Colors!

Daniel Anderson
Week 5, Class 1

UNIVERSITY OF
OREGON

Agenda

- Color basics
  - 3 basic ways color is used
- Color blindness
- Some common problems with color use
- Quick discussion of palettes

Agenda

- Color basics
  - 3 basic ways color is used
- Color blindness
- Some common problems with color use
- Quick discussion of palettes

Learning Objectives

- Understand different types of color palettes
  - ...and when you should use one versus another
- Understand and be able to effectively evaluate concerns related to color blindness
- Be able to fluently change colors/fills within ggplot

Before we get too deep

Some very practical advice

- Keep straight when color is mapped to a variable through `aes` and when it's modifying an element overall
    - Former requires `scale_color_*` or `scale_fill_*` while the latter does not

Before we get too deep

Some very practical advice

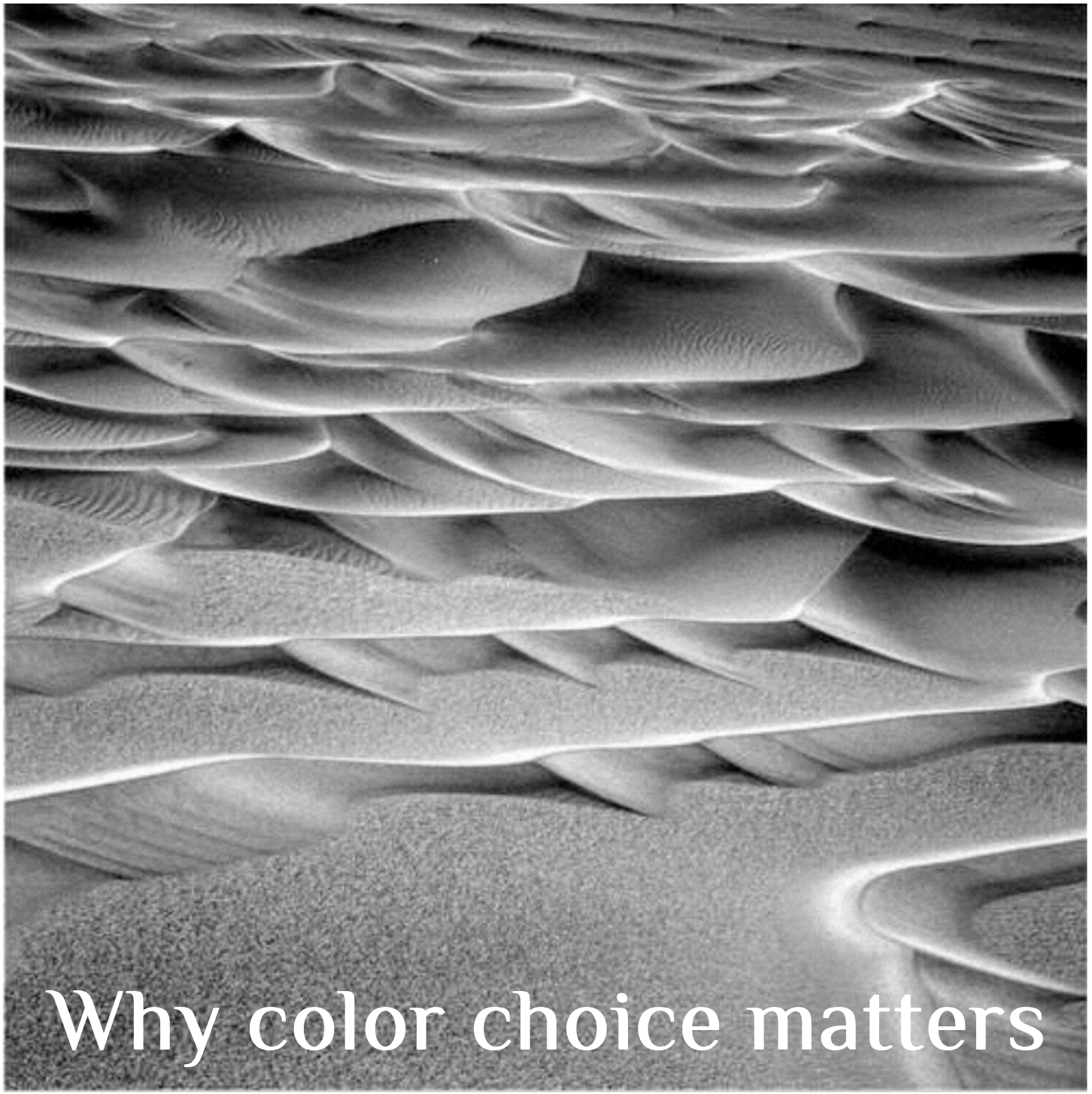
- Keep straight when color is mapped to a variable through `aes` and when it's modifying an element overall
    - Former requires `scale_color_*` or `scale_fill_*` while the latter does not
- Keep straight colors and fills (see former bullet)

Before we get too deep

Some very practical advice

- Keep straight when color is mapped to a variable through `aes` and when it's modifying an element overall
    - Former requires `scale_color_*` or `scale_fill_*` while the latter does not
- Keep straight colors and fills (see former bullet)
- Use advice of others to your advantage (e.g., http://colorbrewer2.org/)

Why color choice matters



Why color choice matters

Another quick example

rayshader

Three fundamental uses of color

Three fundamental uses of color

1. Distinguish groups from each other

Three fundamental uses of color

1. Distinguish groups from each other
2. Represent data valuels

Three fundamental uses of color

1. Distinguish groups from each other
2. Represent data valuels
3. Highlight

Color as a tool to distinguish

Discrete items

- Often no intrinsic order

Discrete items

- Often no intrinsic order

Qualitative color scale

- Finite number of colors
  - Chosen to maximize distinctness, while also be equivalent
  - Equivalent
    - No color should stand out
    - No impression of order

Some examples

Okabe Ito

ColorBrewer Dark2

ggplot2 hue

See more about the Okabe Ito palette origins here: http://jfly.iam.u-tokyo.ac.jp/color/

How do we use them?

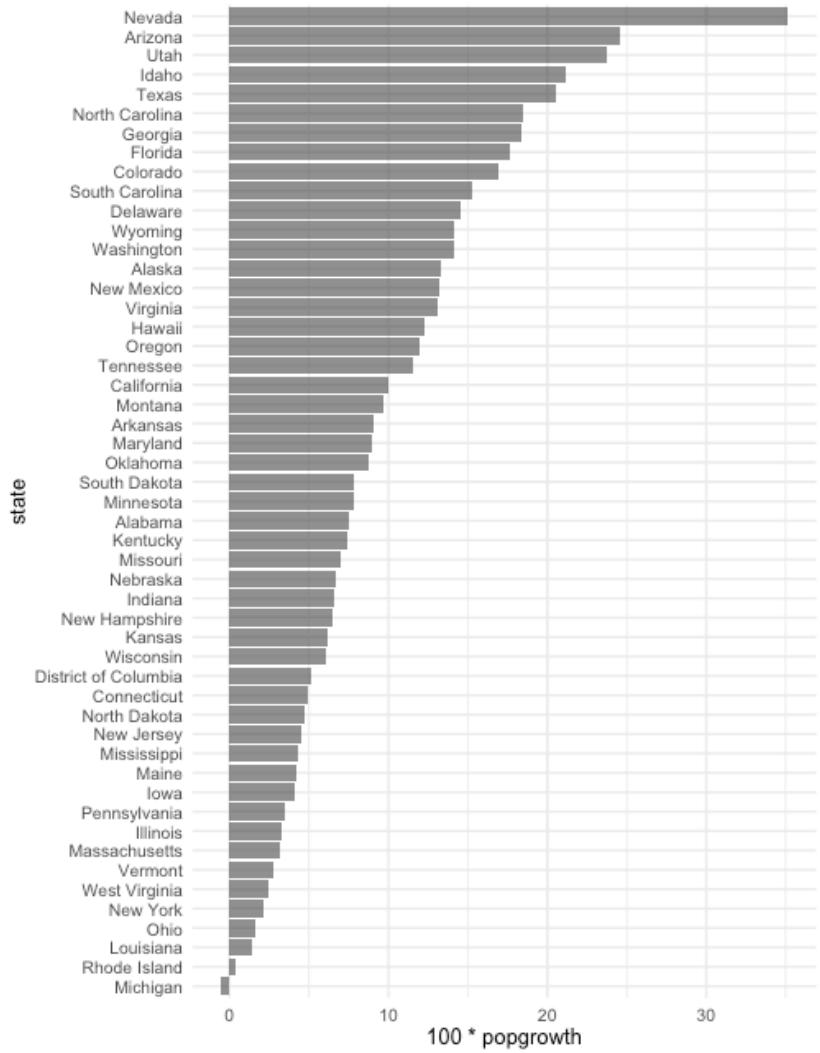
Imagine we have data like this

```
popgrowth_df
```

```
## # A tibble: 51 x 7
##   region   division       state      pop2000  pop2010    popgrowth     area
##   <fct>    <chr>          <fct>        <dbl>    <dbl>        <dbl>    <dbl>
## 1 Midwest  East North Ce… Michigan   9938444  9883640 -0.005514344 56538.92
## 2 Northea… New England    Rhode Is…  1048319  1052567  0.004052202  1033.82
## 3 South    West South Ce… Louisiana  4468976  4533372  0.01440956  43203.93
## 4 Midwest  East North Ce… Ohio      11353140 11536504  0.01615095  40860.73
## 5 Northea… Middle Atlant… New York  18976457 19378102  0.02116544  47126.43
## 6 South    South Atlantic West Vir…  1808344  1852994  0.02469110  24038.21
## # … with 45 more rows
```

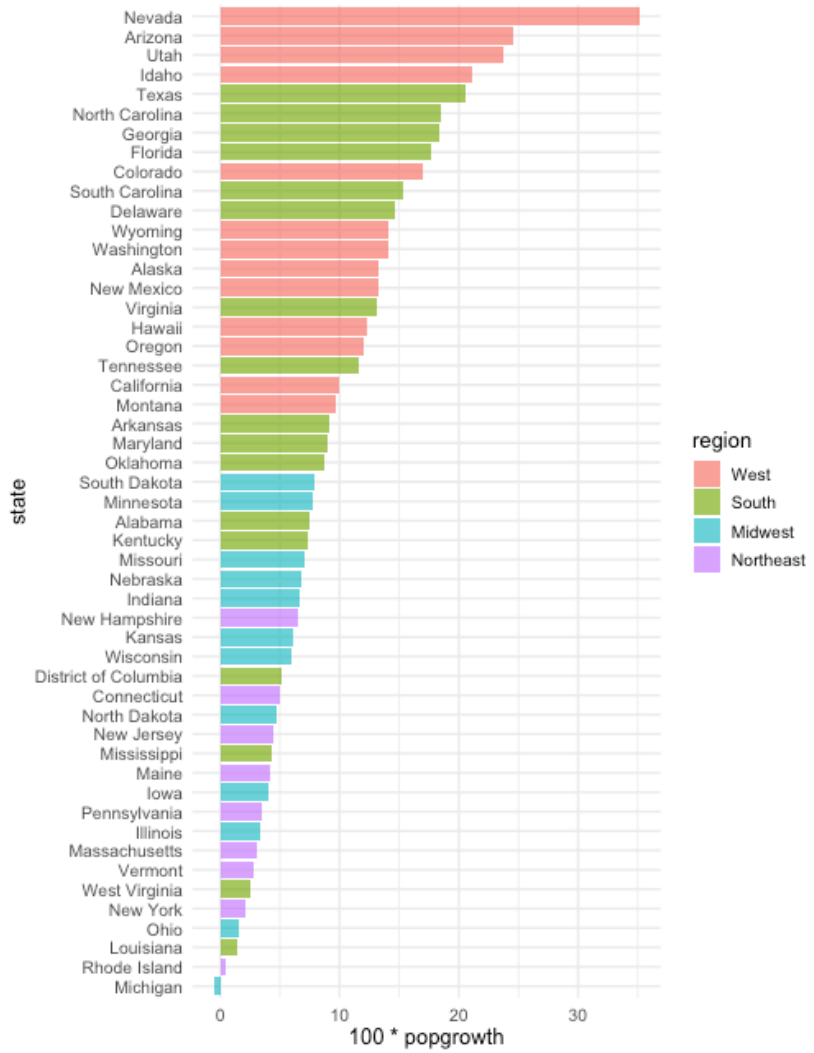
We might build a plot like this

```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(alpha = 0.7) +
  coord_flip()
```

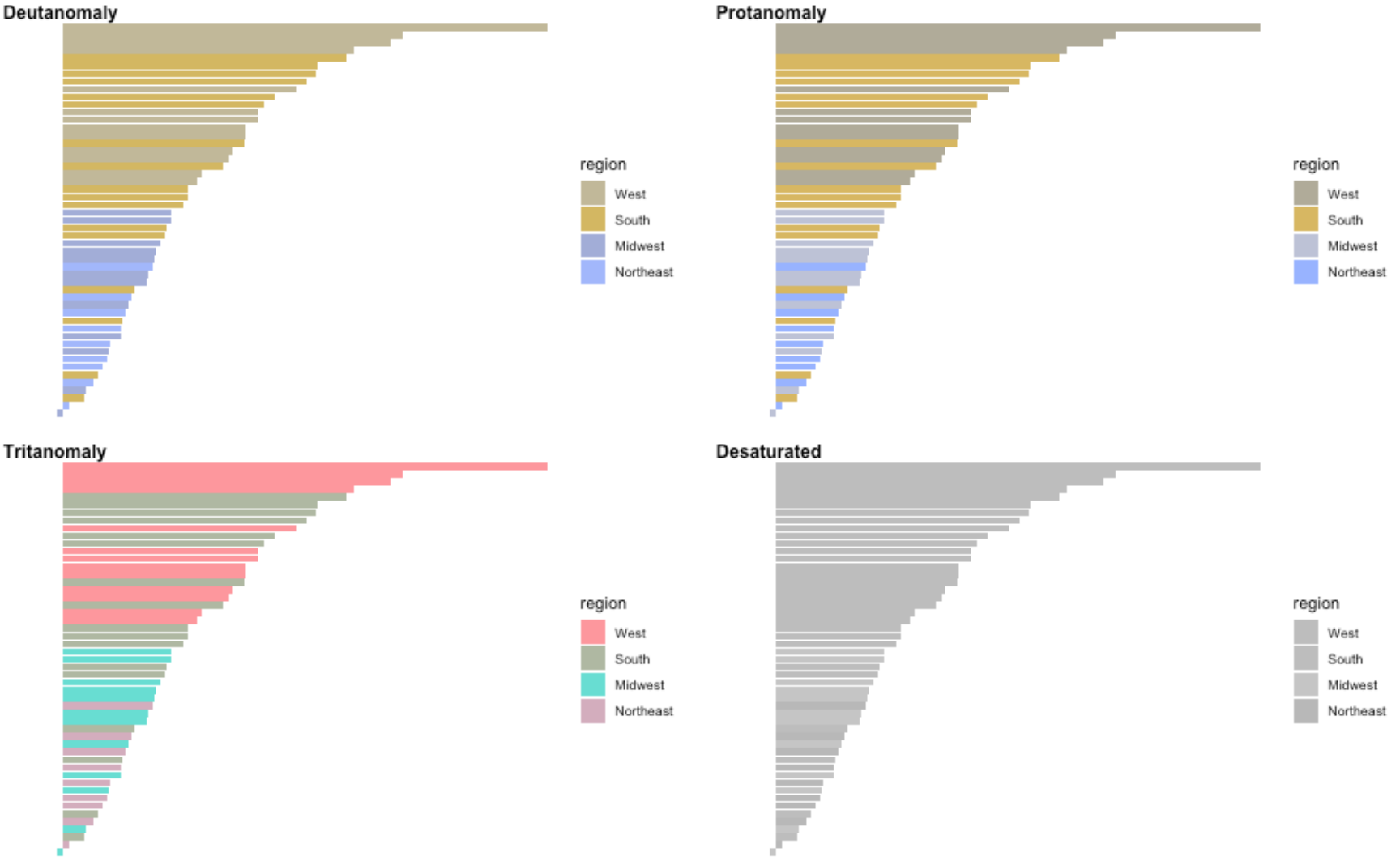


Alternatively, fill by region

```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region),
           alpha = 0.7) +
  coord_flip()
```

