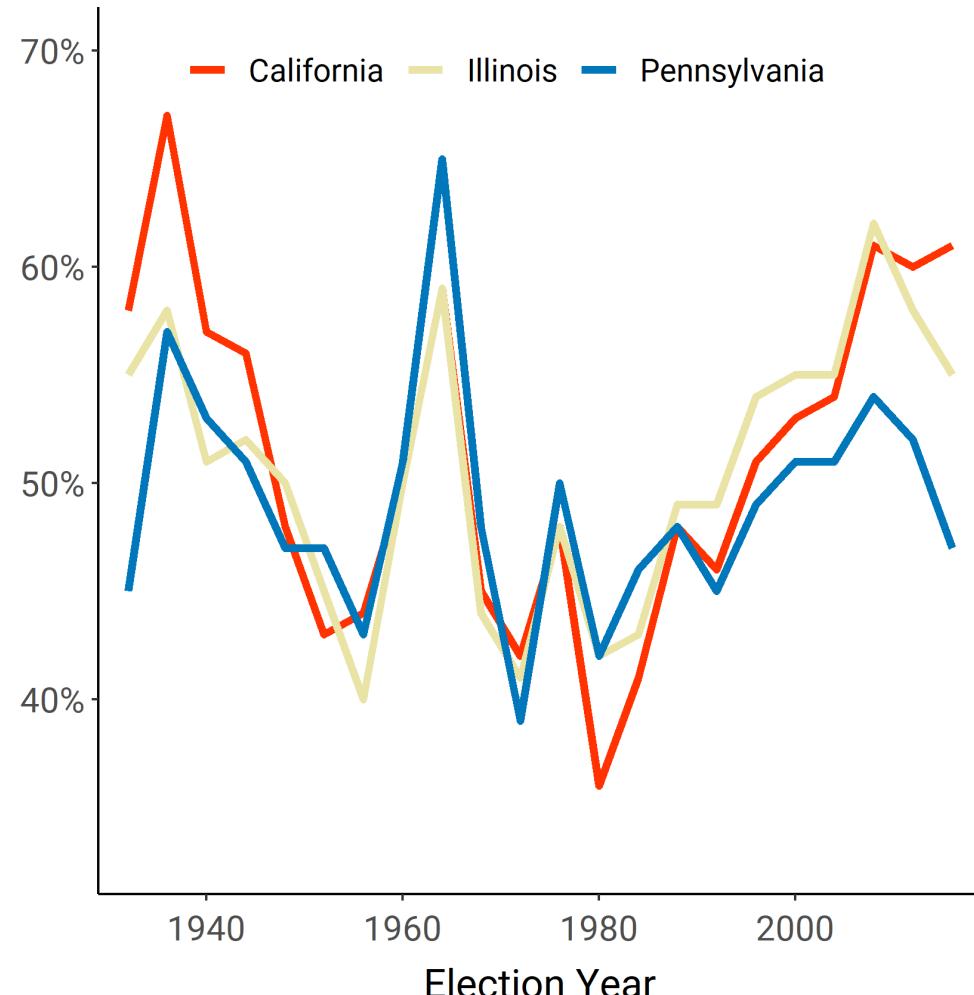
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  scale_color_manual(values = LaCroixColoR::lacroix_palette)
```

## Percent of democrat votes by state

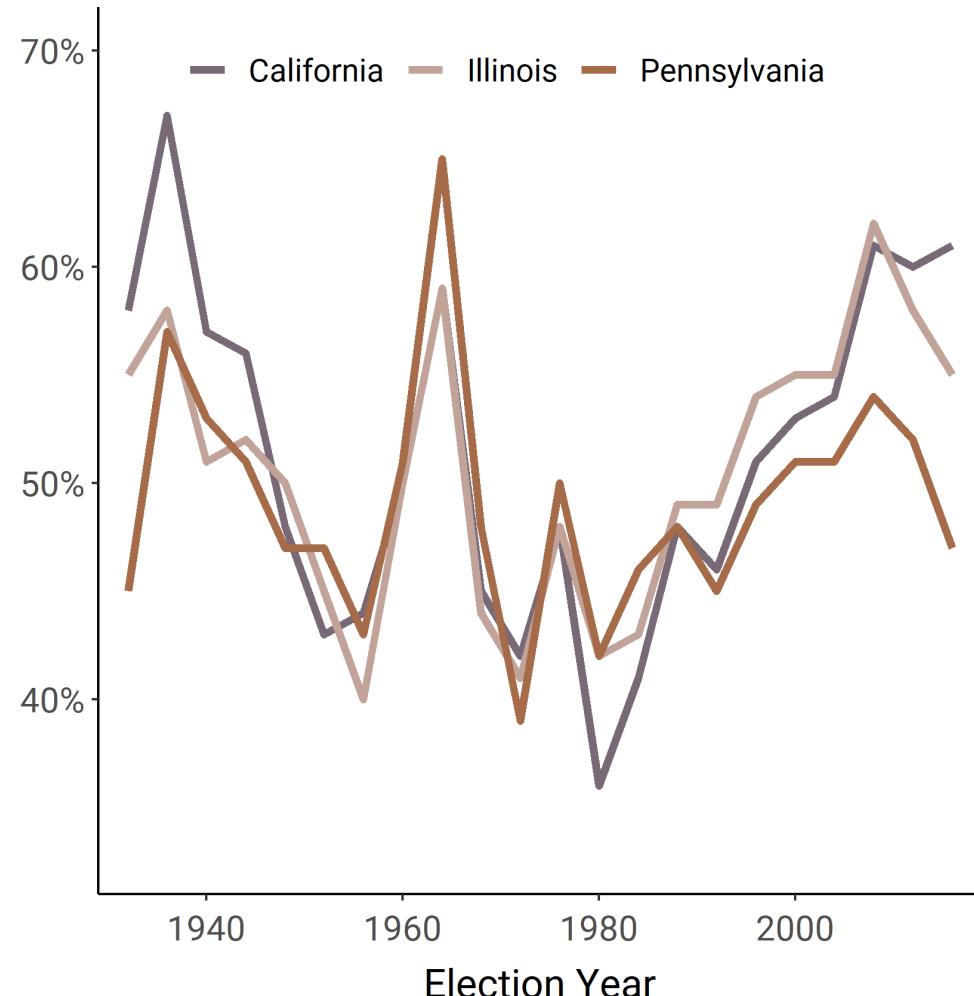
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  scale_color_manual(values = junebug::junebug_palettes$Misc)
```

## Percent of democrat votes by state

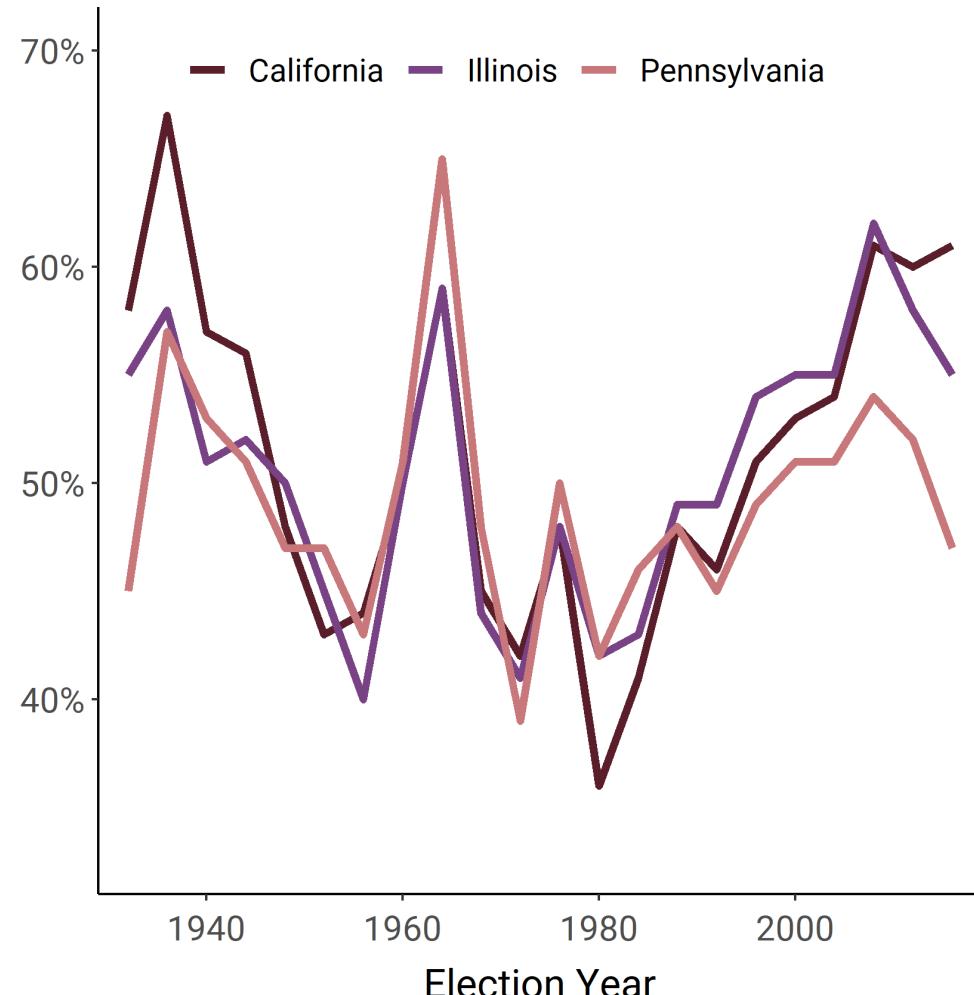
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  scale_color_manual(values = junebug::junebug_palettes$Crar
```

## Percent of democrat votes by state

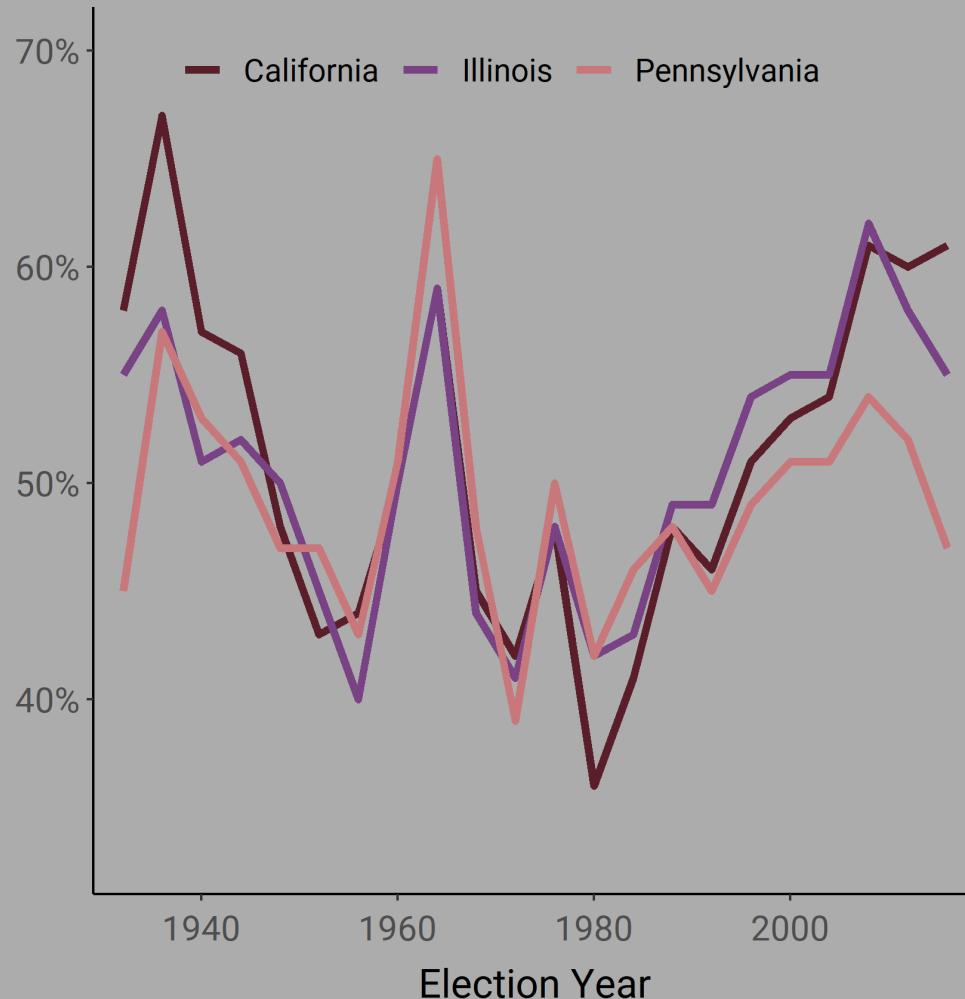
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  theme(plot.background = element_rect(fill = "#ACACAC"), pa  
scale_color_manual(values = junebug::junebug_palettes$Crar
```

## Percent of democrat votes by state

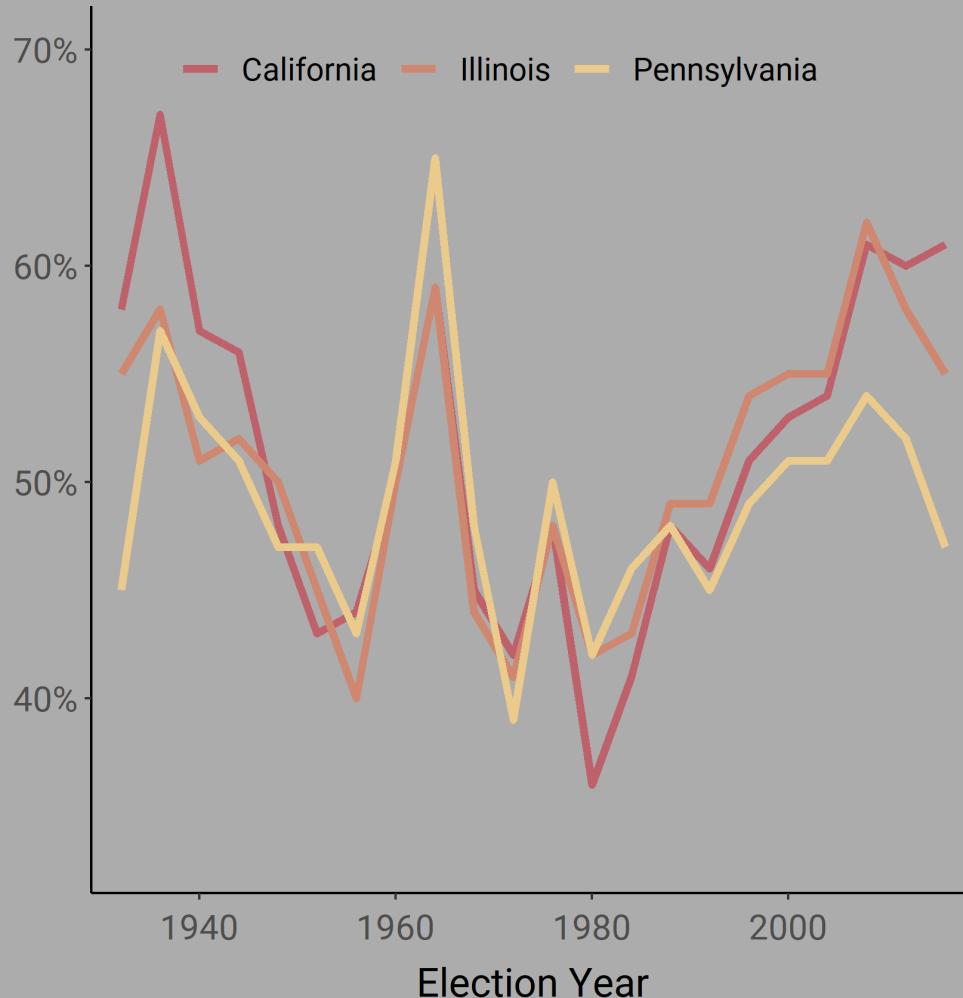
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(panel.background = element_rect(fill = NA)) +  
  theme(legend.background = element_rect(fill = NA)) +  
  theme(axis.line = element_line()) +  
  geom_line(size = 1.5) +  
  theme(plot.background = element_rect(fill = "#ACACAC"), pa  
scale_color_manual(values = nord::nord_palettes$aurora[1:3])
```

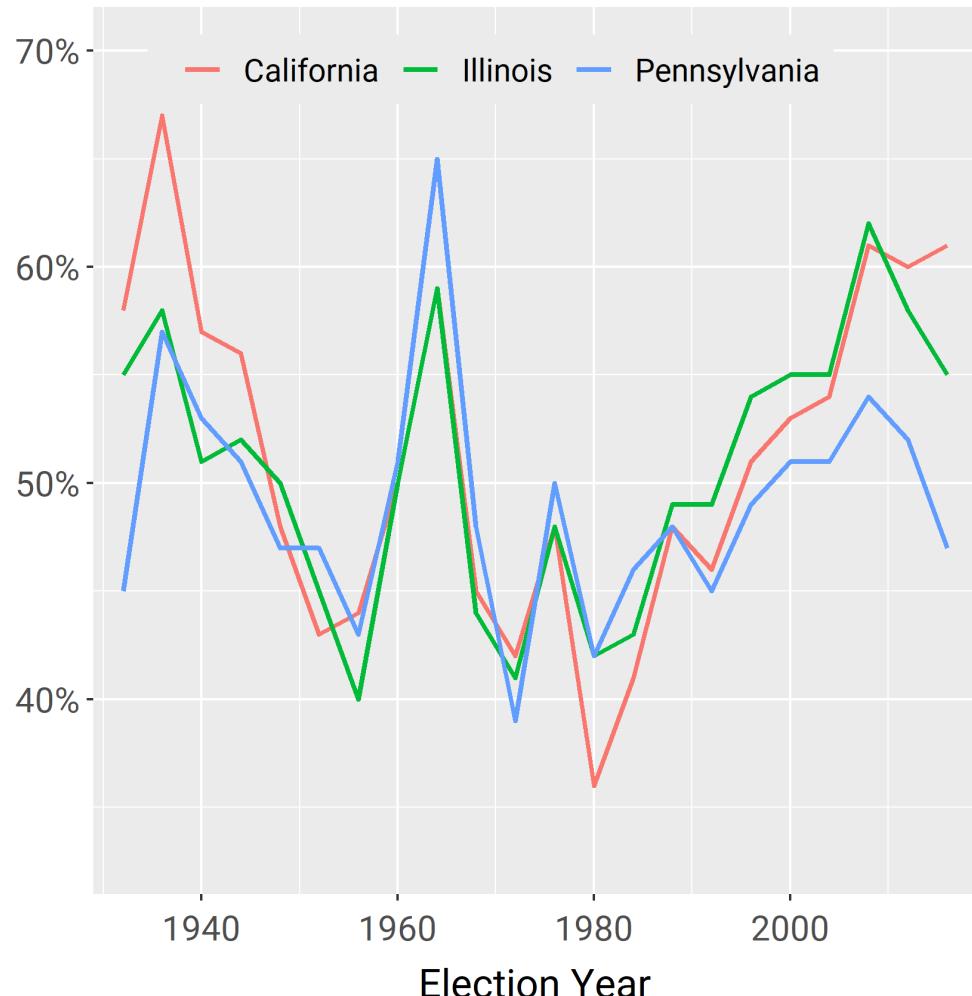
## Percent of democrat votes by state

*We're a swing state! Go vote!*



## Percent of democrat votes by state

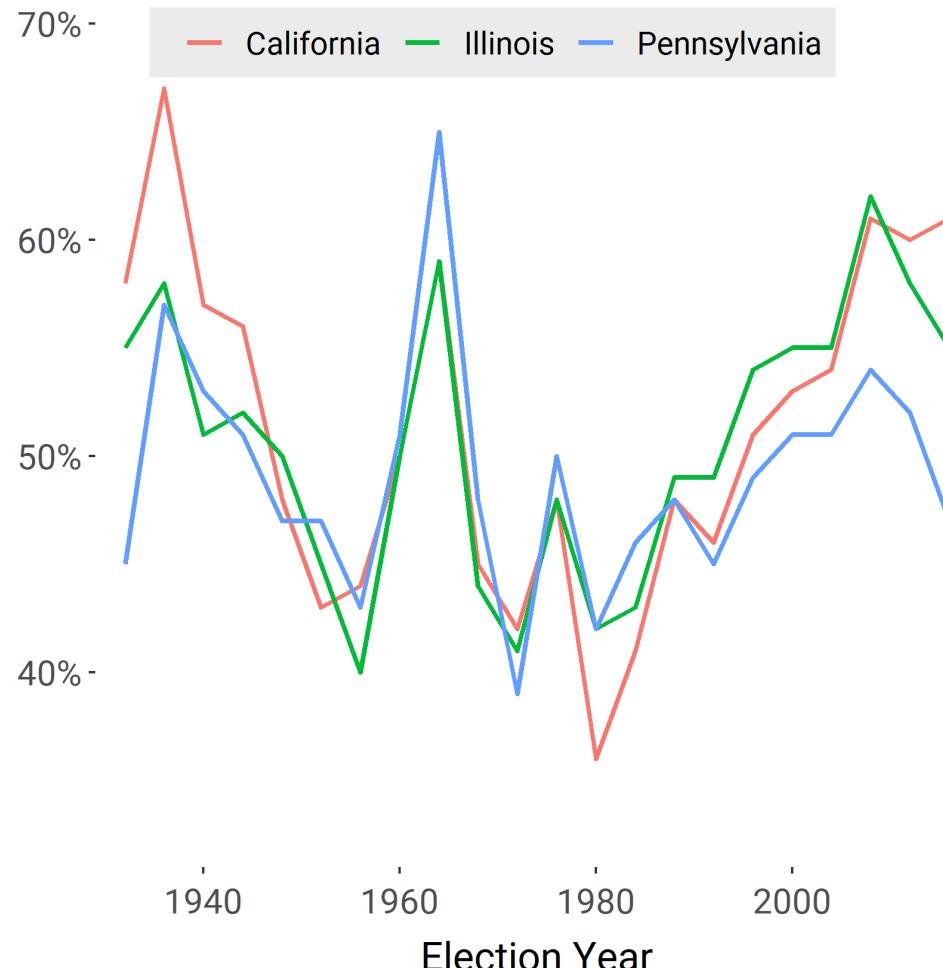
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(  
    panel.background = element_rect(fill = NA)  
)
```

## Percent of democrat votes by state

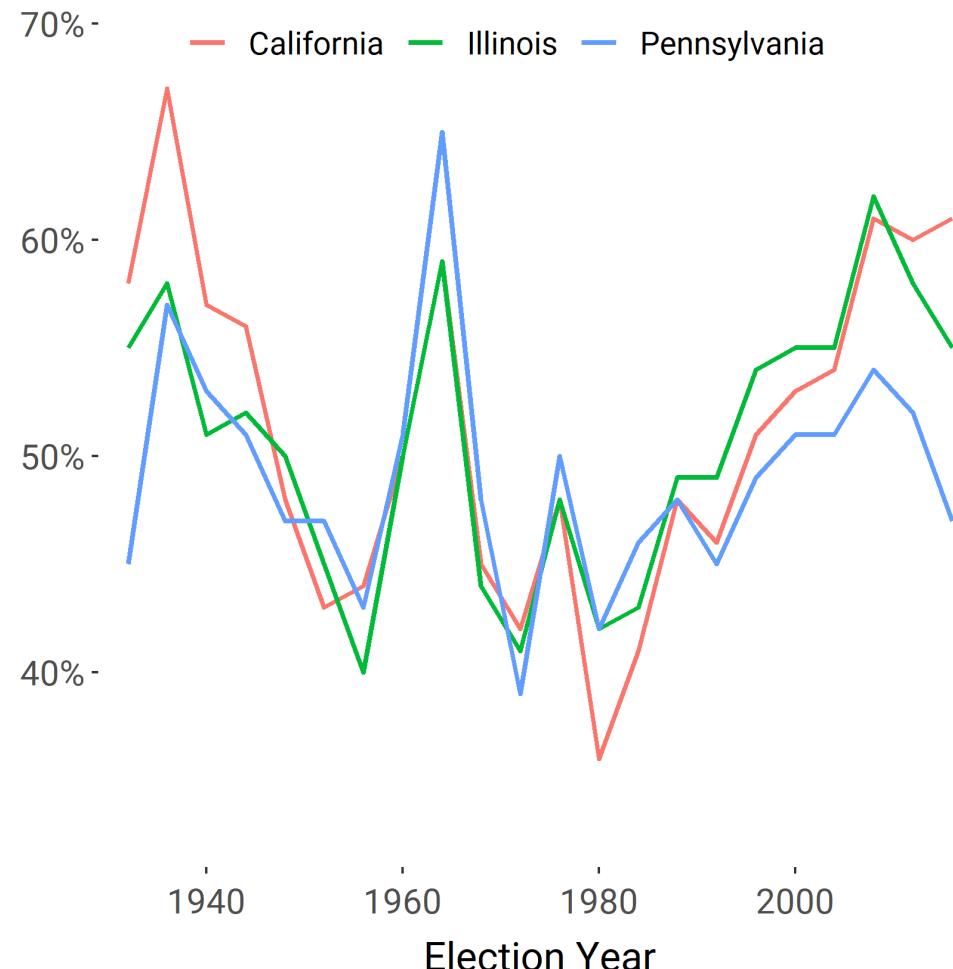
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(  
    panel.background = element_rect(fill = NA)  
) +  
  theme(  
    legend.background = element_rect(fill = NA)  
)
```

## Percent of democrat votes by state

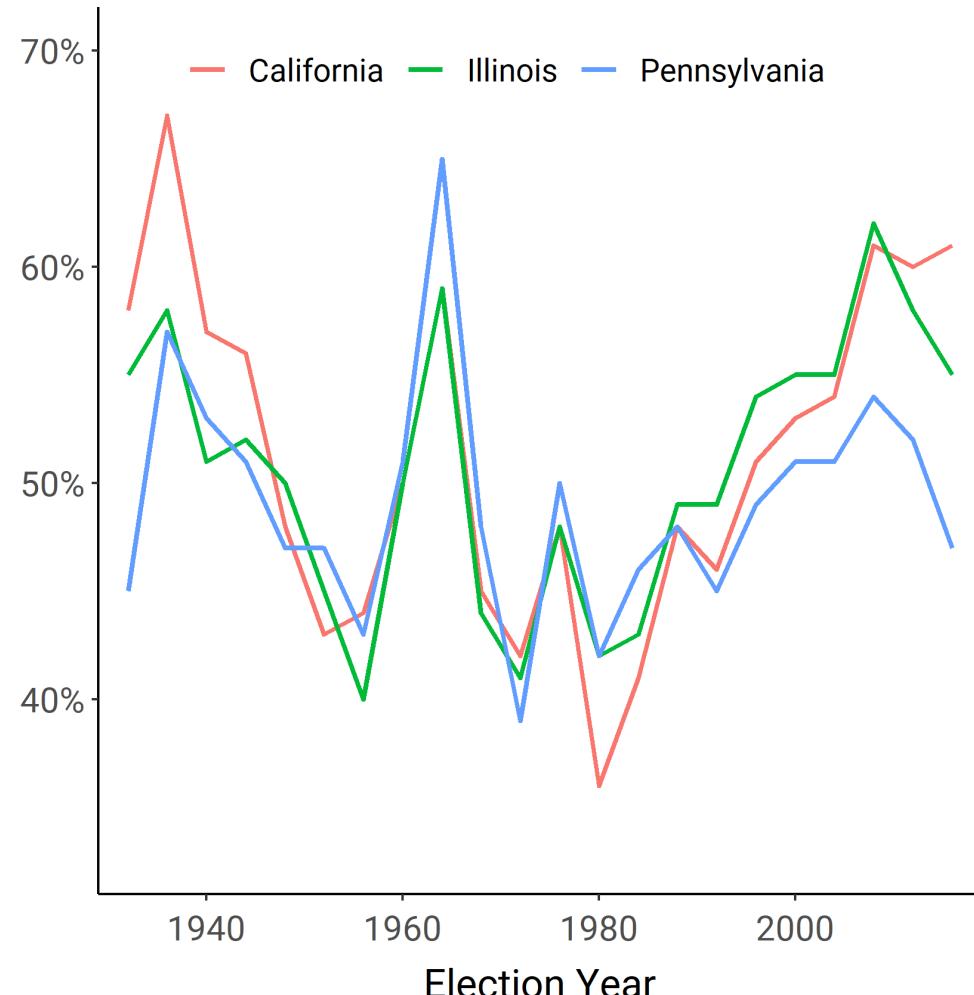
*We're a swing state! Go vote!*



