

Data viz w/ggplot2

Part 2 - Explanatory plots

PA 434 - Federica Fusi

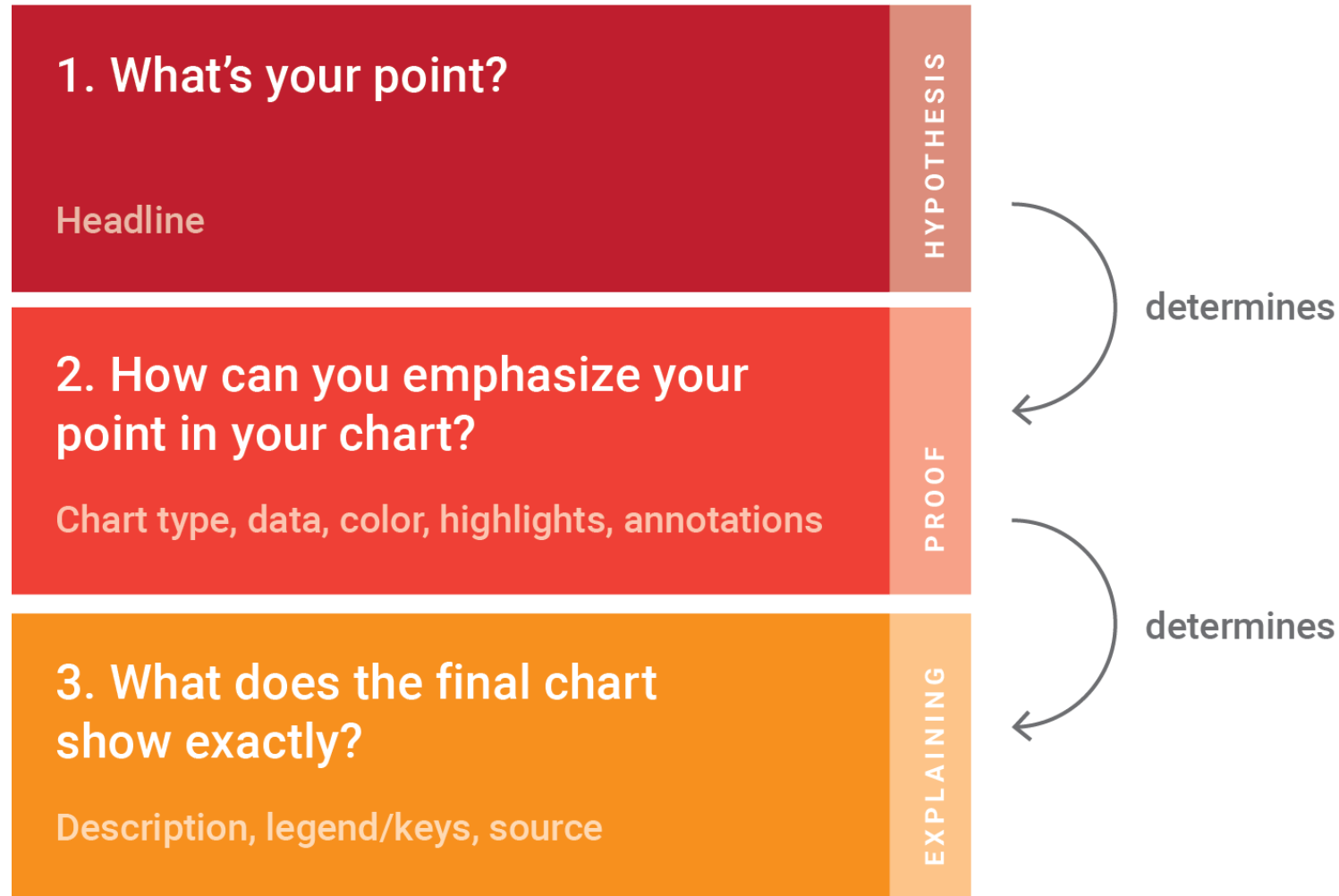
Explanatory data viz

Explanatory data visualization

A change in your audience: From visualization for yourself and your exploration of data to visualization for others

1. Data that are displayed appropriately considering the type of data you have.
2. Axes that are not misleading (e.g., start at zero)
3. Clearly-labeled axes.
4. Title and description need to be clear
5. Be mindful of colors [NEW]
6. Text and other graphic elements are clearly visible [NEW]

Three questions for creating a chart



3 questions to ask yourself

Making your graph prettier

Basic arguments

- **color** applies to points, lines, text, borders
- **fill** applies to any area that can be filled
- **size** specifies the size of your elements
- **shape** applies different shapes to points
- **line** specifies the type of line to be used
- **alpha** changes transparency of your color

By playing around with these arguments you can easily change the look of your graph.

color and fill

There are at least 4 ways to change color to your graph:

- Colors can be indicated by their **R name**. I use this method for exploratory graph as I quickly remember a few basic color names by heart (e.g., "white", "red", "darkgreen", "darkblue",)
- **Hex codes** look like this: "#69b3a2". I prefer this system to customize my graphs using some of the websites above
- **RBG specification** - how much Red, Blue, and Green is contained in each color. This method is also good for customization but you need to report three values so I found it a bit more time consuming than the hex code
- **palettes** are pre-defined sets of colors that are available in different R packages, among which paletteer and brewer. This is a pretty good and time-saving when you have several groups to show (i.e., need to use several colors) or you want to quick fix some colors.

Choose your colors

There are TONS of good resources to choose colors for your graphs - there is no excuse to ugly colors! Some resources I like:

- [Color in R by name](#)
- [Color palette generator](#): upload a picture and get the color combination from that picture
- There are several websites [like this one](#) and [this one](#) where you can find cool color combinations for your graph.

R has also several packages for colors - most of which offer pre-made palette of colors that you can use. **paletteer** collects palettes from other packages into one unique place.

- the package can be installed from [here](#)
- and here you can find the [full list of colors](#)

An example



A color palette from the inauguration

Use of colors

Uses of color in data visualization

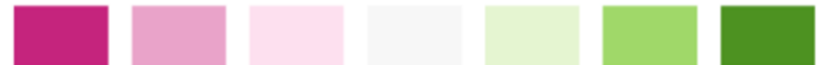
1. Distinguish categories (qualitative)



2. Represent numeric values (sequential)



3. Represent numeric values (diverging)

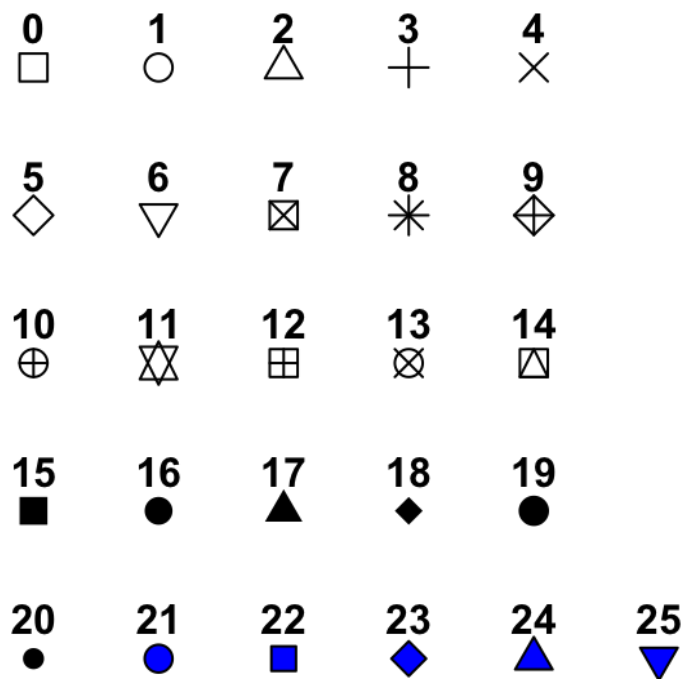


4. Highlight



Shapes

You can visualize the shapes here: `ggpubr::show_point_shapes()`



Shapes in R

Lines

You can visualize the line types here: `ggpubr::show_line_types()`

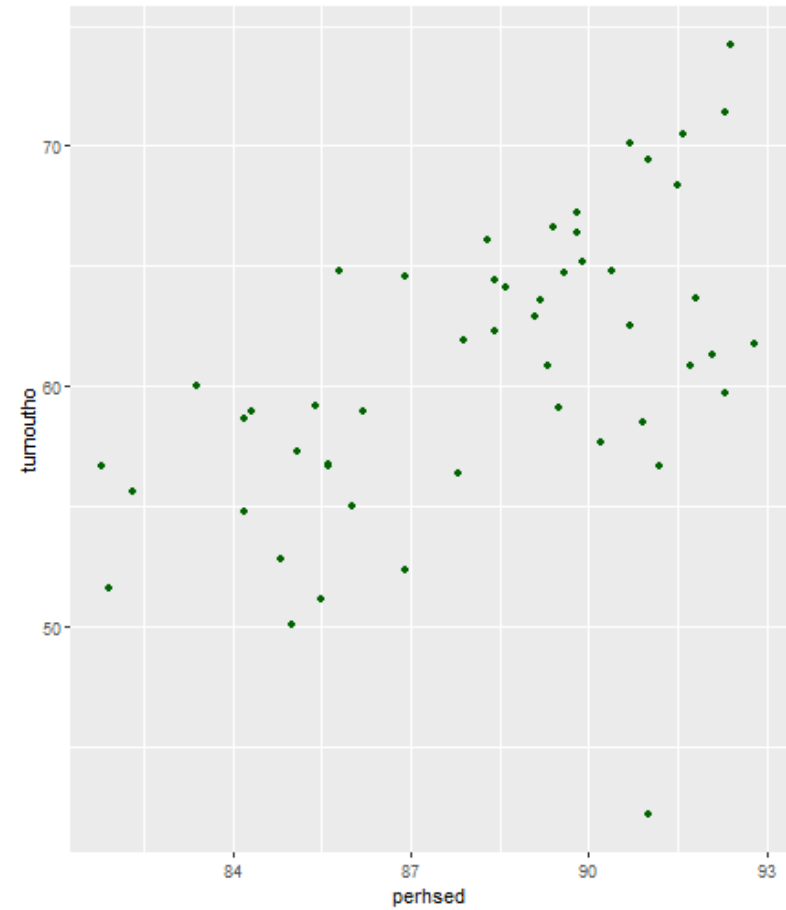


Some examples

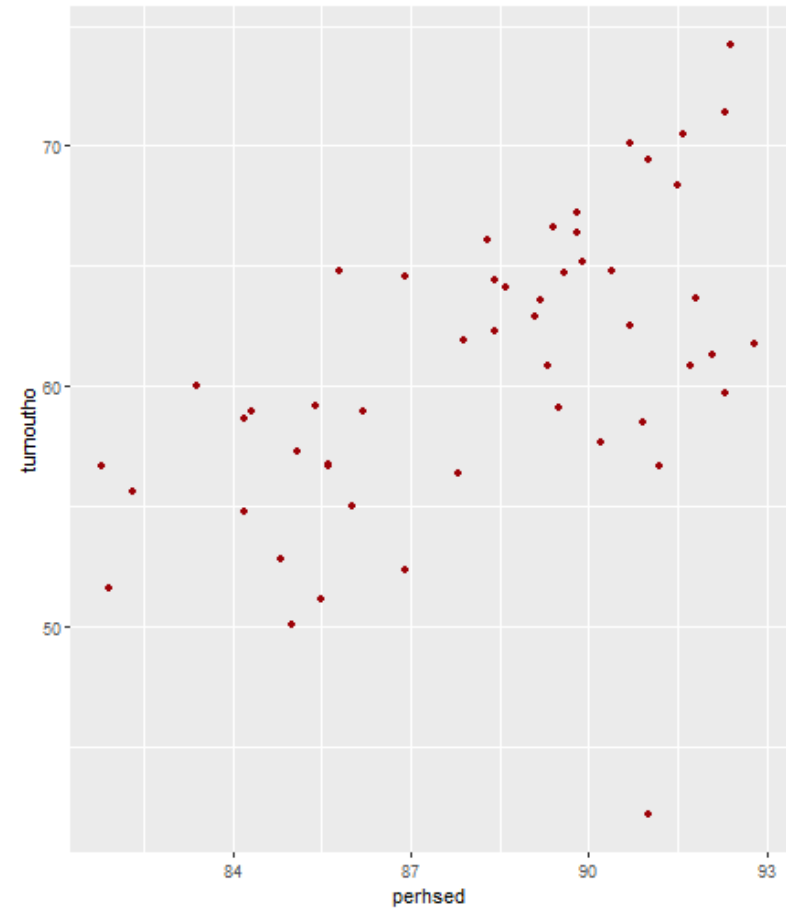
Let's start by looking at how we can specify colors in R.

```
#install.packages(stevedata)  
library("stevedata")
```

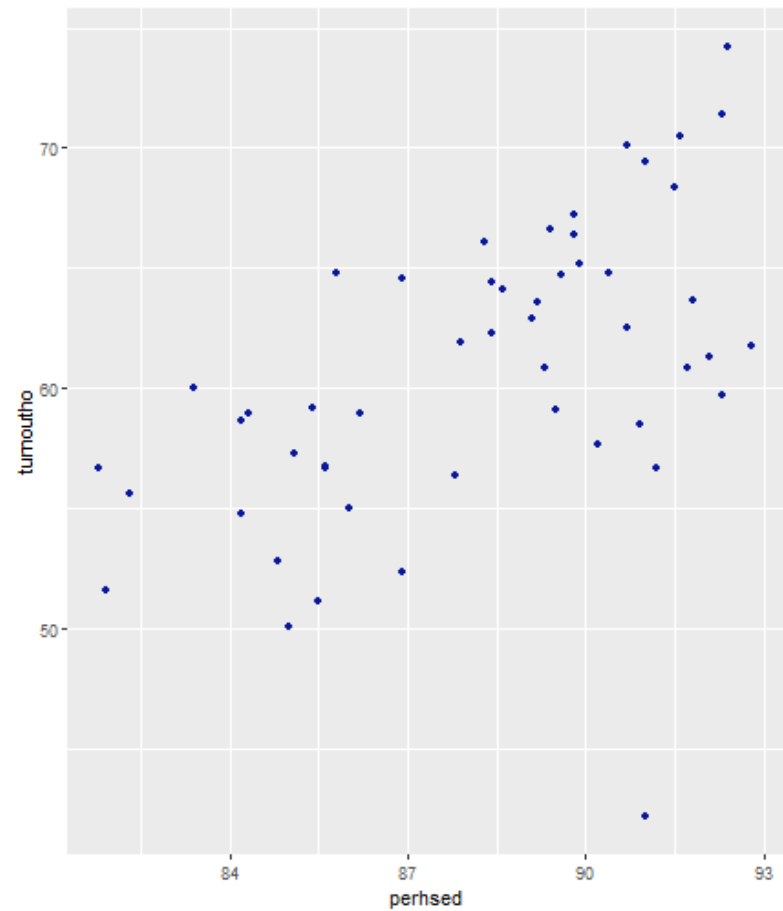
```
ggplot(data = election_turnout) +  
  geom_point(mapping = aes(x = perhsed, y = turnoutho),  
              color = "darkgreen"  
            )
```



```
ggplot(data = election_turnout) +  
  geom_point(mapping = aes(x = perhsed, y = turnoutho),  
              color = "#9d0208" #ROTATE  
            )
```