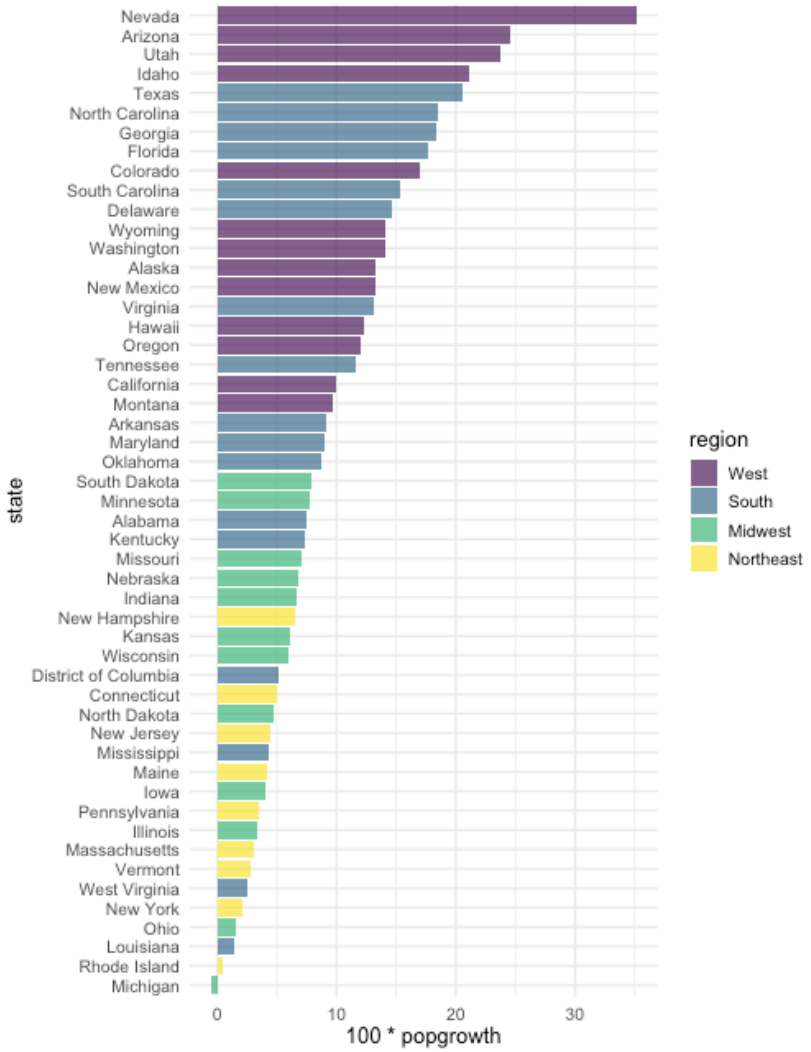


Problem with default palette

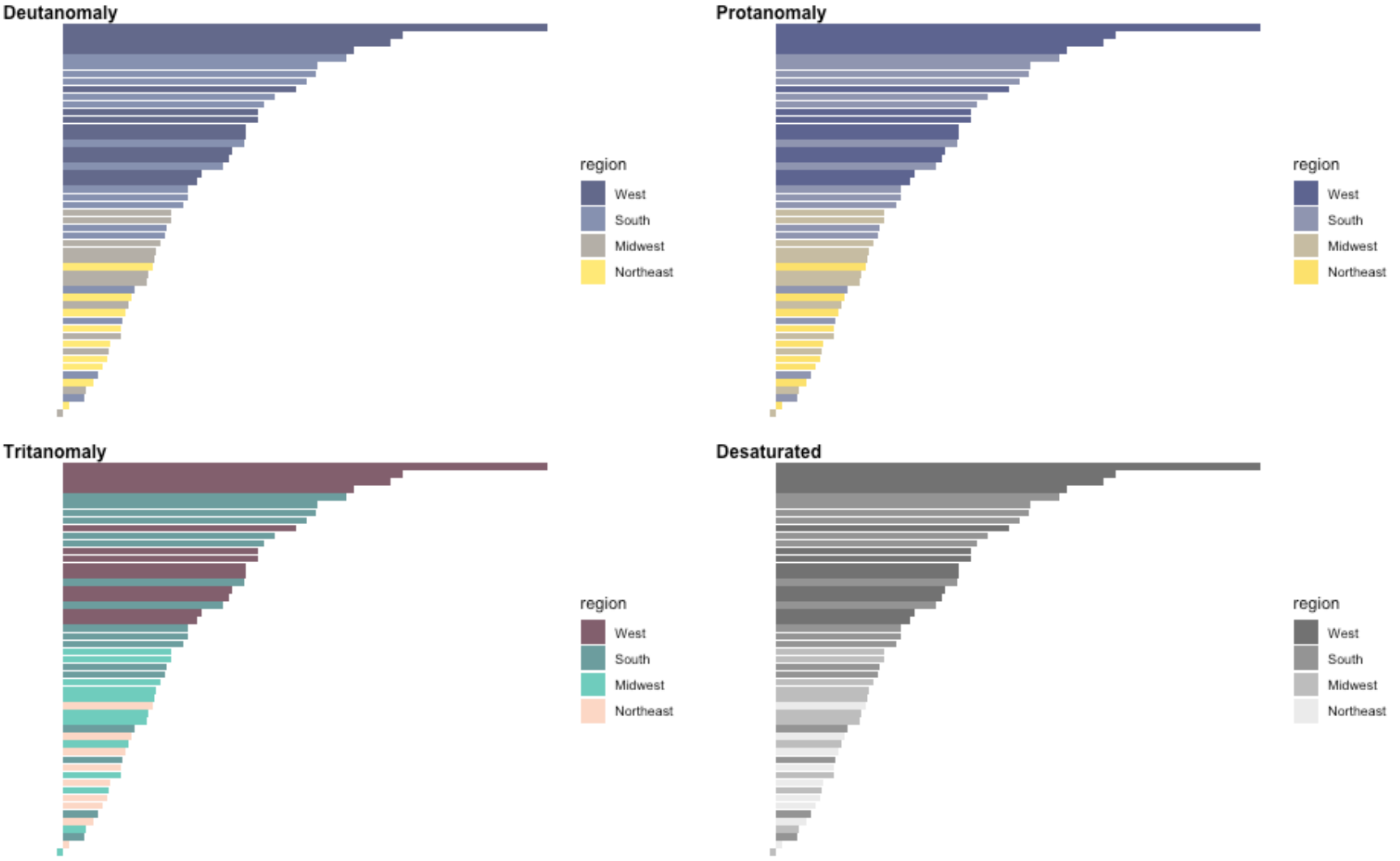


Alternative: viridis

```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region),
           alpha = 0.7) +
  scale_fill_viridis_d() +
  coord_flip()
```



Revised version

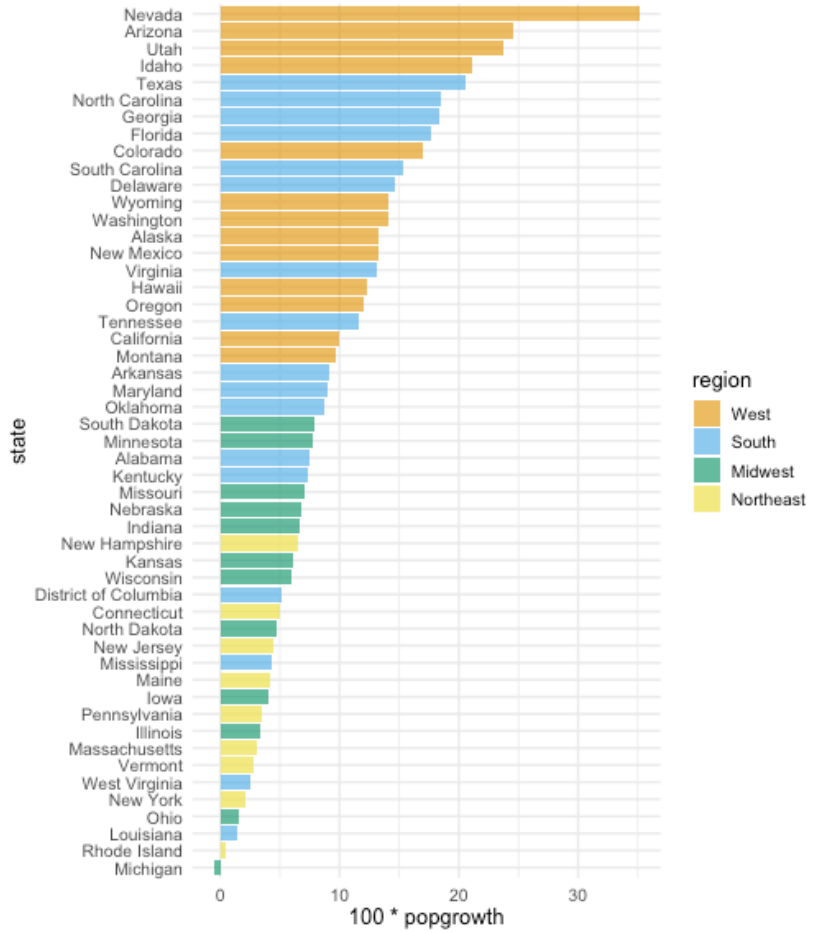


The Okabe Ito palette

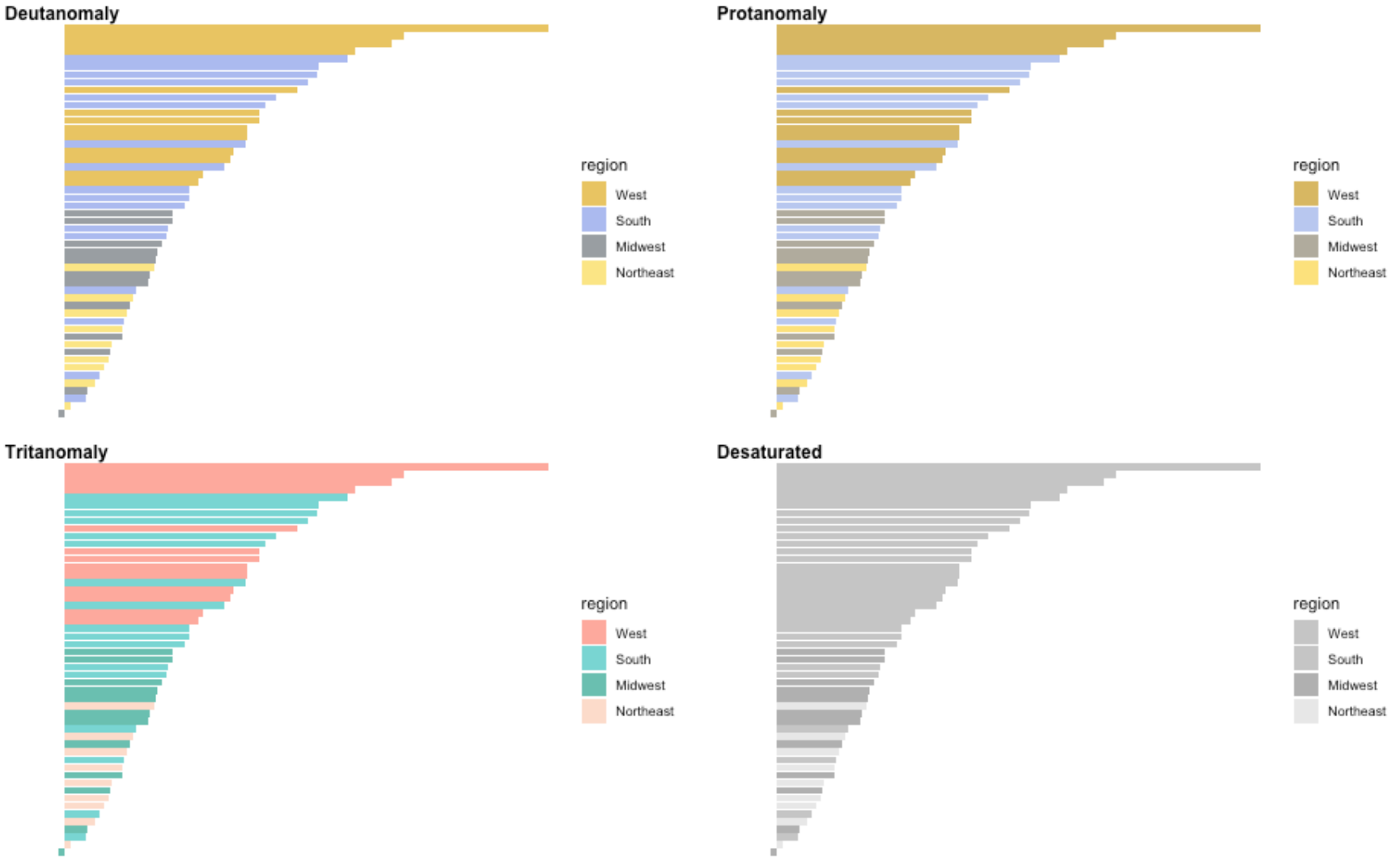
- From Color Universal Design

```
library(colorblindr)

ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region),
           alpha = 0.7) +
  scale_fill_OkabeIto() +
  coord_flip()
```



Okabe Ito for colorblindness



How am I checking for colorblindness?

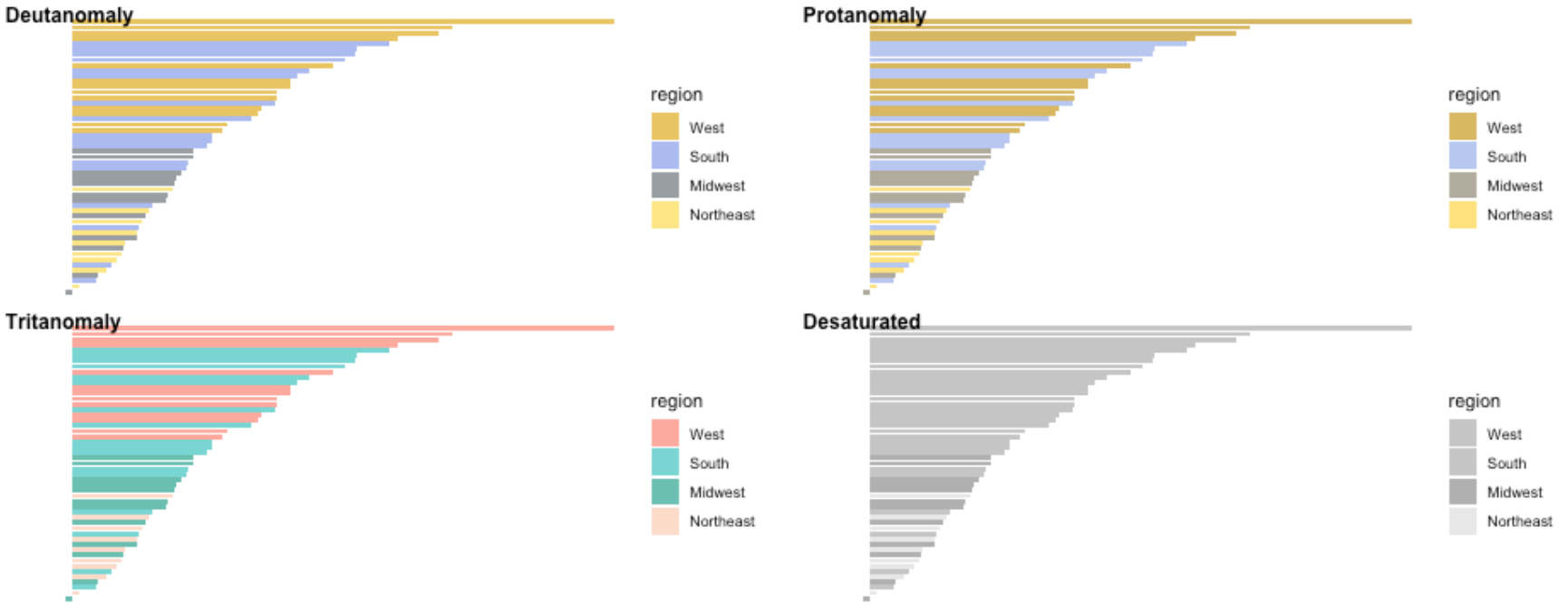
- Also part of the `colorblindr` package (here)
  - depends on the dev versions of `colorspace` and `cowplot`, which are useful packages in their own right

```
devtools::install_github("wilkelab/cowplot")
install.packages("colorspace", repos = "http://R-Forge.R-project.org")

devtools::install_github("clauswilke/colorblindr")
```

```
p <- ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region),
           alpha = 0.7) +
  scale_fill_OkabeIto() +
  coord_flip()  +
  theme_void() # not necessary but I like it

colorblindr::cvd_grid(p)
```