```
state_election_plot_C +  
  theme(  
    panel.background = element_rect(fill = NA)  
) +  
  theme(  
    legend.background = element_rect(fill = NA)  
) +  
  theme(  
    axis.line = element_line()  
)
```

## Percent of democrat votes by state

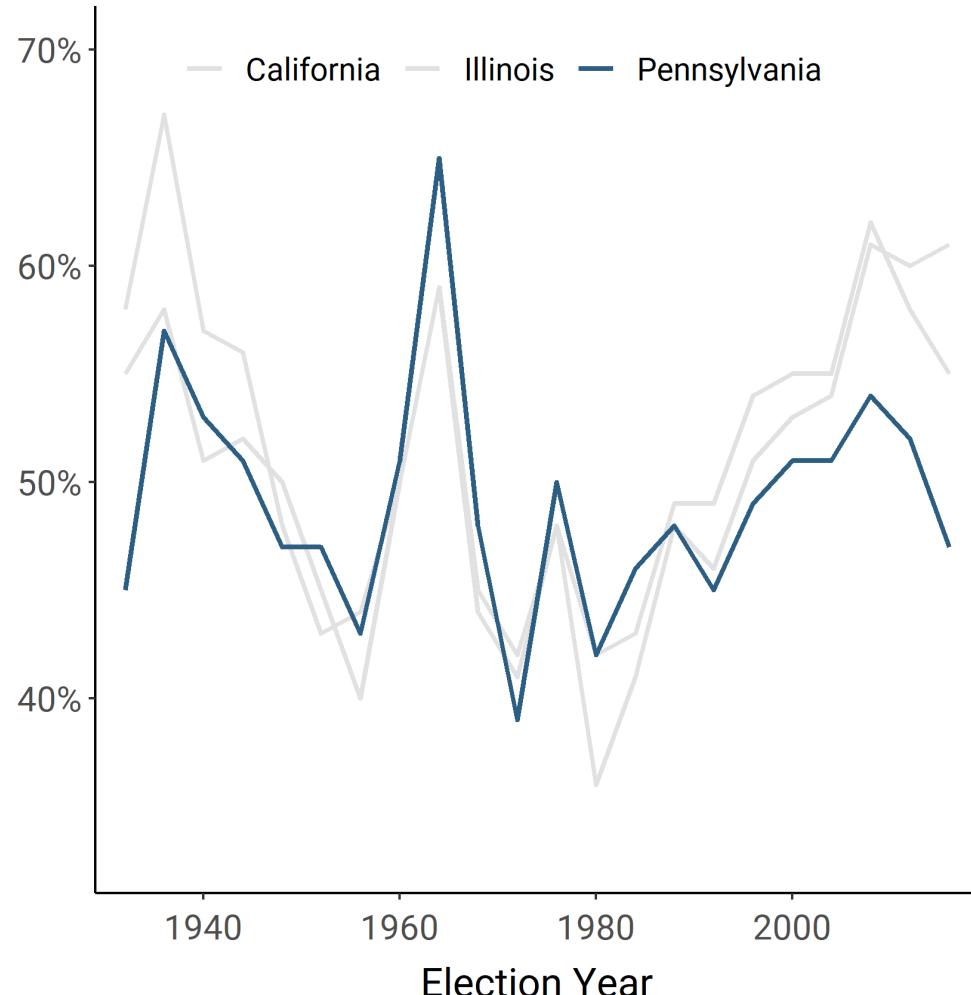
*We're a swing state! Go vote!*



```
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"))  
)
```

## 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()
  )

```

## Percent of democrat votes by state

*We're a swing state! Go vote!*



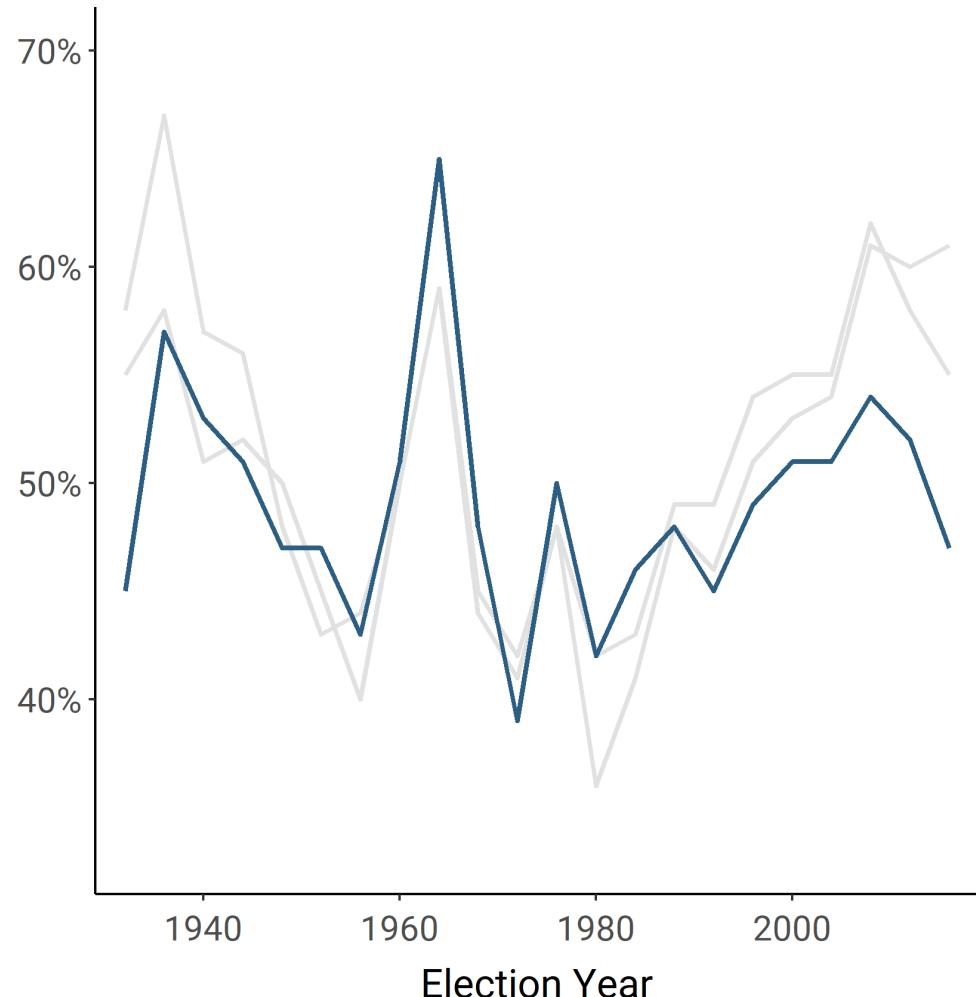
```

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 = "<span style='color:#2b5e82'>Pennsylvania</span> is a swi"

```

## Percent of democrat votes by state

<span style='color:#2b5e82'>Pennsylvania</span> is a swi



```

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 = "<span style='color:#2b5e82'>Pennsylvania</span>")
  theme(
    plot.subtitle = ggtext::element_markdown()
  )

```

## Percent of democrat votes by state

*Pennsylvania is a swing state! Go vote!*



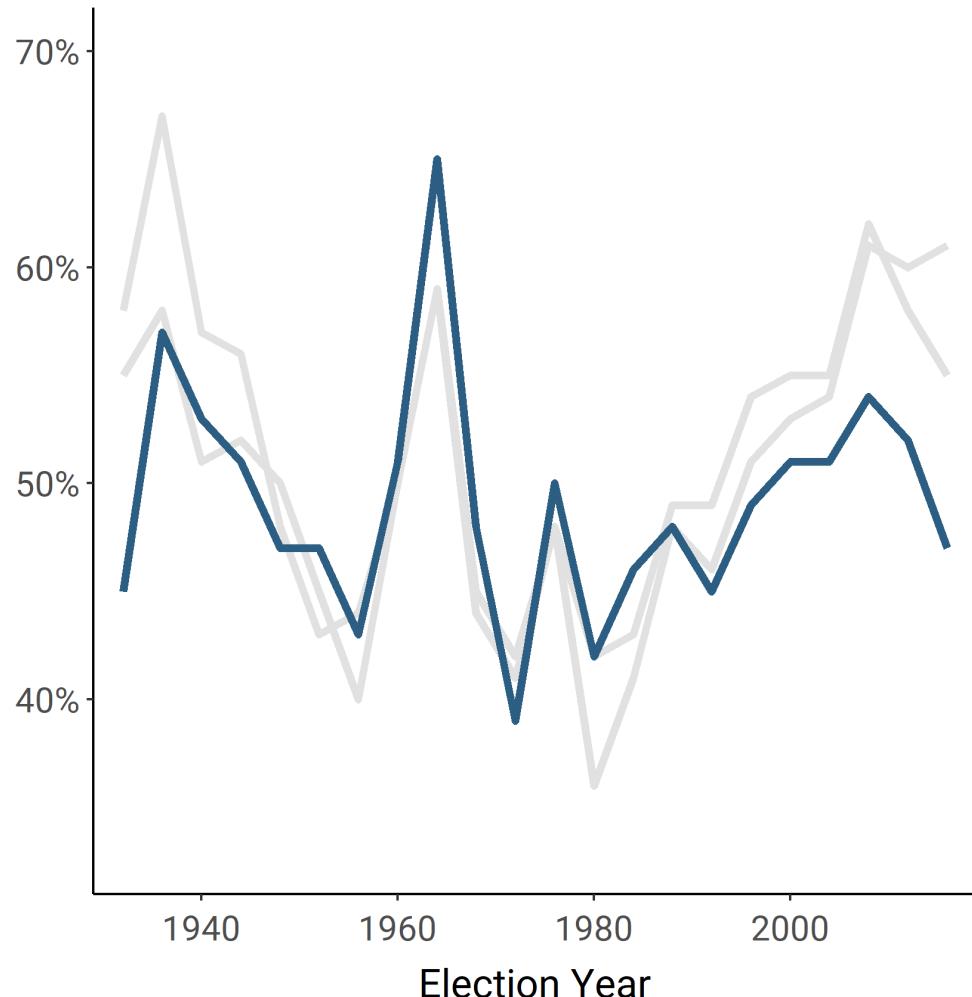
```

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 = "<span style='color:#2b5e82'>Pennsylvania</span>",
       theme(
         plot.subtitle = ggtext::element_markdown()
       ) +
       geom_line(size = 1.5)
  )

```

## Percent of democrat votes by state

*Pennsylvania is a swing state! Go vote!*



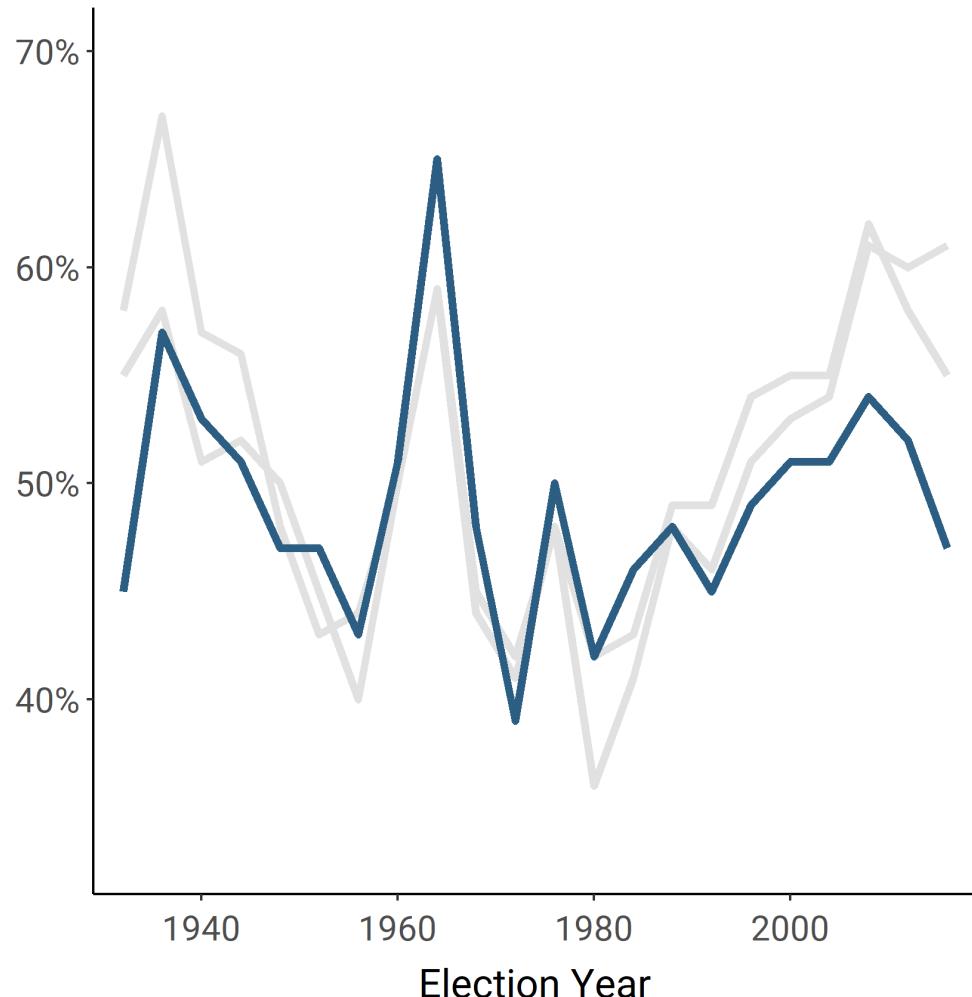
```

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 = "<span style='color:#0E4369'>Pennsylvania</span>")
  theme(
    plot.subtitle = ggtext::element_markdown()
  ) +
  geom_line(size = 1.5)

```

## Percent of democrat votes by state

*Pennsylvania is a swing state! Go vote!*



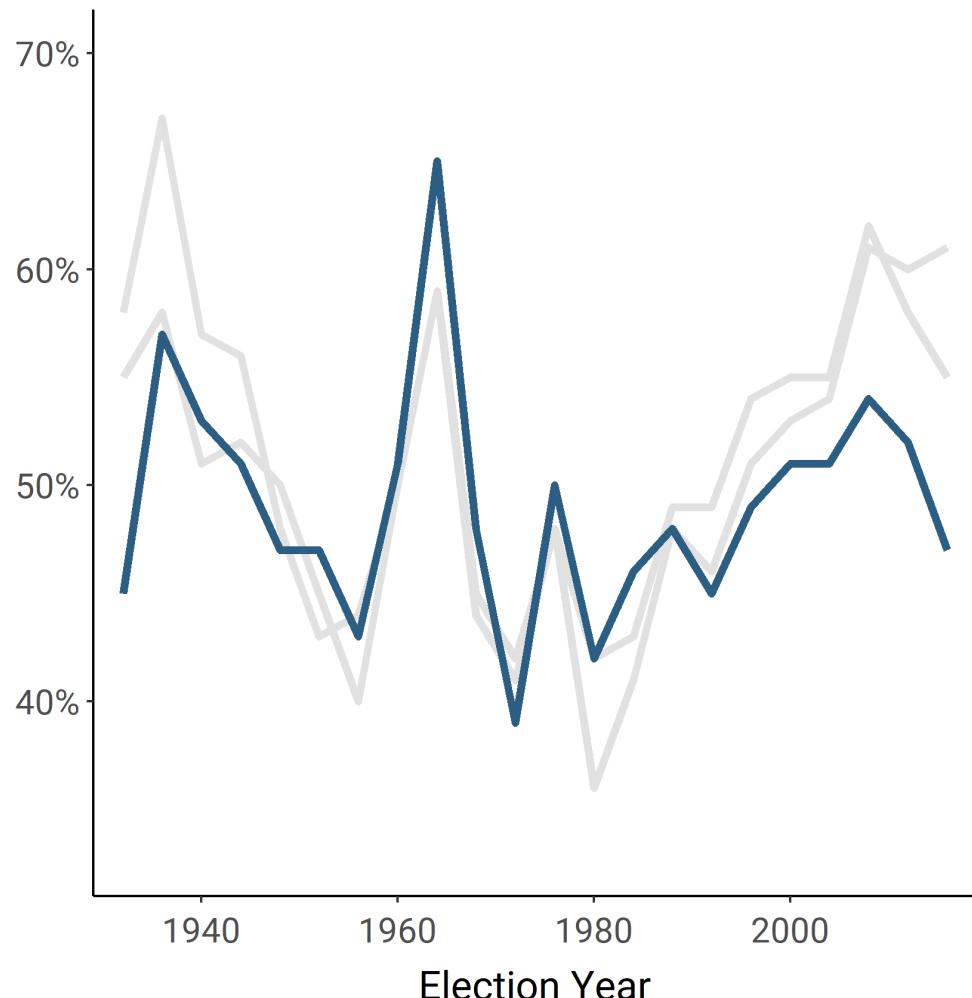
```

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!



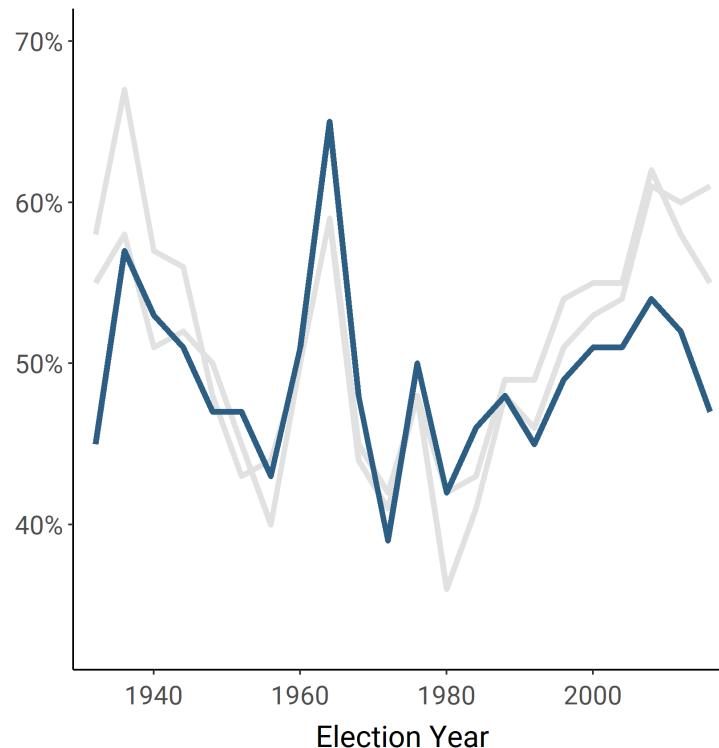
# 4. Colors (End!)