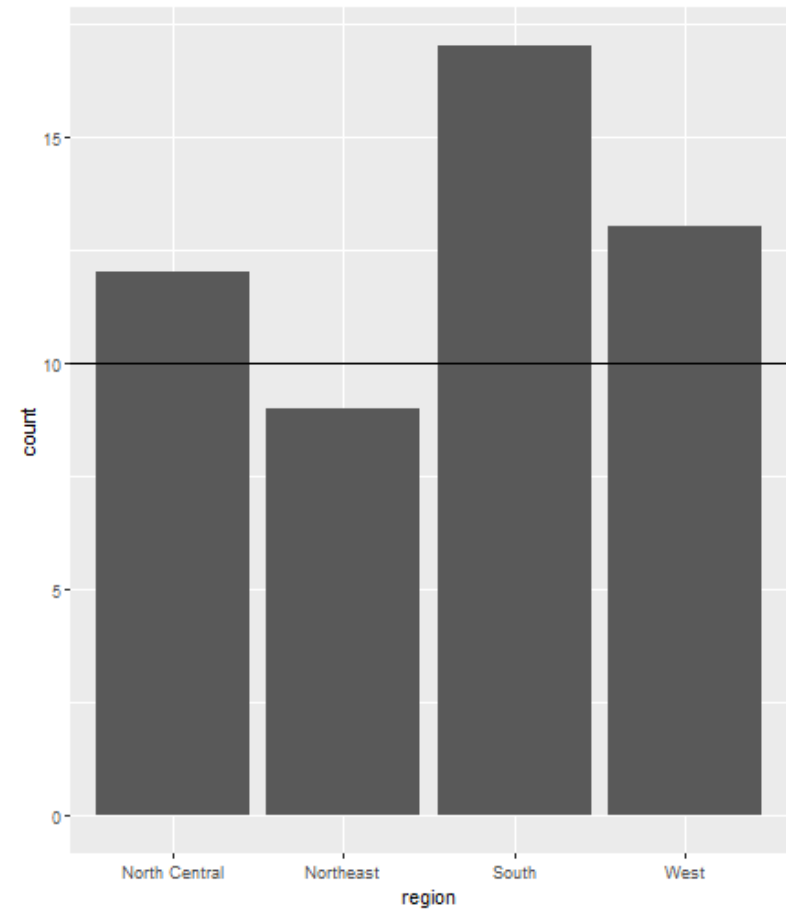



```
ggplot(data = election_turnout) +
  geom_point(mapping = aes(x = perhsed, y = turnoutho),
    color = rgb(9, 24, 154, maxColorValue = 255) #ROTATI
  )
```

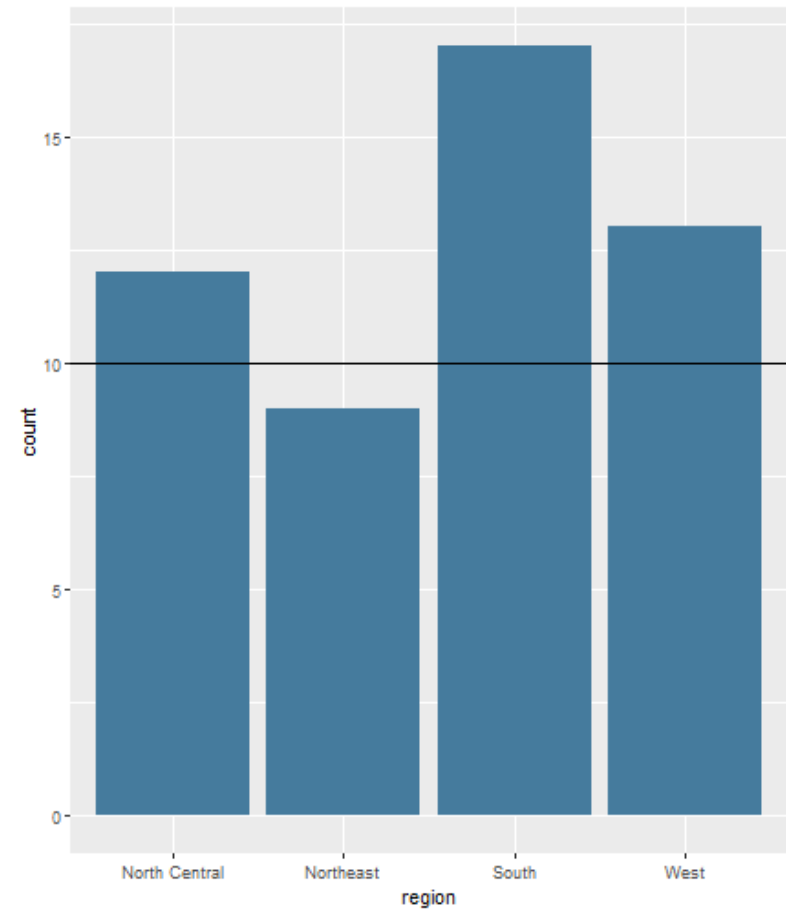


Barplot

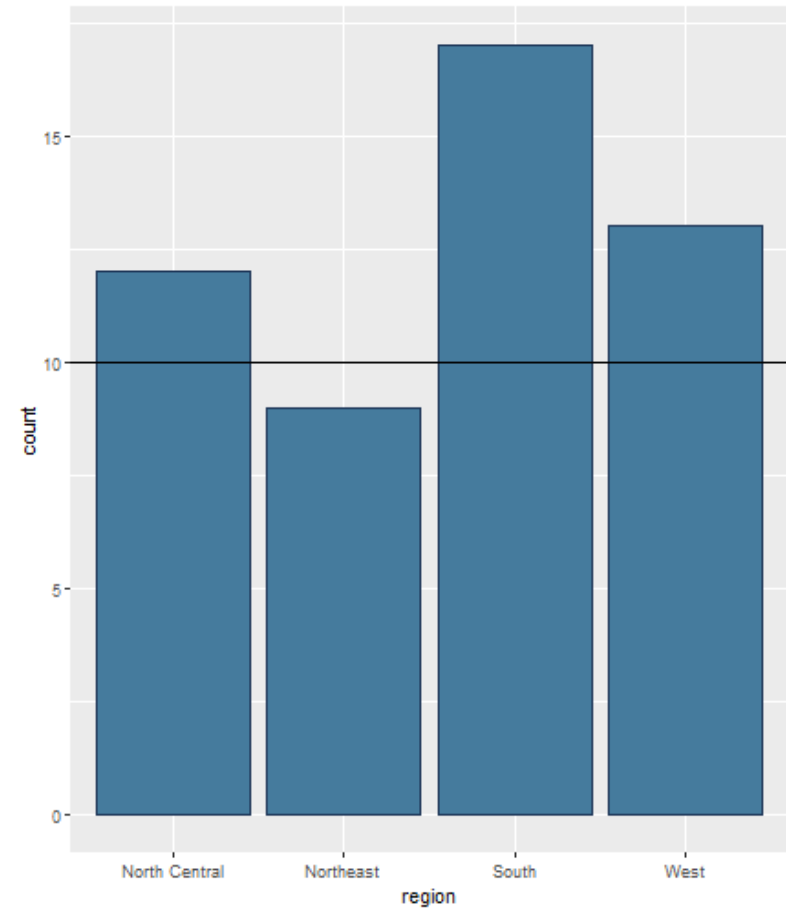
```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region),  
    ) +  
  geom_hline(yintercept = 10,  
    )
```



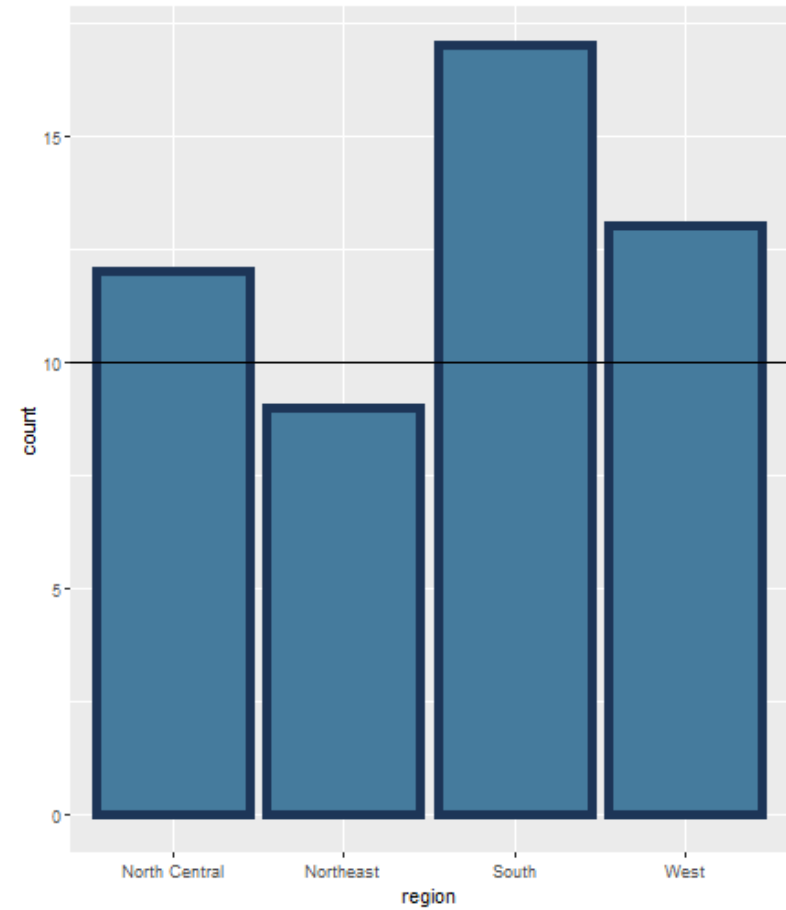
```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region),  
    fill = "#457b9d",  
  ) +  
  geom_hline(yintercept = 10,  
  )
```



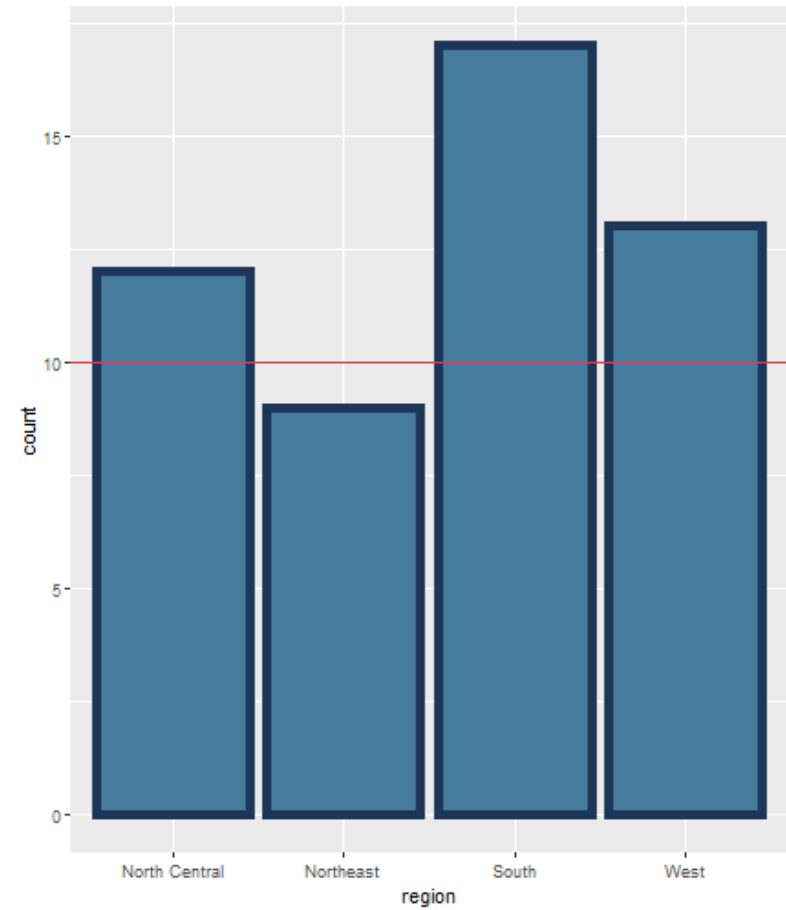
```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region),  
    fill = "#457b9d",  
    color = "#1d3557",  
  ) +  
  geom_hline(yintercept = 10,  
    )
```