Colors to represent continuous values

Sequential scale examples

ColorBrewer Blues

Heat

Viridis

Sequential scale representation

- Which values are larger/smaller

Sequential scale representation

- Which values are larger/smaller
- How distant two values are from each other

Sequential scale representation

- Which values are larger/smaller
- How distant two values are from each other
  - Scale must be perceptually uniform across its entire range

Sequential scale representation

- Which values are larger/smaller
- How distant two values are from each other
  - Scale must be perceptually uniform across its entire range
  - Similar to an interval scale, but for color

Sequential scale representation

- Which values are larger/smaller
- How distant two values are from each other
  - Scale must be perceptually uniform across its entire range
  - Similar to an interval scale, but for color
- Often based on a single **hue**

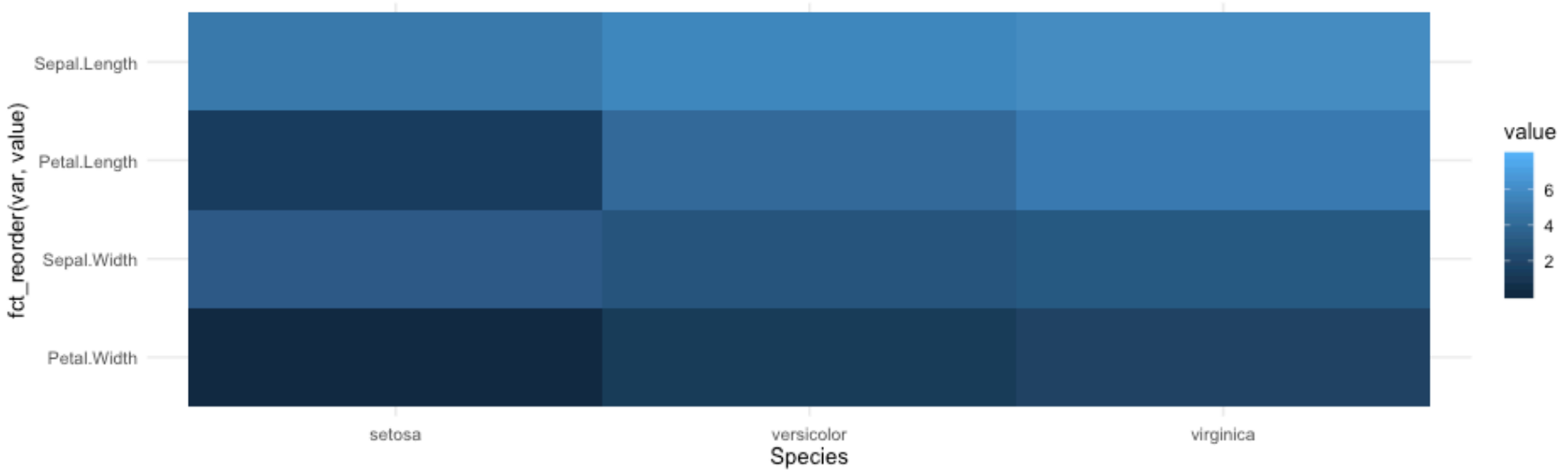
Sequential scale representation

- Which values are larger/smaller
- How distant two values are from each other
  - Scale must be perceptually uniform across its entire range
  - Similar to an interval scale, but for color
- Often based on a single **hue**
- Multi-hue sequential scales tend to follow gradients in the natural world

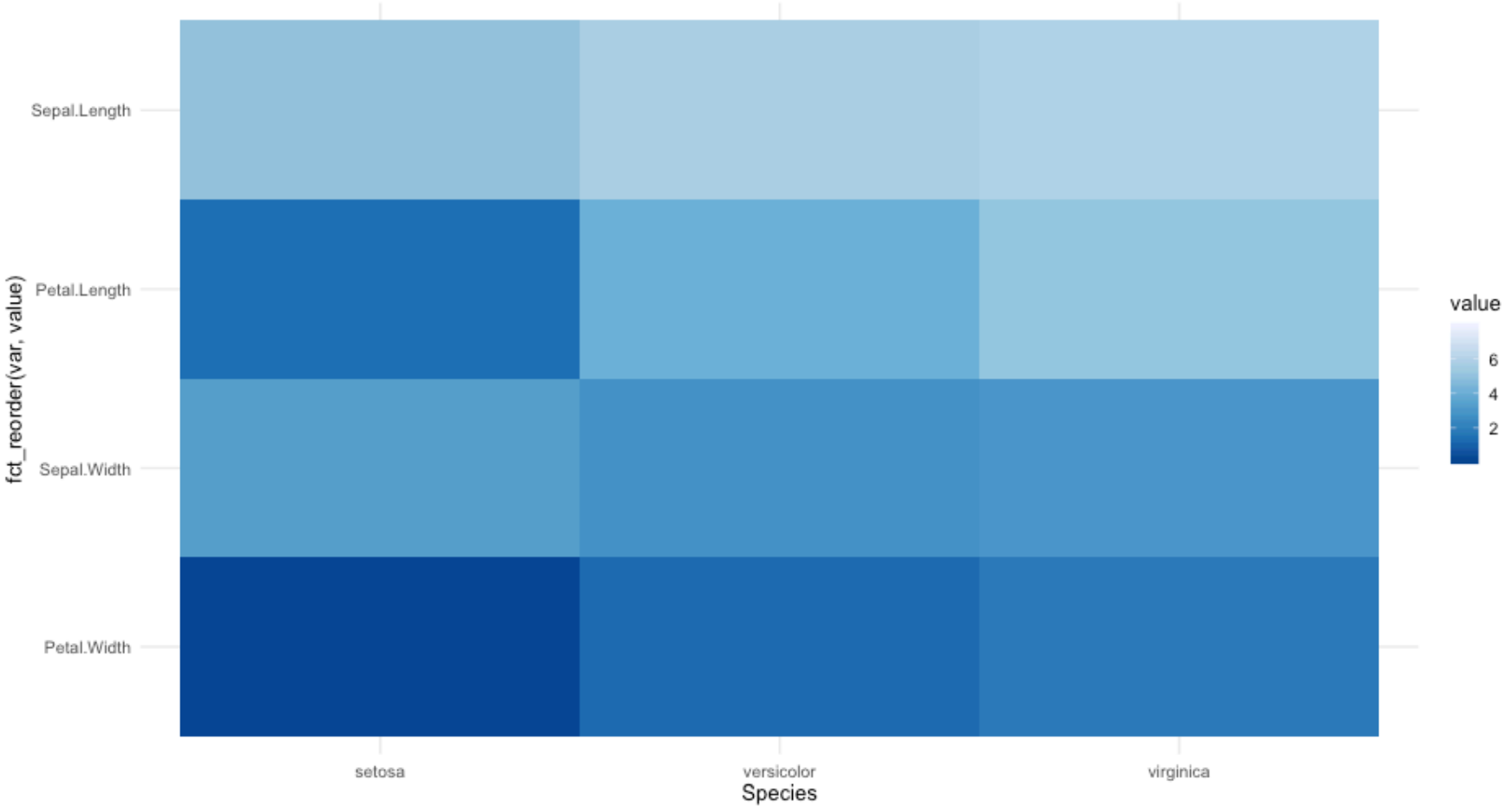
Common use of sequential palettes

Heatmaps

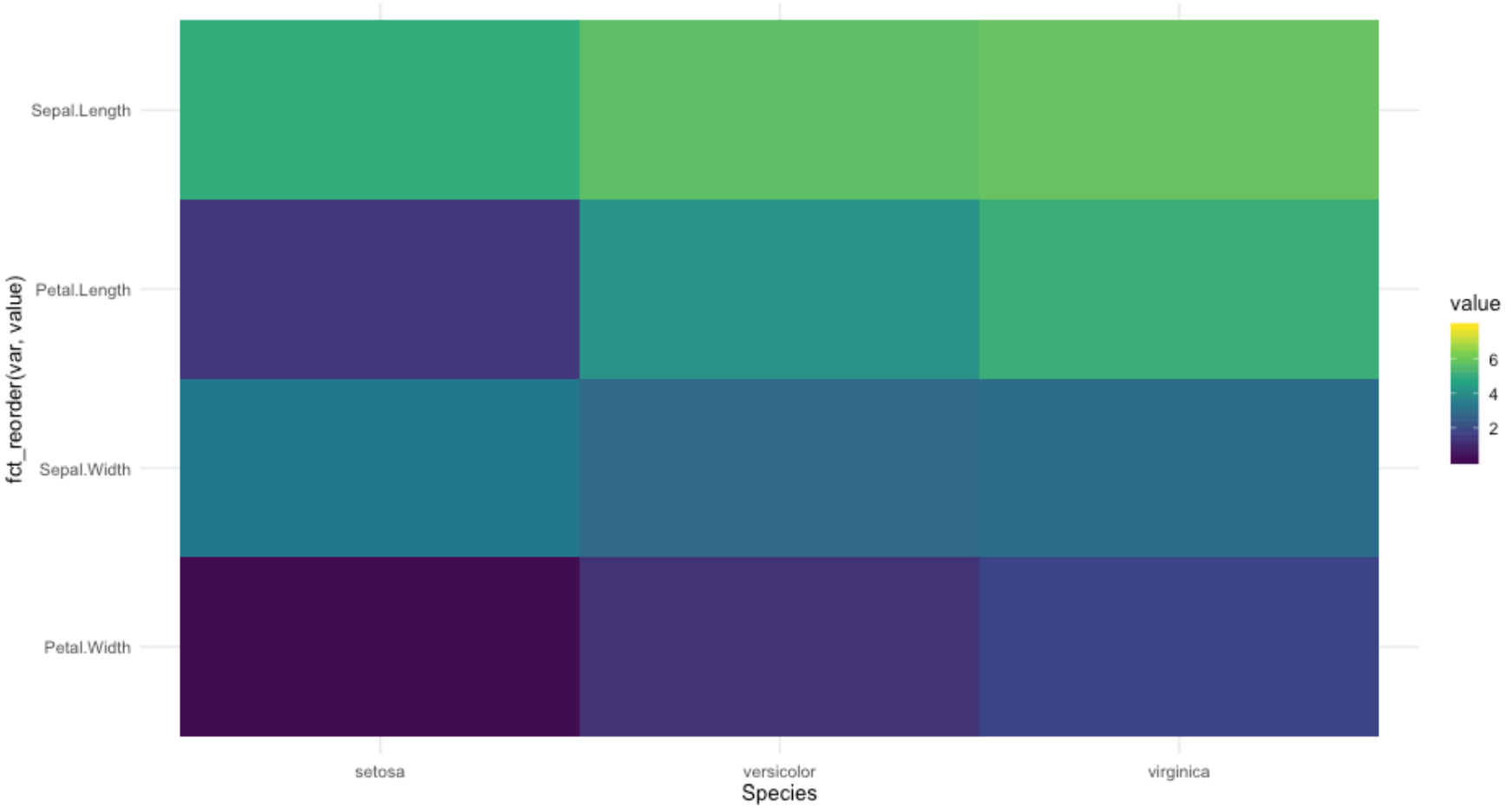
```
iris %>%
    gather(var, value, -Species) %>%
ggplot(aes(Species, fct_reorder(var, value))) +
    geom_tile(aes(fill = value))
```



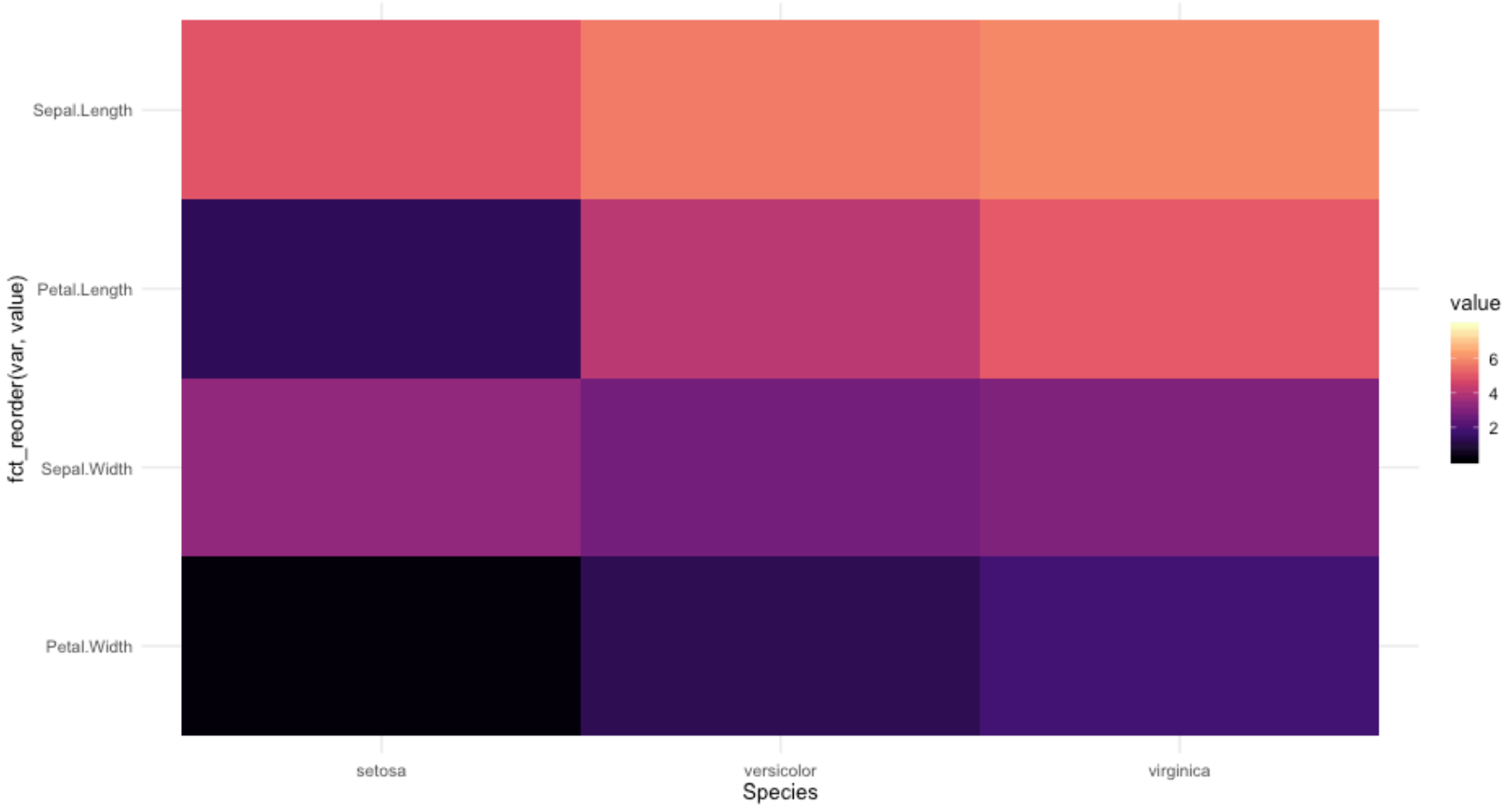
```
iris %>%
    gather(var, value, -Species) %>%
ggplot(aes(Species, fct_reorder(var, value))) +
    geom_tile(aes(fill = value)) +
scale_fill_distiller(palette = "Blues")
```



```
iris %>%
    gather(var, value, -Species) %>%
ggplot(aes(Species, fct_reorder(var, value))) +
    geom_tile(aes(fill = value)) +
scale_fill_viridis_c()
```