```
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
```

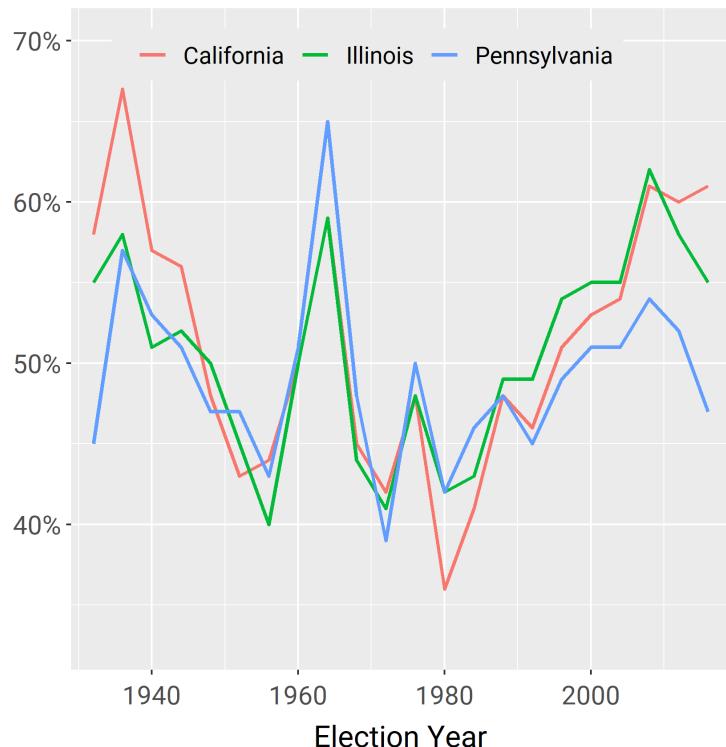
Percent of democrat votes by state  
**Pennsylvania** is a swing state! Go vote!



# 4. Colors (Before-After)

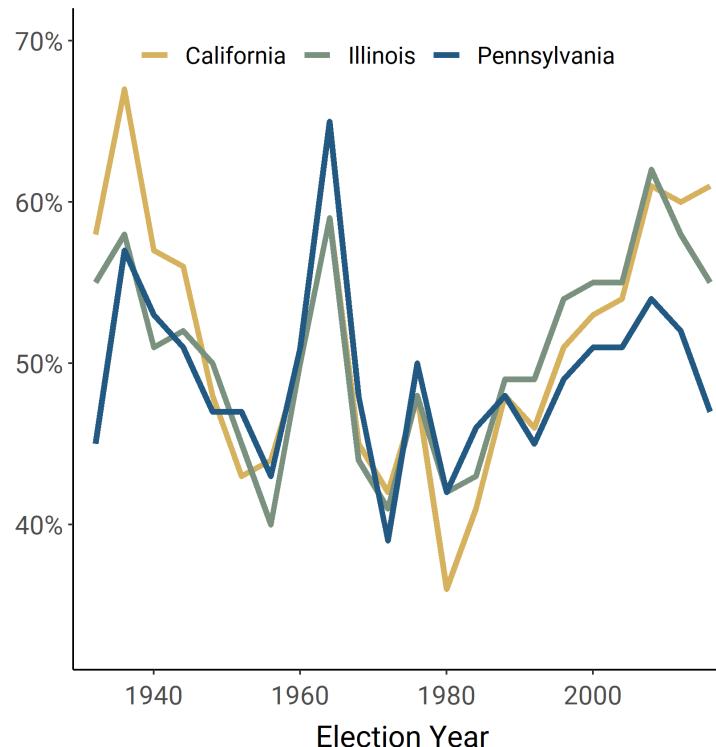
Percent of democrat votes by state

We're a swing state! Go vote!



Percent of democrat votes by state

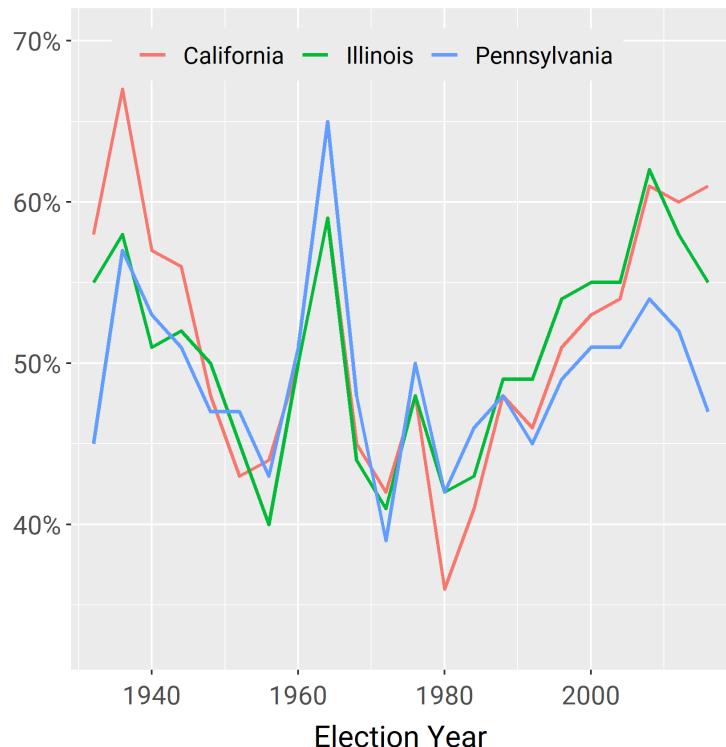
We're a swing state! Go vote!



# 4. Colors (Before-After)

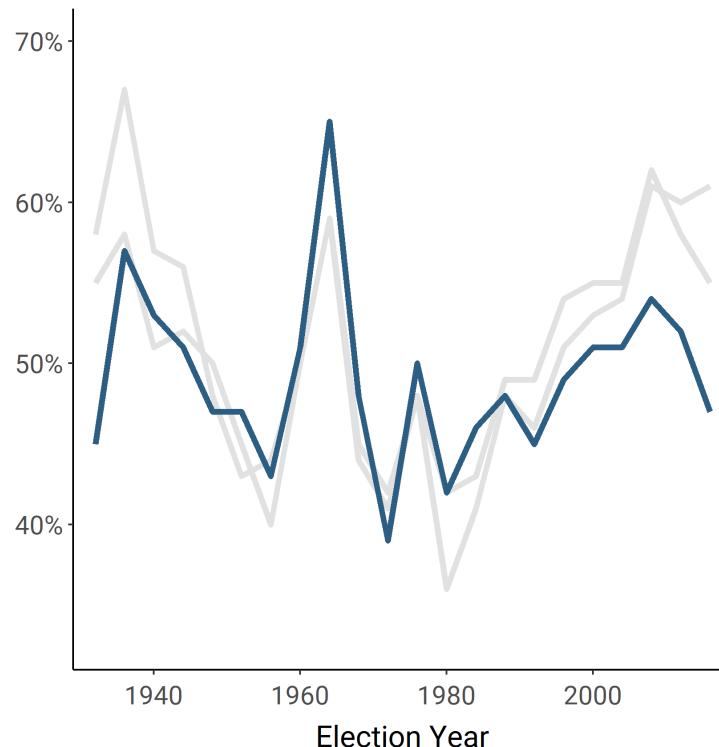
Percent of democrat votes by state

*We're a swing state! Go vote!*

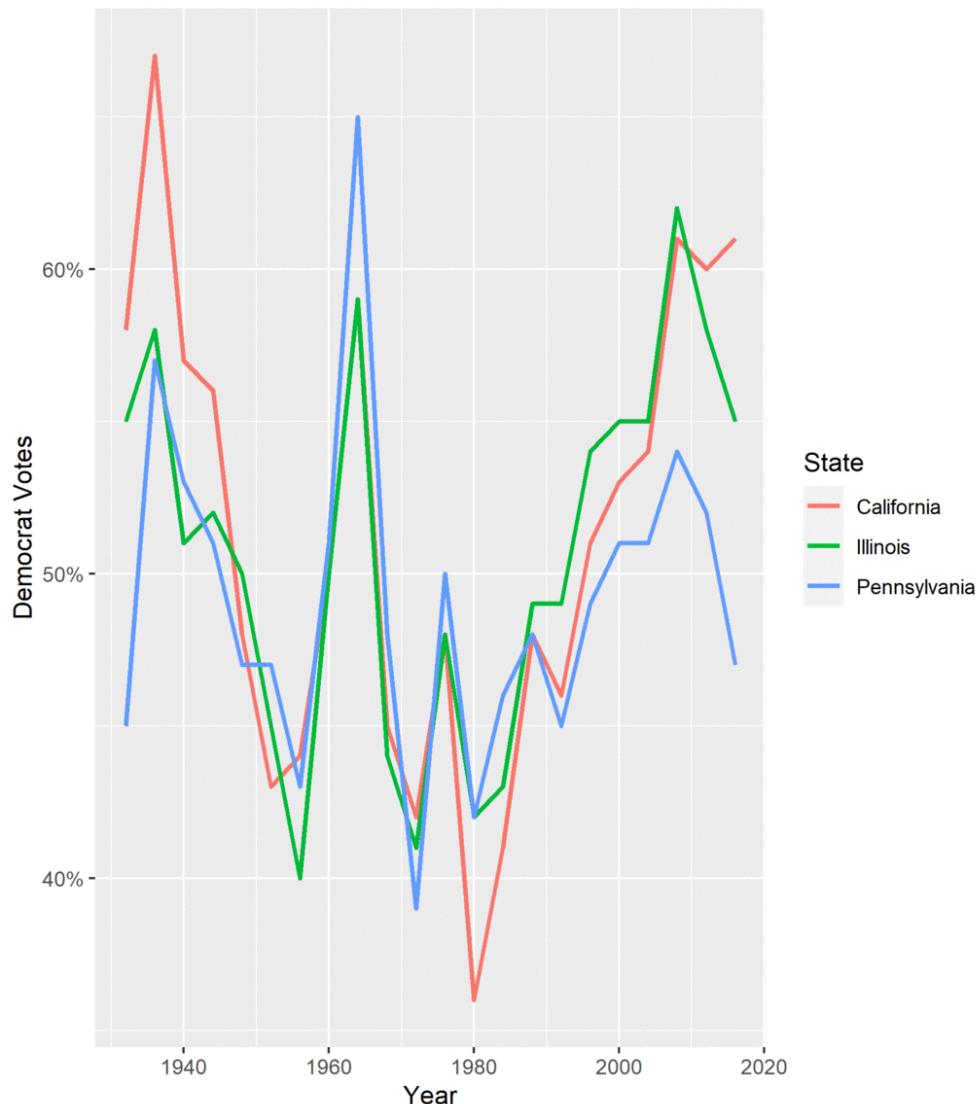


Percent of democrat votes by state

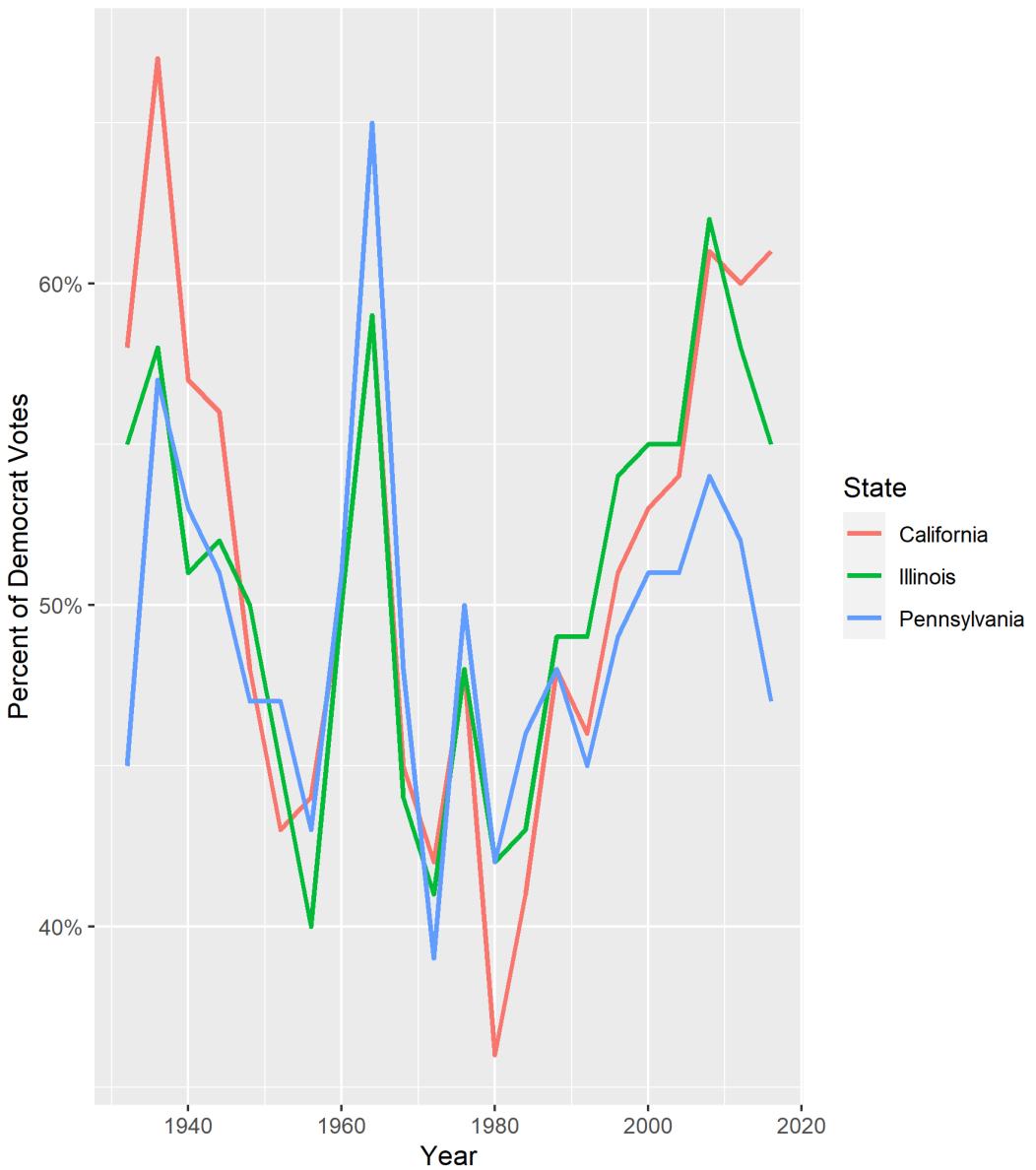
*Pennsylvania is a swing state! Go vote!*



Go Vote!

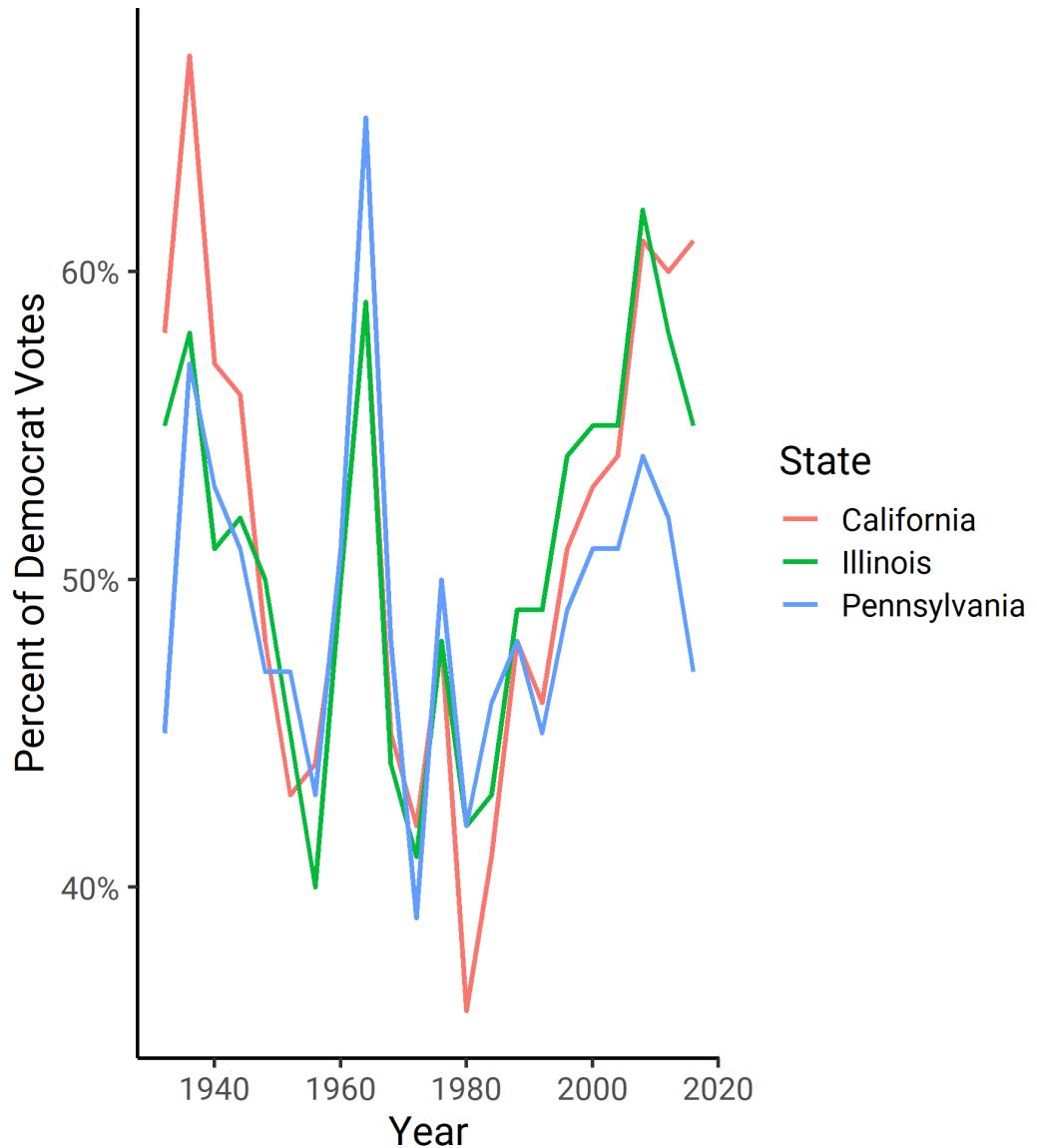


## Go Vote!

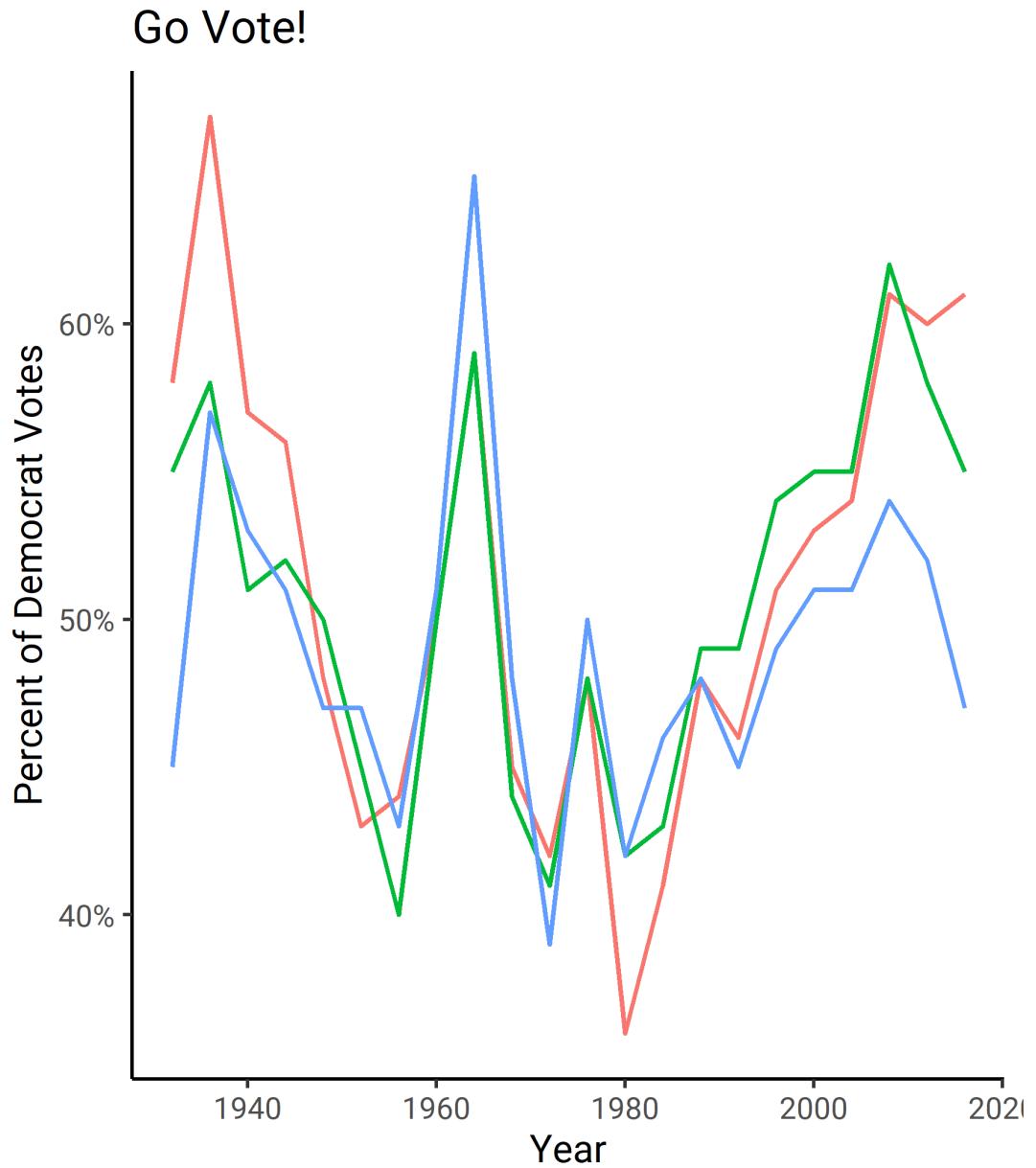


```
state_election_plot +  
  theme_classic(  
    base_family = "Roboto",  
    base_size = 16  
)
```

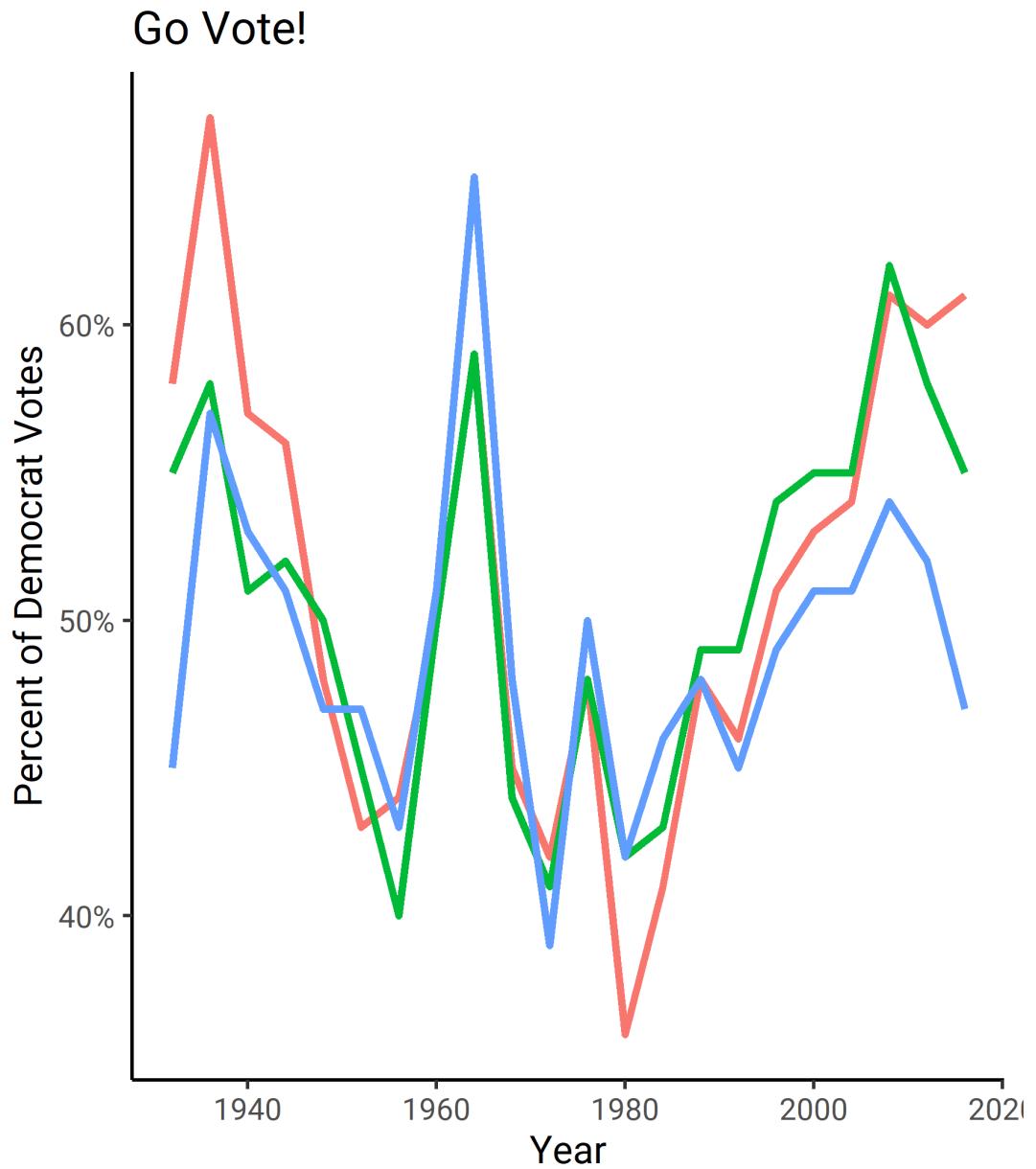
## Go Vote!



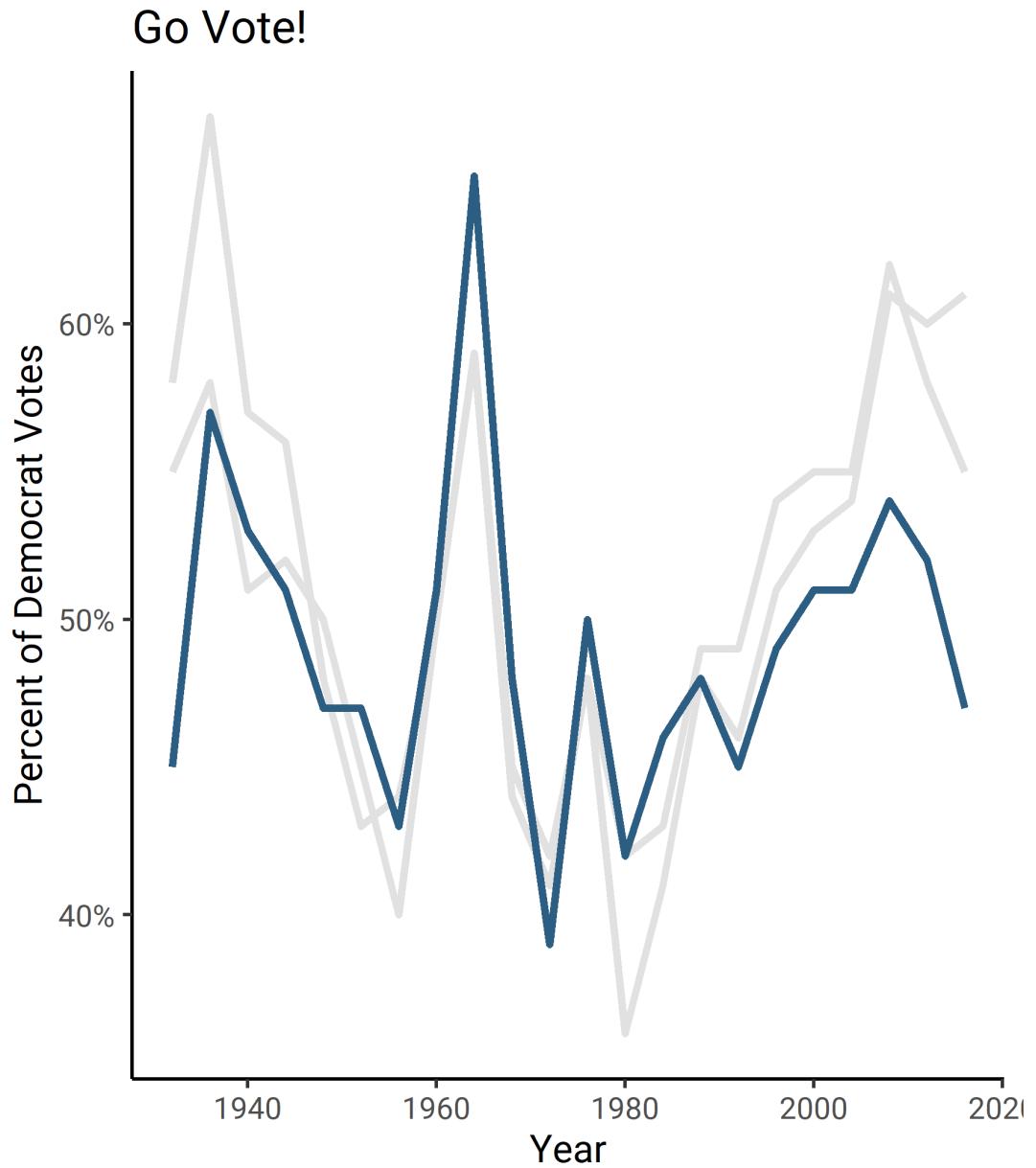
```
state_election_plot +  
  theme_classic(  
    base_family = "Roboto",  
    base_size = 16  
  ) +  
  guides(color = guide_none())
```



```
state_election_plot +  
  theme_classic(  
    base_family = "Roboto",  
    base_size = 16  
  ) +  
  guides(color = guide_none()) +  
  geom_line(size = 1.5)
```