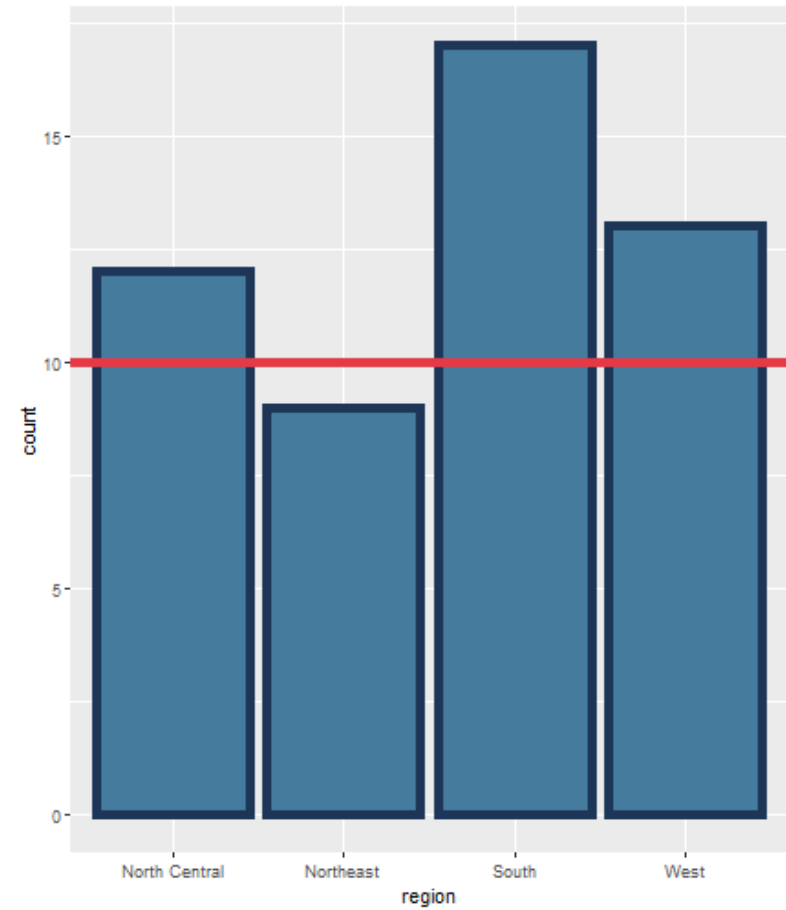
```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region),  
            fill = "#457b9d",  
            color = "#1d3557",  
            size = 2  
          ) +  
  geom_hline(yintercept = 10,  
            )
```



```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region),  
    fill = "#457b9d",  
    color = "#1d3557",  
    size = 2  
  ) +  
  geom_hline(yintercept = 10,  
    col = "#e63946",  
  )
```

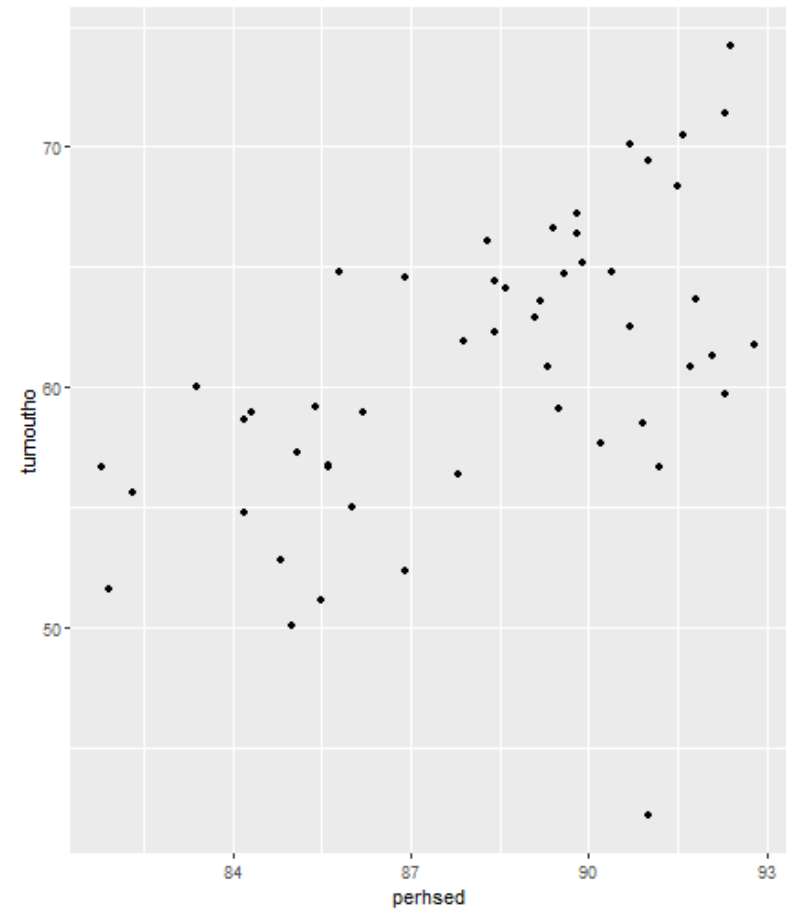


```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region),  
    fill = "#457b9d",  
    color = "#1d3557",  
    size = 2  
  ) +  
  geom_hline(yintercept = 10,  
    col = "#e63946",  
    size = 2  
  )
```

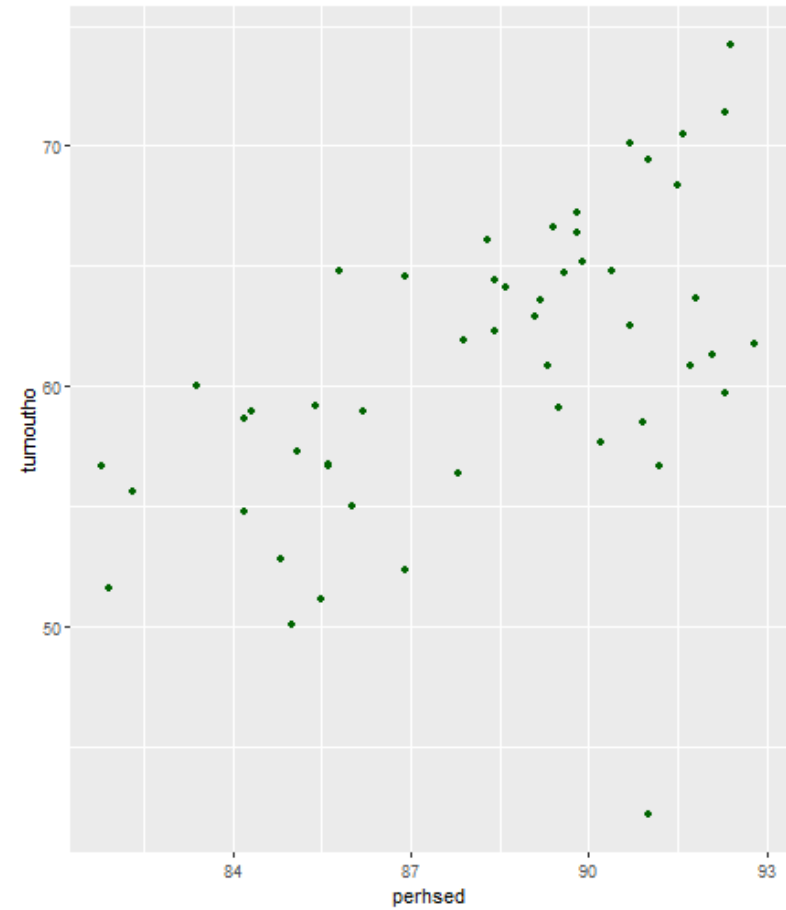


Scatterplot

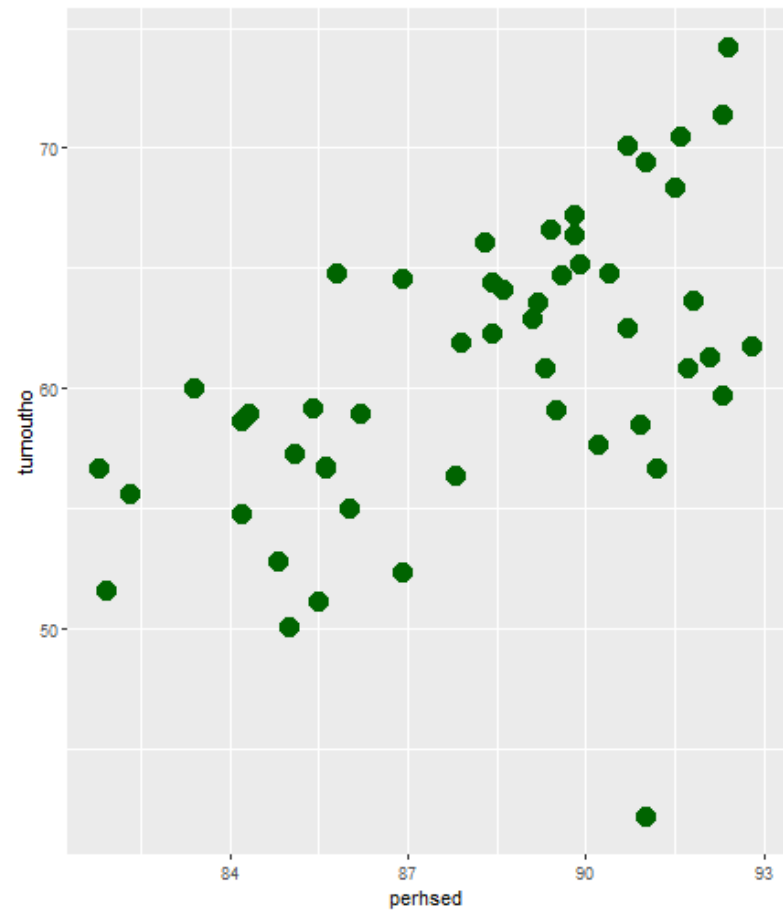
```
election_turnout %>%  
  ggplot() +  
  geom_point(mapping =  
    aes(x = perhsed,  
        y = turnoutho),  
  )
```



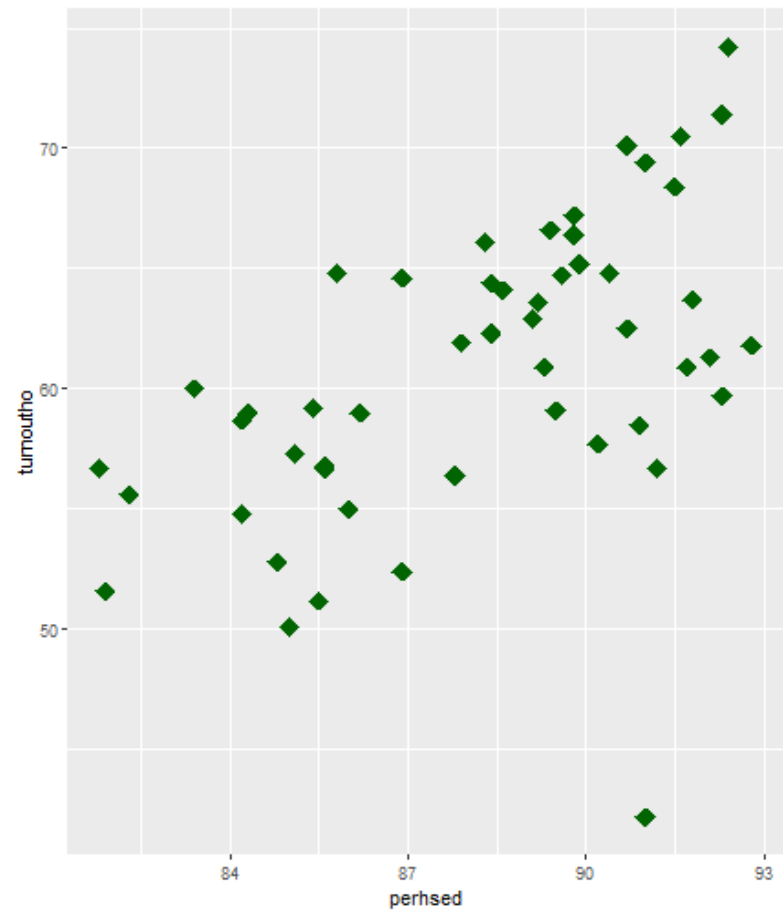
```
election_turnout %>%  
  ggplot() +  
  geom_point(mapping =  
    aes(x = perhsed,  
        y = turnoutho),  
    color = "darkgreen",  
  )
```



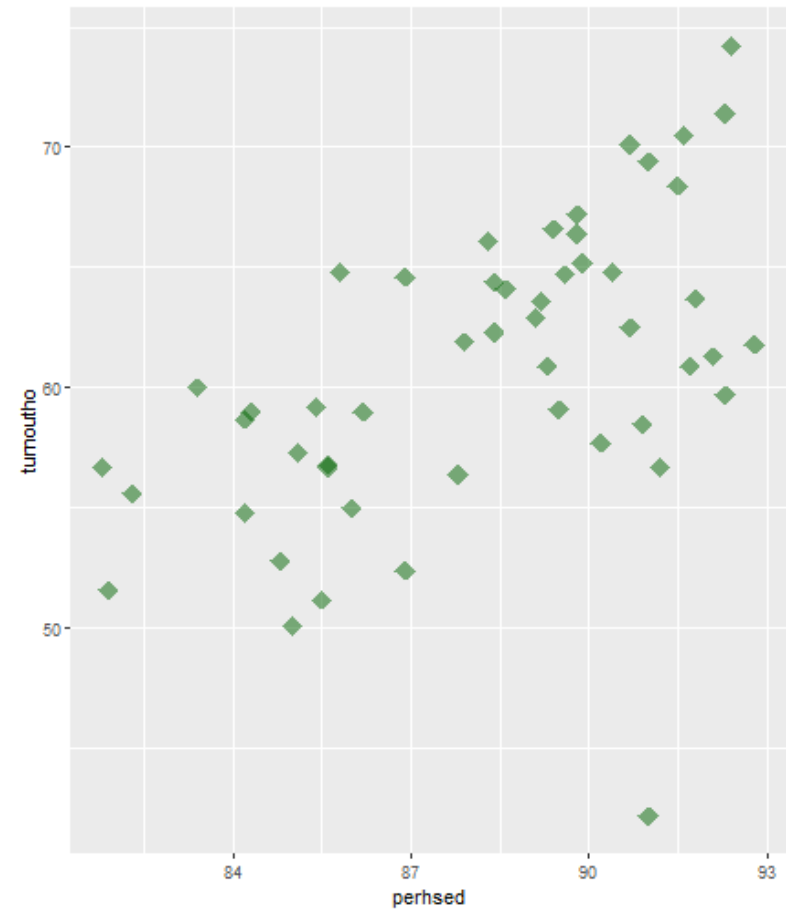
```
election_turnout %>%  
  ggplot() +  
  geom_point(mapping =  
    aes(x = perhsed,  
        y = turnoutho),  
    color = "darkgreen",  
    size = 5,  
  )
```



```
election_turnout %>%  
  ggplot() +  
  geom_point(mapping =  
    aes(x = perhsed,  
        y = turnoutho),  
    color = "darkgreen",  
    size = 5,  
    shape = 18  
  )
```



```
election_turnout %>%  
  ggplot() +  
  geom_point(mapping =  
    aes(x = perhsed,  
        y = turnoutho),  
    color = "darkgreen",  
    alpha = 0.5,  
    size = 5,  
    shape = 18  
  )
```



alpha

This parameter sets the transparency of your color

It is particularly useful in case of **overplotting** - there are too many observations on one graph and they overlap on one another. The overlap prevents the reader to distinguish the number of observations on a graph. Transparency helps to visualize them.