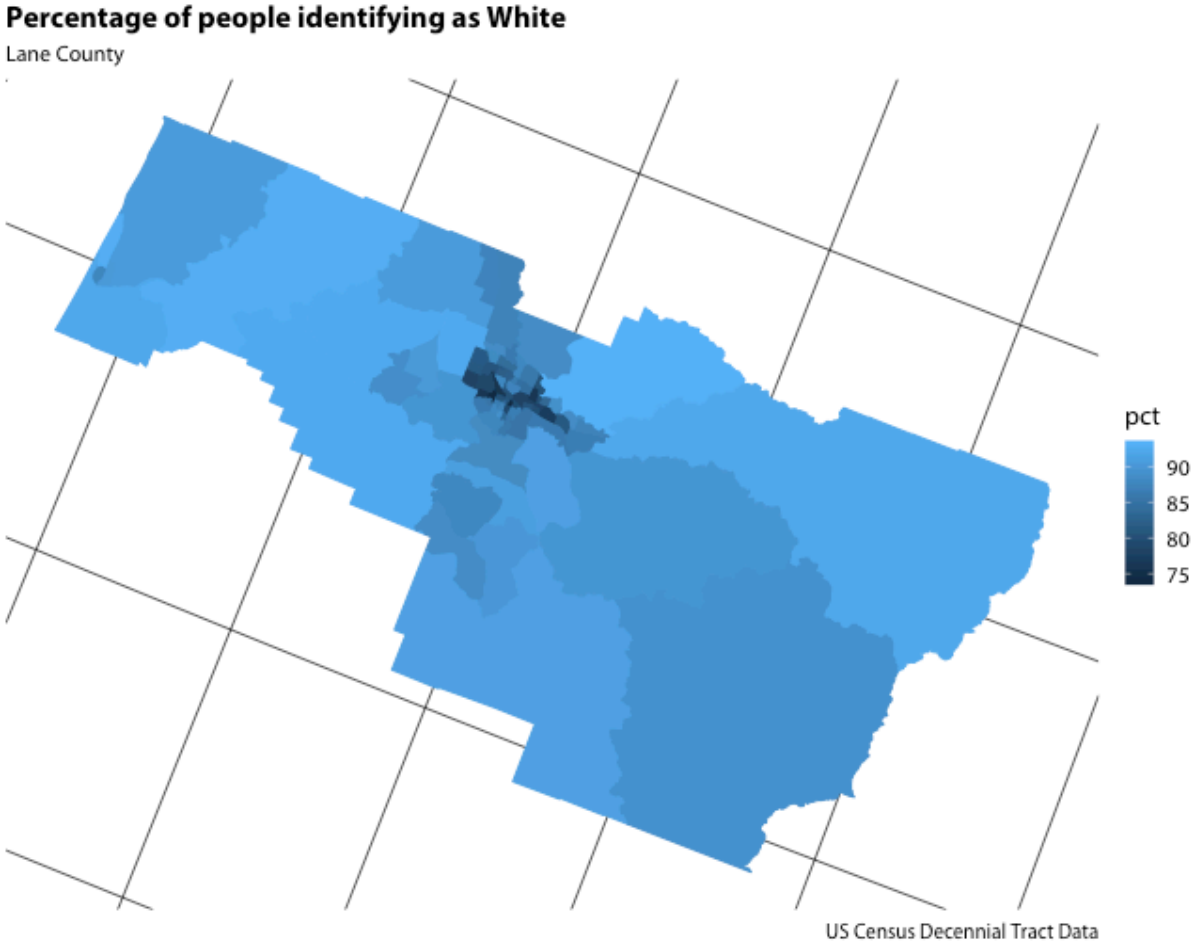
```
iris %>%
    gather(var, value, -Species) %>%
ggplot(aes(Species, fct_reorder(var, value))) +
    geom_tile(aes(fill = value)) +
scale_fill_viridis_c(option = "magma")
```



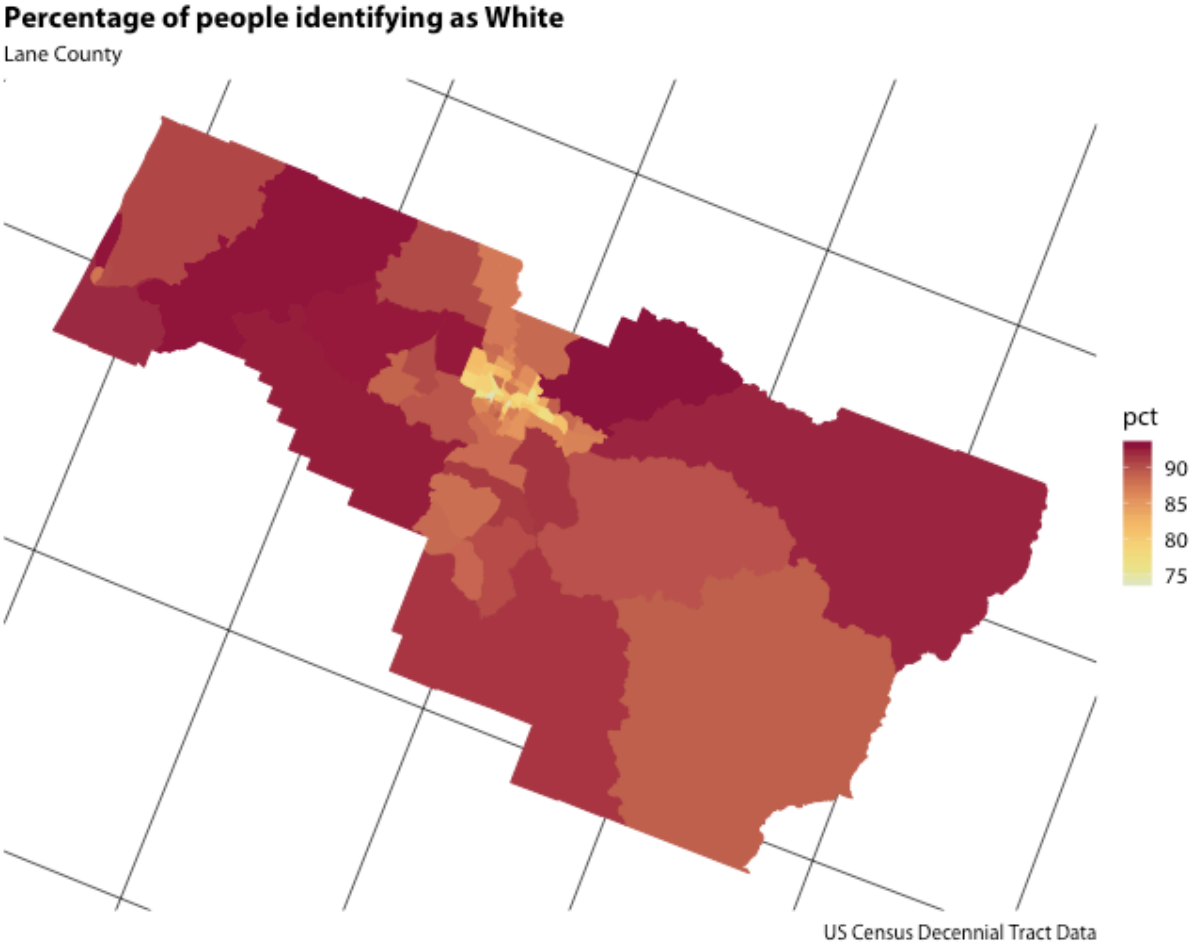
`option = c("viridis", "magma", "inferno", "plasma")`

Another common use

Choropleths



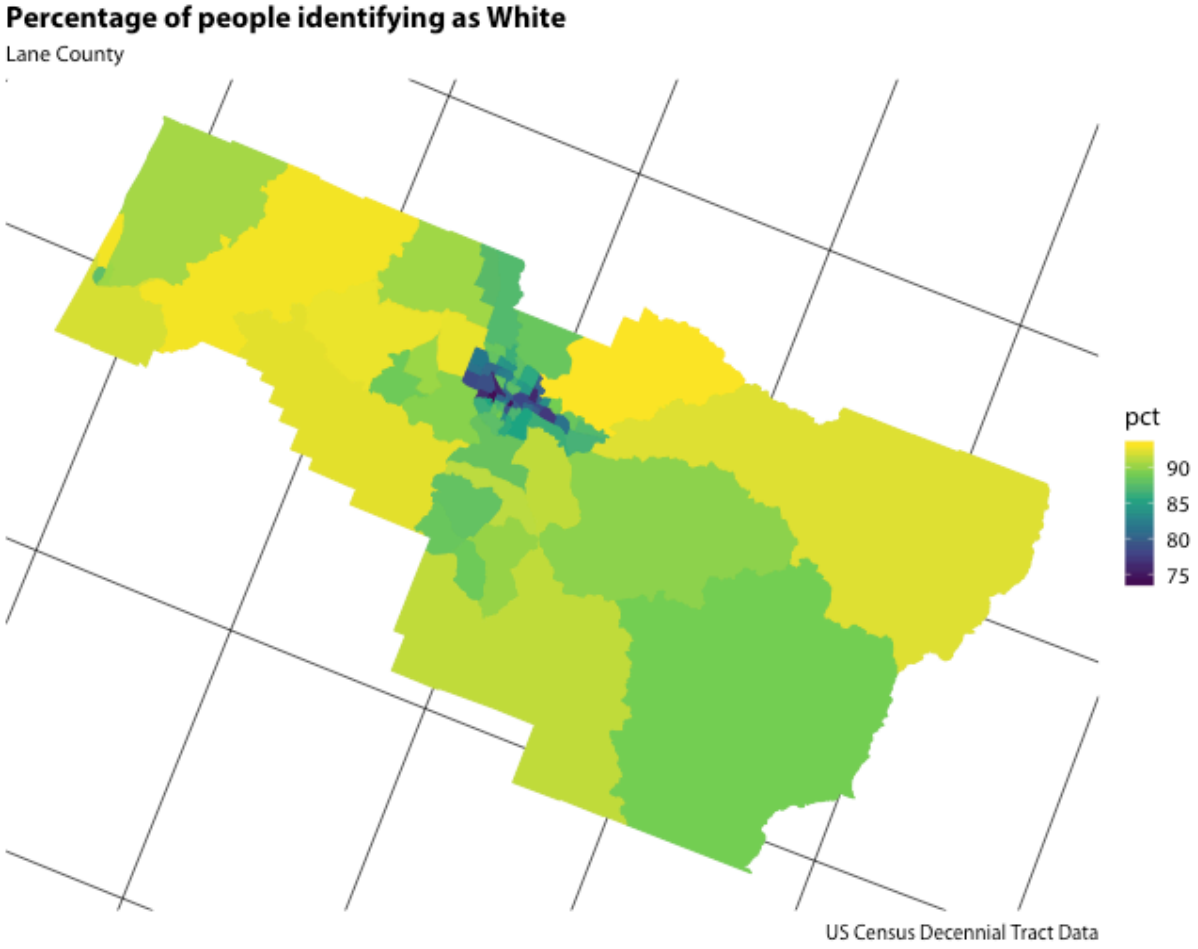
Heat palette



Options

- `scale_fill_continuous_sequential("Heat")`
- `scale_color_continuous_sequential("Heat")`
- `scale_fill_discrete_sequential("Heat")`
- `scale_color_discrete_sequential("Heat")`