```
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")  
  )
```



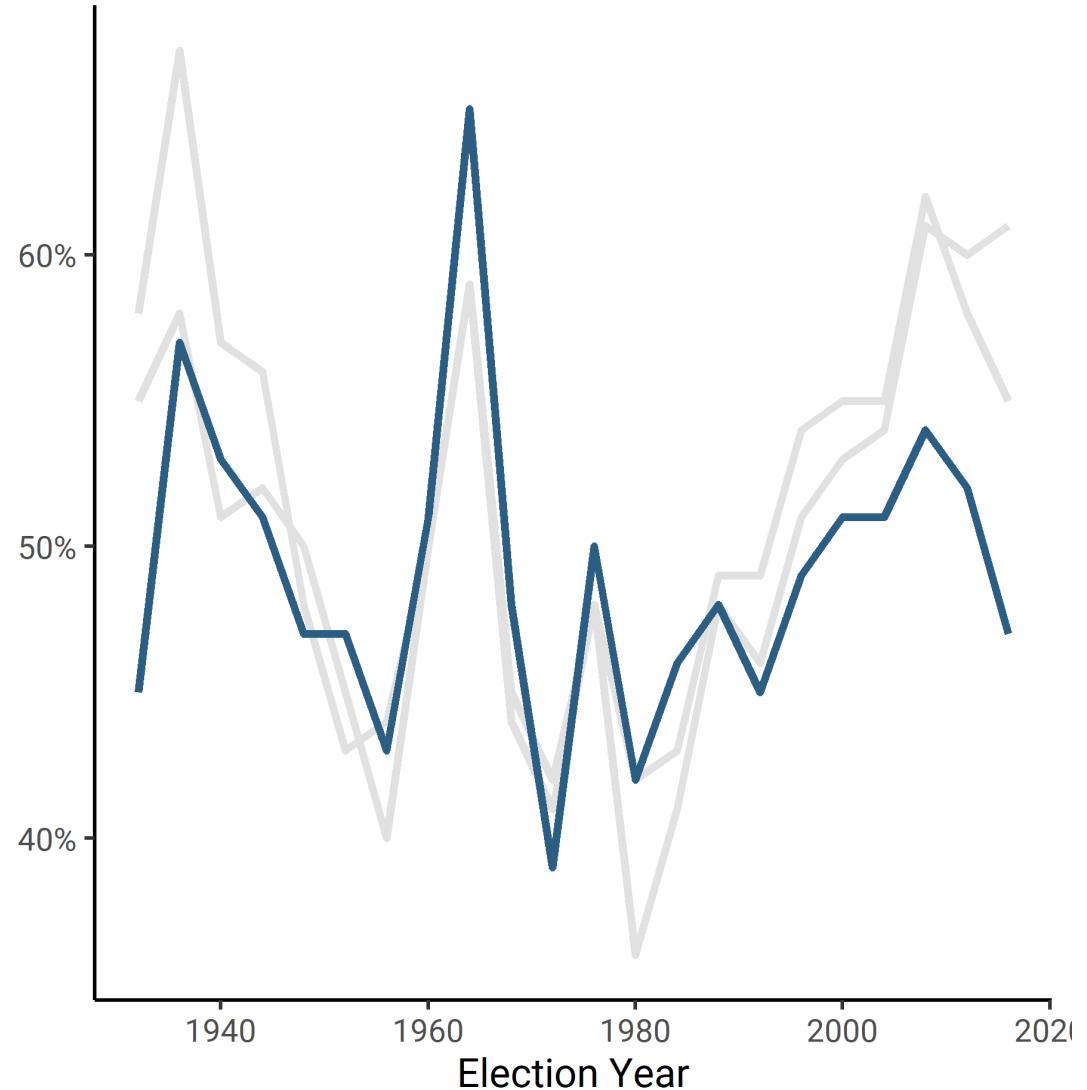
```

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> is"
  )

```

## Percent of democrat votes by state

<strong style='color:#0E4369'>Pennsylvania</strong> is



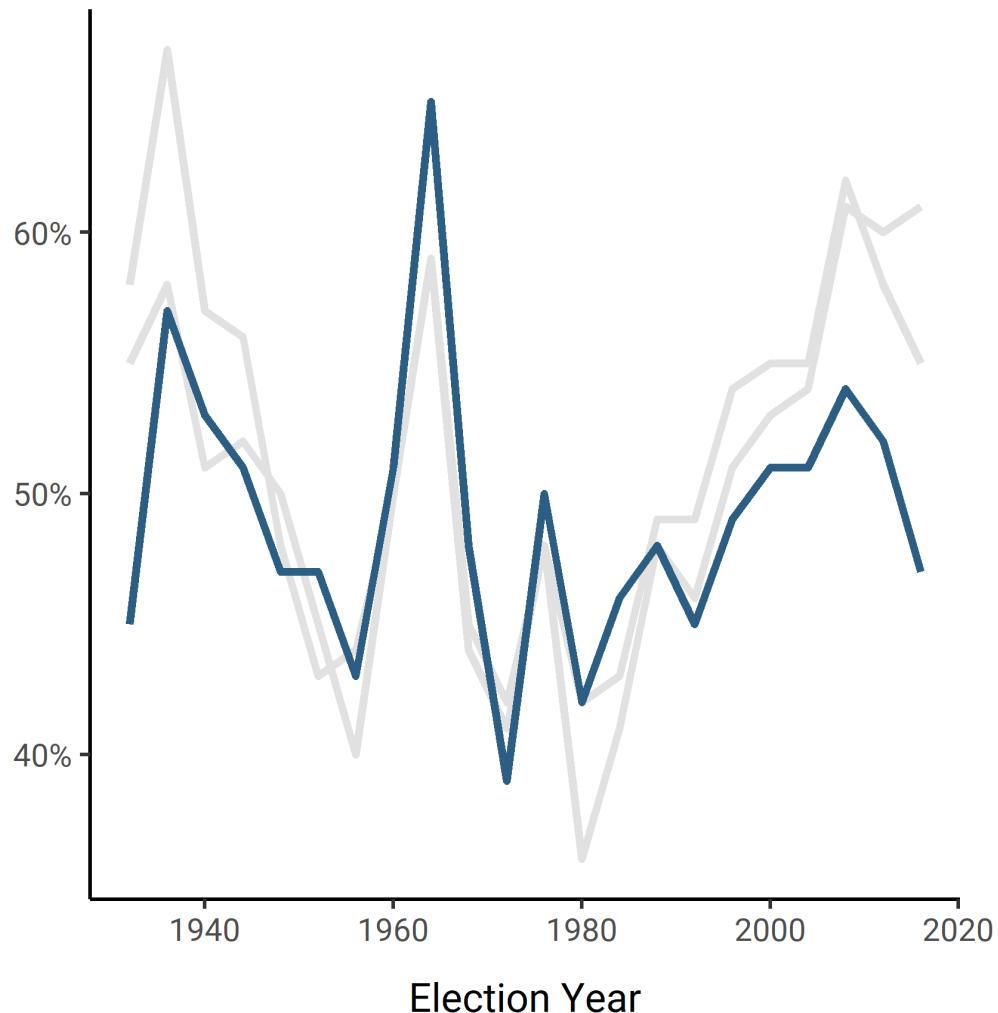
```

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

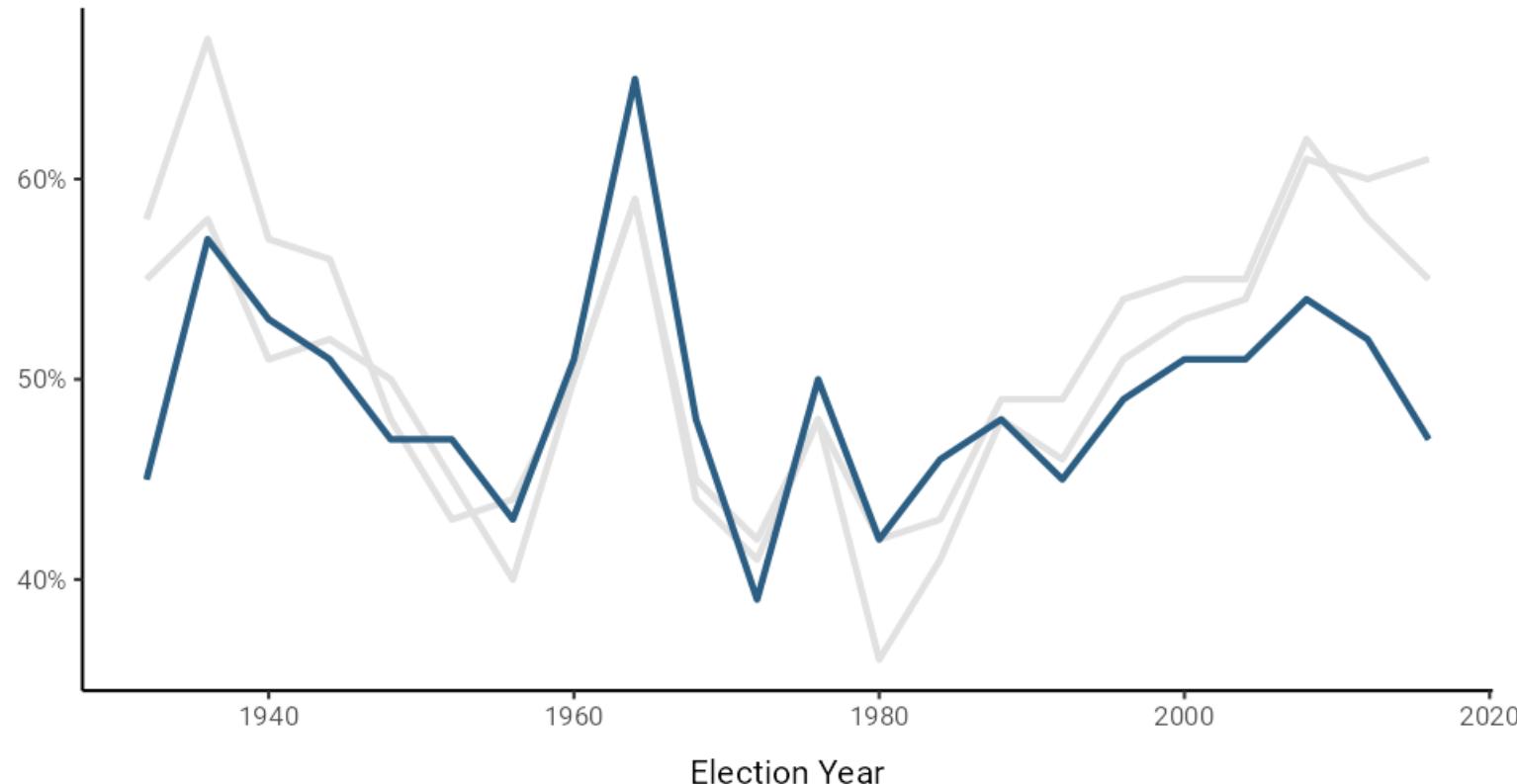
The screenshot shows the RStudio interface with the following components:

- Code Editor:** The main window displays R code for creating a plot. The code includes setting up a ggplot object, defining a theme with various aesthetic parameters like margin, font family, and size, and adding a subtitle. It also includes a chunk reveal command and a section titled "5(?). Plot quality".
- Environment Browser:** Located at the top right, it shows the global environment with two objects: "state\_election\_plot" (a gg object) and "state\_election\_votes" (a tbl\_df object).
- Plots Viewer:** At the bottom right, it shows a list of plots and their details, such as "5(?). Plot quality" (gg), "Showcasing" (gg), and "First, save our theme!" (gg).
- Console:** The bottom left shows the R console output, which is identical to the code in the editor, indicating the plot has been successfully generated.

## Point-and-click method in RStudio

### Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!

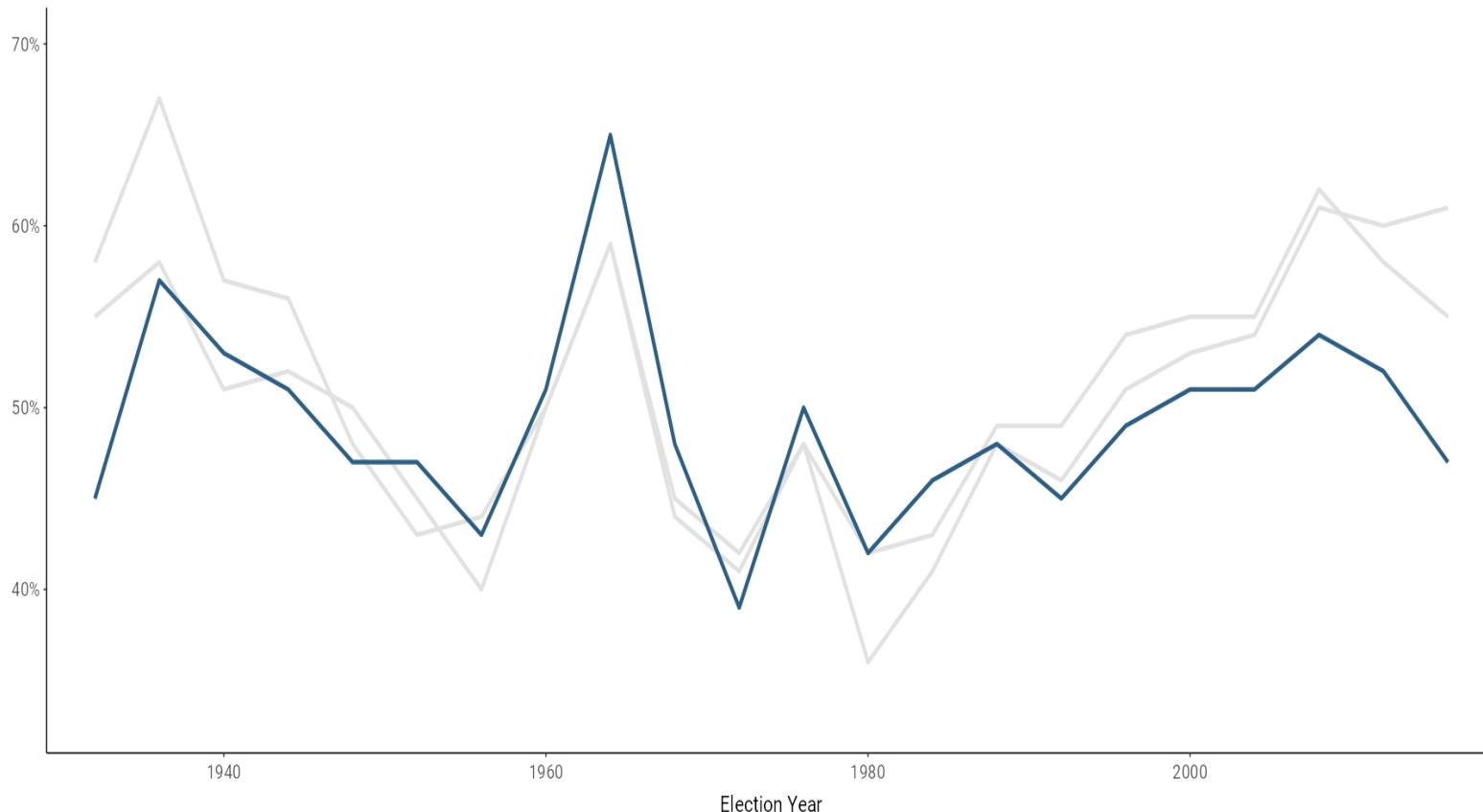


# 5(?). Plot quality: Scaling

## 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
```

# 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
```

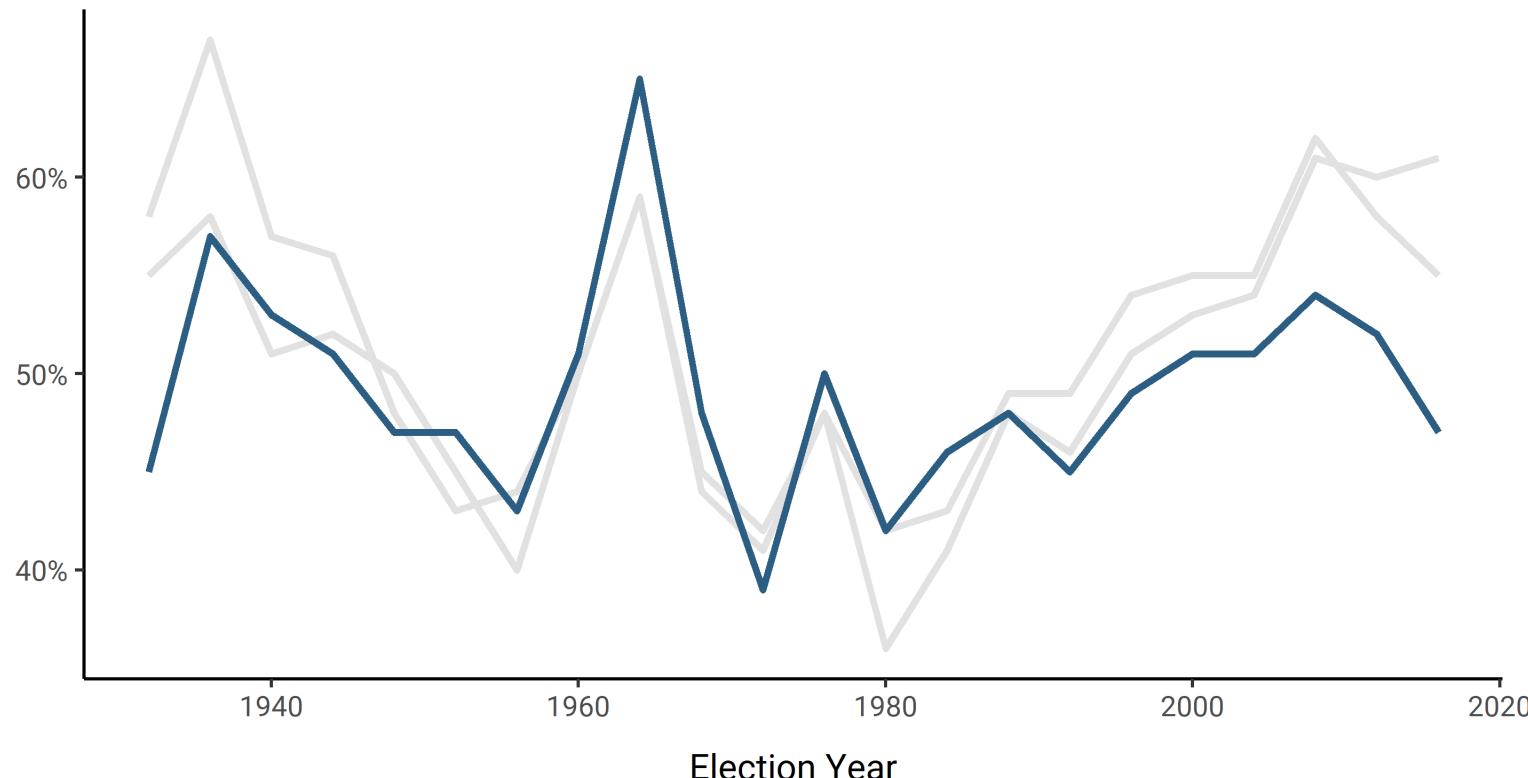
Use better-quality, OS-independent graphic devices (works for any figure):

```
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())
```

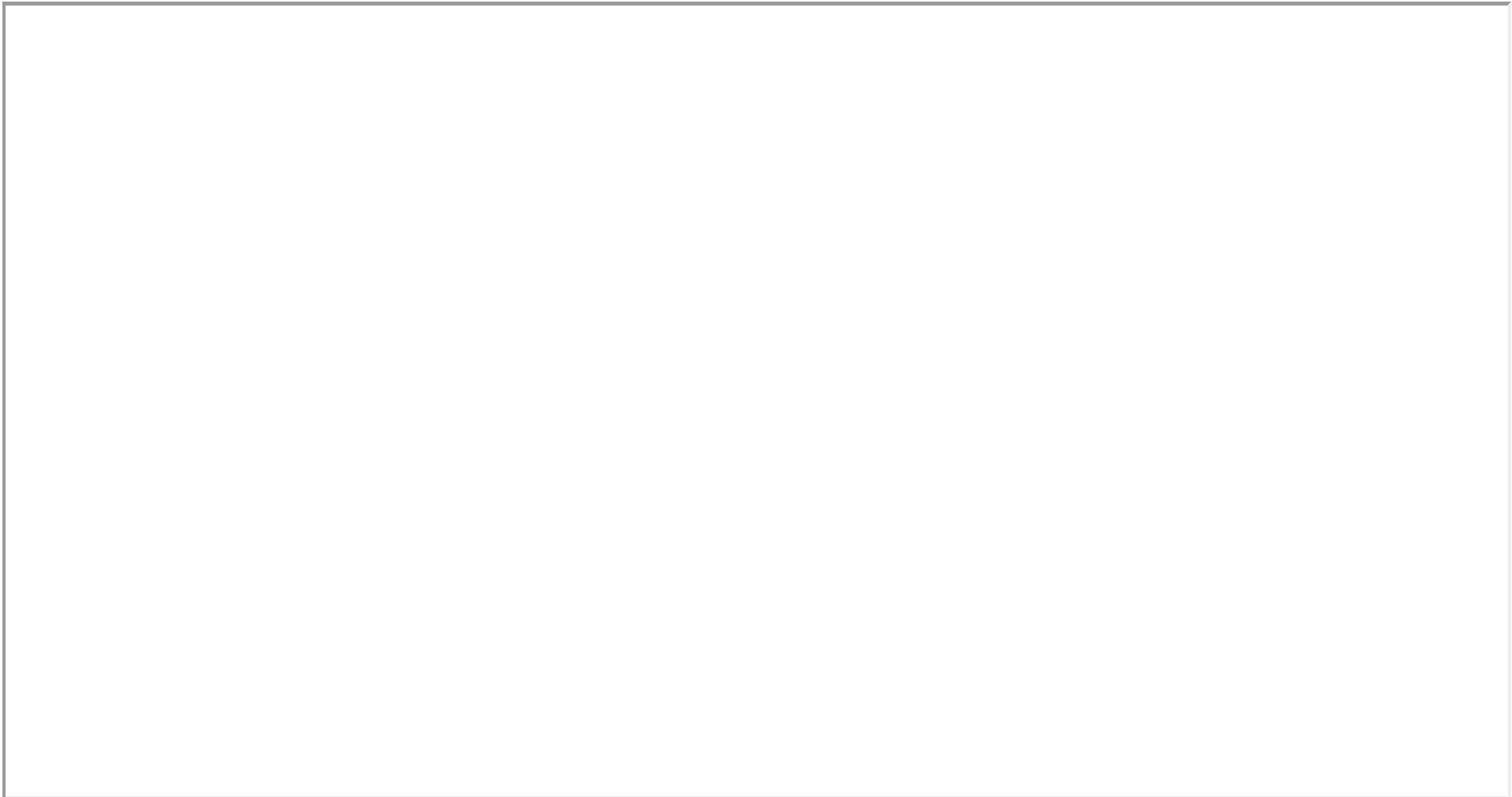
ggsave() png

## Percent of democrat votes by state

Pennsylvania is a swing state! Go vote!

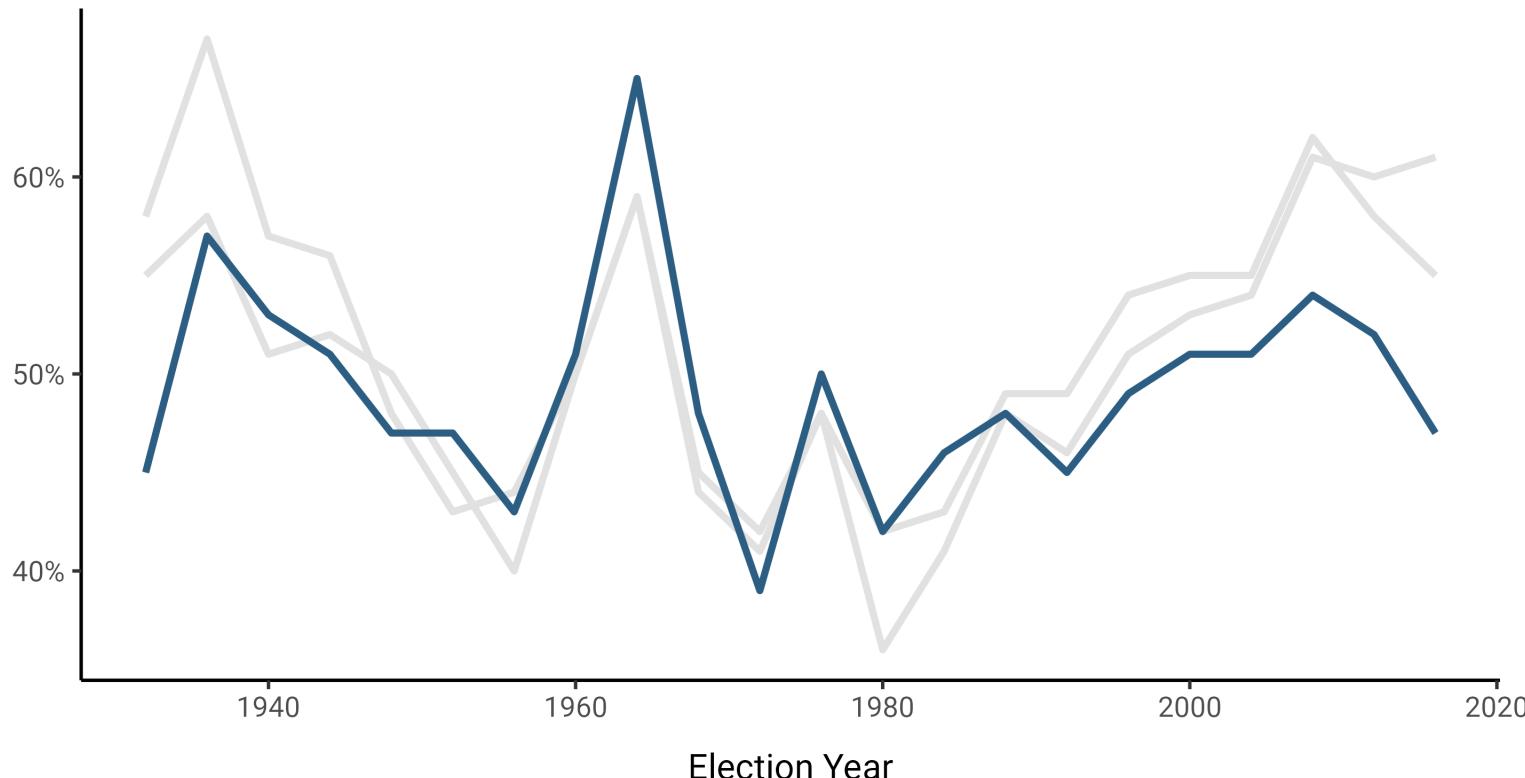


## **ggsave() pdf**



## 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()
```

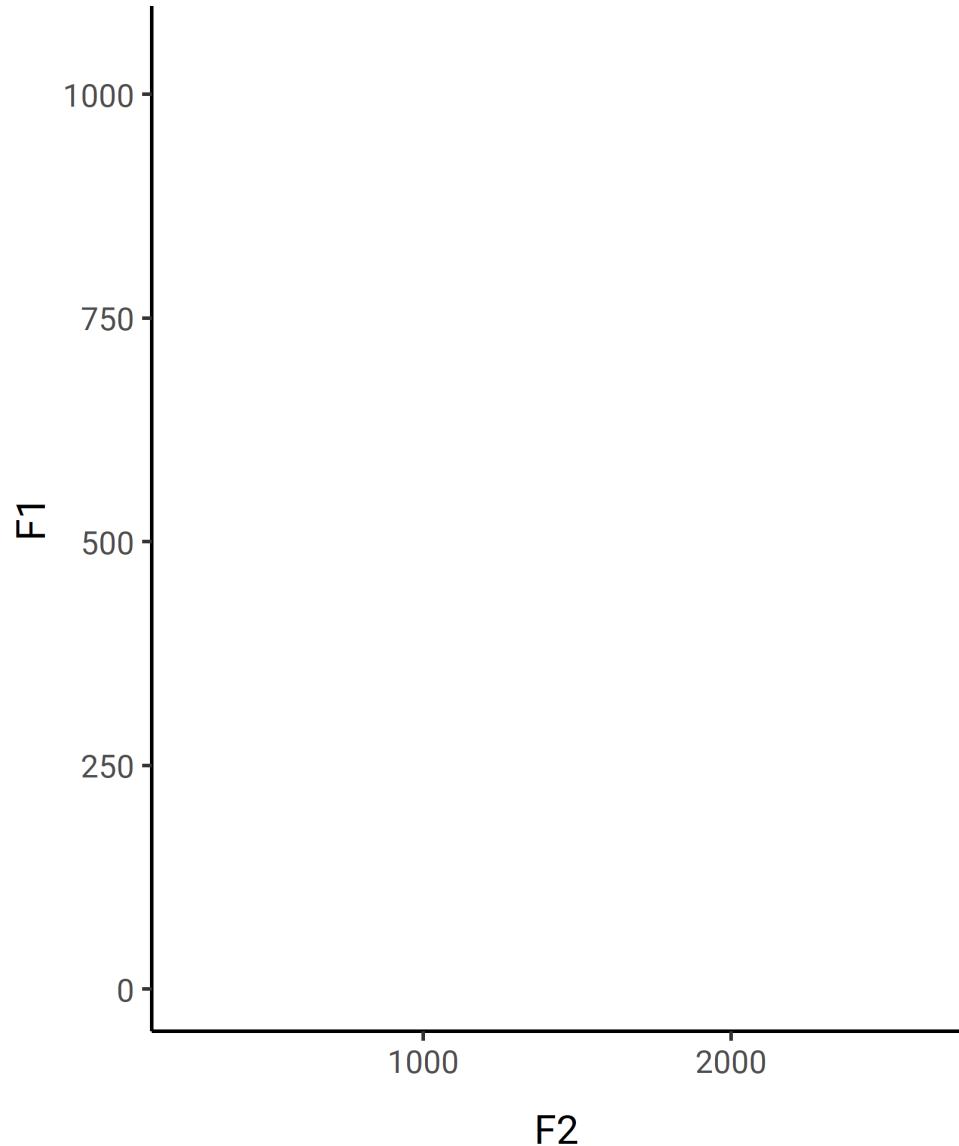
# **Vowel Plots: Vowel Formants**