- Darker color means that there are multiple observations on one another

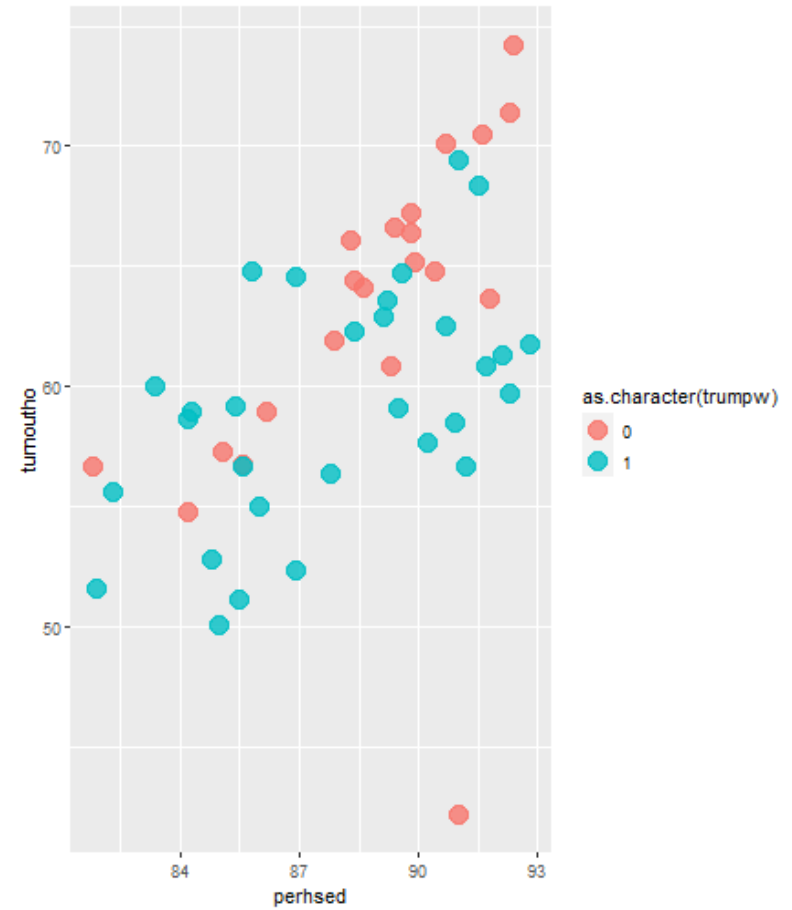
I also use this if I am showing two groups on a graph (old vs. new groups or predicted vs. fitted values).

Colors in aes

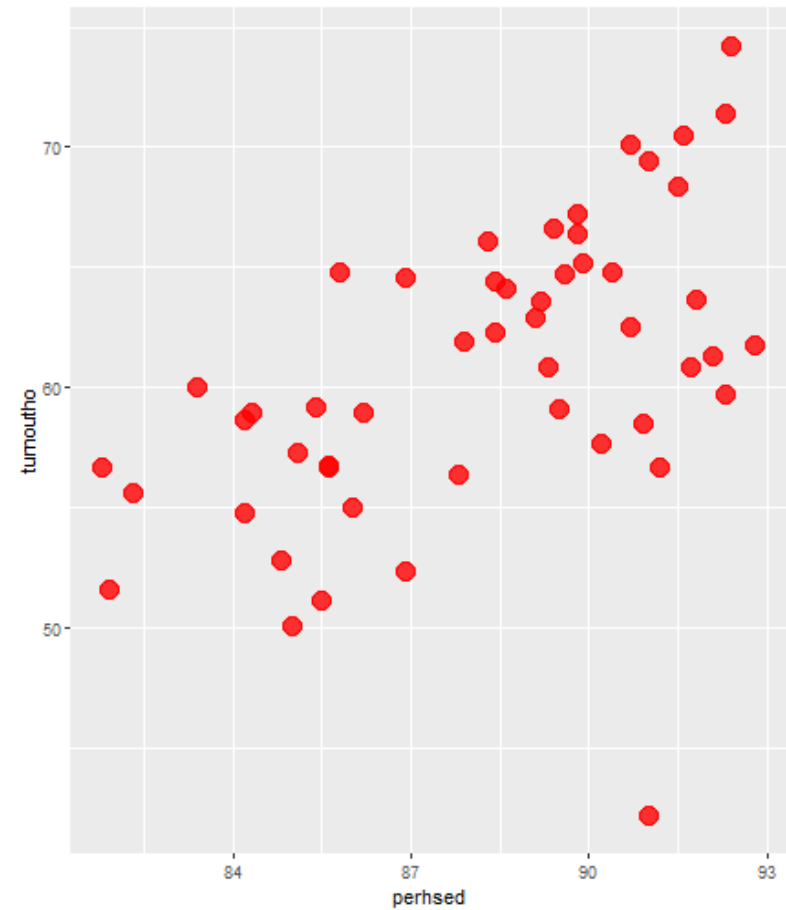
Remember that we used color and fill inside the **aes** field to specify whether the observations on our graph should have been grouped according to a categorical variable.

In that case, changing colors works slightly differently...


```
election_turnout %>%
  ggplot() +
  geom_point(mapping =
    aes(x = perhsed,
        y = turnoutho,
        color = as.character(trumpw),
        size = 5,
        alpha = 0.8,
    )
  )
```



```
election_turnout %>%
  ggplot() +
  geom_point(mapping =
    aes(x = perhsed,
        y = turnoutho,
        color = as.character(trumpw),
        size = 5,
        alpha = 0.8,
        color = "red"
    )
  )
```



Scales

Scales

Scales in ggplot2 control the mapping from data to aesthetics. They take your data in **aes** and turn it into something that you can see.

Remember in aes you can map:

- position (x or y)
- shape
- size
- color
- line

As long as you have mapped a variable to an aesthetic with `aes()`, you can use the `scale_*`() functions to deal with it.

We already some scales last time to modify the position argument in aes:

- `scale_y_continuous`
- `scale_x_continuous`
- `scale_y_discrete`
- `scale_x_discrete`

Scales

Here are some scales that you can use to change color when color or fill are set up within the aes command.

Change aes arguments

- `scale_fill_manual`
- `scale_color_manual`
- `scale_shape_manual`
- `scale_size_continuous`
- `scale_alpha_manual`

Use of palettes

- `scale_fill_PALETTENAMES`
- `scale_color_PALETTENAMES`

Reverse your scale

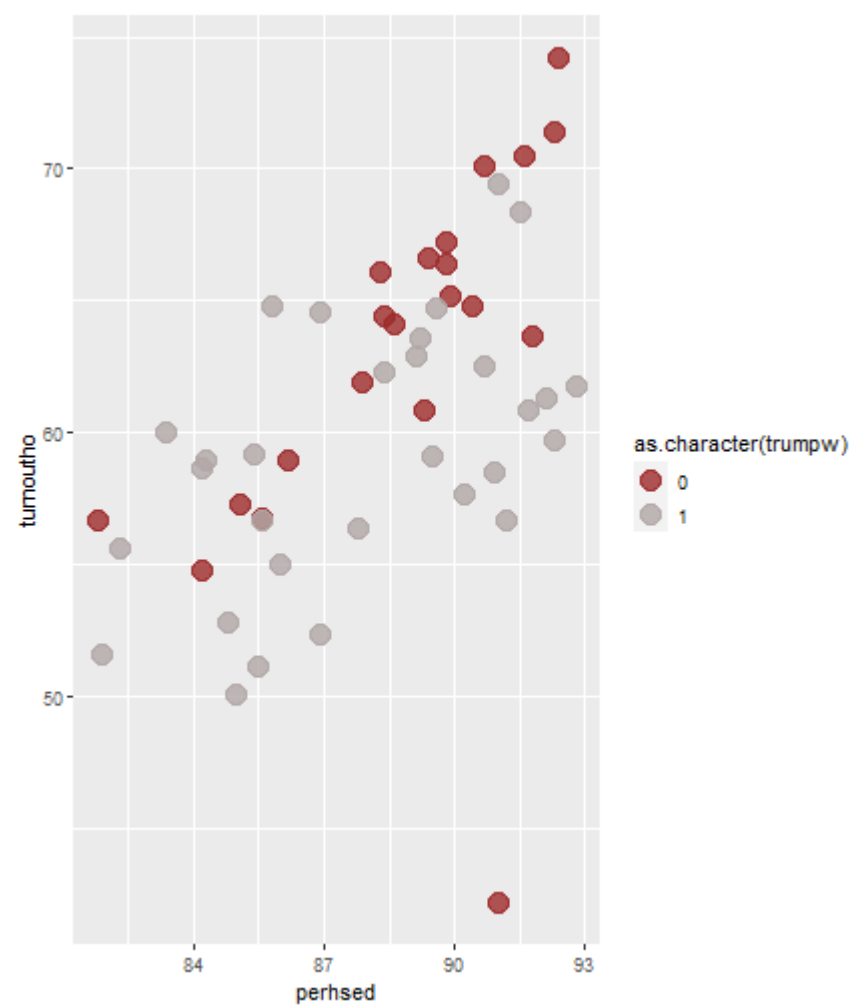
- `scale_x_reverse`
- `scale_y_reverse`

More scales can be found [here](#)

Examples

```
plot_2 =  
election_turnout %>%  
  ggplot() +  
  geom_point(mapping =  
    aes(x = perhsed,  
        y = turnoutho,  
        color = as.character(trumpw)),  
    size = 5,  
    alpha = 0.8) +  
  scale_color_manual(values = c("#9e2a2b", "#b1a7a6"))
```

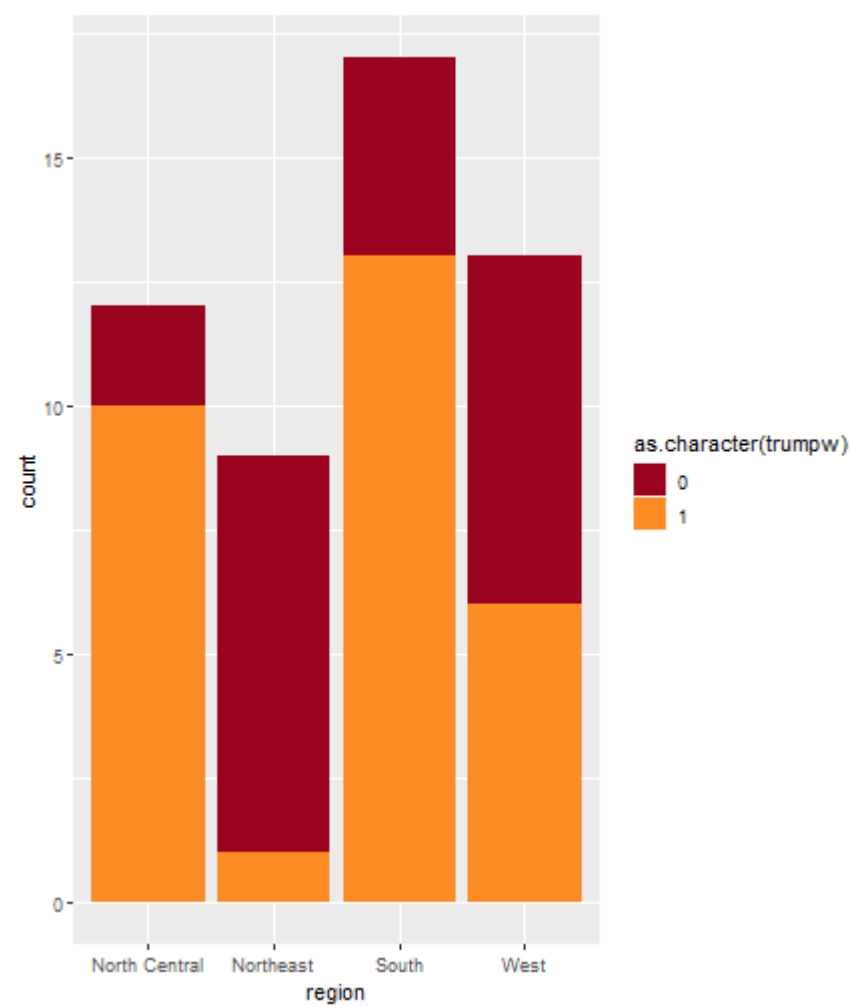
plot_2



Example2

```
plot_3 =  
election_turnout %>%  
  ggplot() +  
  geom_bar(mapping =  
    aes(x = region,  
        fill = as.character(trumpw))) +  
  scale_fill_manual(values = c("#9a031e", "#fb8b24"))
```