viridis palette



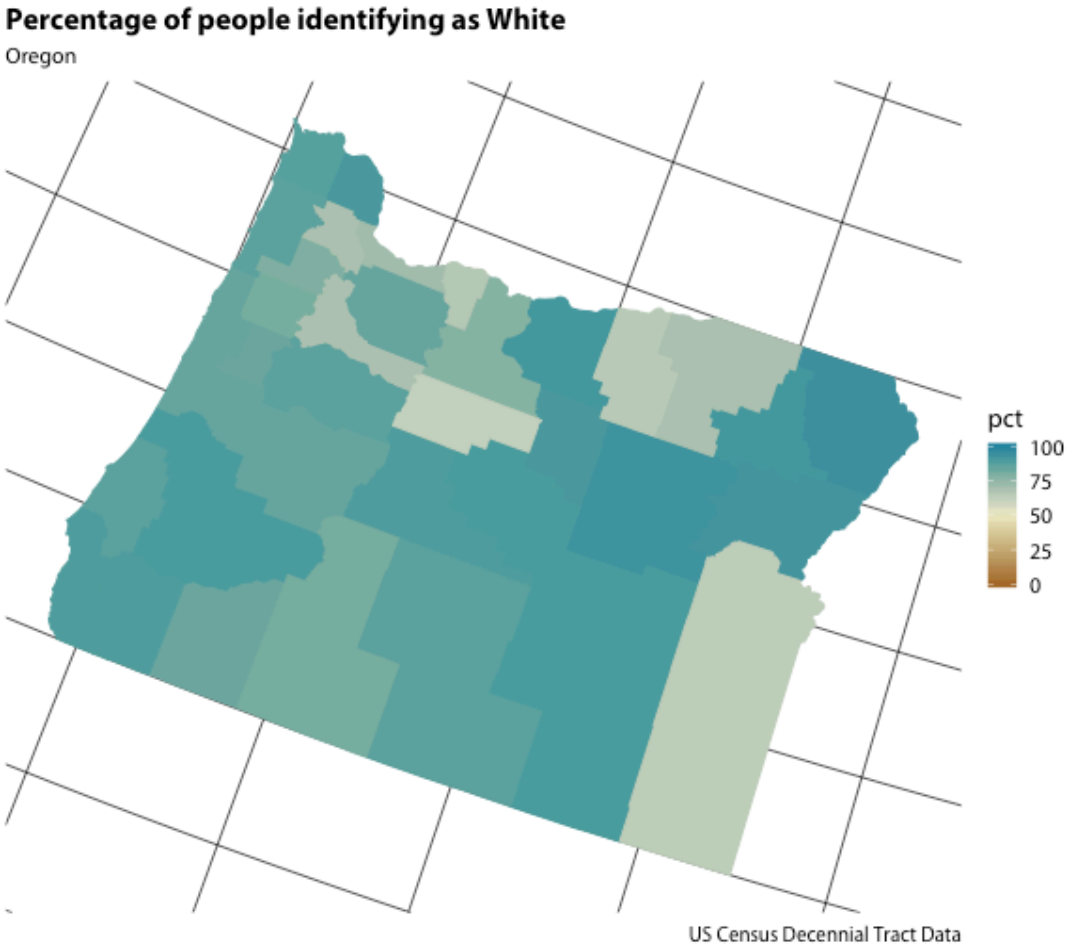
Diverging palettes

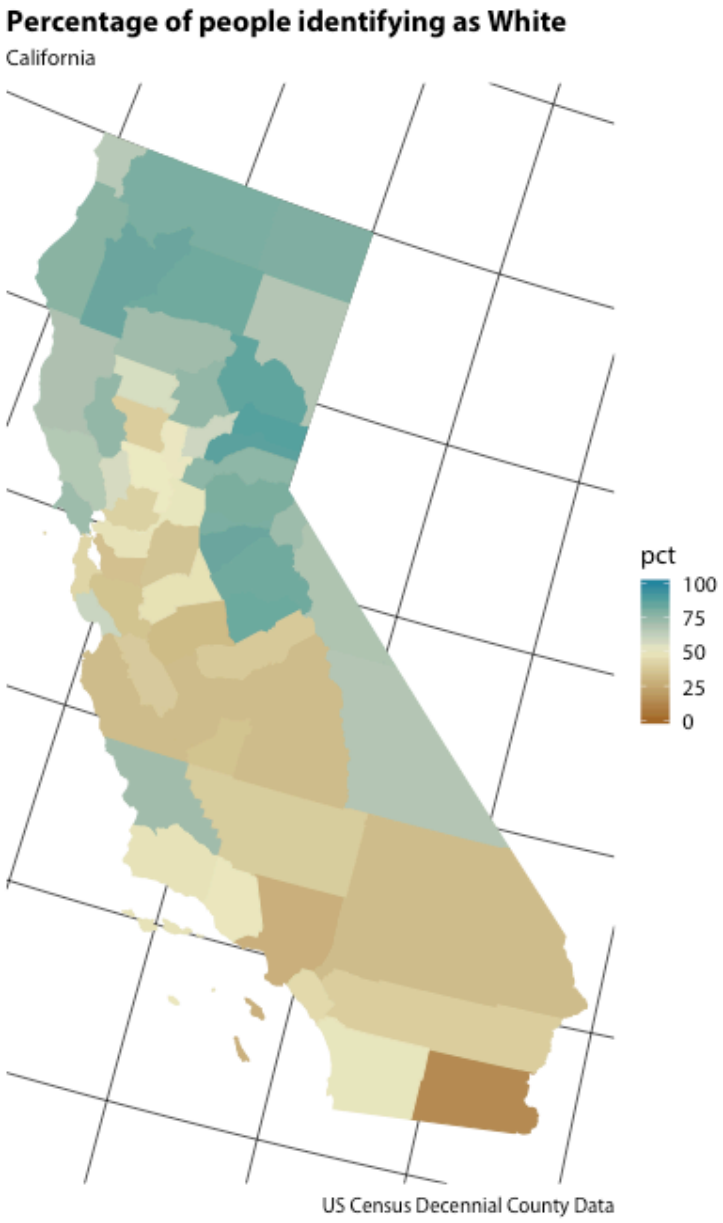
CARTO Earth

ColorBrewer PiYG

Blue-Red

Earth palette



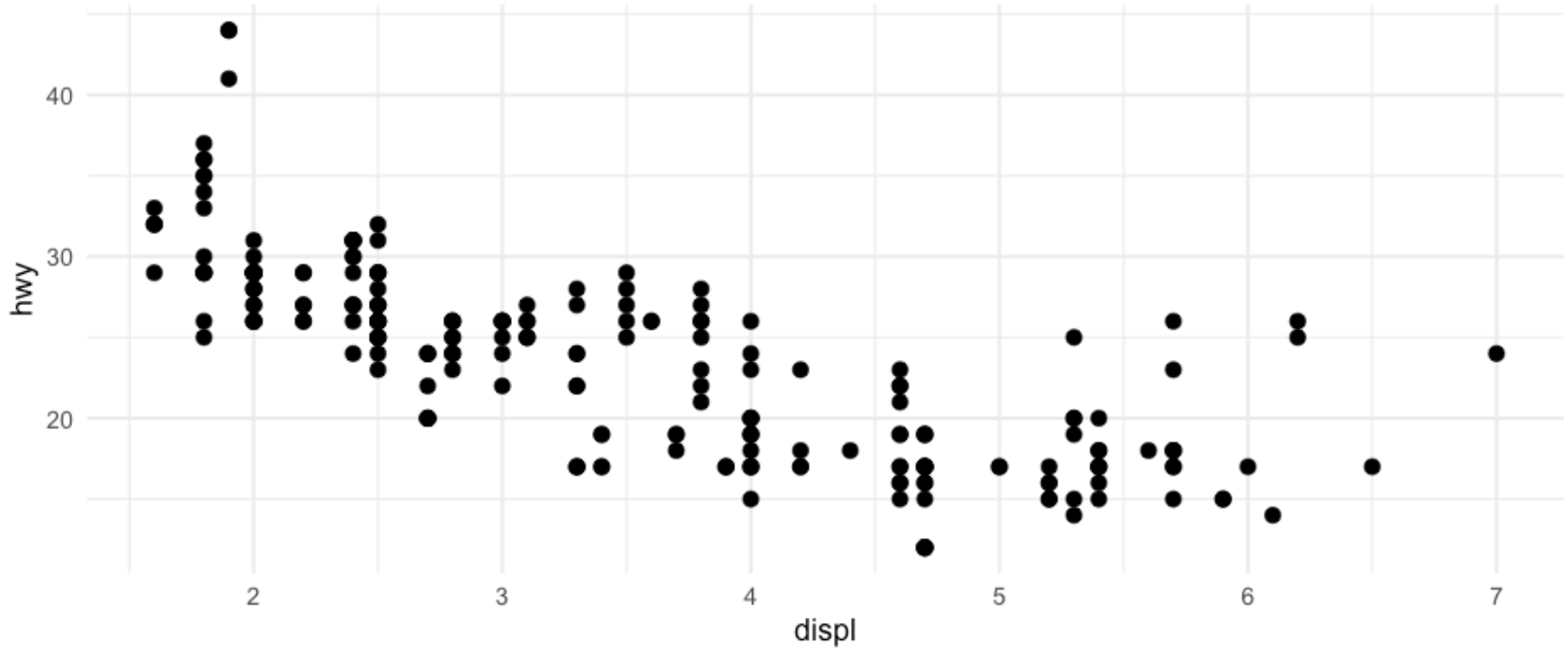

Percentage of people identifying as White
California
pct
100
75
50
25
0
US Census Decennial County Data

Color as a tool to highlight

MPG data

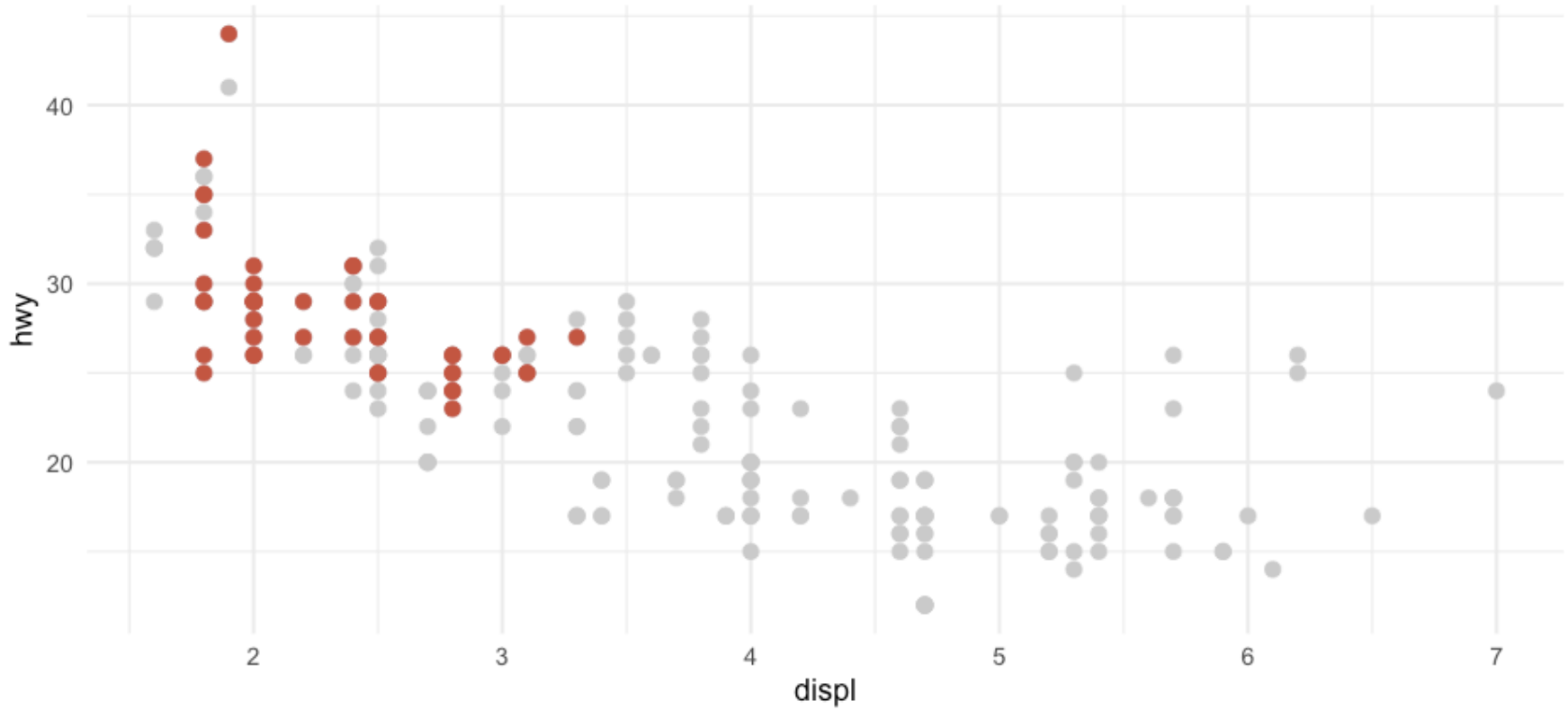
Basic scatterplot of weight to highway mpg

```
ggplot(mpg, aes(displ, hwy)) +
    geom_point()
```



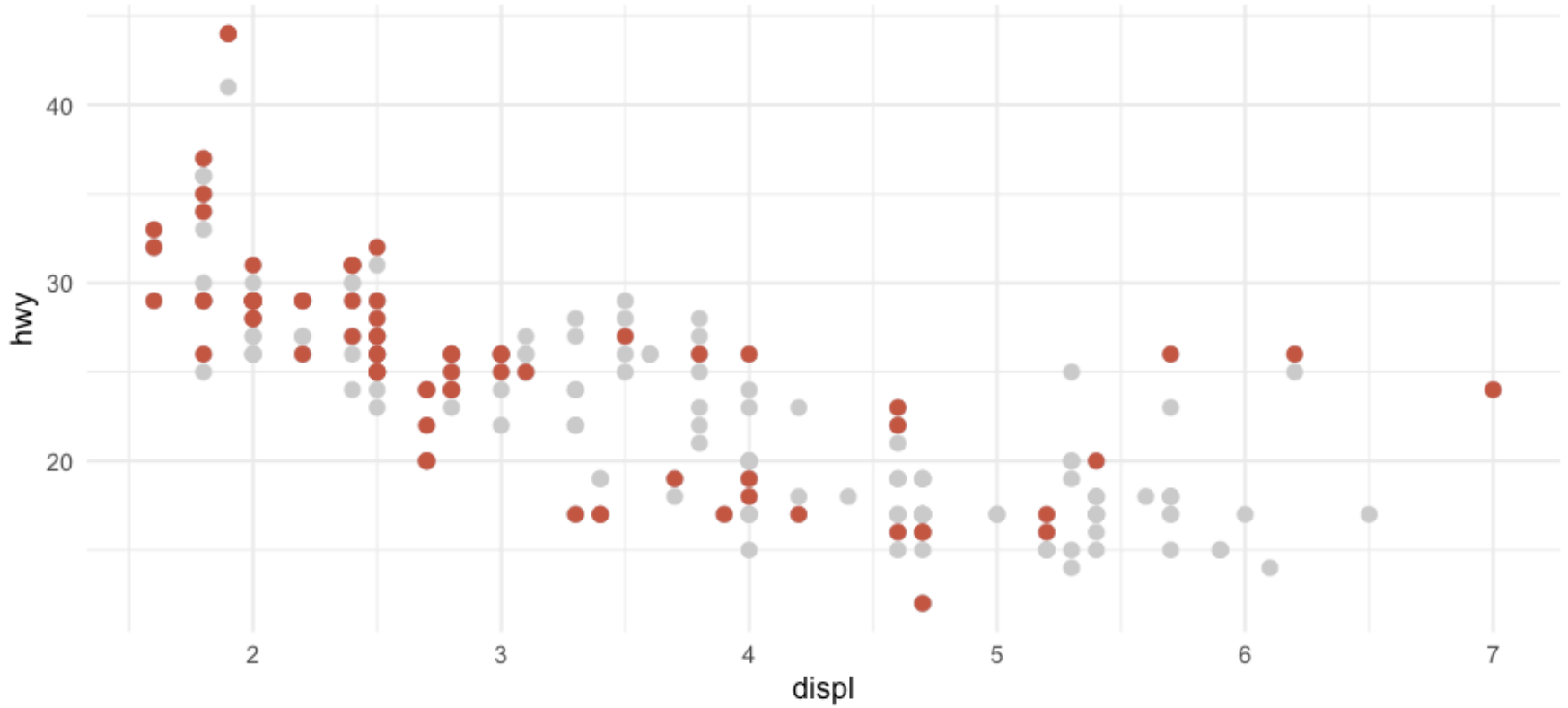
Highlight compact cars

```
ggplot(mpg, aes(displ, hwy)) +
    geom_point(color = "gray80") +
    geom_point(data = filter(mpg, class == "compact"),
               color = "#C55644")
```



Highlight manual cars

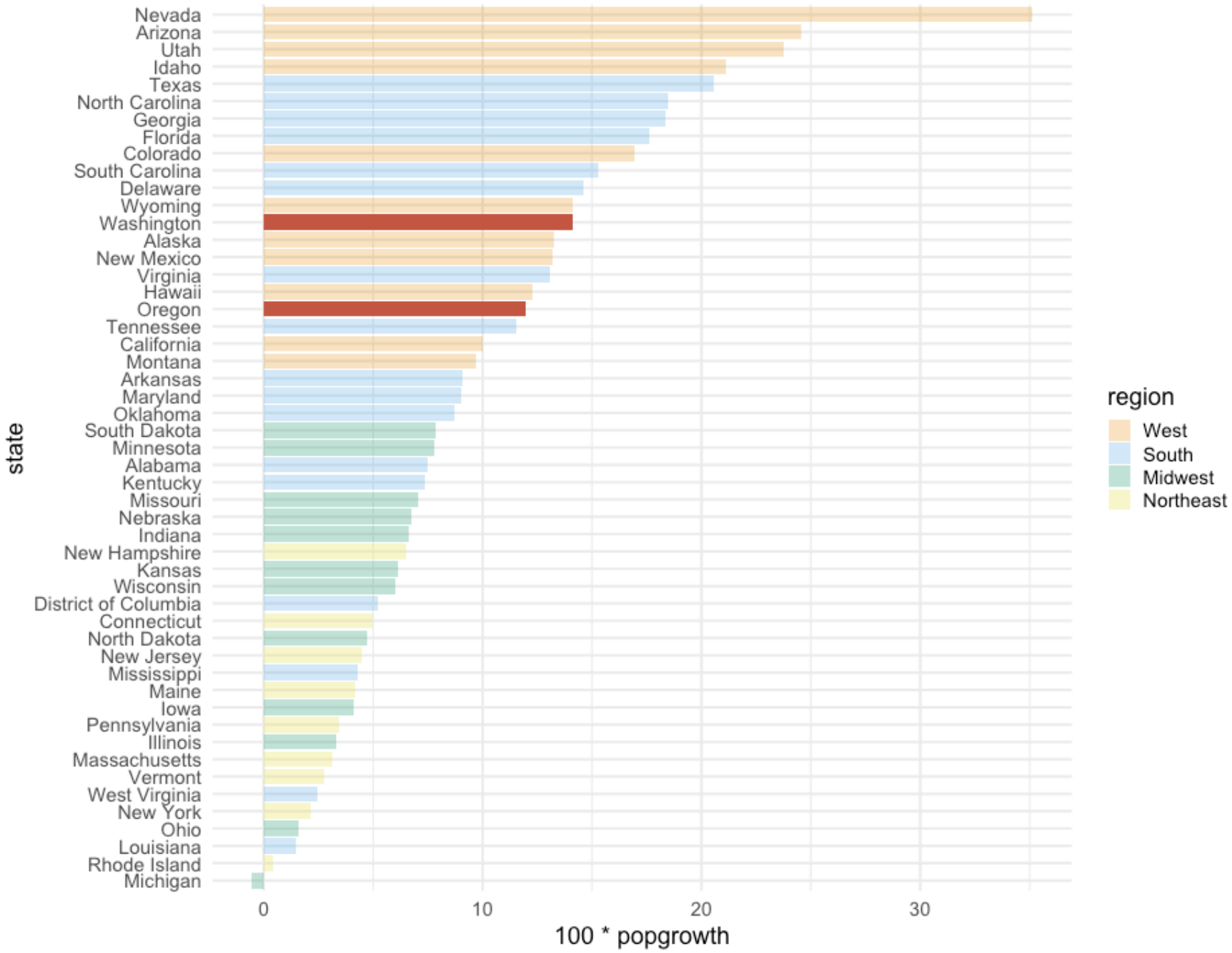
```
ggplot(mpg, aes(displ, hwy)) +
    geom_point(color = "gray80") +
    geom_point(data = filter(mpg, str_detect(trans, "manual")),
               color = "#C55644")
```



Back to our states plot

Highlight Oregon and Washington

```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region),
           alpha = 0.3) +
    geom_col(data = filter(popgrowth_df,
                           state == "Oregon" |
                           state == "Washington"),
           fill = "#C55644") +
  scale_fill_OkabeIto() +
  coord_flip()
```