```
sim_vowel_data
```

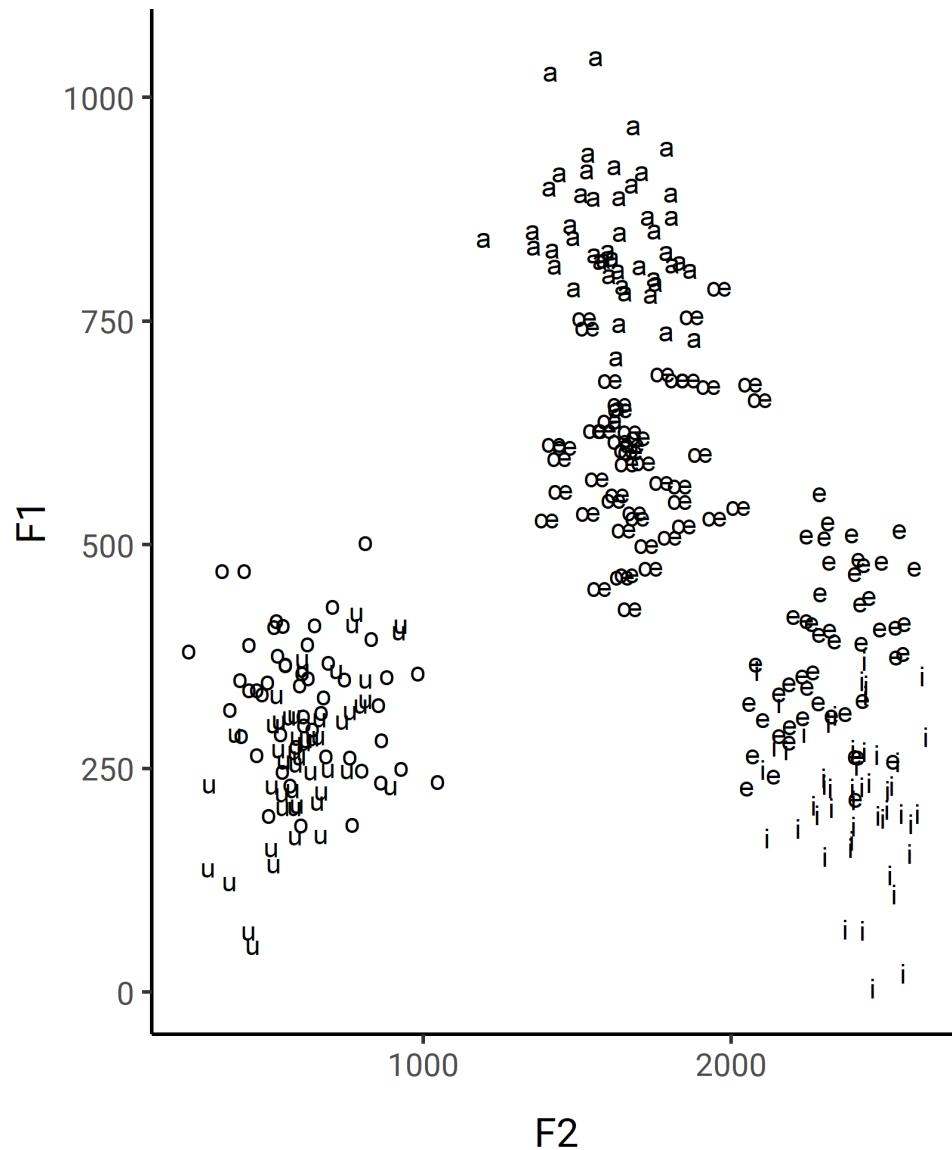
```
# A tibble: 300 x 3
  Vowel    F1     F2
  <chr> <dbl> <dbl>
1 i      371.  2431.
2 i      332.  2435.
3 i      286.  2630.
4 i      110.  2527.
5 i      302.  2315.
6 i      199.  2279.
7 i      152.  2303.
8 i      173.  2116.
9 i      258.  2540.
10 i     200.  2604.
# ... with 290 more rows
```

```
sim_vowel_data %>%
```

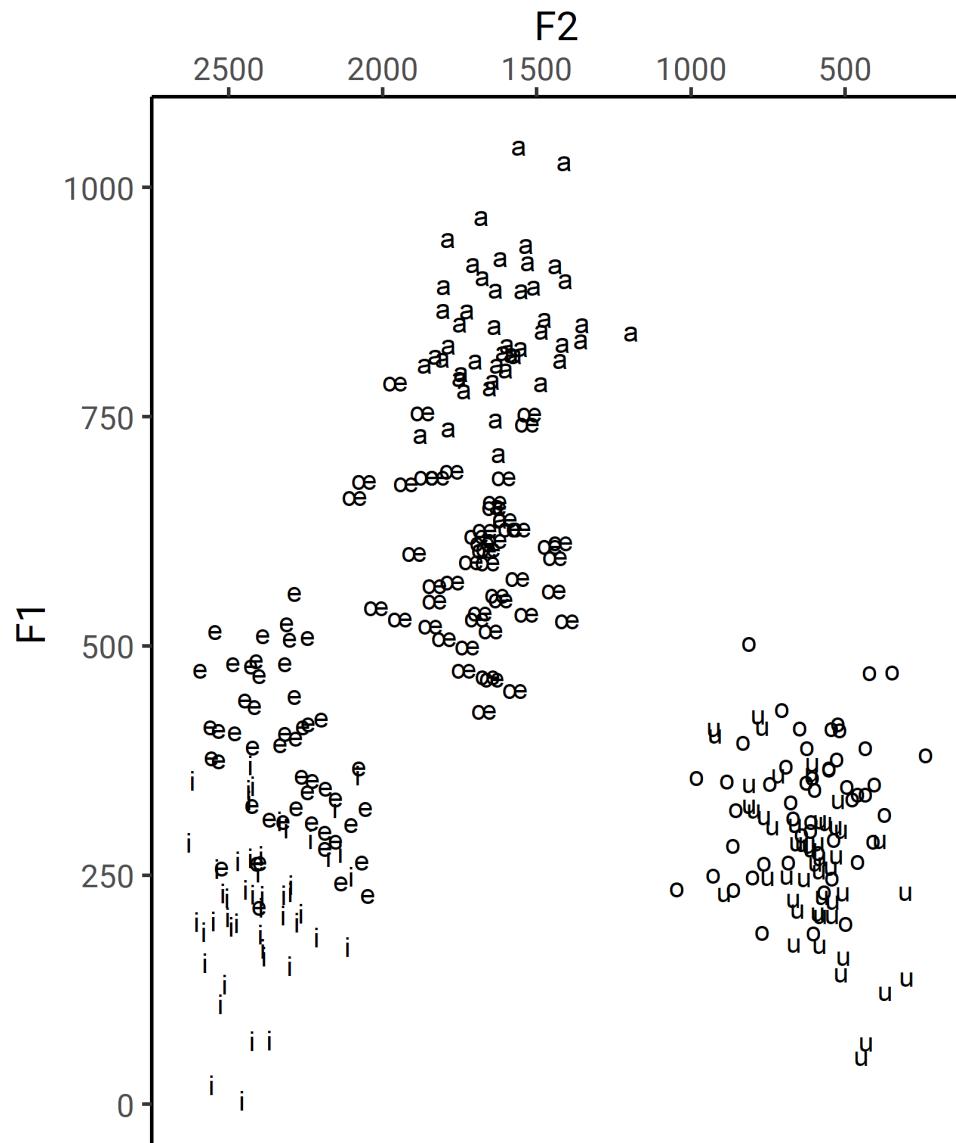
```
  ggplot(aes(x = F2, y = F1))
```



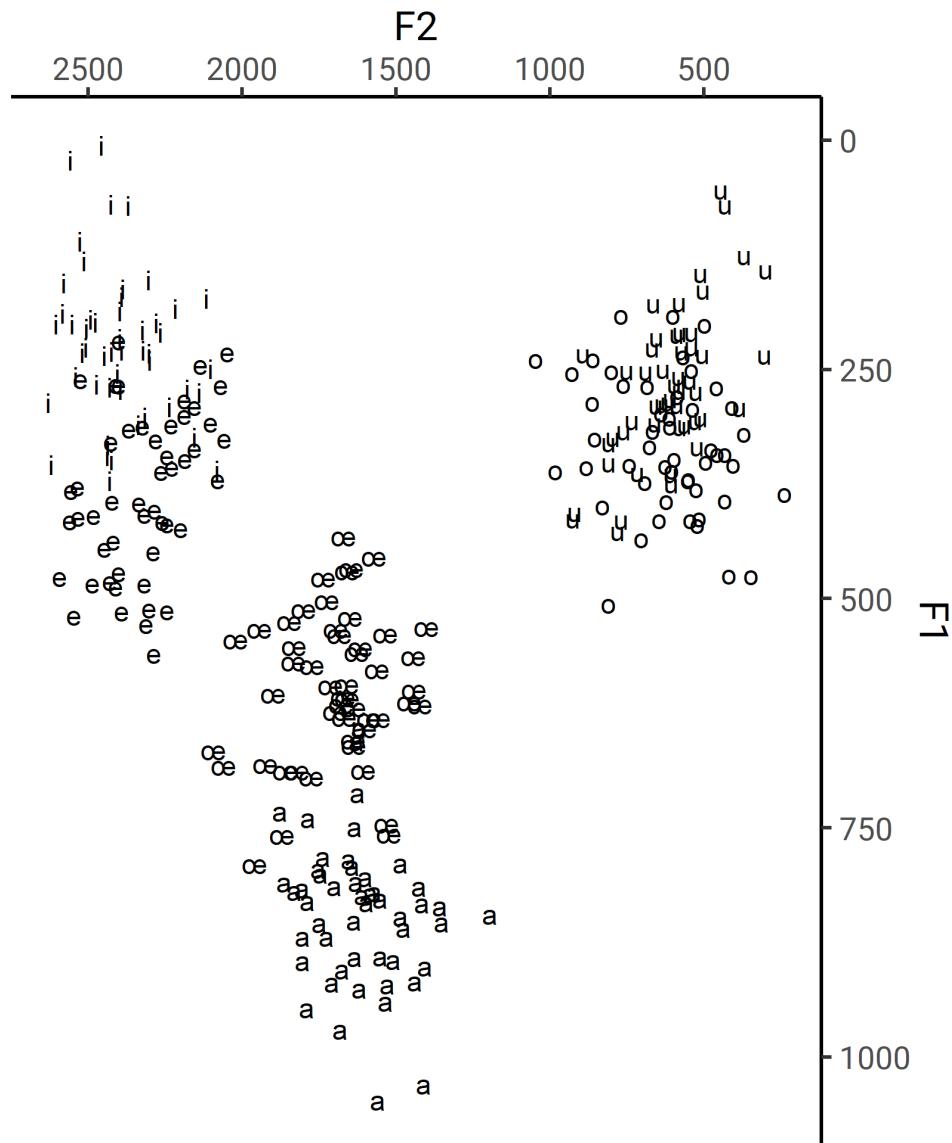
```
sim_vowel_data %>%
  ggplot(aes(x = F2, y = F1)) +
  geom_text(aes(label = Vowel))
```



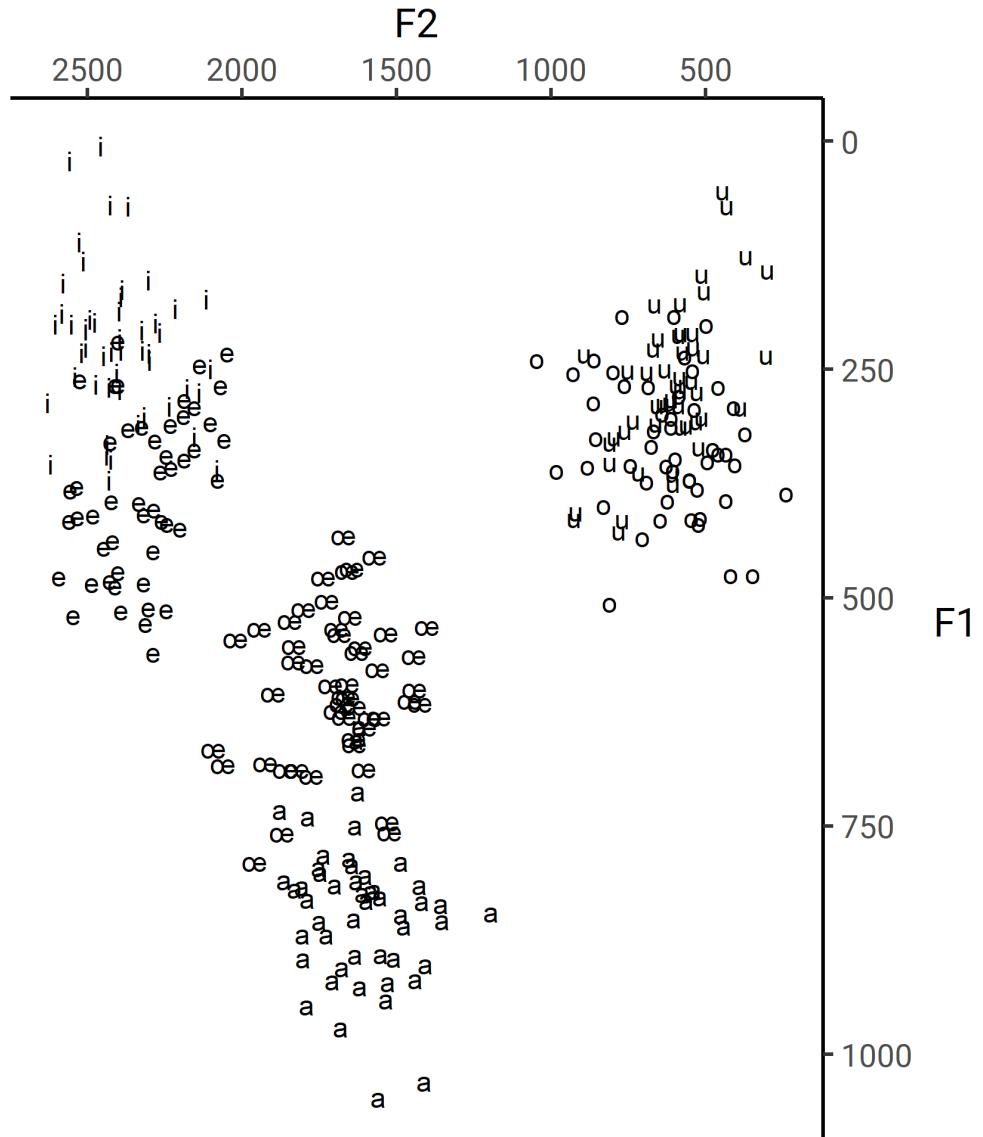
```
sim_vowel_data %>%
  ggplot(aes(x = F2, y = F1)) +
  geom_text(aes(label = Vowel)) +
  scale_x_reverse(
    position = "top",
    breaks = pretty_breaks(5)
  )
```



```
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")
```



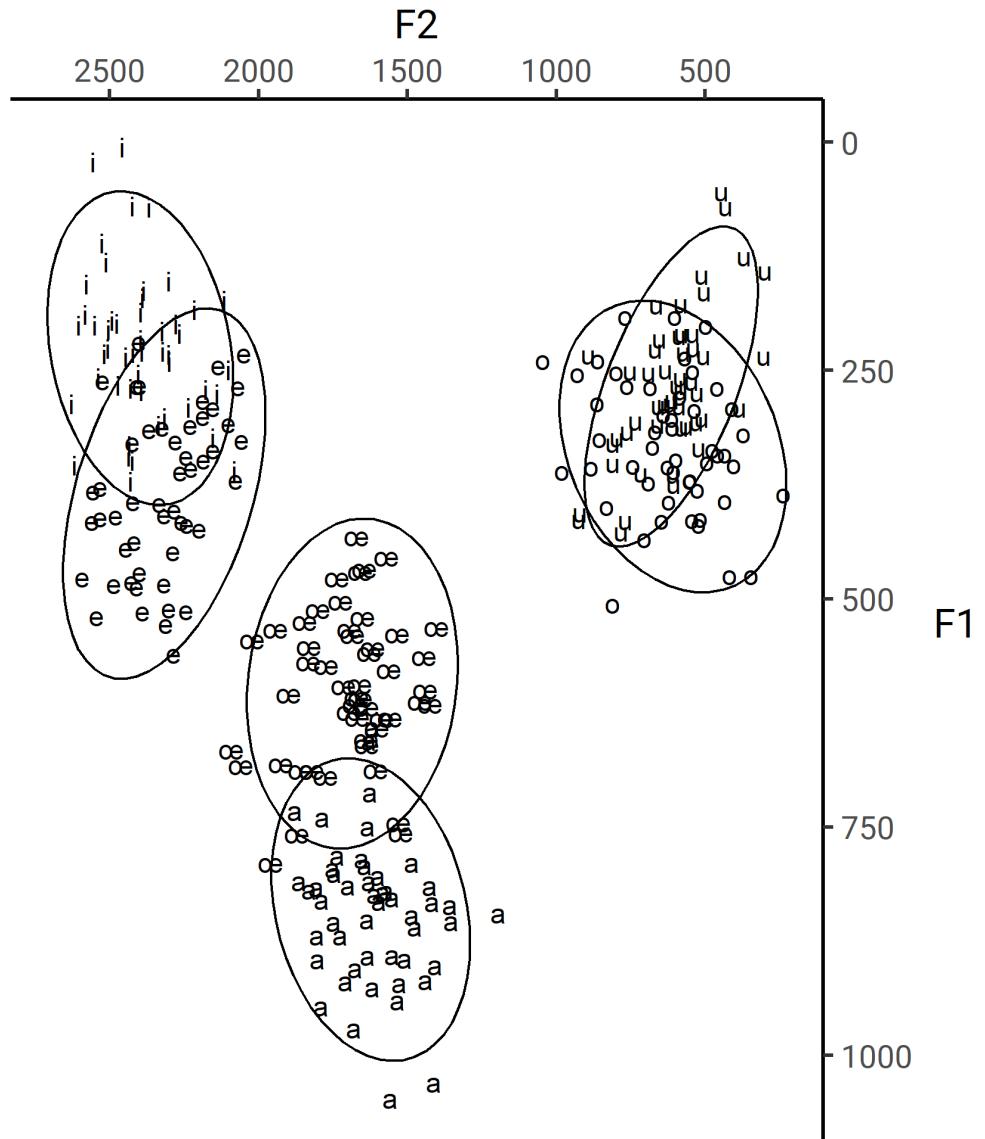
```
sim_vowel_data %>%
  ggplot(aes(x = F2, y = F1)) +
  geom_text(aes(label = Vowel)) +
  scale_x_reverse(
    position = "top",
    breaks = pretty_breaks(5)
  ) +
  scale_y_reverse(position = "right") +
  theme(
    plot.margin = margin(.5,.5, 1, 1, unit = "cm"),
    axis.title.x.top = element_text(margin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



```

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))

```



# Vowel Plots: Vowel Space

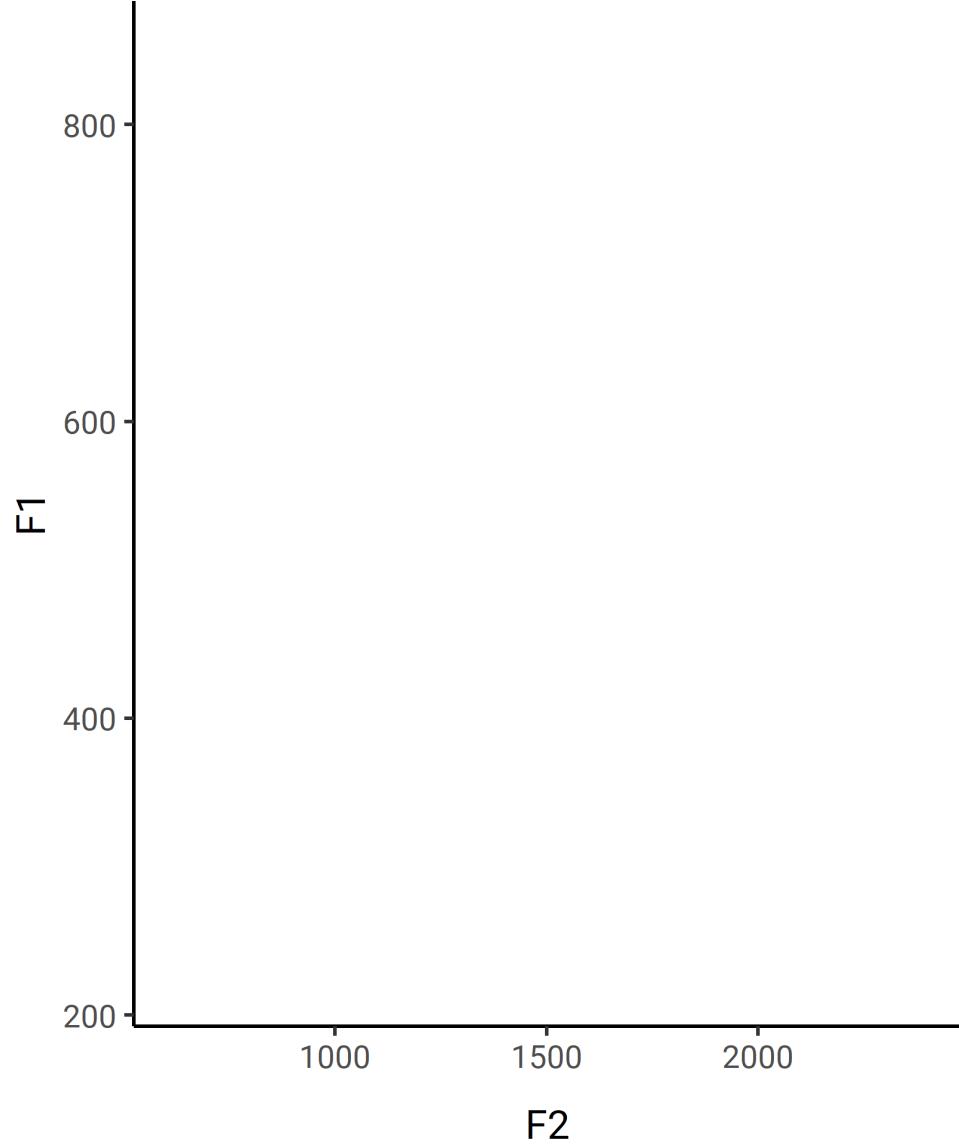
```
sim_vowel_data_all # A tibble: 800 x 3
#> # ... with 790 more rows
#> # ... with 3 variables: Vowel <chr>, F1 <dbl>, F2 <dbl>
#> #   i      371. 2431.
#> #   i      332. 2435.
#> #   i      286. 2630.
#> #   i      110. 2527.
#> #   i      302. 2315.
#> #   i      199. 2279.
#> #   i      152. 2303.
#> #   i      173. 2116.
#> #   i      258. 2540.
#> #   i      200. 2604.
```

```
sim_vowel_data_all %>%  
  group_by(Vowel)  
  
# A tibble: 800 x 3  
# Groups:   Vowel [16]  
  Vowel     F1     F2  
  <chr> <dbl> <dbl>  
1 i       371.  2431.  
2 i       332.  2435.  
3 i       286.  2630.  
4 i       110.  2527.  
5 i       302.  2315.  
6 i       199.  2279.  
7 i       152.  2303.  
8 i       173.  2116.  
9 i       258.  2540.  
10 i      200.  2604.  
# ... with 790 more rows
```