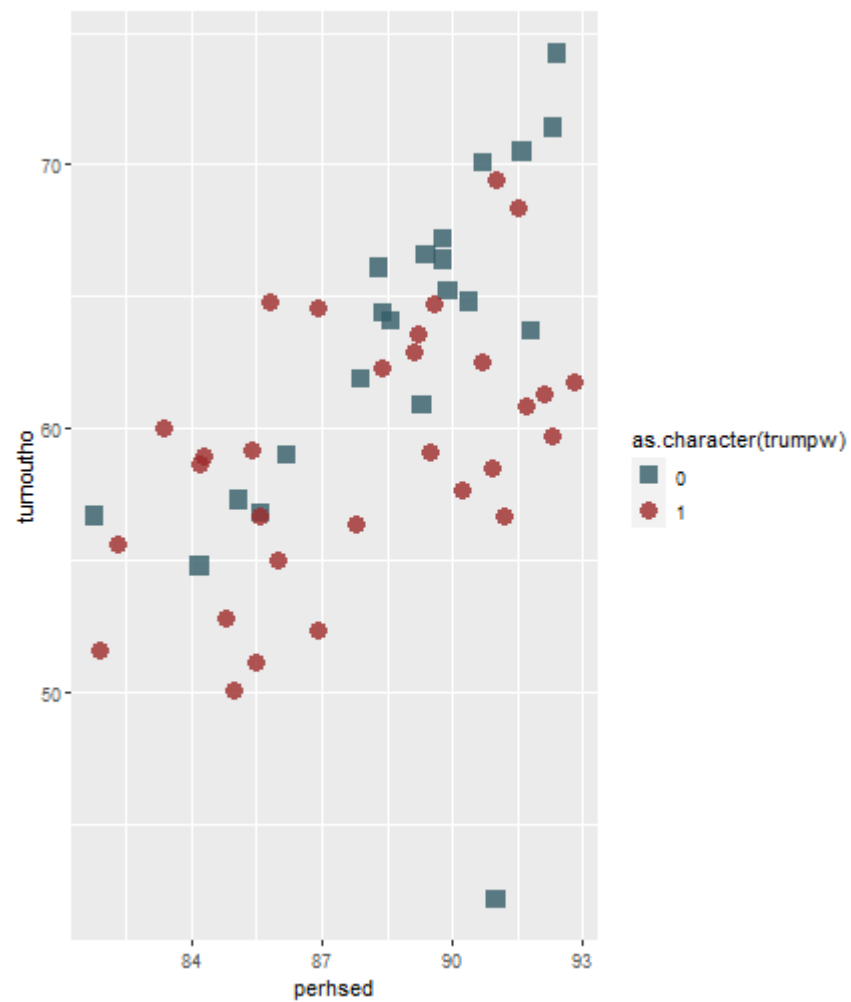
plot_3



Example 3

```
plot_4 =  
election_turnout %>%  
  ggplot() +  
  geom_point(mapping = aes(x = perhsed, y = turnoutho,  
                           shape = as.character(trumpw),  
                           color = as.character(trumpw)),  
             size = 4,  
             alpha = 0.8) +  
  scale_shape_manual(values = c(15, 16)) +  
  scale_color_manual(values = c("#335c67", "#9e2a2b"))
```

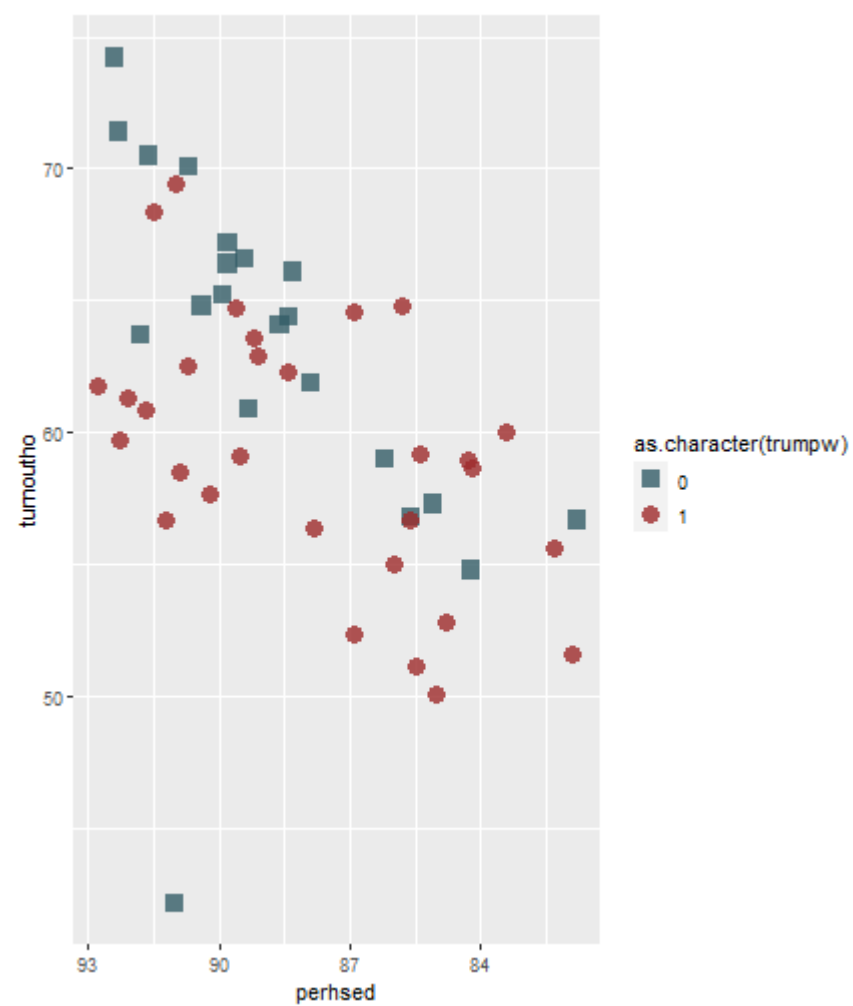
plot_4



reverse

```
plot_5 =  
election_turnout %>%  
  ggplot() +  
  geom_point(mapping = aes(x = perhsed, y = turnoutho,  
                           shape = as.character(trumpw),  
                           color = as.character(trumpw)),  
             size = 4,  
             alpha = 0.8) +  
  scale_shape_manual(values = c(15, 16)) +  
  scale_color_manual(values = c("#335c67", "#9e2a2b")) +  
  scale_x_reverse()
```

plot_5



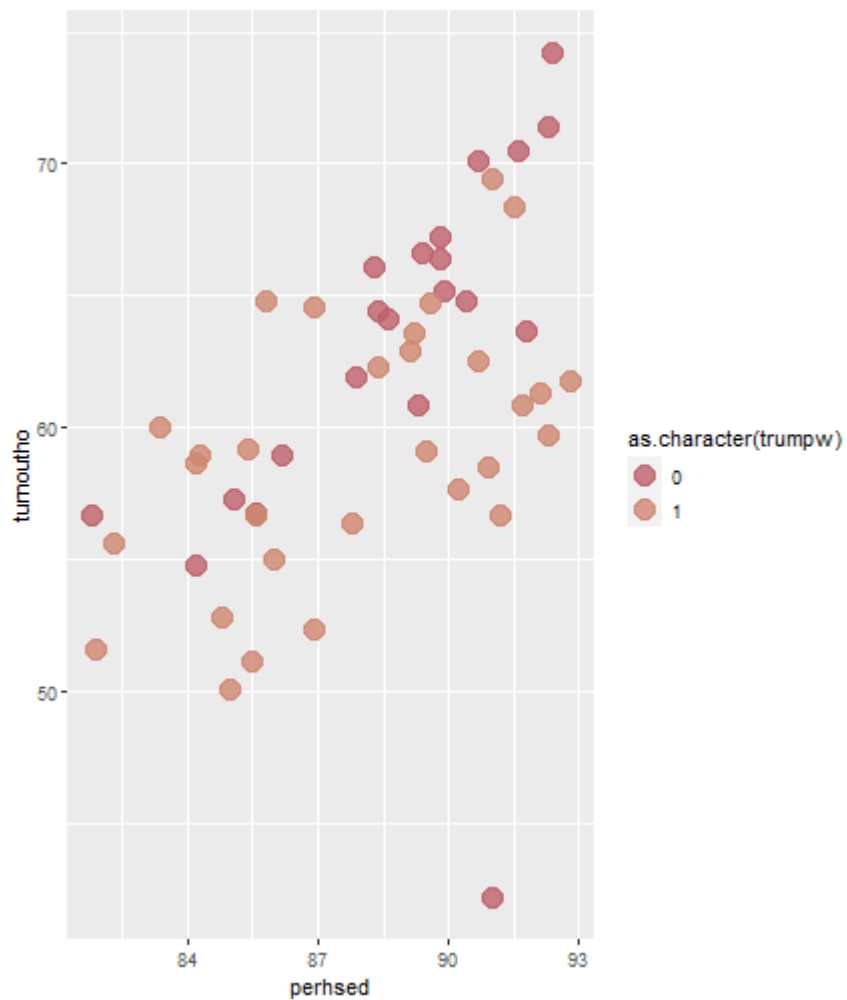
palettes

```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +  
  scale_fill_brewer()
```

```
ggplot(data = election_turnout) +  
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +  
  scale_fill_brewer(palette = "Spectral")
```




```
ggplot(data = election_turnout) +  
  geom_point(mapping = aes(x = perhsed, y = turnoutho,  
                           color = as.character(trumpw)), size = 5, alpha = 0.8) +  
  scale_color_paletteer_d("nord::aurora")
```



Themes

Themes

You can change the overall appearance of your graph, such as a background, borders and so on.

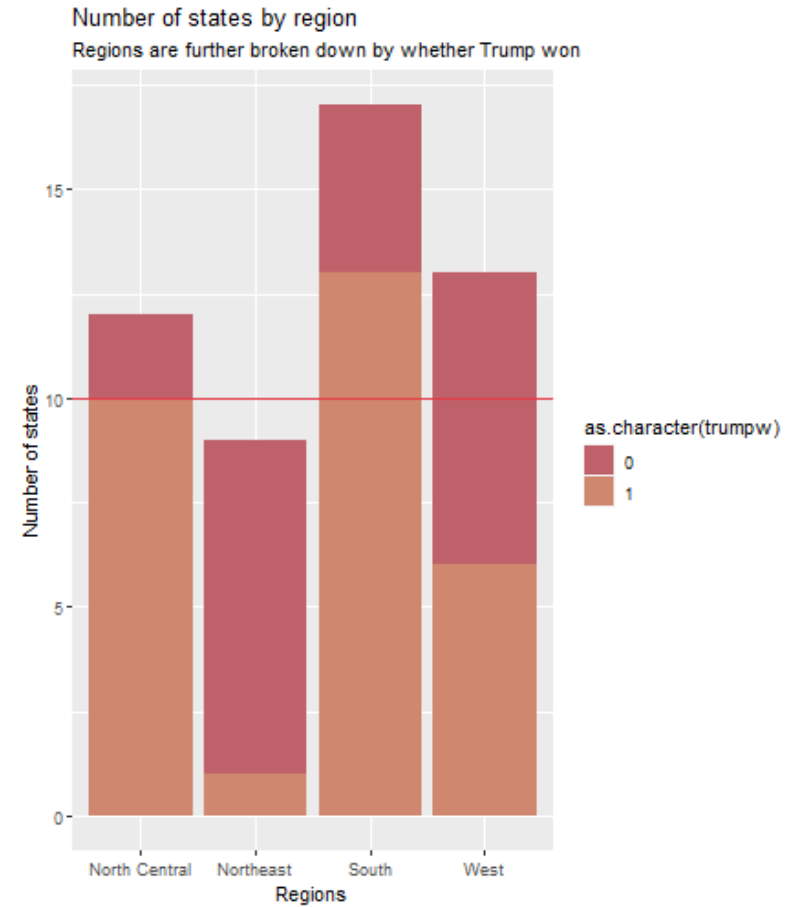
There are 8 themes built into `ggplot2` that you can use to make your graph a better fit for your report.

- `theme_grey`
- `theme_gray`
- `theme_bw`
- `theme_linedraw`
- `theme_light`
- `theme_dark`
- `theme_minimal`
- `theme_classic`
- `theme_void`
- `theme_test`