Nevada
Arizona
Utah
Idaho
Texas
North Carolina
Georgia
Florida
Colorado
South Carolina
Delaware
Wyoming
Washington
Alaska
New Mexico
Virginia
Hawaii
Oregon
Tennessee
California
Montana
Arkansas
Maryland
Oklahoma
South Dakota
Minnesota
Alabama
Kentucky
Missouri
Nebraska
Indiana
New Hampshire
Kansas
Wisconsin
District of Columbia
Connecticut
North Dakota
New Jersey
Mississippi
Maine
Iowa
Pennsylvania
Illinois
Massachusetts
Vermont
West Virginia
New York
Ohio
Louisiana
Rhode Island
Michigan
state
0
10
20
30
100 * popgrowth
region
West
South
Midwest
Northeast

Color labels

```
states <- unique(popgrowth_df$state)

label_color <- ifelse(states == "Oregon" | states == "Washington",
                "#C55644",
                "gray30")
label_color
```

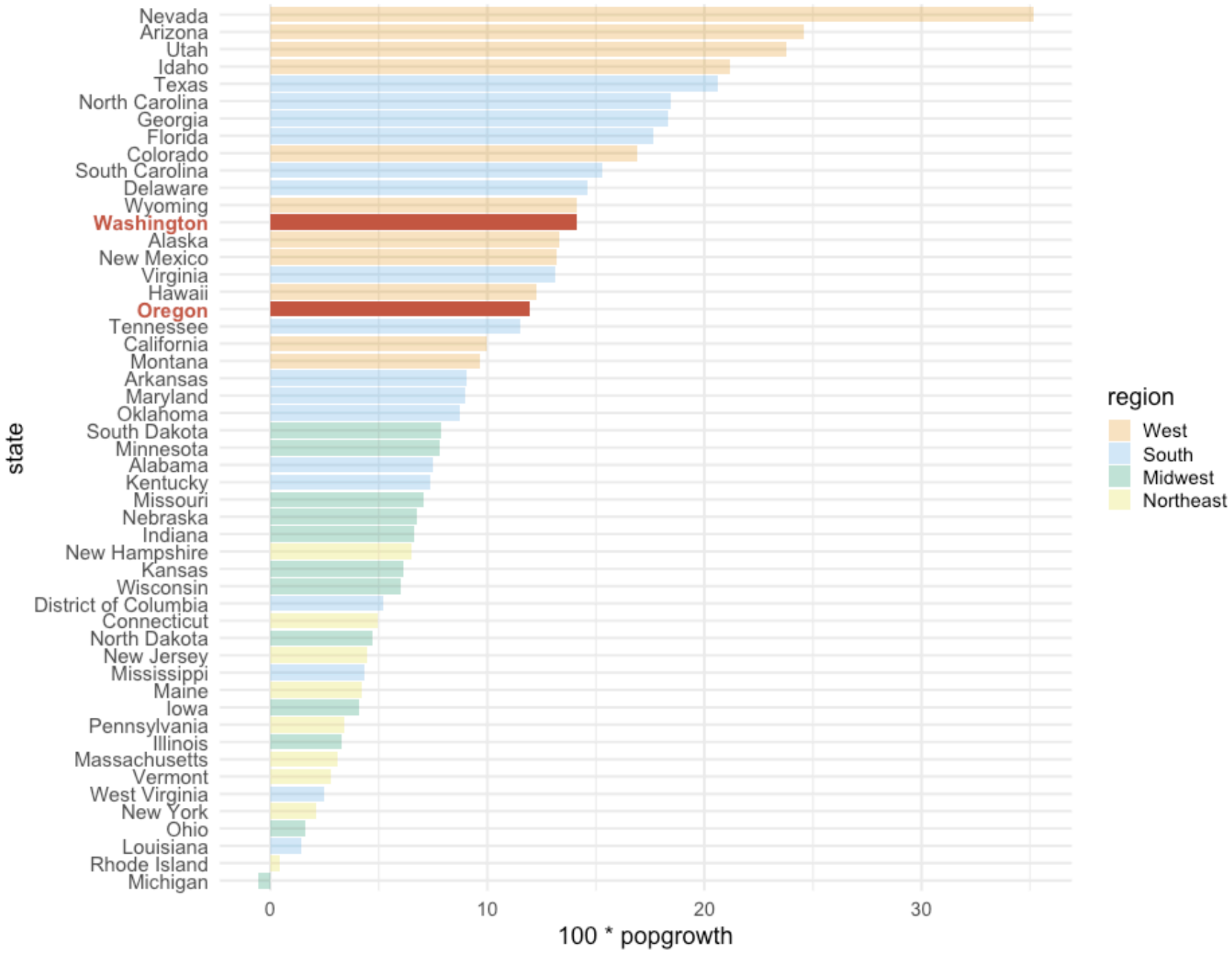
```
##  [1] "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"
##  [8] "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"
## [15] "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"
## [22] "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"
## [29] "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "#C55644" "gray30"
## [36] "gray30"  "gray30"  "gray30"  "#C55644" "gray30"  "gray30"  "gray30"
## [43] "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"  "gray30"
## [50] "gray30"  "gray30"
```

```
label_face <- ifelse(states == "Oregon" | states == "Washington",
                "bold",
                "plain")

label_face
```

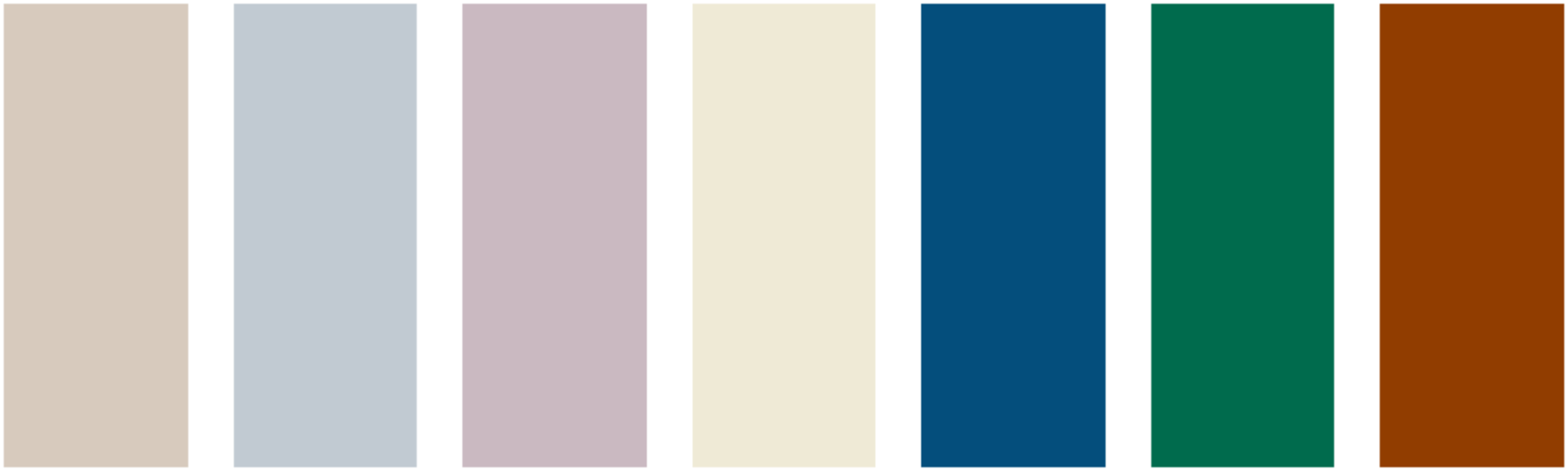
```
##  [1] "plain" "plain" "plain" "plain" "plain" "plain" "plain" "plain"
##  [9] "plain" "plain" "plain" "plain" "plain" "plain" "plain" "plain"
```

```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region),
           alpha = 0.3) +
    geom_col(data = filter(popgrowth_df,
                           state == "Oregon" |
                           state == "Washington"),
           fill = "#C55644") +
  scale_fill_OkabeIto() +
  coord_flip() +
  theme(axis.text.y = element_text(color = label_color,
                                   face = label_face))
```

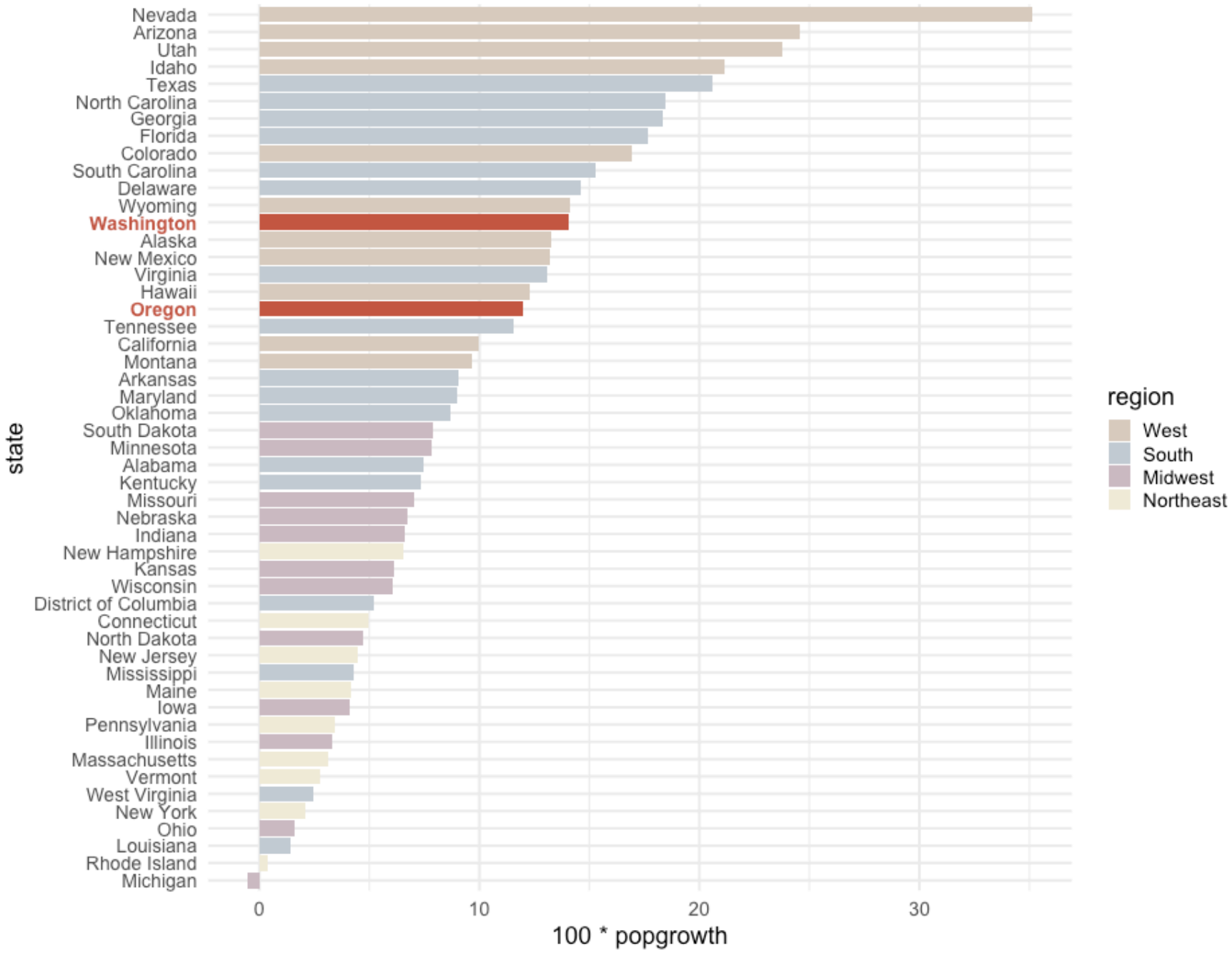

Nevada
Arizona
Utah
Idaho
Texas
North Carolina
Georgia
Florida
Colorado
South Carolina
Delaware
Wyoming
Washington
Alaska
New Mexico
Virginia
Hawaii
Oregon
Tennessee
California
Montana
Arkansas
Maryland
Oklahoma
South Dakota
Minnesota
Alabama
Kentucky
Missouri
Nebraska
Indiana
New Hampshire
Kansas
Wisconsin
District of Columbia
Connecticut
North Dakota
New Jersey
Mississippi
Maine
Iowa
Pennsylvania
Illinois
Massachusetts
Vermont
West Virginia
New York
Ohio
Louisiana
Rhode Island
Michigan
state
0
10
20
30
100 * popgrowth
region
West
South
Midwest
Northeast

Even better

```
accent_OkabeIto <- palette_OkabeIto[c(1, 2, 7, 4, 5, 3, 6)]
accent_OkabeIto[1:4] <- desaturate(lighten(accent_OkabeIto[1:4], .4), .8)
accent_OkabeIto[5:7] <- darken(accent_OkabeIto[5:7], .3)
gg_color_swatches(7) +
    scale_fill_manual(values = accent_OkabeIto)
```



```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region)) +
    geom_col(data = filter(popgrowth_df,
                           state == "Oregon" |
                           state == "Washington"),
           fill = "#C55644") +
  scale_fill_manual(values = accent_OkabeIto) +
  coord_flip() +
  theme(axis.text.y = element_text(color = label_color,
                                   face = label_face))
```


Nevada
Arizona
Utah
Idaho
Texas
North Carolina
Georgia
Florida
Colorado
South Carolina
Delaware
Wyoming
Washington
Alaska
New Mexico
Virginia
Hawaii
Oregon
Tennessee
California
Montana
Arkansas
Maryland
Oklahoma
South Dakota
Minnesota
Alabama
Kentucky
Missouri
Nebraska
Indiana
New Hampshire
Kansas
Wisconsin
District of Columbia
Connecticut
North Dakota
New Jersey
Mississippi
Maine
Iowa
Pennsylvania
Illinois
Massachusetts
Vermont
West Virginia
New York
Ohio
Louisiana
Rhode Island
Michigan
state
0
10
20
30
100 * popgrowth
region
West
South
Midwest
Northeast

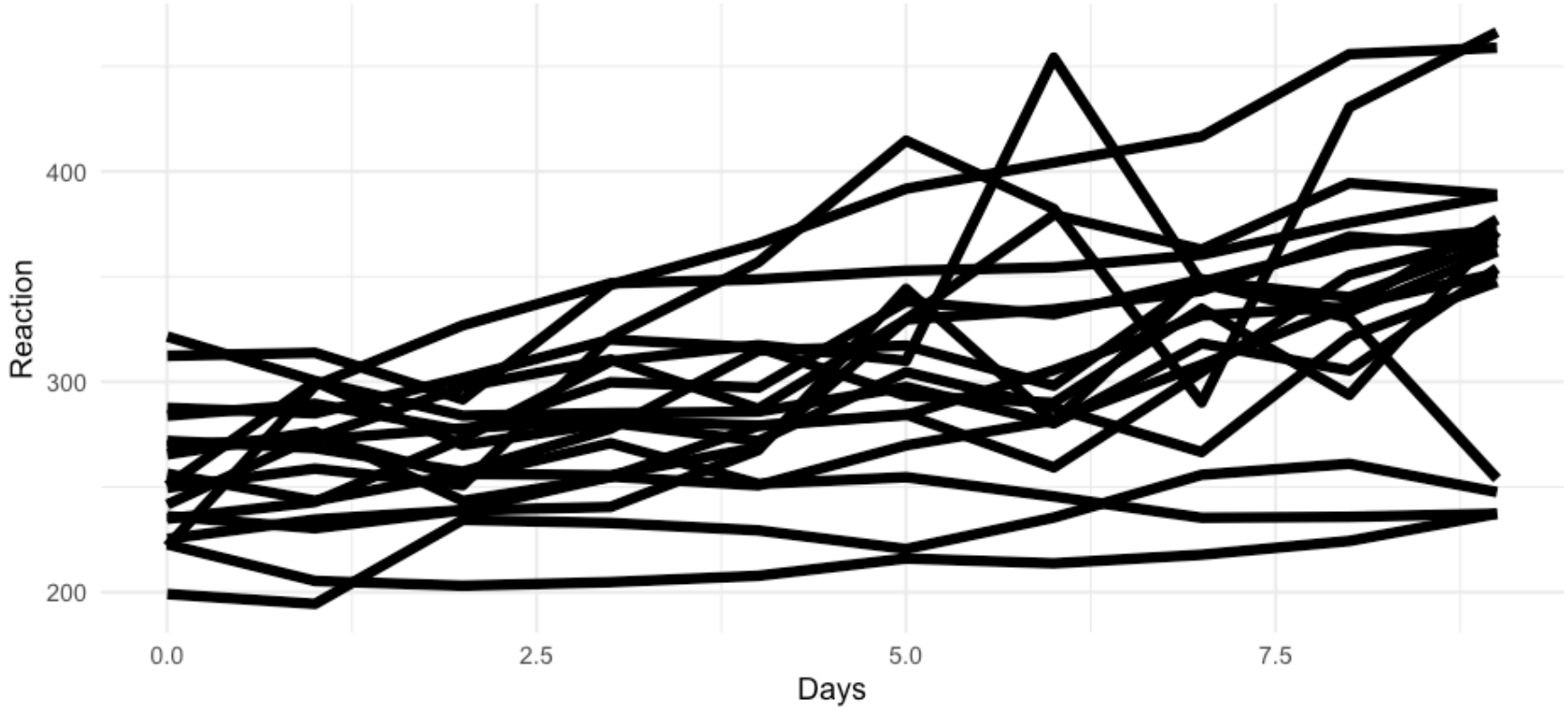
Last example

```
data(sleepstudy, package = "lme4")
head(sleepstudy)
```