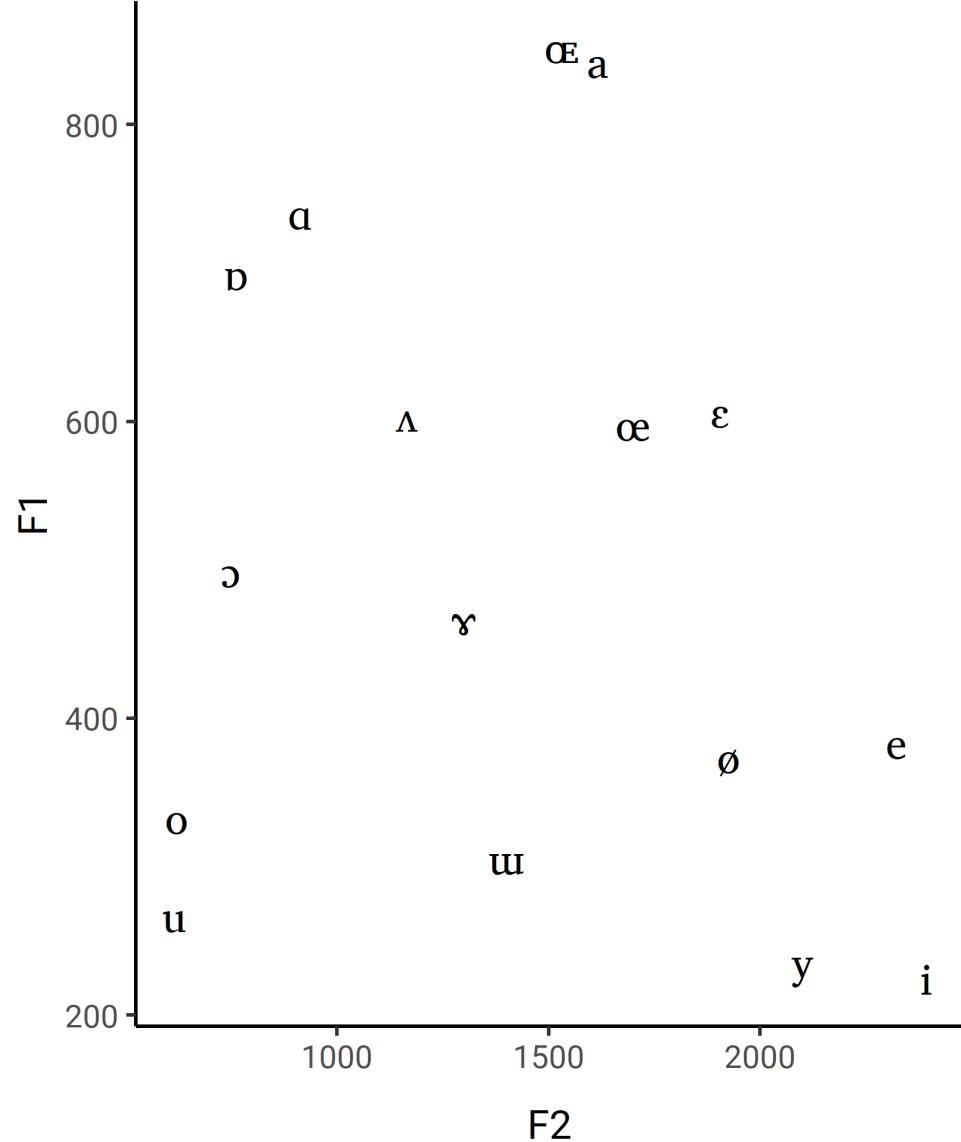
```
sim_vowel_data_all %>%  
  group_by(Vowel) %>%  
  summarize(across(c(F1, F2), mean), .groups = 'drop')
```

```
# A tibble: 16 x 3  
  Vowel     F1     F2  
  <chr> <dbl> <dbl>  
1 <U+0251>    740.  912.  
2 <U+0252>    699.  762.  
3 <U+0254>    498.  748.  
4 <U+025B>    606. 1905.  
5 <U+0264>    466. 1300.  
6 <U+026F>    305. 1400.  
7 <U+0276>    852. 1532.  
8 <U+028C>    603. 1165.  
9 a          842. 1617.  
10 e         383. 2321.  
11 i         224. 2394.  
12 o         332.  620.  
13 ø         373. 1924.  
14 œ         597. 1699.  
15 u         266.  614.  
16 y         235. 2099.
```

```
sim_vowel_data_all %>%
  group_by(Vowel) %>%
  summarize(across(c(F1, F2), mean), .groups = 'drop') %>%
  ggplot(aes(x = F2, y = F1))
```



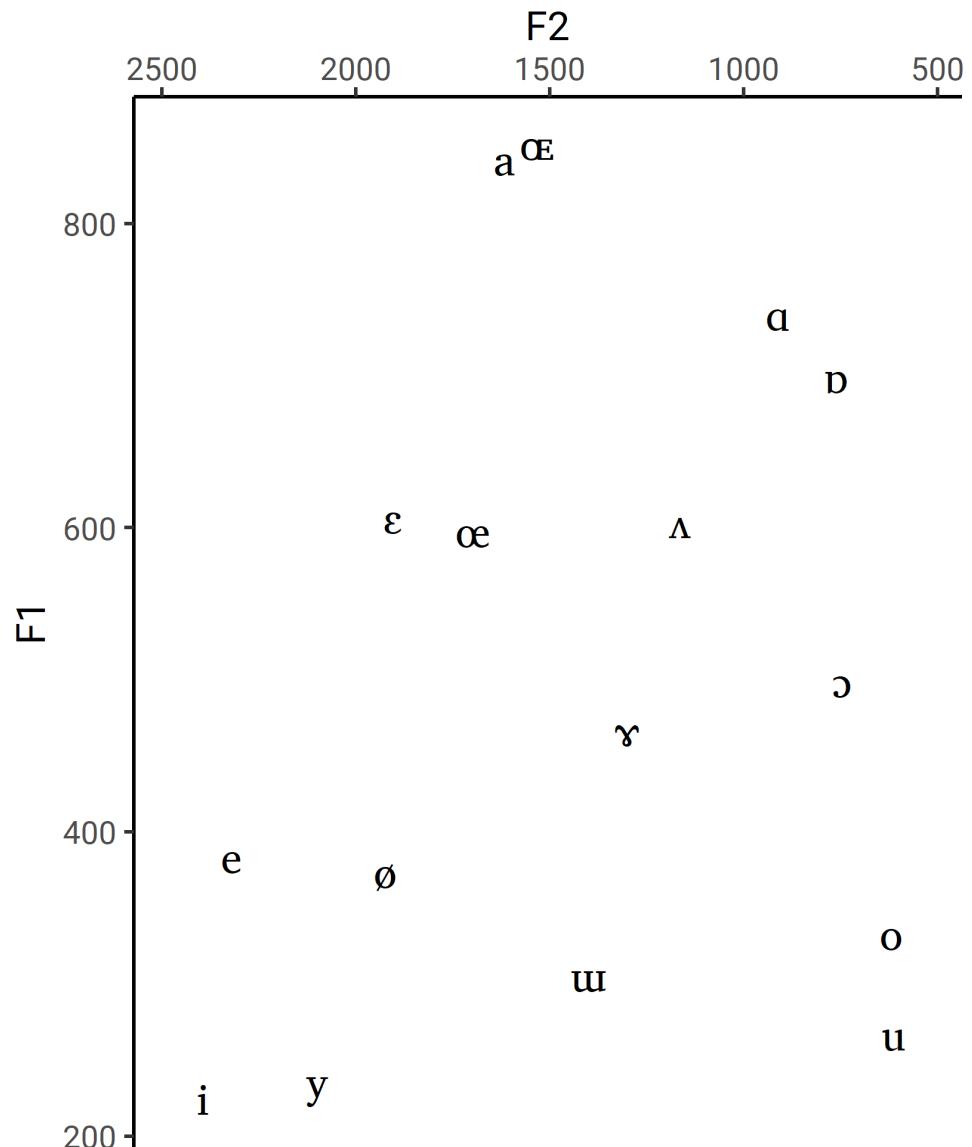
```
sim_vowel_data_all %>%
  group_by(Vowel) %>%
  summarize(across(c(F1, F2), mean), .groups = 'drop') %>%
  ggplot(aes(x = F2, y = F1)) +
  geom_text(aes(label = Vowel), size = 6, family = "Charis S
```



```

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 SC")
  scale_x_reverse(
    position = "top",
    breaks = pretty_breaks(5),
    expand = expansion(.1)
  )

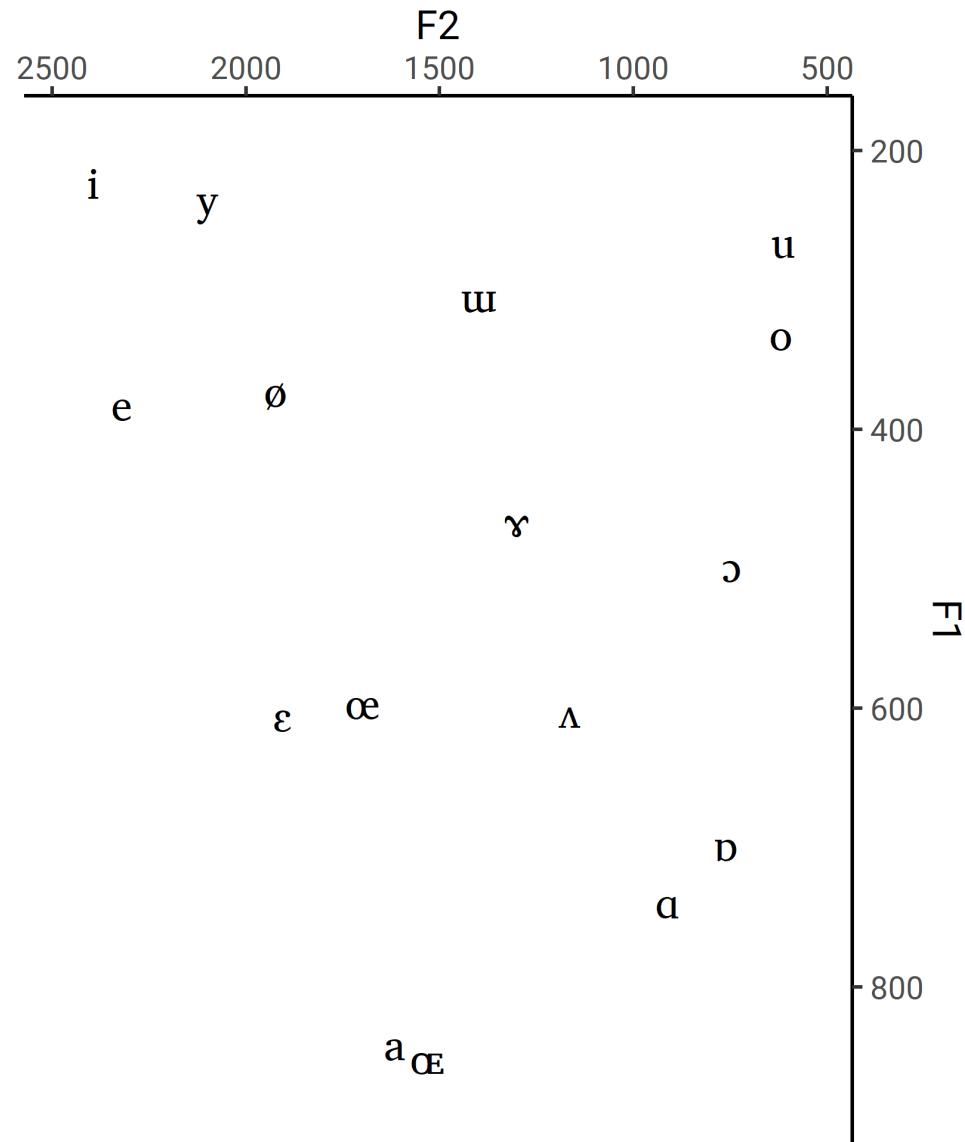
```



```

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)
  )

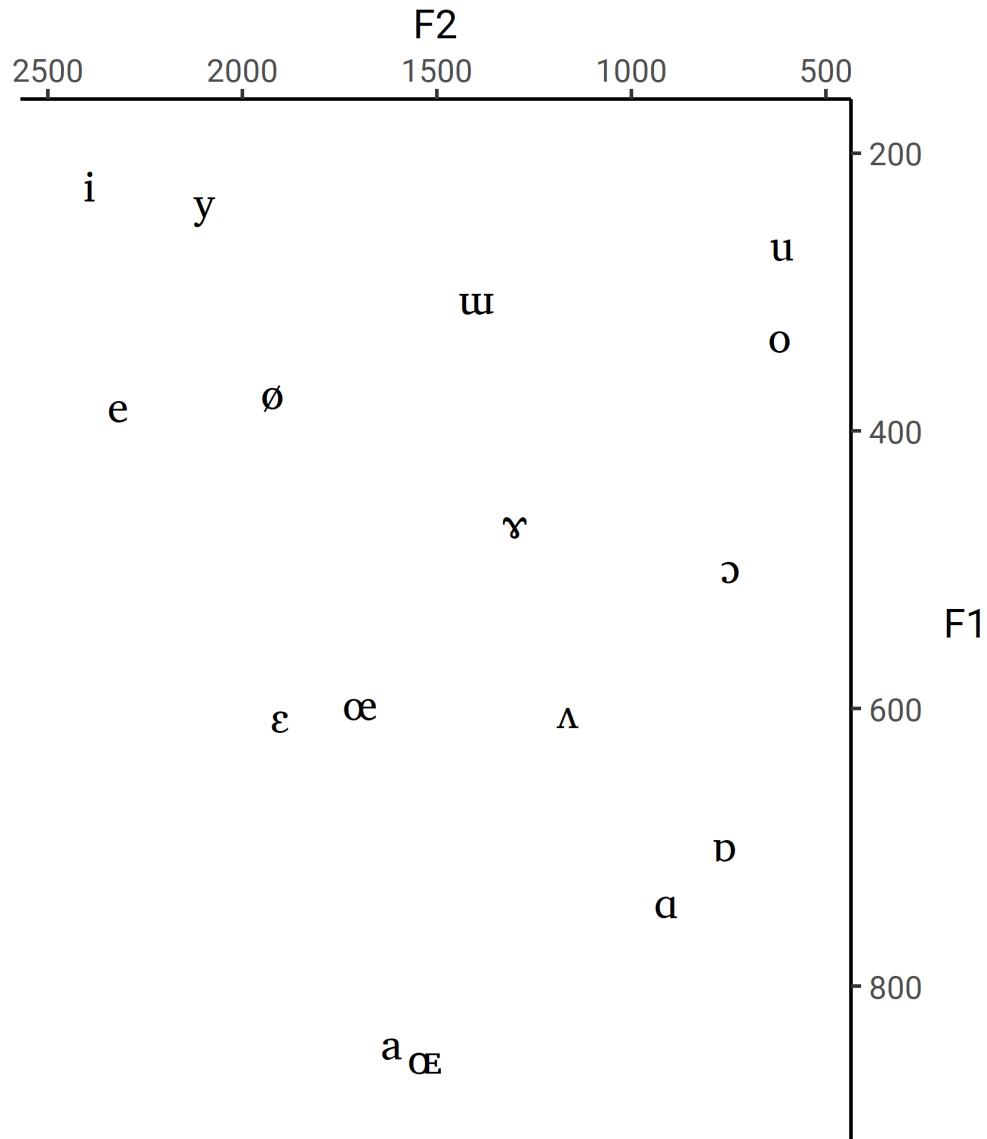
```



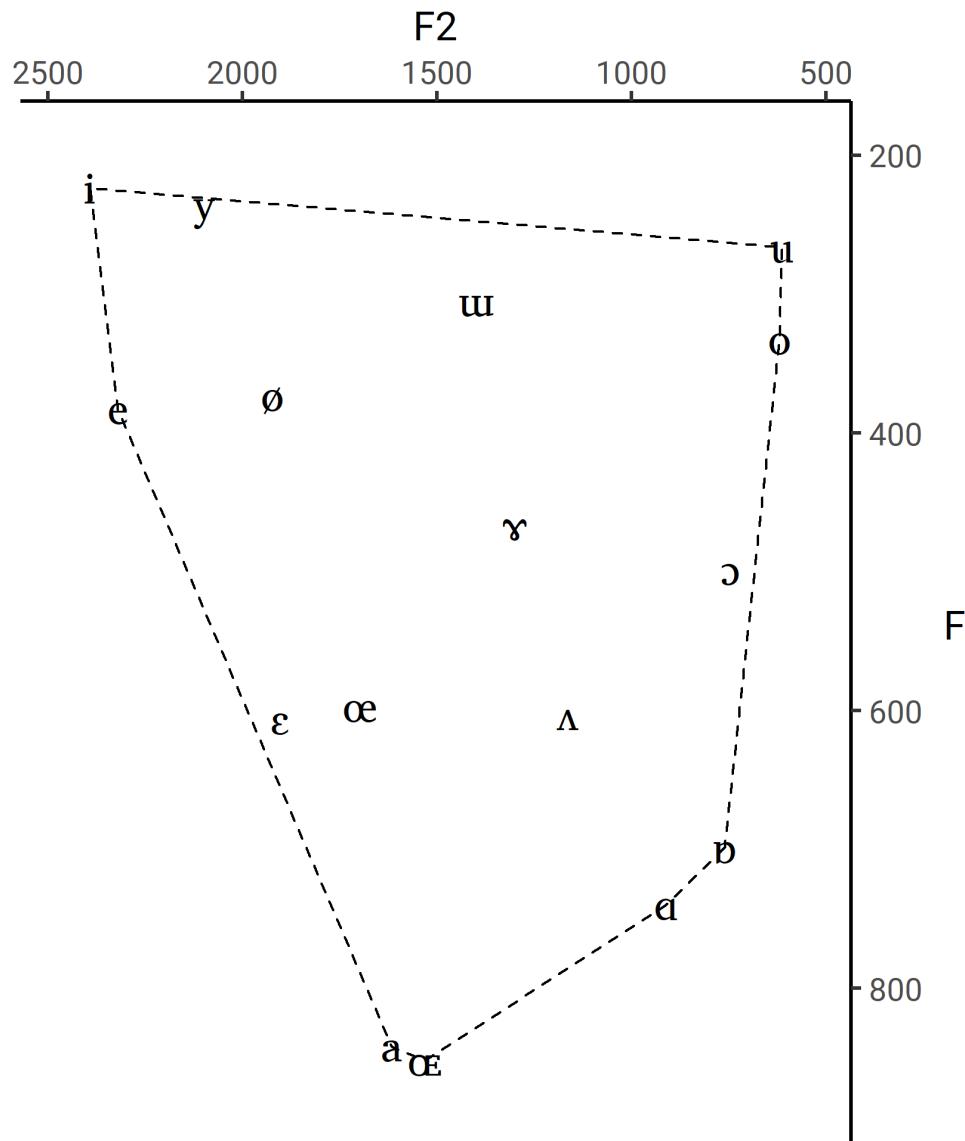
```

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 =
    )
  )

```



```
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
  )
)
```



# Bar Plot of Proportions

```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant
```

```
# A tibble: 4,516 x 10
  Value Item    Cond   Group Subject      Time Answer Accuracy Type
  <chr> <chr>   <chr>  <chr>  <chr>       <dbl> <chr>   <dbl> <chr>
1 Yes   Awaken~ Subje~ B     5cfa77e907f7dd~  917  No     0 Crit
2 No    Awaken~ Subje~ B     5c1d21196036e4~  2624 No     1 Crit
3 No    Awaken~ Subje~ B     5d730aab6b4b16~  2809 No     1 Crit
4 No    Awaken~ Subje~ B     5ddef0fae454cd~  1587 No     1 Crit
5 No    Awaken~ Subje~ B     5e7a8e8b11bf56~  1697 No     1 Crit
6 No    Awaken~ Verb      A     5d4613a2da9cb6~  1484 No     1 Crit
7 No    Awaken~ Verb      A     579e1e2b275be6~  1556 No     1 Crit
8 No    Awaken~ Verb      A     5dfe69ed11d879~  1699 No     1 Crit
9 No    Awaken~ Verb      A     5b9442cecd3808~  2159 No     1 Crit
10 No   Awaken~ Verb     A     5e7f3328849917~  1132 No     1 Crit
# ... with 4,506 more rows
```

```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant  
filter(Type == "Critical")
```

```
# A tibble: 1,466 x 10
  Value Item    Cond   Group Subject          Time Answer Accuracy Type
  <chr> <chr>   <chr> <chr>  <chr>        <dbl> <chr>    <dbl> <chr>
1 Yes   Awaken~ Subje~ B      5cfa77e907f7dd~     917 No       0 Crit
2 No    Awaken~ Subje~ B      5c1d21196036e4~    2624 No       1 Crit
3 No    Awaken~ Subje~ B      5d730aab6b4b16~    2809 No       1 Crit
4 No    Awaken~ Subje~ B      5ddef0fae454cd~    1587 No       1 Crit
5 No    Awaken~ Subje~ B      5e7a8e8b11bf56~    1697 No       1 Crit
6 No    Awaken~ Verb      A      5d4613a2da9cb6~    1484 No       1 Crit
7 No    Awaken~ Verb      A      579e1e2b275be6~    1556 No       1 Crit
8 No    Awaken~ Verb      A      5dfe69ed11d879~    1699 No       1 Crit
9 No    Awaken~ Verb      A      5b9442cecd3808~    2159 No       1 Crit
10 No   Awaken~ Verb     A      5e7f3328849917~   1132 No       1 Crit
# ... with 1,456 more rows
```

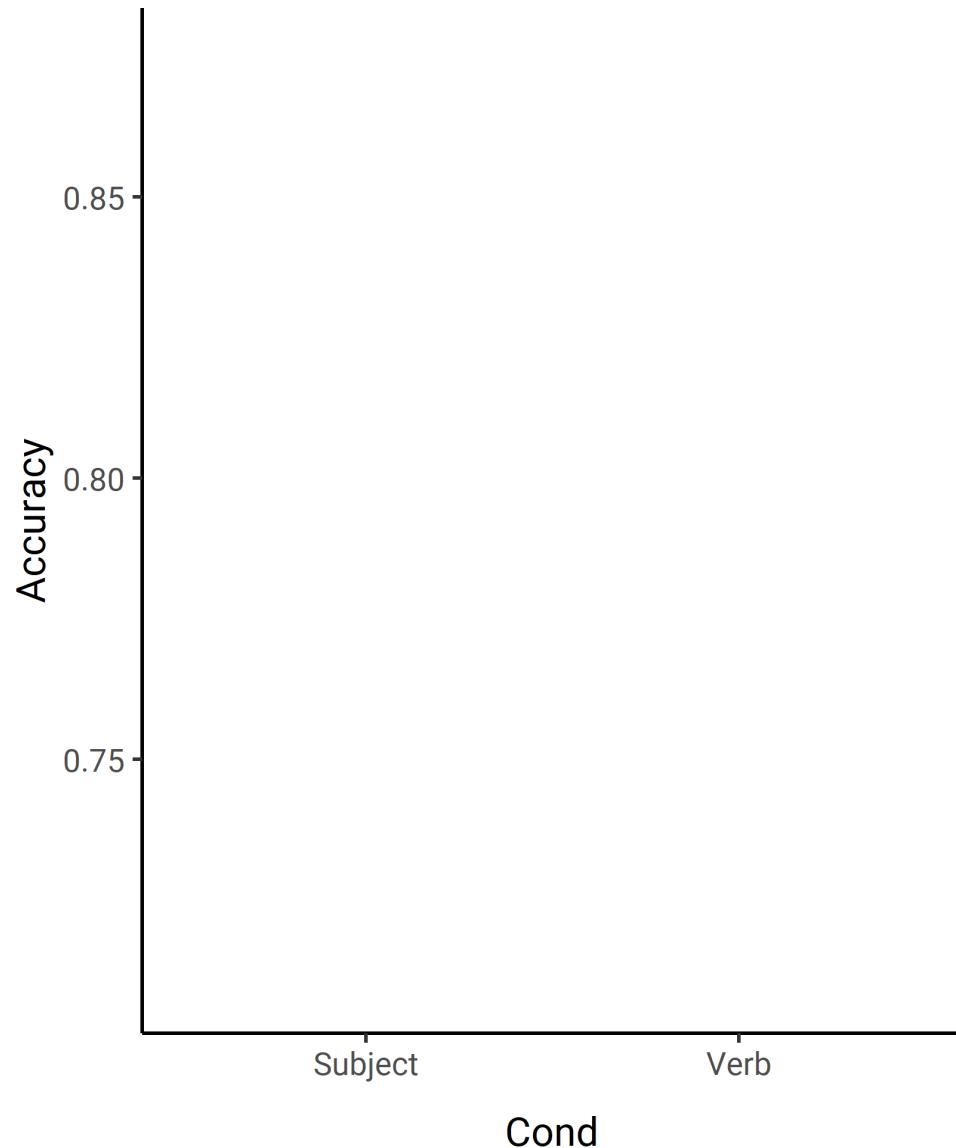
```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant  
filter(Type == "Critical") %>%  
group_by(Cond, Group)
```

```
# A tibble: 1,466 x 10  
# Groups:   Cond, Group [4]  
  Value Item    Cond  Group Subject          Time Answer Accuracy Type  
  <chr> <chr>   <chr> <chr> <chr>           <dbl> <chr>    <dbl> <chr>  
1 Yes   Awaken~ Subje~ B      5cfa77e907f7dd~  917 No     0 Crit  
2 No    Awaken~ Subje~ B      5c1d21196036e4~  2624 No     1 Crit  
3 No    Awaken~ Subje~ B      5d730aab6b4b16~  2809 No     1 Crit  
4 No    Awaken~ Subje~ B      5ddef0fae454cd~  1587 No     1 Crit  
5 No    Awaken~ Subje~ B      5e7a8e8b11bf56~  1697 No     1 Crit  
6 No    Awaken~ Verb       A      5d4613a2da9cb6~  1484 No     1 Crit  
7 No    Awaken~ Verb       A      579e1e2b275be6~  1556 No     1 Crit  
8 No    Awaken~ Verb       A      5dfe69ed11d879~  1699 No     1 Crit  
9 No    Awaken~ Verb       A      5b9442cecd3808~  2159 No     1 Crit  
10 No   Awaken~ Verb      A      5e7f3328849917~  1132 No     1 Crit  
# ... with 1,456 more rows
```