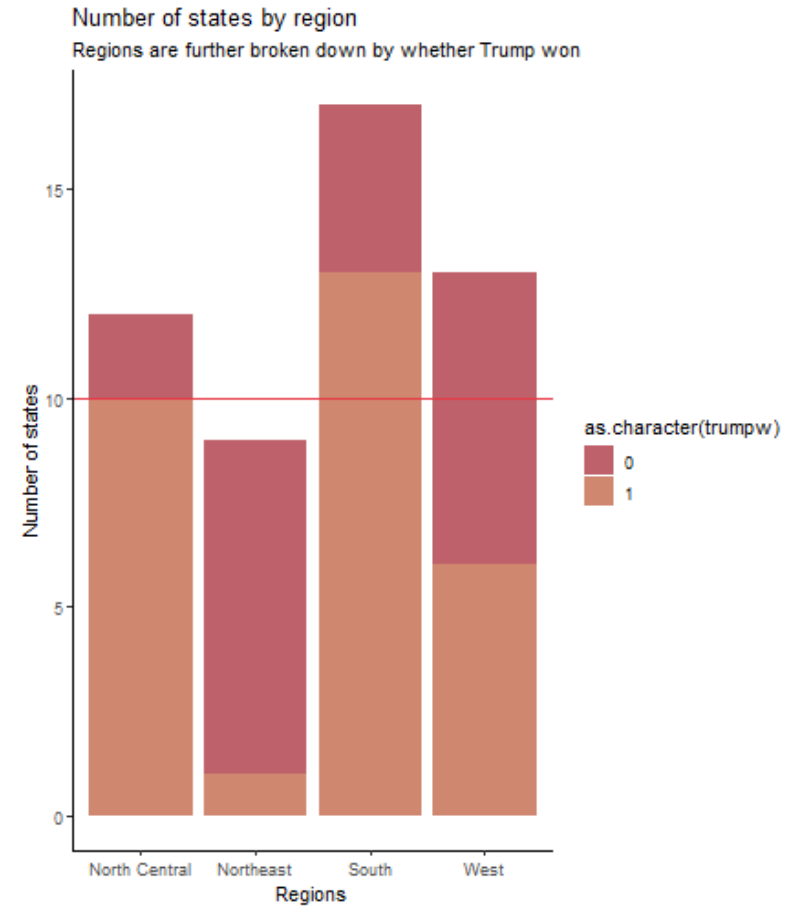
Themes

Let's see this in practice

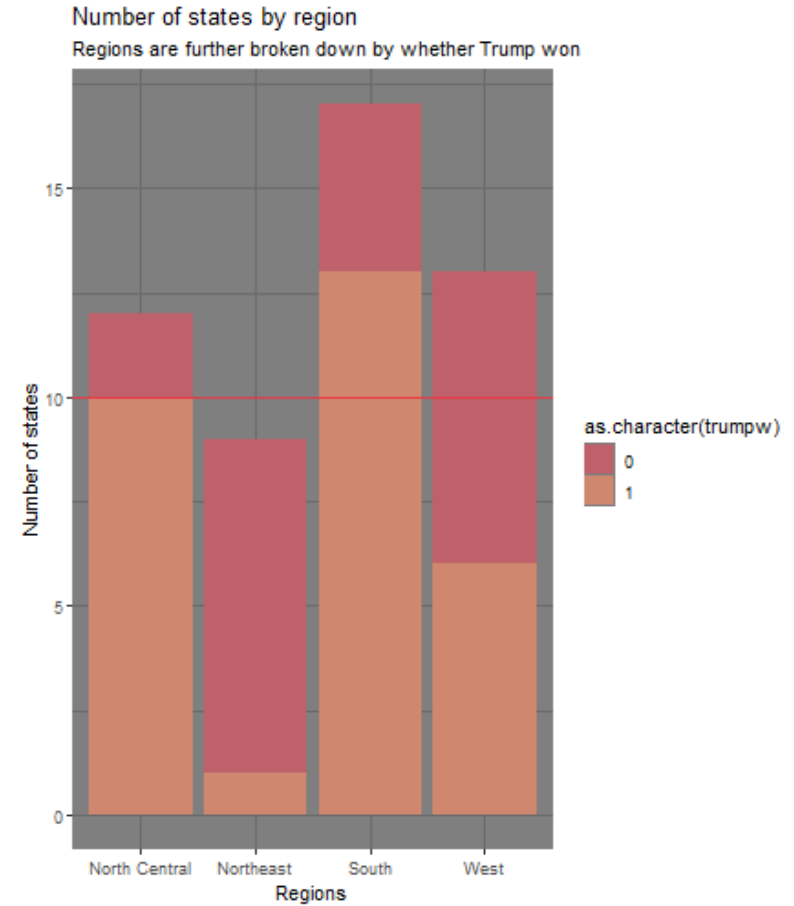
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +
  geom_hline(yintercept = 10, col = "#e63946") +
  scale_fill_paletteer_d("nord::aurora") +
  labs(title = "Number of states by region",
       subtitle = "Regions are further broken down by whether Trump",
       ylab("Number of states") +
       xlab("Regions") +
       theme_grey()
```



```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +
  geom_hline(yintercept = 10, col = "#e63946") +
  scale_fill_paletteer_d("nord::aurora") +
  labs(title = "Number of states by region",
        subtitle = "Regions are further broken down by whether Trump",
        ylab("Number of states") +
        xlab("Regions") +
        theme_classic()
```

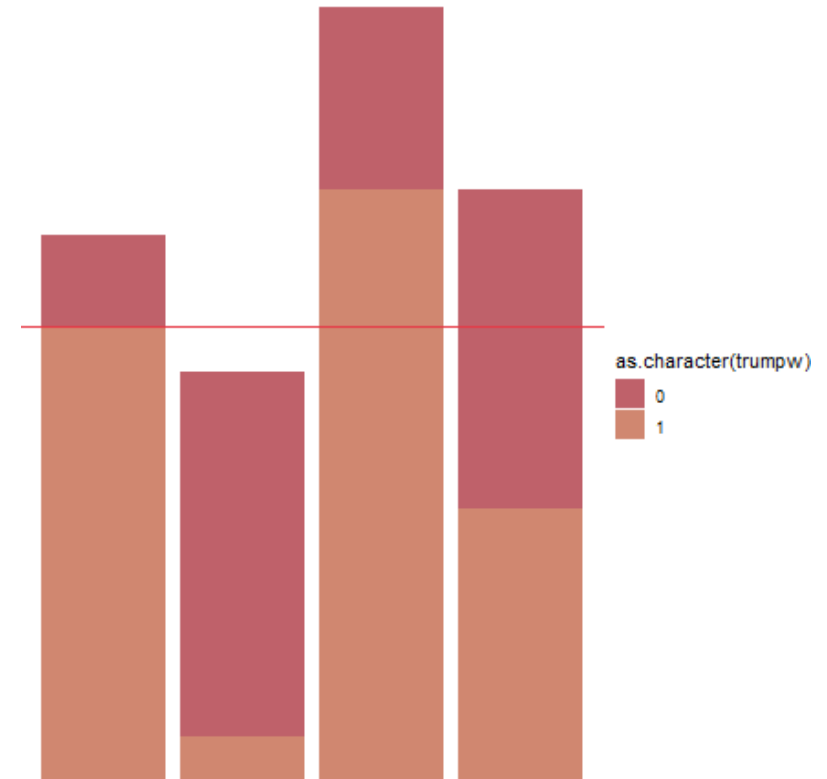


```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw)),
    geom_hline(yintercept = 10, col = "#e63946") +
    scale_fill_paletteer_d("nord::aurora") +
    labs(title = "Number of states by region",
      subtitle = "Regions are further broken down by whether Trump
    ylab("Number of states") +
    xlab("Regions") +
    theme_dark()
```



```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +
  geom_hline(yintercept = 10, col = "#e63946") +
  scale_fill_paletteer_d("nord::aurora") +
  labs(title = "Number of states by region",
        subtitle = "Regions are further broken down by whether Trump",
        ylab("Number of states") +
        xlab("Regions") +
        theme_void()
```

Number of states by region
Regions are further broken down by whether Trump won



Customized themes - arguments

You can also customized your theme. There are **multiple arguments within the function `theme()`** that you can modify. Here is a partial list:

Axes (for all axis elements, you can always specify `.y` or `.x`)

- `axis.title` - *axis labels*
- `axis.text` - *axis ticks' labels*
- `axis.ticks` - *tick marks along axes*
- `axis.line` - *line along axed*

Legend

- `legend.background` - *background of the legend*
- `legend.margin` - *margin around the legend*
- `legend.text` - *legend item lables*
- `legend.title` - *legend title*
- `legend.position` - *legend position*

Customized themes - arguments

Panel

- `panel.background`
- `panel.border`
- `panel.grid`

Plot

- `plot.background`
- `plot.title`
- `plot.subtitle`
- `plot.caption`

Customized themes - elements

To modify these elements there are 4 common elements to use that can be applied to a graph.

- **element_text()** to modify the text of the plot.title, plot.subtitle, and plot.caption, axis.title.x(.y) or axis.text.x(.y)
 - family = "Times New Roman", *specify the font to be used*
 - face = "bold", face = "italic"...*specify the face of the text*
 - color = "red", *specify the color*
 - size = 14, *specify the size of the font*
 - hjust = 10, vjust = 10, *set the horizontal and vertical position in the graph*
 - angle = 30, *specify the angle of the text*

Customized themes - elements

- **element_line()** to modify components as axis lines, major and minor grid lines
 - color
 - size
 - linetype
- **element_rect()** to modify rectangle components such as plot, panel, or legend background
 - fill
 - color
 - size
 - linetype
- **element_blank()** to remove an element from the graph's theme
 - examples: `axis.ticks = element_blank()`

```

full_plot = ggplot(data = election_turnout) +

  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) + # aes settings

  scale_fill_manual(values = c("#e09f3e", "#540b0e")) + #scales to change aes settings - fill
  scale_y_continuous(breaks = seq(0, 20, by = 2)) + #scales to change aes settings - position

  labs(title = "Number of states by region", #title
  subtitle = "Regions are further broken down by whether Trump won") + #subtitle

  ylab("Number of states") + #y lables
  xlab("Regions") + #x labels

  theme_classic() + #theme
  theme(
    plot.title = element_text(face = "bold", size = 16), #change plot title format
    plot.subtitle = element_text(face = "italic", size = 14), #change subtitle format

    axis.title = element_text(size = 14, face = "bold"), #axis title
    axis.text = element_text(size = 12, face = "bold"), #axis lables
    axis.line = element_blank(), #remove the axis line
    axis.ticks = element_blank(), #remove the axis ticks

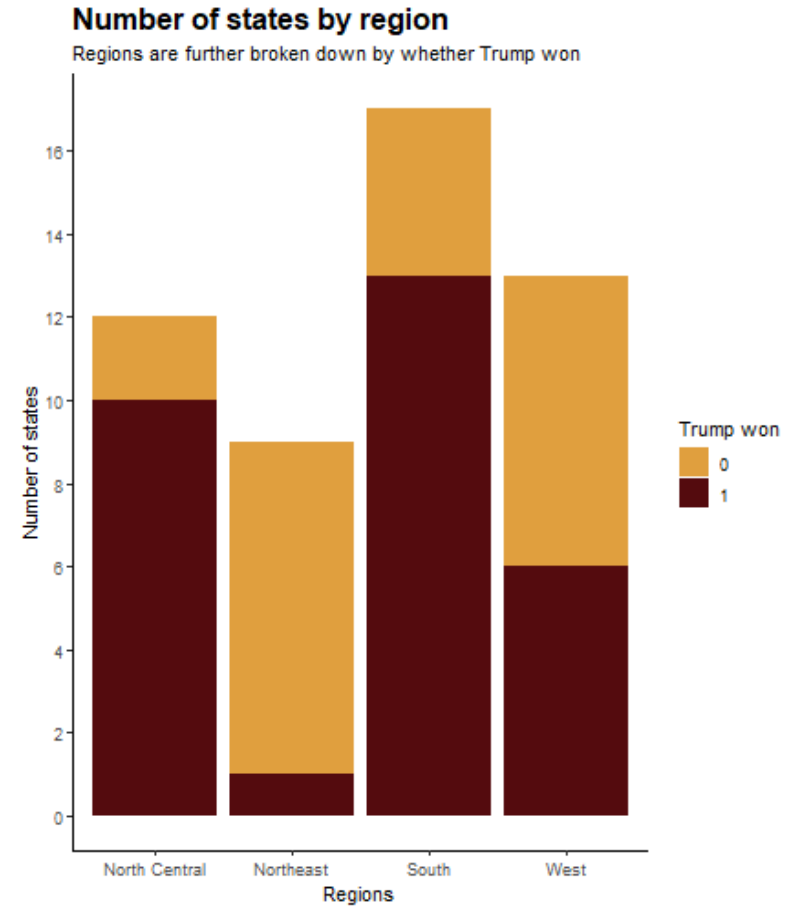
    legend.title = element_text(size = 12, face = "bold", color = "white"), #change legend title
    legend.text = element_text(size = 12, face = "bold", color = "white"), #change legend text
    legend.background = element_rect(fill = "#335c67")) + #change background