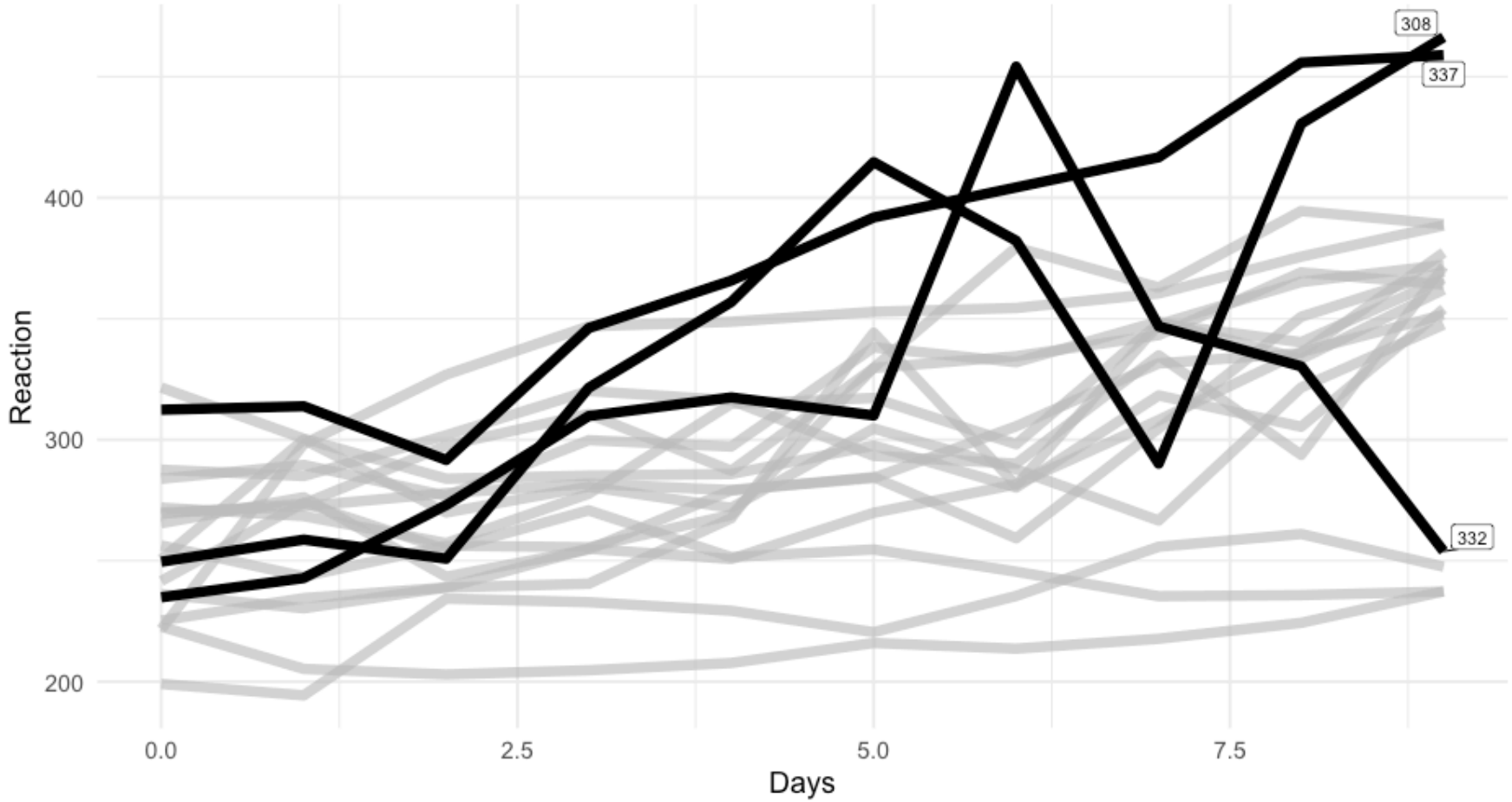
```
##   Reaction Days Subject
## 1      250    0     308
## 2      259    1     308
## 3      251    2     308
## 4      321    3     308
## 5      357    4     308
## 6      415    5     308
```

Plot by subject

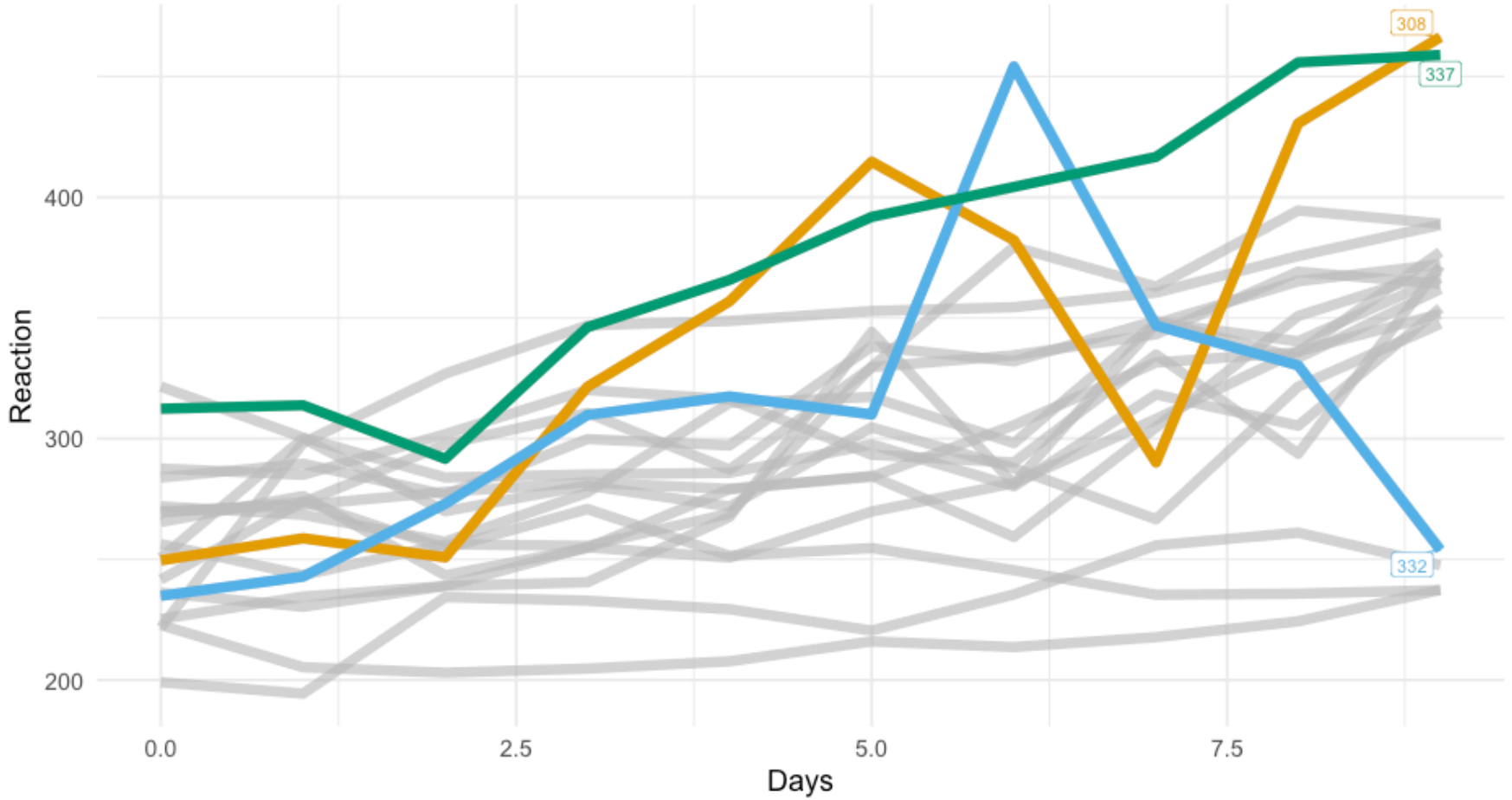
```
ggplot(sleepstudy, aes(Days, Reaction, group = Subject)) +
    geom_line()
```



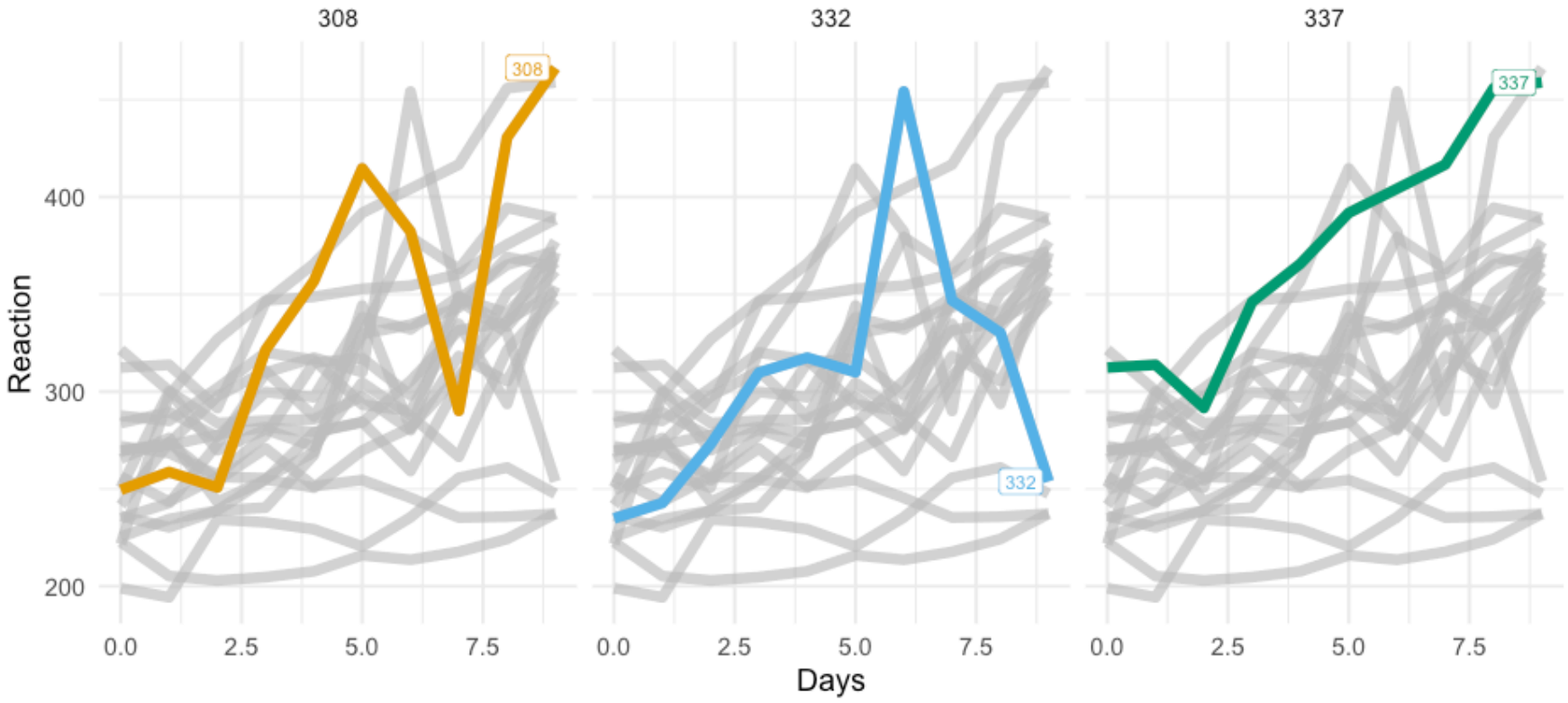
```r
library(gghighlight)
ggplot(sleepstudy, aes(Days, Reaction, group = Subject)) +
    geom_line() +
    gghighlight(max(Reaction) > 400)
```



```
library(gghighlight)
ggplot(sleepstudy, aes(Days, Reaction, color = Subject)) +
    geom_line() +
gghighlight(max(Reaction) > 400) +
    scale_color_OkabeIto()
```

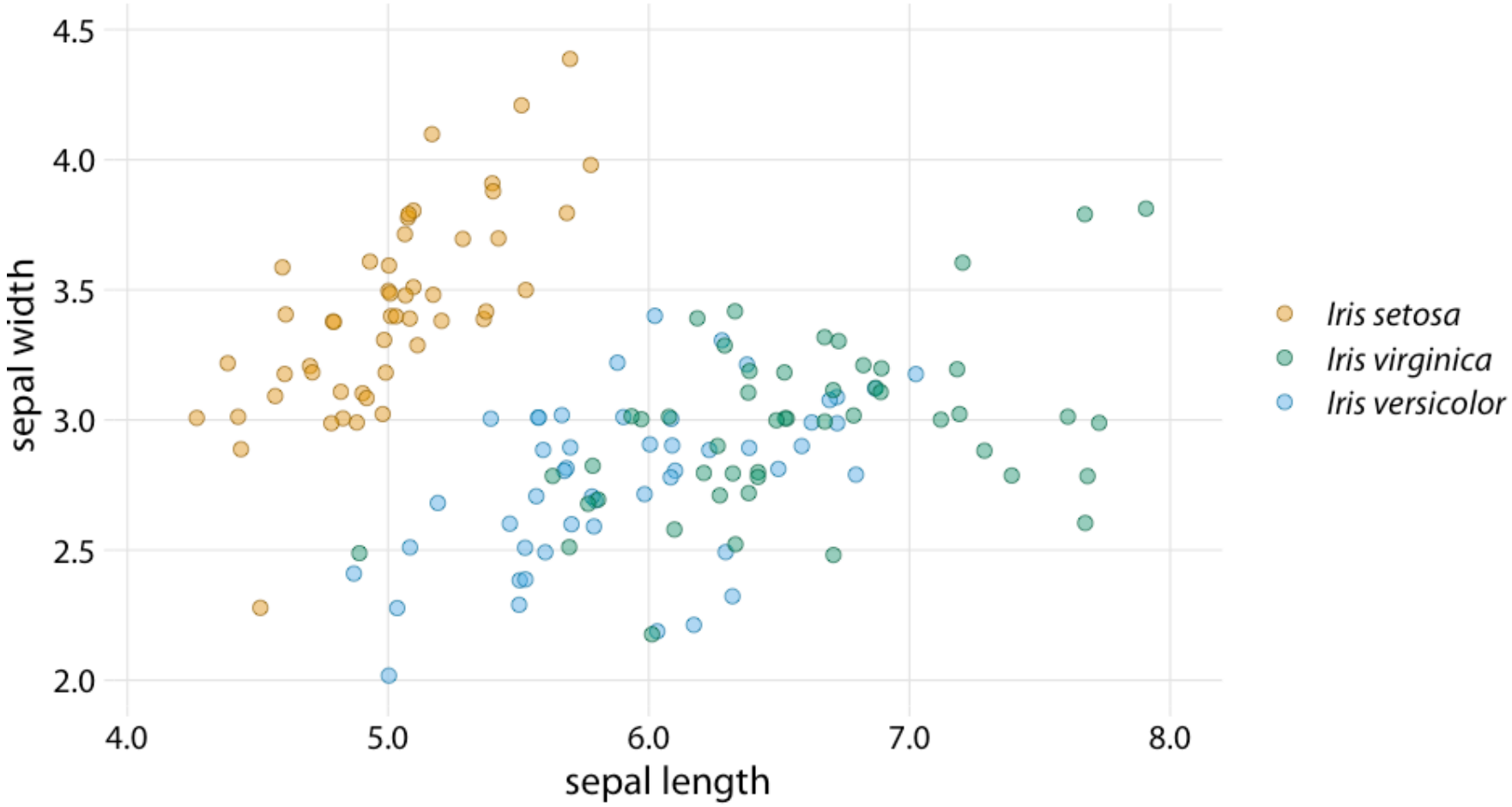


```
library(gghighlight)
ggplot(sleepstudy, aes(Days, Reaction, color = Subject)) +
    geom_line() +
facet_wrap(~Subject) +
gghighlight(max(Reaction) > 400) +
    scale_color_OkabeIto()
```