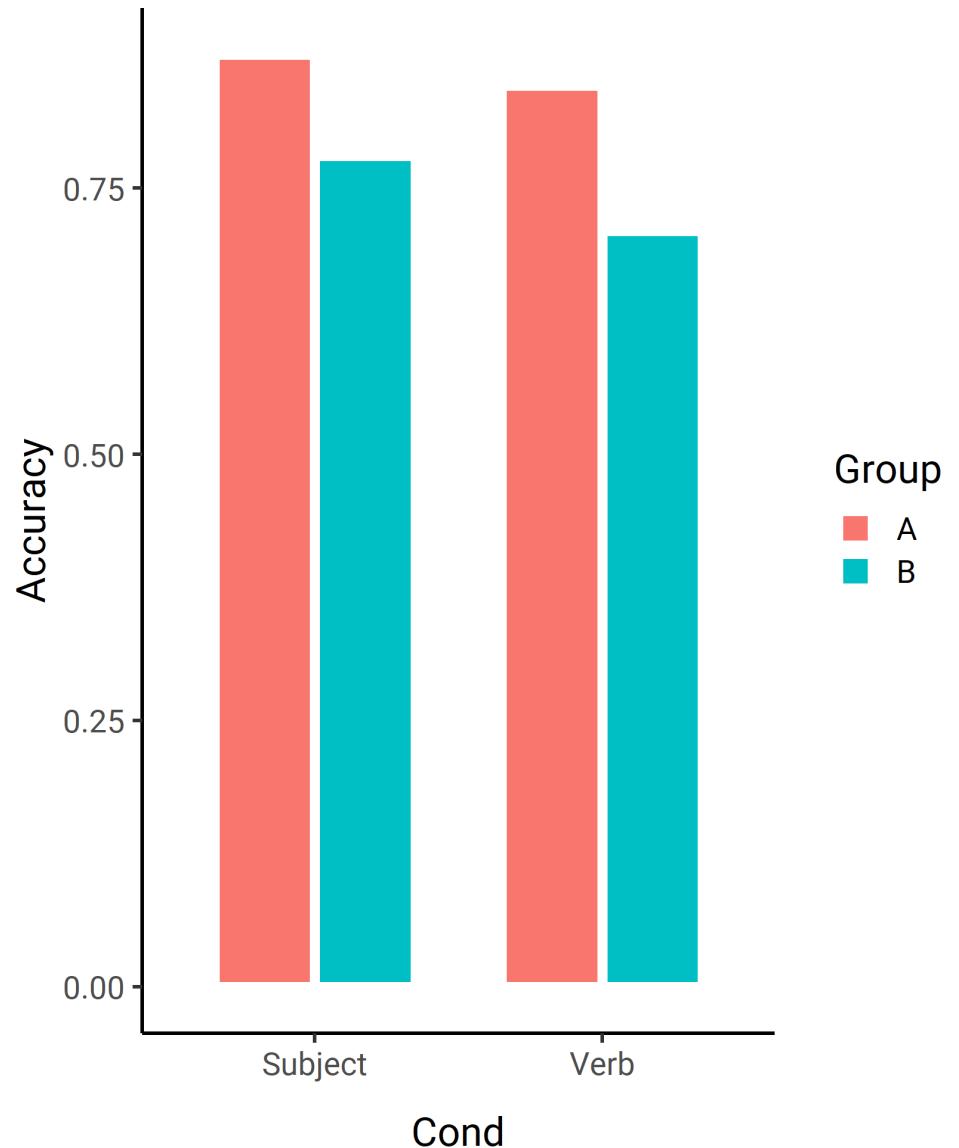
```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant  
filter(Type == "Critical") %>%  
group_by(Cond, Group) %>%  
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups = "none")
```

```
# A tibble: 4 x 3  
  Cond     Group Accuracy  
  <chr>    <chr>    <dbl>  
1 Subject  A        0.875  
2 Subject  B        0.780  
3 Verb     A        0.846  
4 Verb     B        0.710
```

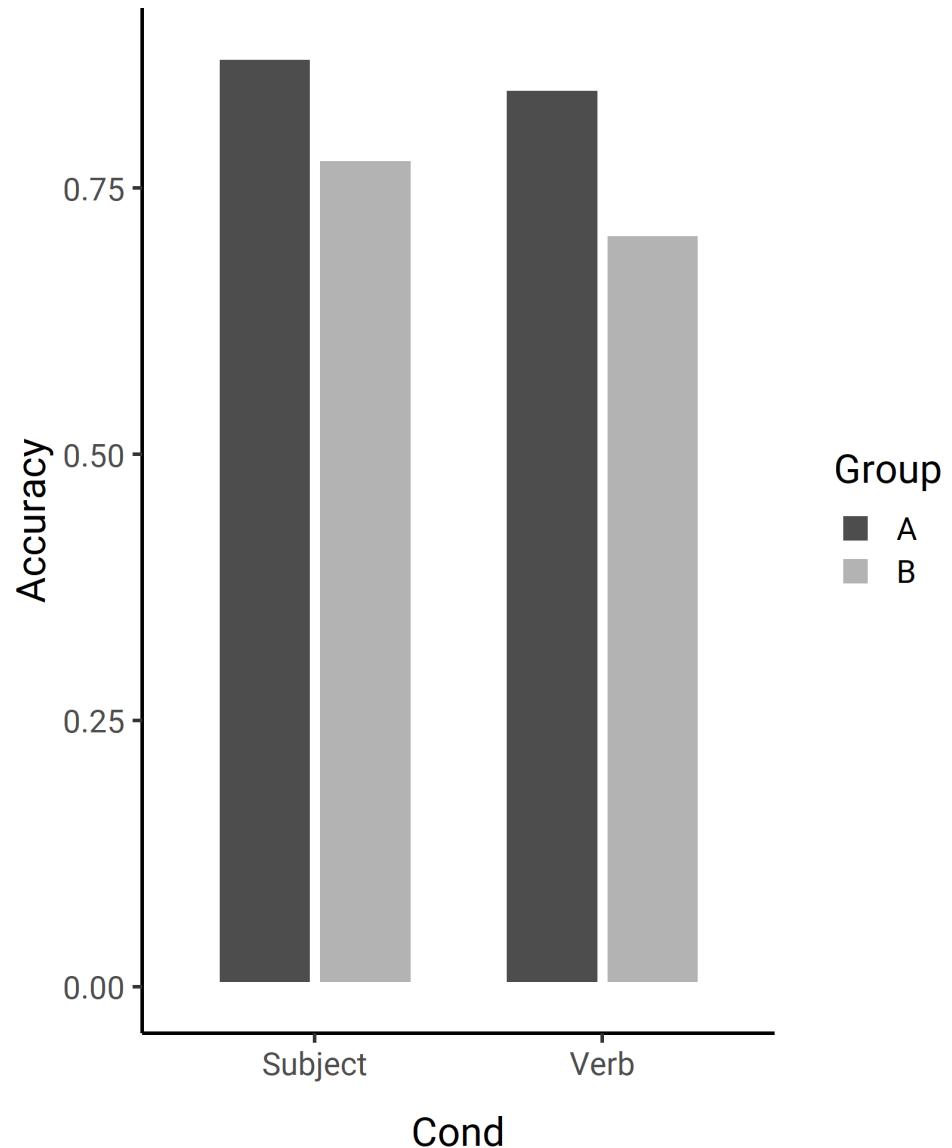
```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant  
filter(Type == "Critical") %>%  
group_by(Cond, Group) %>%  
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups = "none")  
ggplot(aes(x = Cond, y = Accuracy, fill = Group))
```



```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant  
filter(Type == "Critical") %>%  
group_by(Cond, Group) %>%  
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups = "none")  
ggplot(aes(x = Cond, y = Accuracy, fill = Group)) +  
geom_col(position = "dodge", color = "white", width = .7,
```

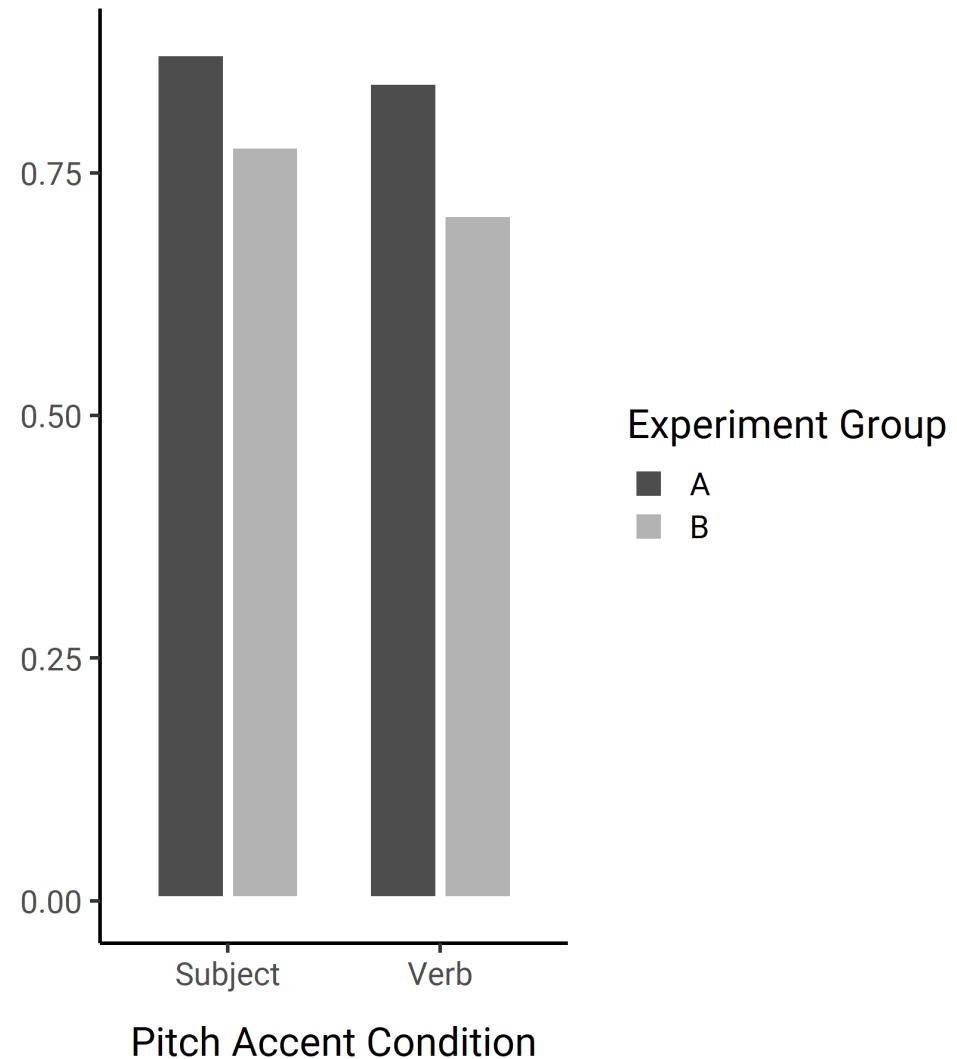


```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant...  
filter(Type == "Critical") %>%  
group_by(Cond, Group) %>%  
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups = "first")  
ggplot(aes(x = Cond, y = Accuracy, fill = Group)) +  
geom_col(position = "dodge", color = "white", width = .7,  
scale_fill_manual(values = c("grey30", "grey70"))
```



```
read_csv("https://raw.githubusercontent.com/yjunechoe/Semant  
filter(Type == "Critical") %>%  
group_by(Cond, Group) %>%  
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups = "first")  
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 = "Accuracy on Comprehension Task",  
x = "Pitch Accent Condition", y = NULL,  
fill = "Experiment Group"  
)
```

## Accuracy on Comprehension Task

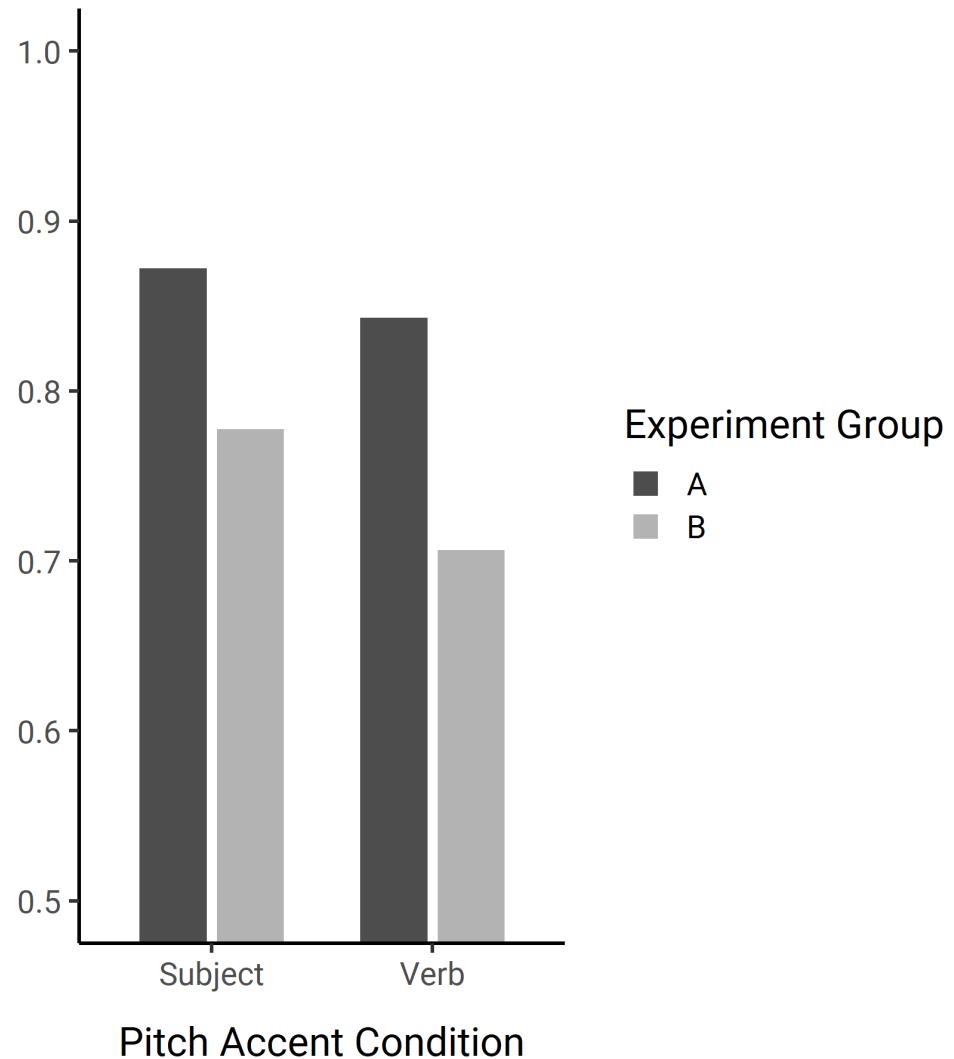


```

read_csv("https://raw.githubusercontent.com/yjunechoe/Semant
filter(Type == "Critical") %>%
group_by(Cond, Group) %>%
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups =
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 = "Accuracy on Comprehension Task",
  x = "Pitch Accent Condition", y = NULL,
  fill = "Experiment Group"
) +
coord_cartesian(ylim = c(0.5, 1))

```

## Accuracy on Comprehension Task

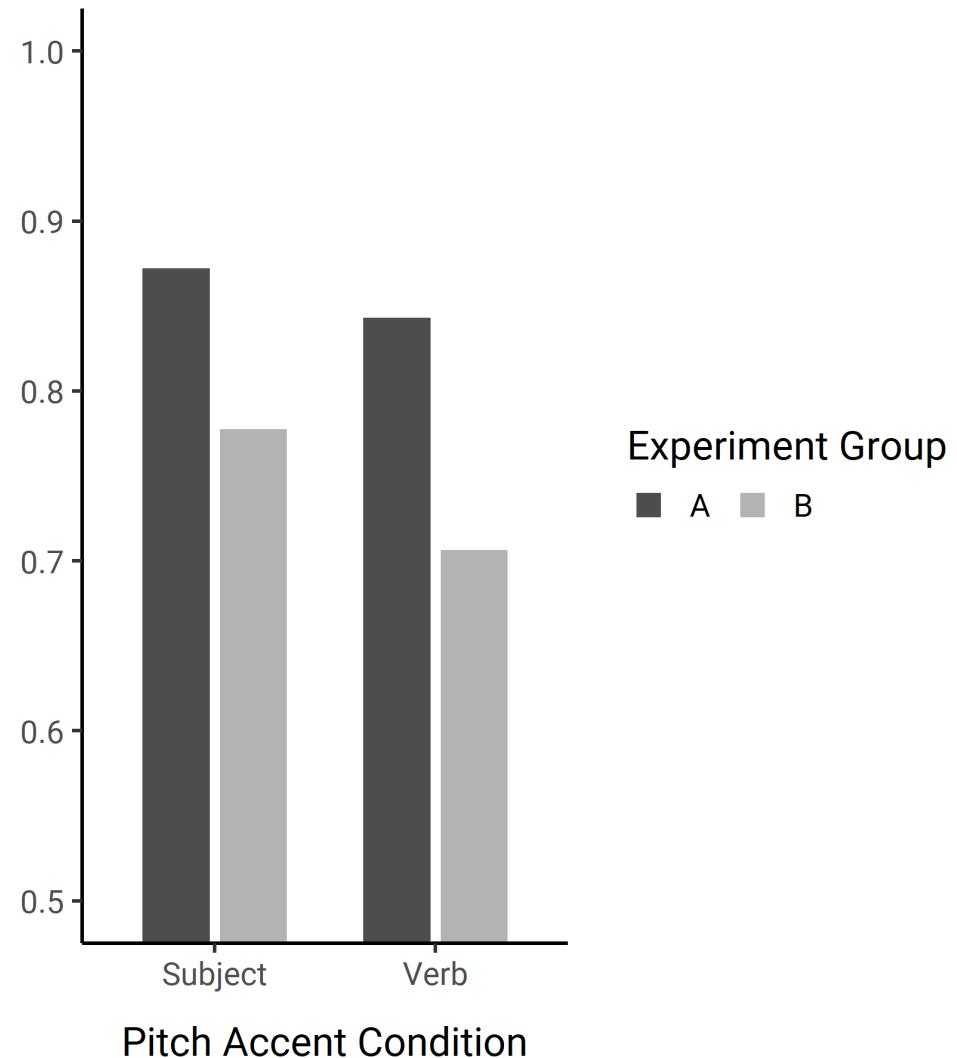


```

read_csv("https://raw.githubusercontent.com/yjunechoe/Semant
filter(Type == "Critical") %>%
group_by(Cond, Group) %>%
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups =
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 = "Accuracy on 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 =

```

## Accuracy on Comprehension Task

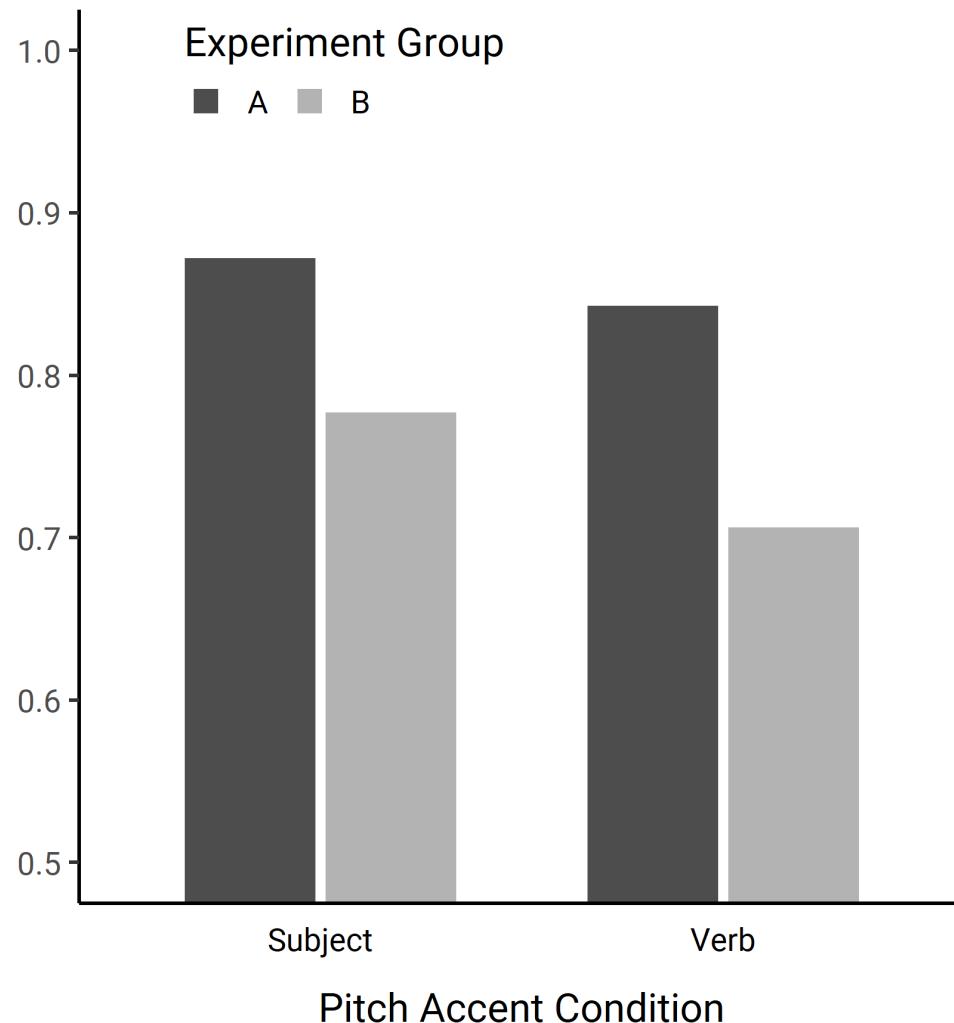


```

read_csv("https://raw.githubusercontent.com/yjunechoe/Semant
filter(Type == "Critical") %>%
group_by(Cond, Group) %>%
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups =
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 = "Accuracy on 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 =
theme(
  axis.ticks.x = element_blank(),
  axis.text.x = element_text(color = "black", margin = margin(
    legend.position = c(.3, .93),
    plot.title = element_text(
      margin = margin(b = 1, unit = "cm")
    )
)

```

# Accuracy on Comprehension Task

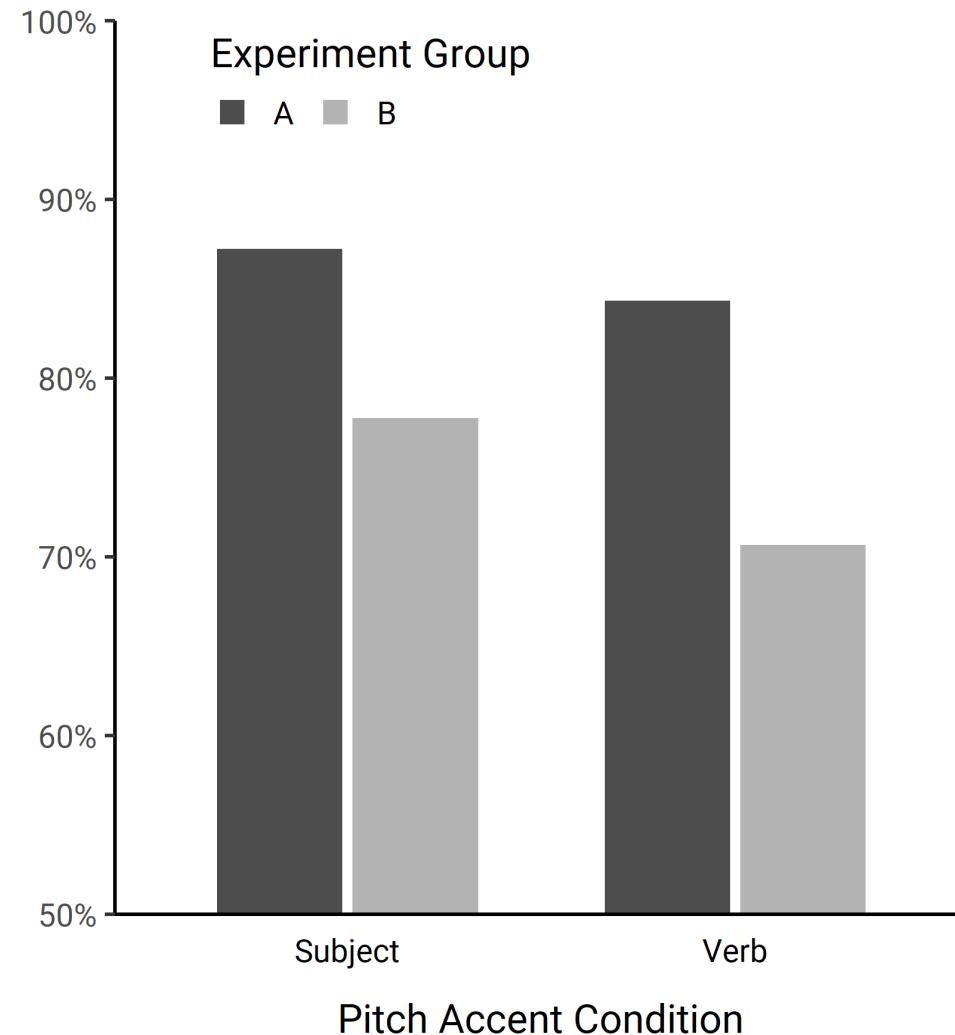


```

read_csv("https://raw.githubusercontent.com/yjunechoe/Semant
filter(Type == "Critical") %>%
group_by(Cond, Group) %>%
summarize(Accuracy = mean(Accuracy, na.rm = TRUE), .groups =
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 = "Accuracy on 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 =
theme(
  axis.ticks.x = element_blank(),
  axis.text.x = element_text(color = "black", margin = margin(
    legend.position = c(.3, .93),
    plot.title = element_text(
      margin = margin(b = 1, unit = "cm")
    )
  )
) +
scale_y_continuous(
  expand = expansion(0, 0),
  labels = percent_format(accuracy = 1)
)