  guides(fill = guide_legend("Trump won", nrow = 2)) #change the legend

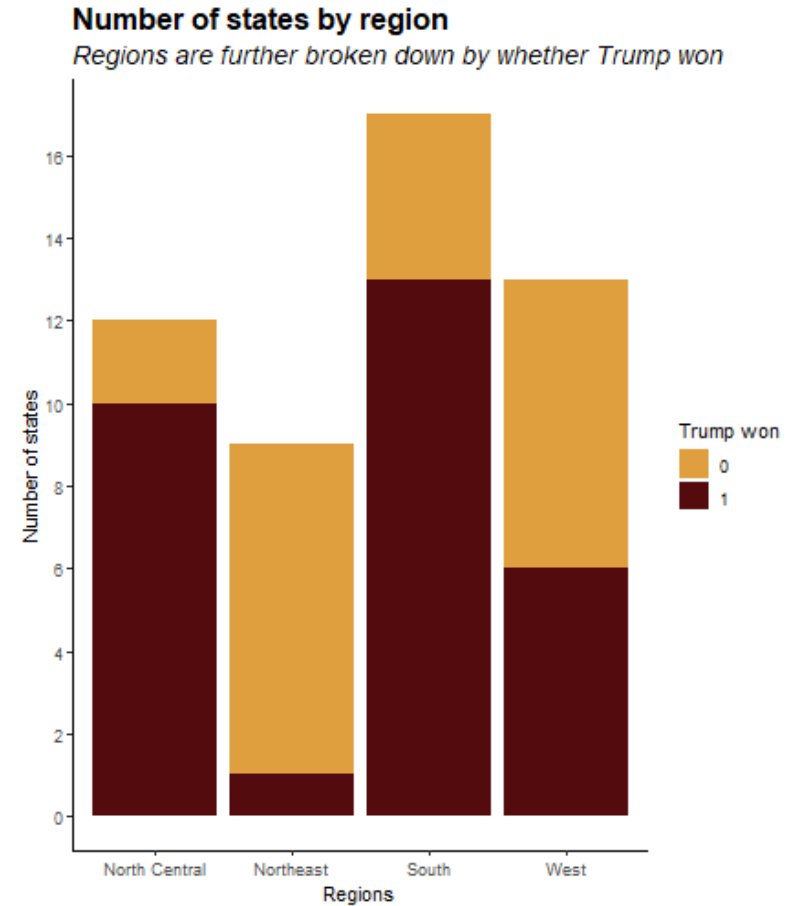
```



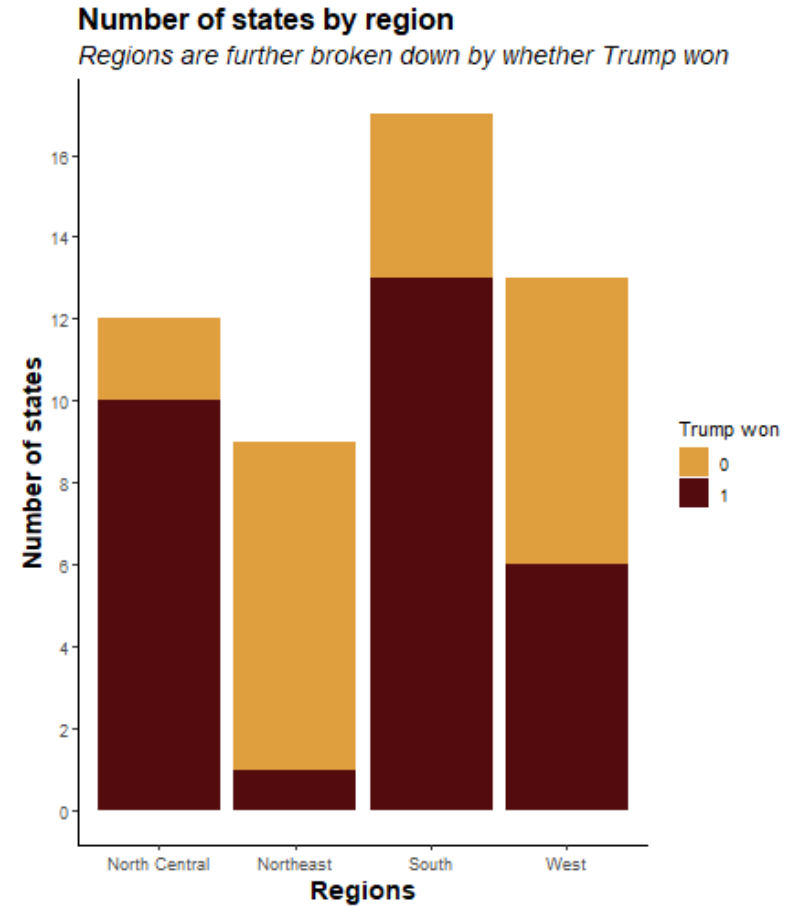
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw)),
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
  subtitle = "Regions are further broken down by whether Trump won",
  ylab("Number of states") +
  xlab("Regions") +
  theme_classic() +
  theme(
  plot.title = element_text(face = "bold", size = 16),
  ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



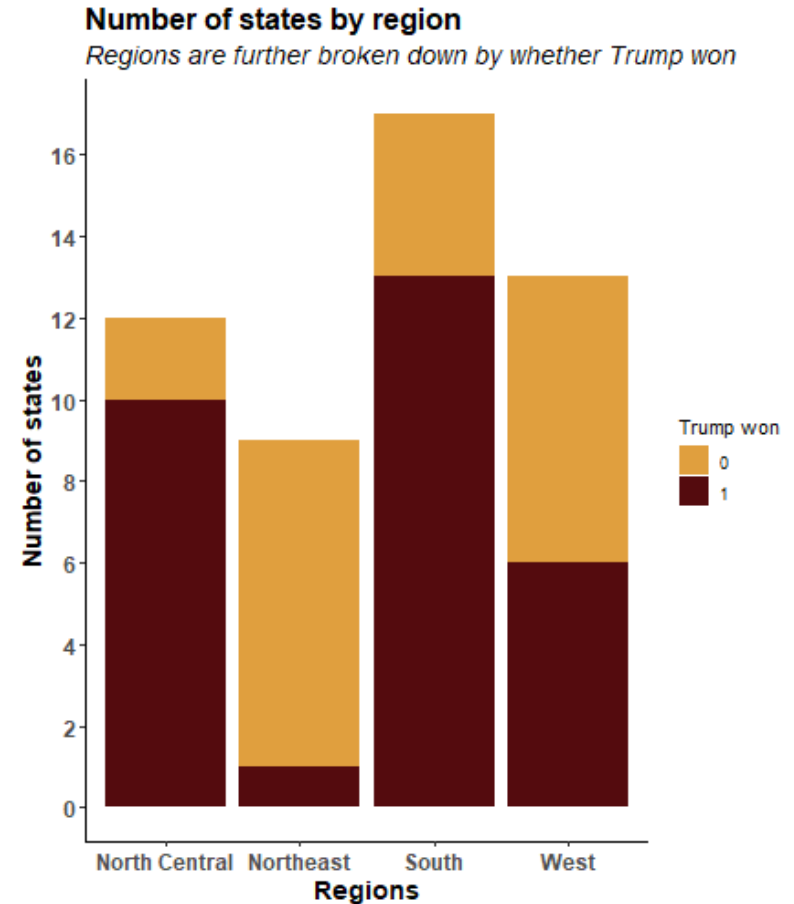
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
       subtitle = "Regions are further broken down by whether Trump won",
       ylab("Number of states") +
       xlab("Regions") +
       theme_classic() +
       theme(
         plot.title = element_text(face = "bold", size = 16),
         plot.subtitle = element_text(face = "italic", size = 14),
       ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



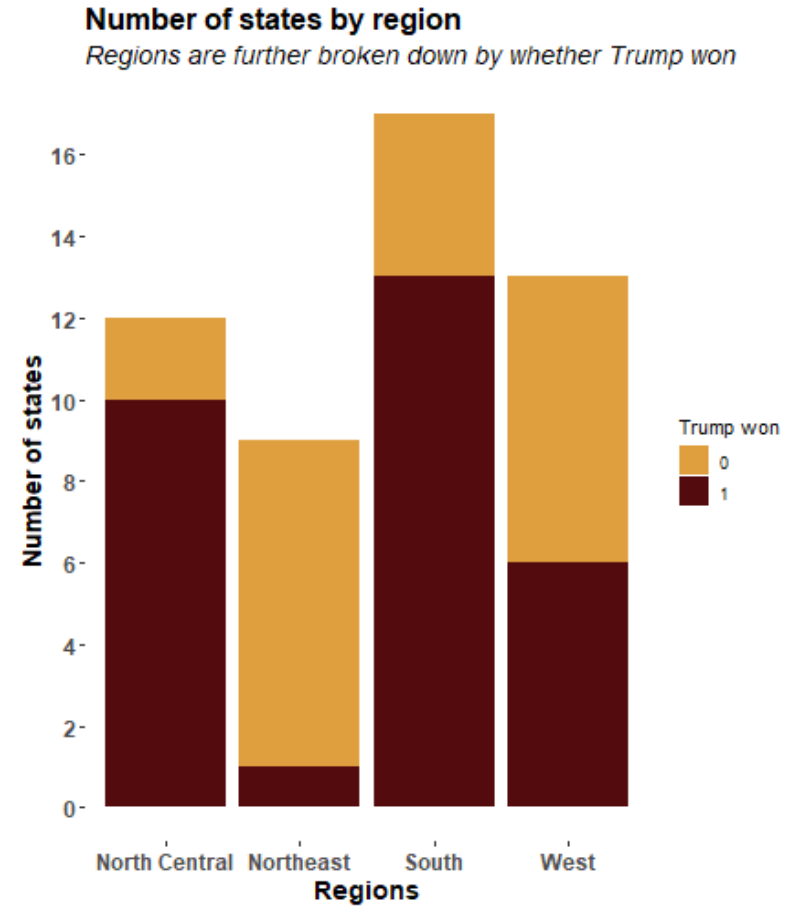

```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
       subtitle = "Regions are further broken down by whether Trump won",
       ylab("Number of states") +
       xlab("Regions") +
       theme_classic() +
       theme(
         plot.title = element_text(face = "bold", size = 16),
         plot.subtitle = element_text(face = "italic", size = 14),
         axis.title = element_text(size = 14, face = "bold"),
       ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



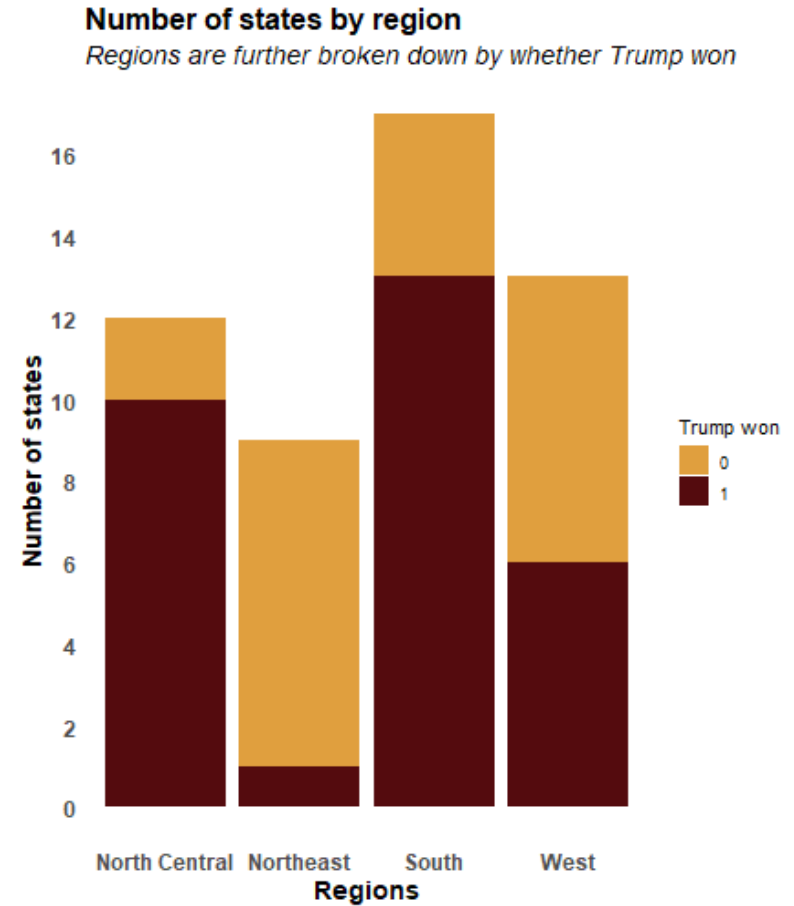
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw)),
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
  subtitle = "Regions are further broken down by whether Trump won",
  ylab("Number of states") +
  xlab("Regions") +
  theme_classic() +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(face = "italic", size = 14),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
  ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



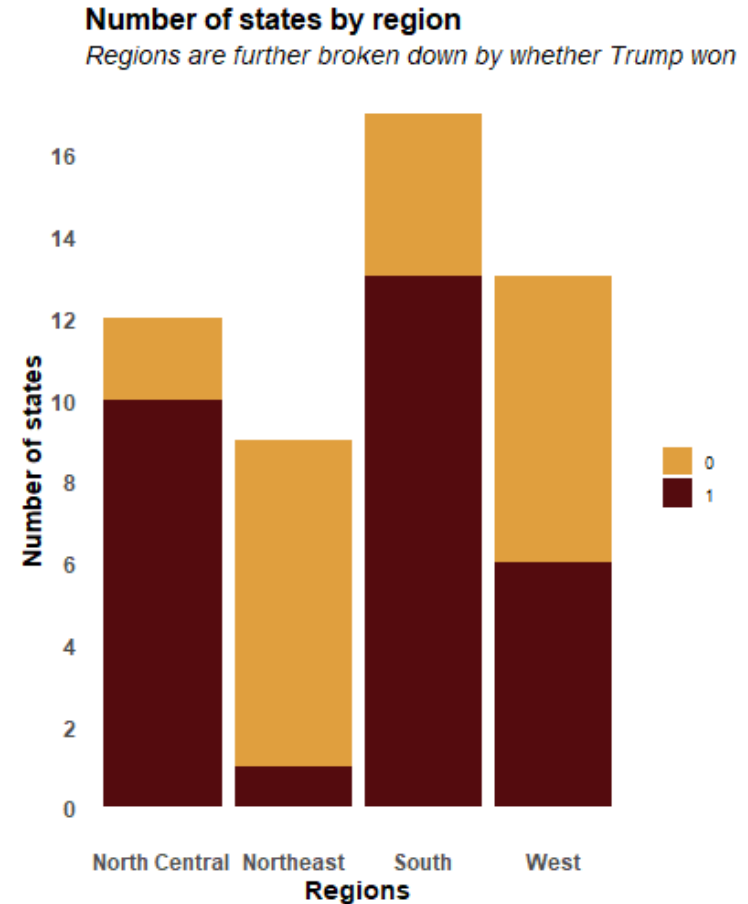
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw)),
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
  subtitle = "Regions are further broken down by whether Trump won",
  ylab("Number of states") +
  xlab("Regions") +
  theme_classic() +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(face = "italic", size = 14),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
    axis.line = element_blank(),
  ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



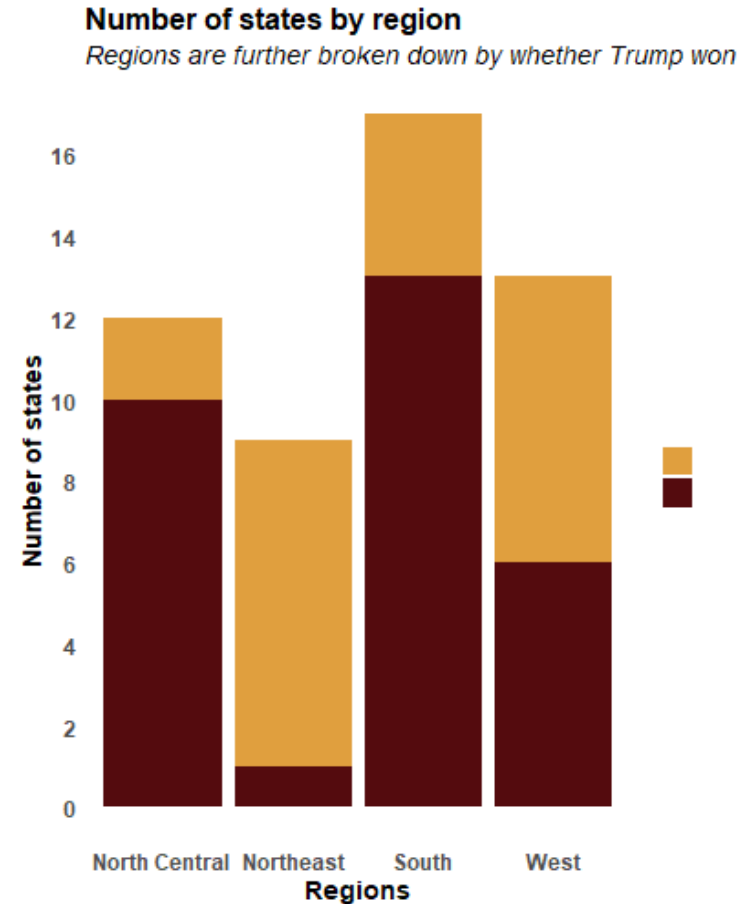
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw)),
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
  subtitle = "Regions are further broken down by whether Trump won",
  ylab("Number of states") +
  xlab("Regions") +
  theme_classic() +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(face = "italic", size = 14),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
    axis.line = element_blank(),
    axis.ticks = element_blank(),
  ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