```

# Accuracy on Comprehension Task



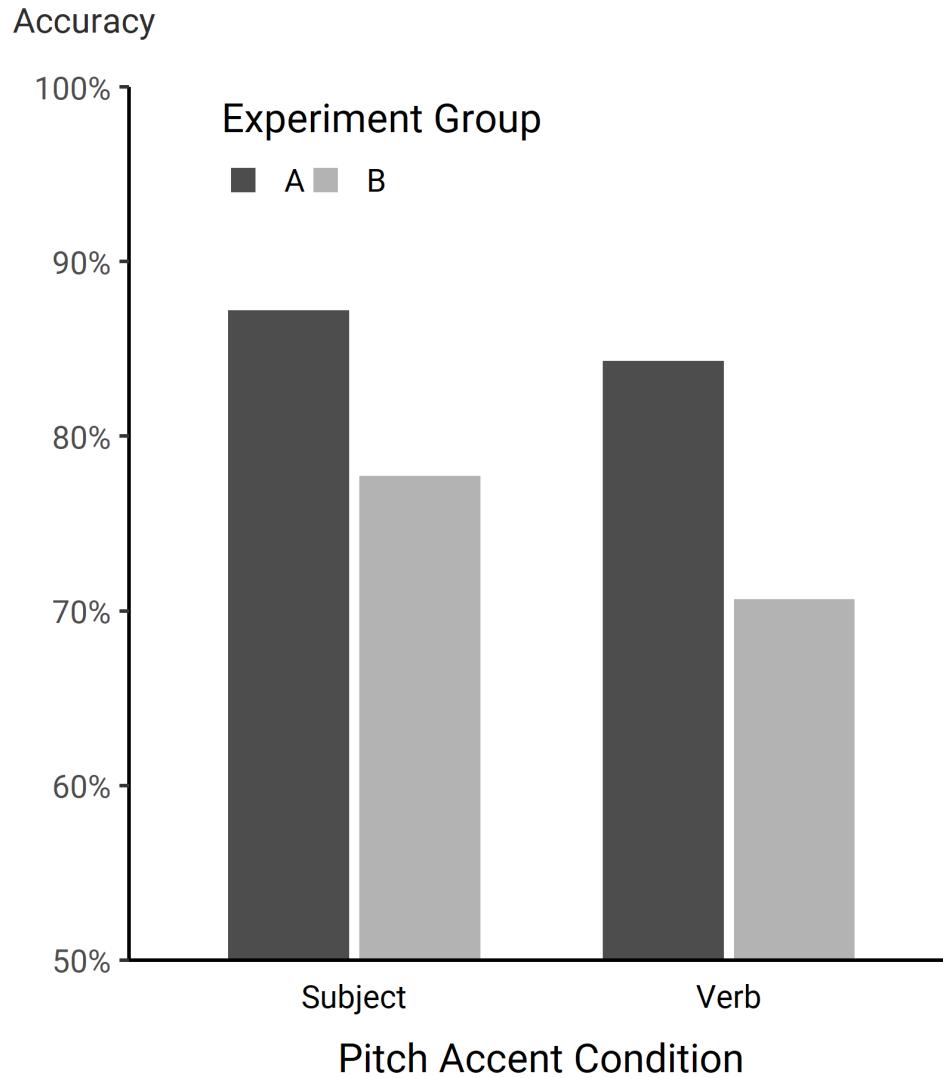
```

p <- read_csv("https://raw.githubusercontent.com/yjunechoe/9-10-2023/main/critical_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)
)

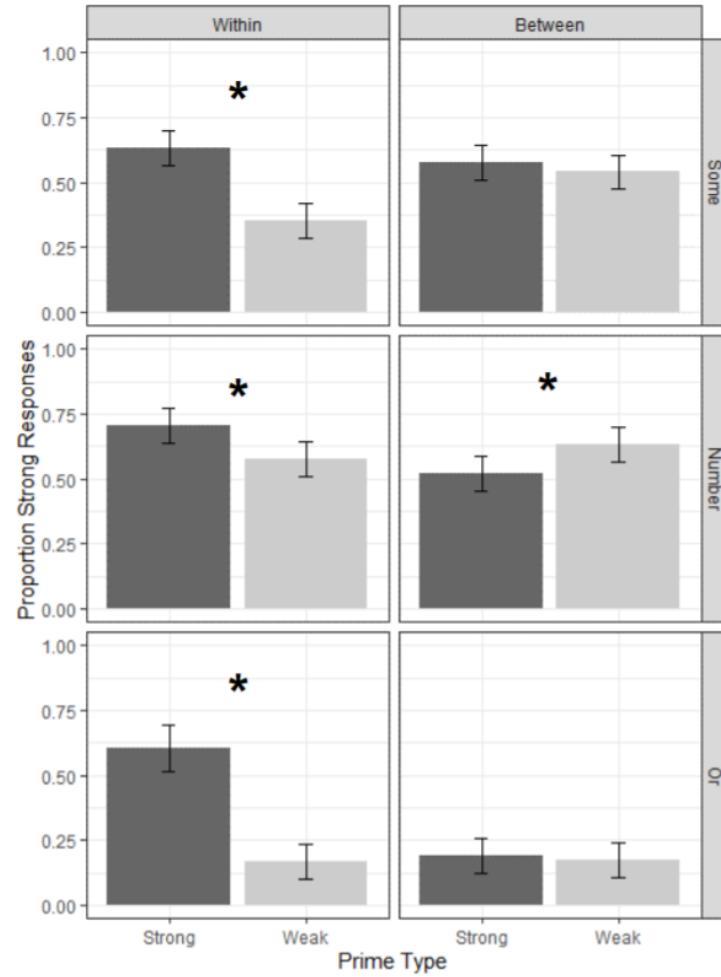
library(cowplot)
ggdraw(p) +
  draw_label("Accuracy", x = .125, y = .88, fontfamily = "Rockwell")

```

# Comprehension Task



# Multiple Categorical Levels



Source: Husband & Patson (2020)

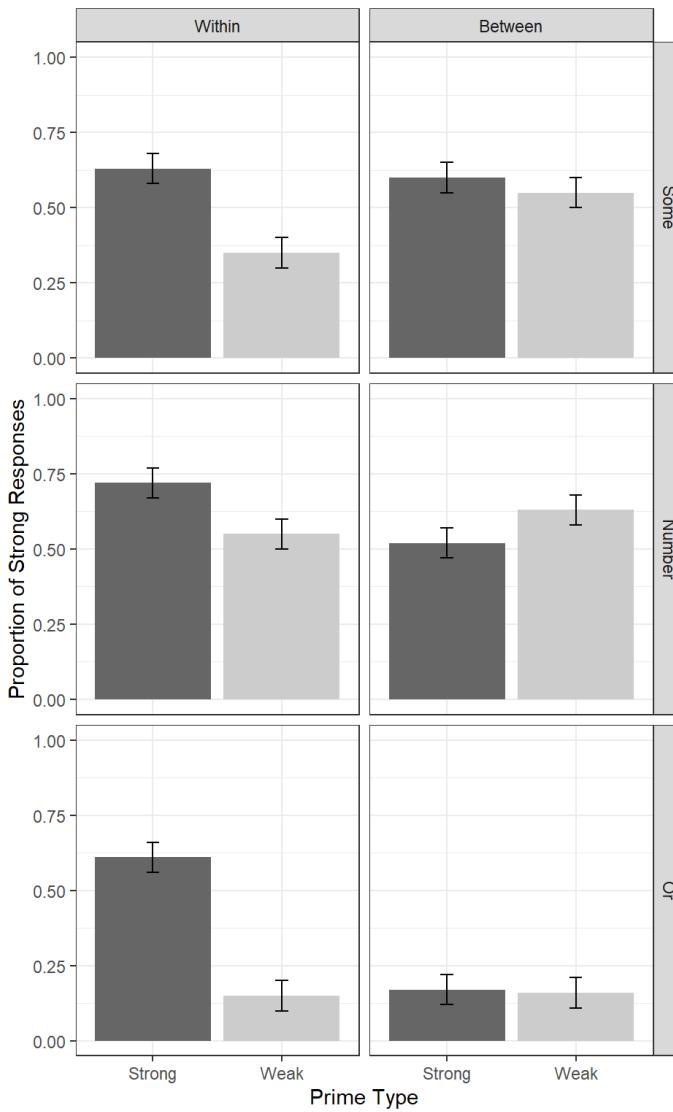
```

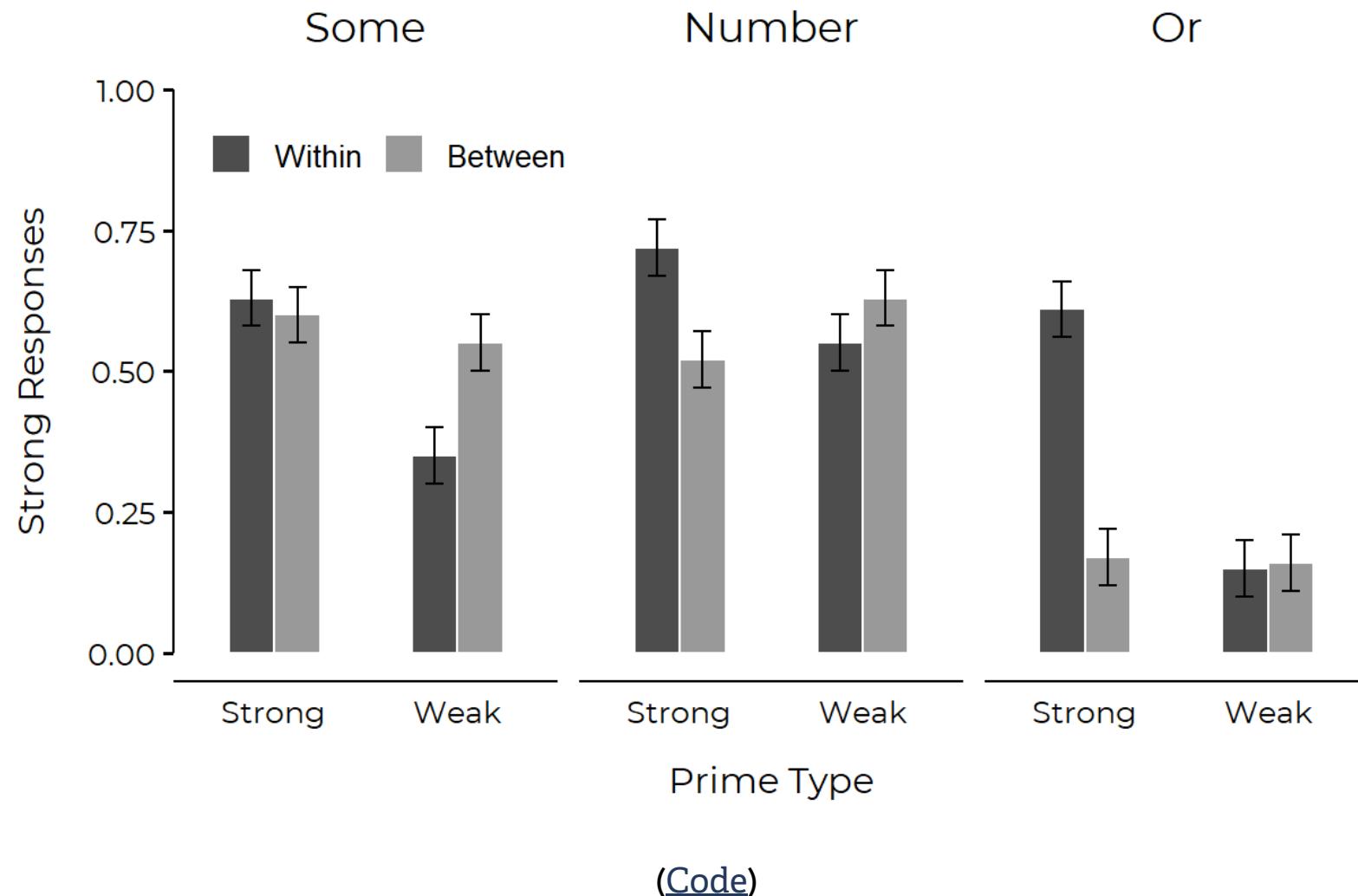
df <- crossing(level_1 = fct_inorder(c("Within", "Between"),
                                      level_2 = fct_inorder(c("Some", "Number")),
                                      level_3 = factor(c("Strong", "Weak"))))

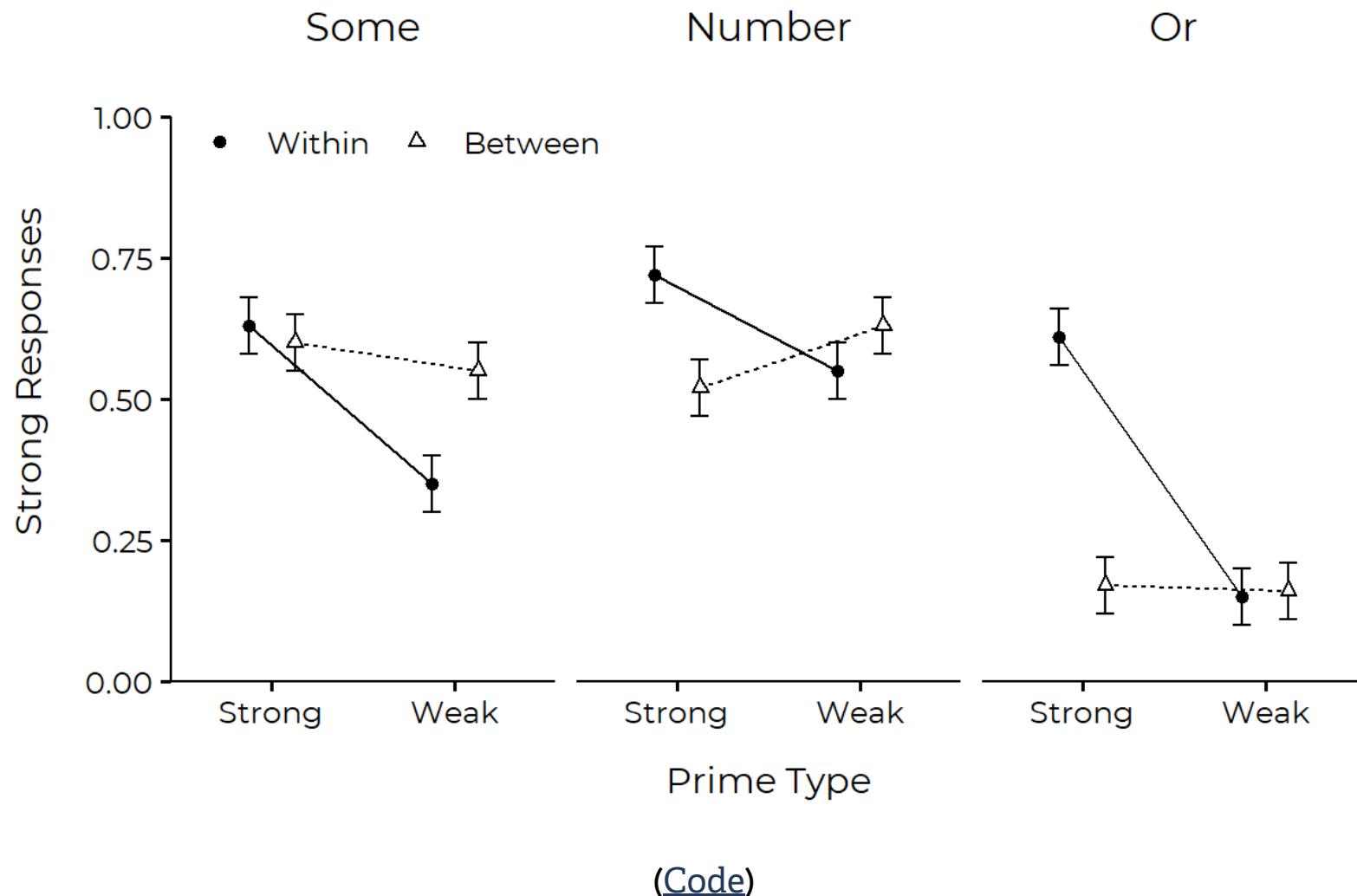
df$barheight <- c(.63, .35, .72, .55, .61, .15, .60, .52)

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()

```







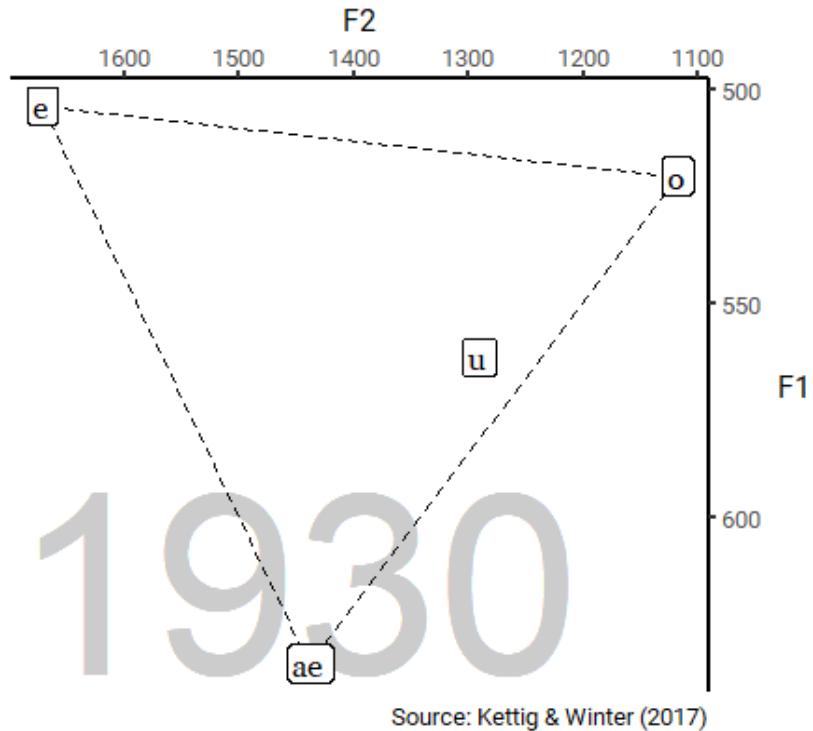
# Animations: Vowel shift

```

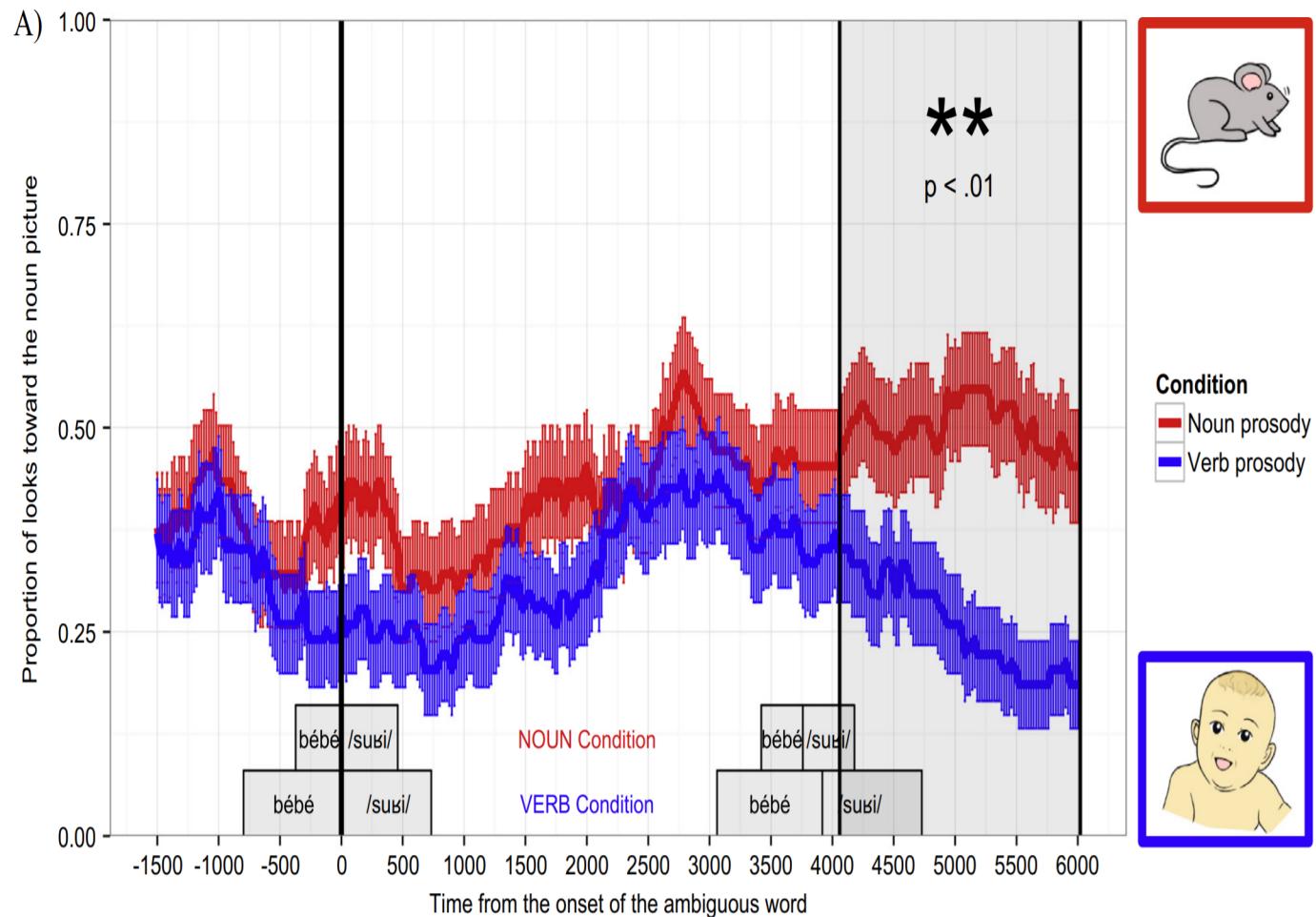
read_csv("https://raw.githubusercontent.com/bodowinter/
group_by(Gender, Vowel, BirthDecade = 10 * BirthYear
summarize(across(F1:F2, mean), .groups = 'drop') %>%
filter(Gender == "F") %>%
ggplot(aes(F2, F1)) +
geom_text(aes(1450, 615, label = as.character(BirthDe
          color = "gray80", size = 48) +
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

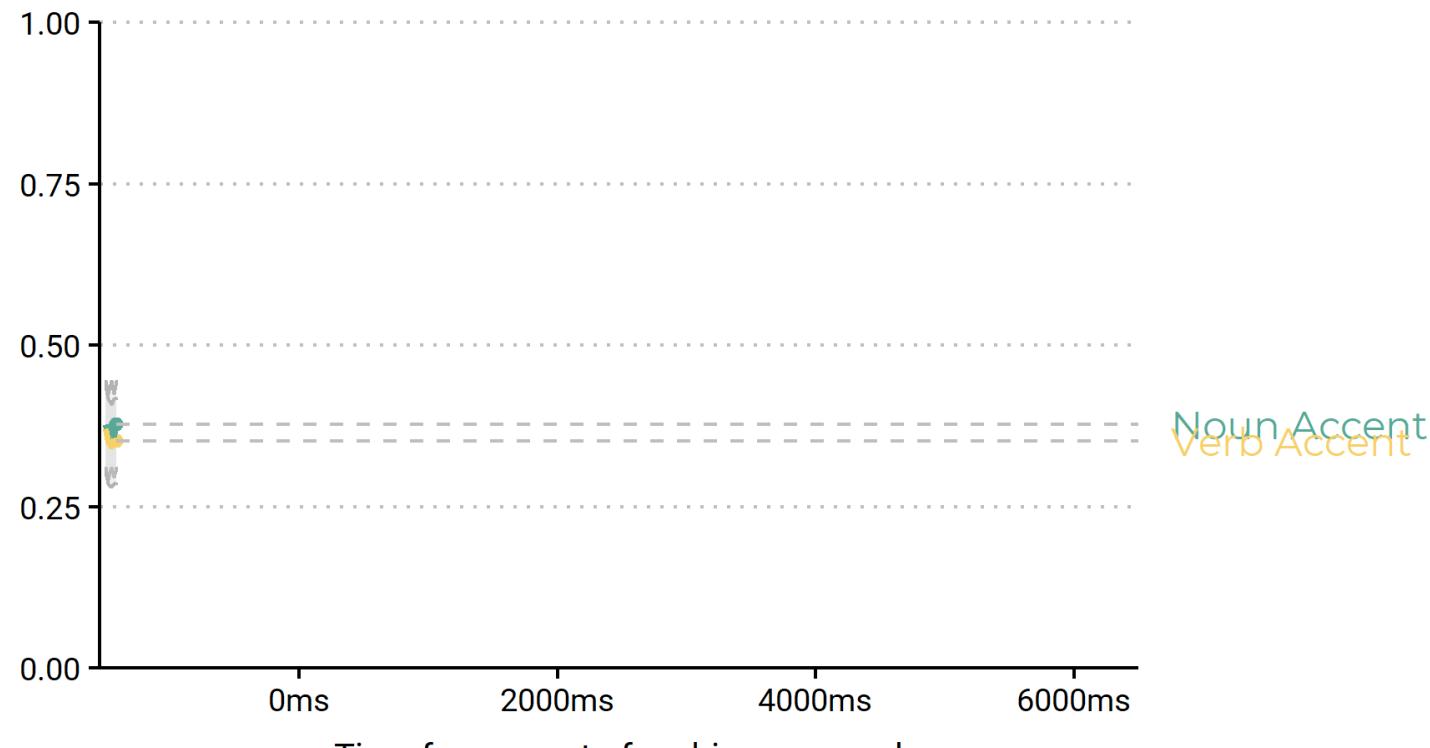


# Animations: Eye-tracking



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](#)