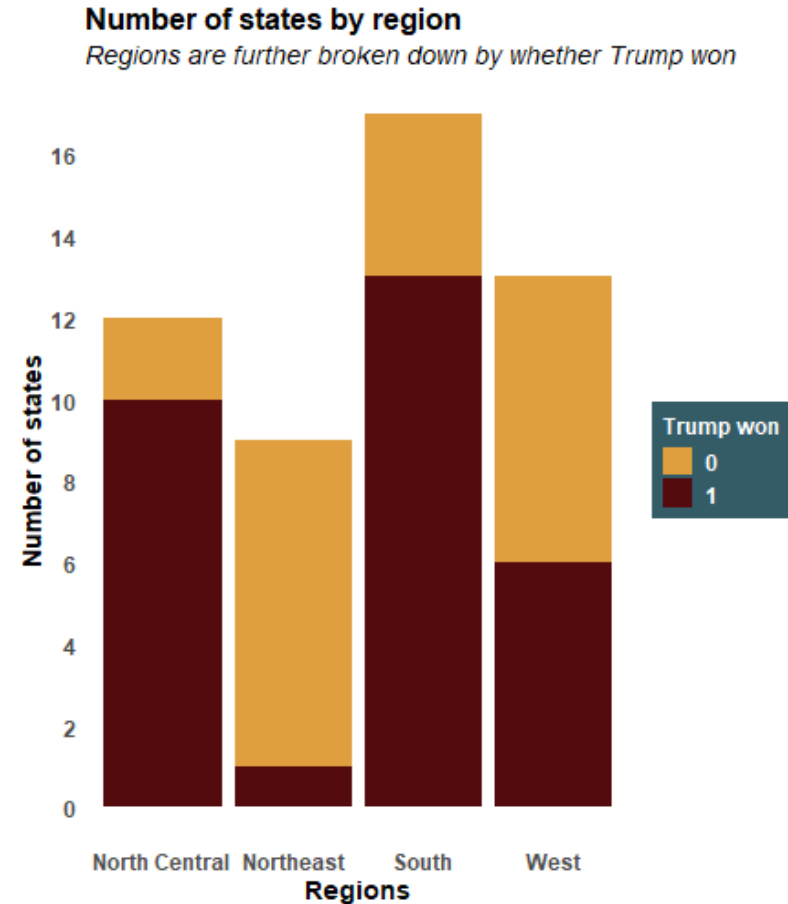
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw))) +
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
       subtitle = "Regions are further broken down by whether Trump won",
       ylab("Number of states") +
       xlab("Regions") +
       theme_classic() +
       theme(
         plot.title = element_text(face = "bold", size = 16),
         plot.subtitle = element_text(face = "italic", size = 14),
         axis.title = element_text(size = 14, face = "bold"),
         axis.text = element_text(size = 12, face = "bold"),
         axis.line = element_blank(),
         axis.ticks = element_blank(),
         legend.title = element_text(size = 12, face = "bold", color = "white"),
         legend.text = element_text(size = 10, face = "normal", color = "white")
       )
  ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



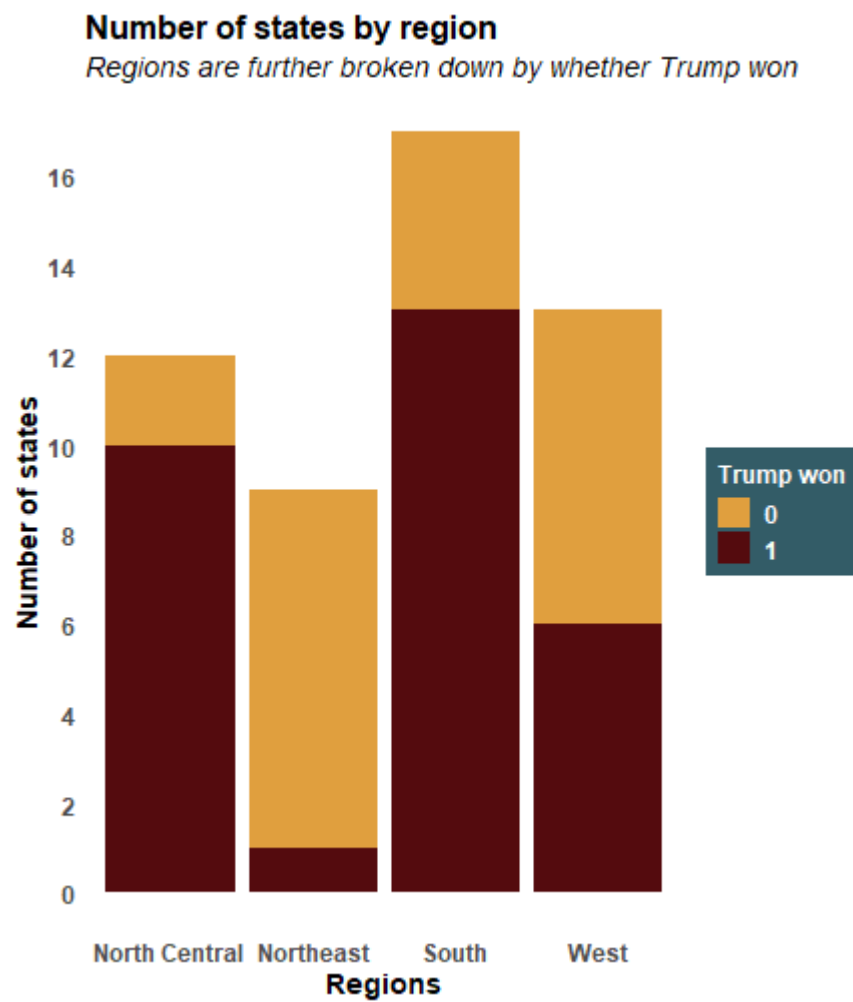
```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw)),
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
  subtitle = "Regions are further broken down by whether Trump won",
  ylab("Number of states") +
  xlab("Regions") +
  theme_classic() +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(face = "italic", size = 14),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
    axis.line = element_blank(),
    axis.ticks = element_blank(),
    legend.title = element_text(size = 12, face = "bold", color = "white"),
    legend.text = element_text(size = 12, face = "bold", color = "white"),
  ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



```
ggplot(data = election_turnout) +
  geom_bar(mapping = aes(x = region, fill = as.character(trumpw)),
  scale_fill_manual(values = c("#e09f3e", "#540b0e")) +
  scale_y_continuous(breaks = seq(0, 20, by = 2)) +
  labs(title = "Number of states by region",
  subtitle = "Regions are further broken down by whether Trump won",
  ylab("Number of states") +
  xlab("Regions") +
  theme_classic() +
  theme(
    plot.title = element_text(face = "bold", size = 16),
    plot.subtitle = element_text(face = "italic", size = 14),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12, face = "bold"),
    axis.line = element_blank(),
    axis.ticks = element_blank(),
    legend.title = element_text(size = 12, face = "bold", color = "white"),
    legend.text = element_text(size = 12, face = "bold", color = "white"),
    legend.background = element_rect(fill = "#335c67")
  ) + guides(fill = guide_legend("Trump won", nrow = 2))
```



```
full_plot
```



Saving your theme

This is especially helpful if you are writing a report or a full document.

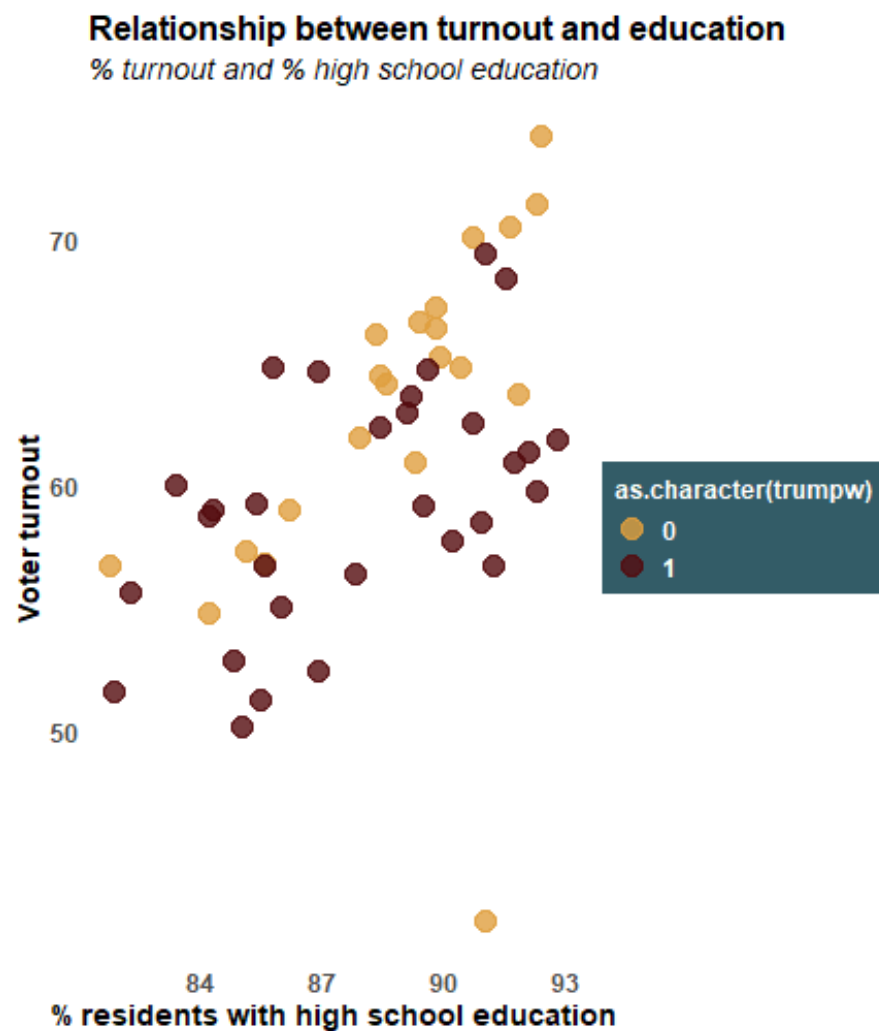
```
my_new_theme <-  
  theme_classic() +  
  theme(  
    plot.title = element_text(face = "bold", size = 16),  
    plot.subtitle = element_text(face = "italic", size = 14),  
  
    axis.title = element_text(size = 14, face = "bold"),  
    axis.text = element_text(size = 12, face = "bold"),  
    axis.line = element_blank(),  
    axis.ticks = element_blank(),  
  
    legend.title = element_text(size = 12, face = "bold", color = "white"),  
    legend.text = element_text(size = 12, face = "bold", color = "white"),  
    legend.background = element_rect(fill = "#335c67"))
```

Saving your theme

```
plot_1 =  
election_turnout %>%  
  ggplot() +  
  geom_point(mapping = aes(x = perhsed, y = turnoutho,  
                           color = as.character(trumpw)),  
             size = 5,  
             alpha = 0.8) +  
  scale_color_manual(values = c("#e09f3e", "#540b0e")) +  
  
  labs(title = "Relationship between turnout and education",  
       subtitle = "% turnout and % high school education") +  
  
  ylab("Voter turnout") +  
  xlab("% residents with high school education") +
```

```
my_new_theme
```

plot_1



Additional themes

There are several packages with customized themes that you can use inspired by newspapers, other software, TV series...

ggthemes - <https://yutannihilation.github.io/allYourFigureAreBelongToUs/ggthemes/> hrbrthemes - <https://github.com/hrbrmstr/hrbrthemes> ggpomological (an older feeling...) - <https://www.garrickadenbuie.com/project/ggpomological/> ggtech (inspired by tech companies) - <https://github.com/ricardo-bion/ggtech>

To use them:

```
install.packages("ggthemes")
library("ggthemes")

election_turnout %>%
  ggplot() +
  geom_point(mapping = aes(x = perhsed, y = turnoutho,
                          shape = as.character(trumpw),
                          size = 4,
                          alpha = 0.8)) +
  scale_shape_manual(values = c(15, 16)) +
  theme_stata()
```

Save your plot

To save your plot, you can use **ggsave** as shown here.

```
ggsave(filename = "my_first_plot.png", # you can also use .pdf
        plot = plot_1)

ggsave(filename = "my_first_plot.png",
        plot = plot_1,
        width = 10,
        height = 10,
        units = "in")
```

Saving your plot

Here are some general guidelines:

.JPEG (maintains colors)

- Lots of colors (like a photograph)
- Images on the Internet
- Billboard size

.PNG (compress a few colors)

- Few colors
- Not a photograph
- To be used in Word or other document

.PDF or .SVG

- Plan to use in Adobe
- Plan to use in multiple sizes

A few final notes

There is SO MUCH MORE to data viz! This was a two-session crash course on the basics of ggplot2.

We will keep working with plots as we move forward and build off this knowledge.

There are also tons of resources for you to play around data visualization.

One of them to draw inspiration + technical ideas: **#TidyTuesday**. This is a weekly online "challenge" where a dataset is proposed and anyone can use it to make cool data viz. You can find a [collection of past challenges at this website](#).

Some data visualizations are really complex (i.e., we haven't covered those packages nor functions) but others are cool examples of how you can do nice graphs with pretty much what we learned so far.

We will use some datasets from the TidyTuesday challenge in the next couple of weeks....

Assignment 7

Have some fun and work on a TidyTuesday challenge!

We are going to use the **Transit Cost Project** data. The original [source of data is here](#) but we are going to use the **TidyTuesday dataset**.

The dataset is already tidy - each row is a transit project. There can be multiple projects in the same city and same country.

Your goal is to produce a nice plot that could be potentially used for explanatory purposes with your readers.

Submit your code along with a picture of your final graph

You can absolutely draw inspiration (but not plagiarize) from the TidyTuesday website - they have images + codes!

Evaluation will be based on your ability to choose an appropriate data visualization (e.g., right graph for the data), apply the basic rules to make a good plot (e.g., axes, titles...), use of color strategically (e.g., to highlight groups) in the graph, apply a pleasant themes, and overall create a plot that communicate *information* to the reader.

Note: I am not expecting more than what we learned so far! If this is your first time doing data viz force yourself to start 'small' and rely on your knowledge to create the graph. If you are more experienced, feel free to pick around and experiment a new package of your interest.