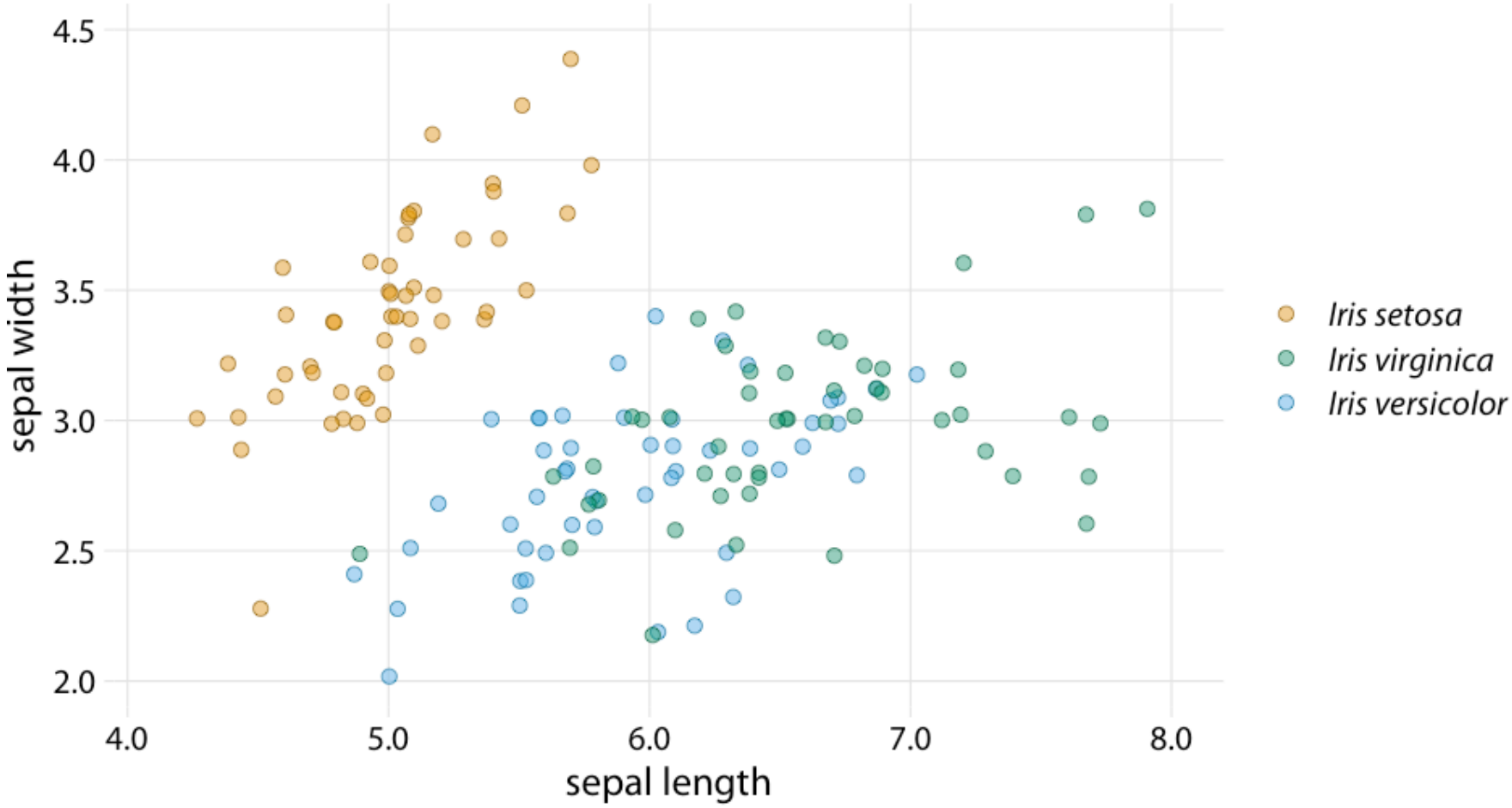


A few other things to consider

Double encodings



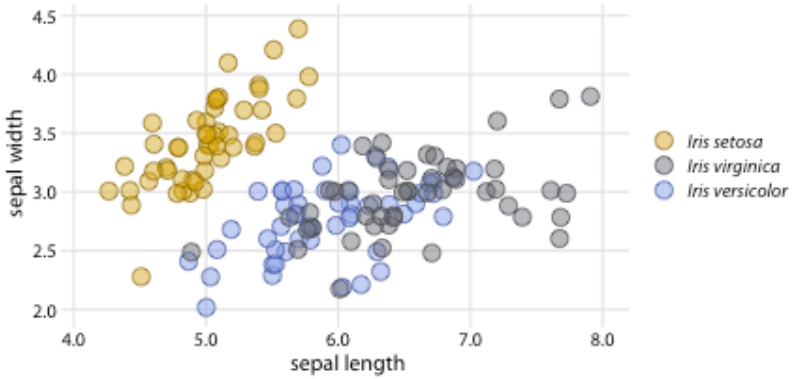
Double encodings



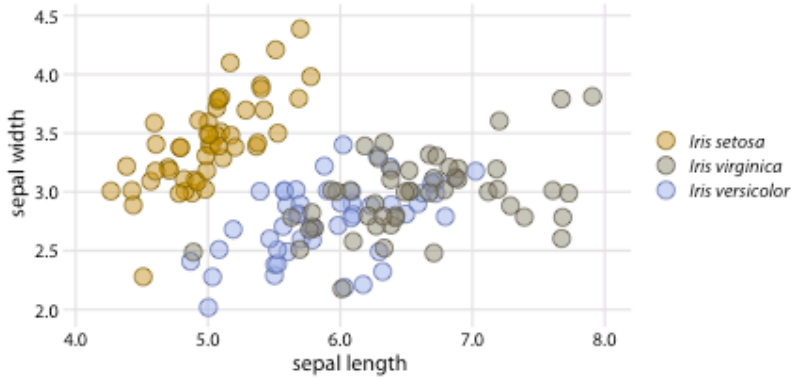
This plot is less than ideal. Why?

Color blindness

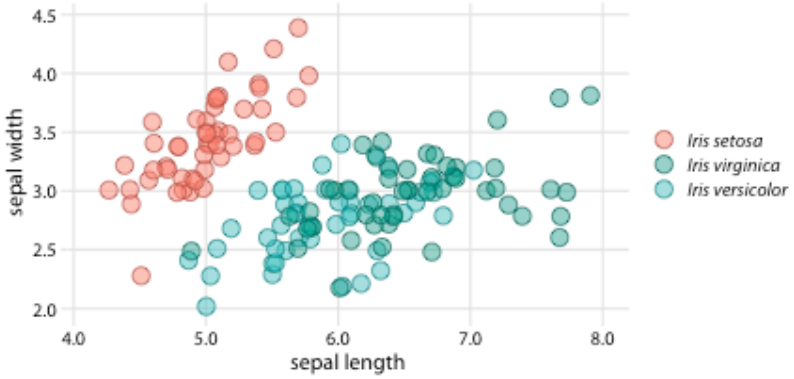
Deutanomaly



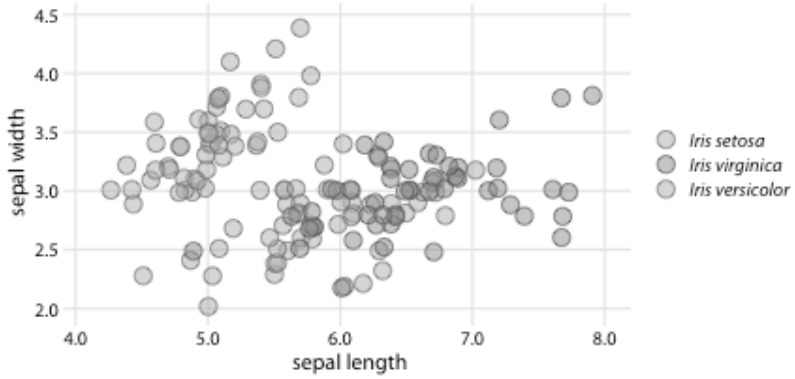
Protanomaly



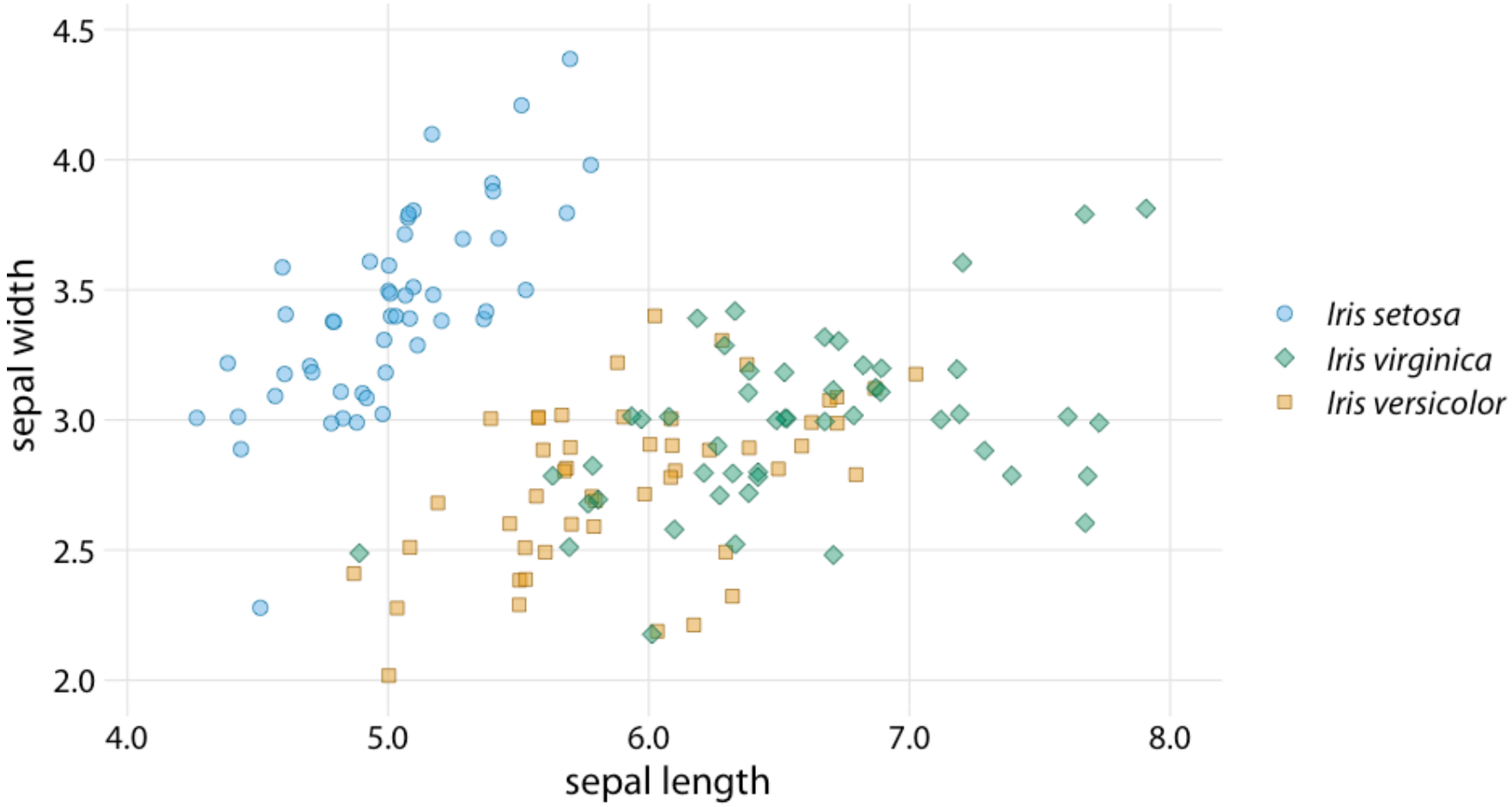
Tritanomaly



Desaturated

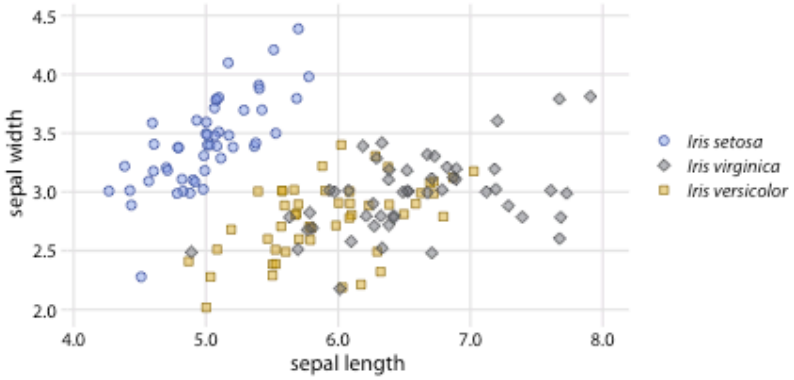


Better version

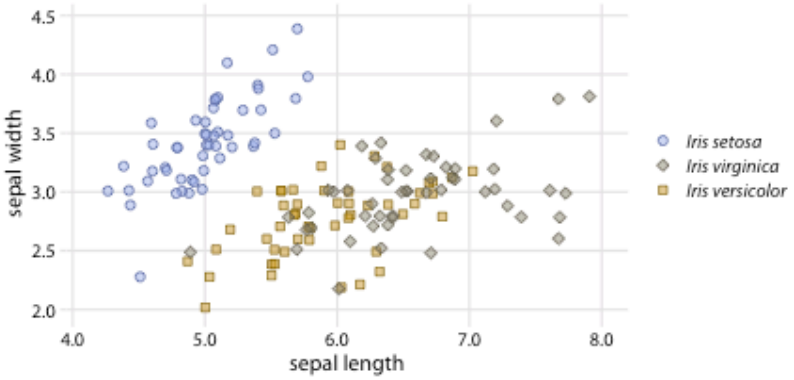


Color blindness check

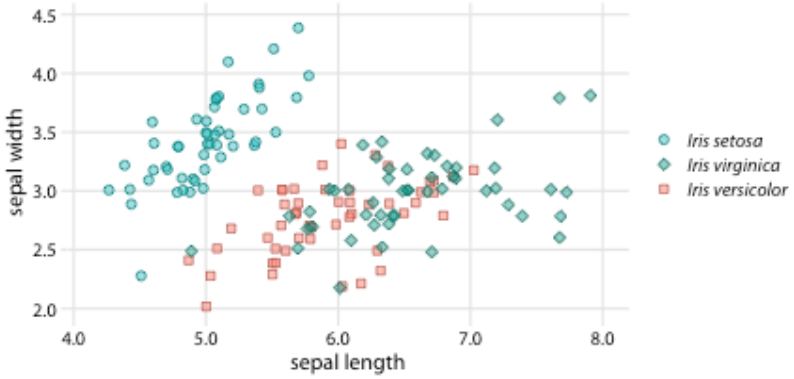
deuteranomaly



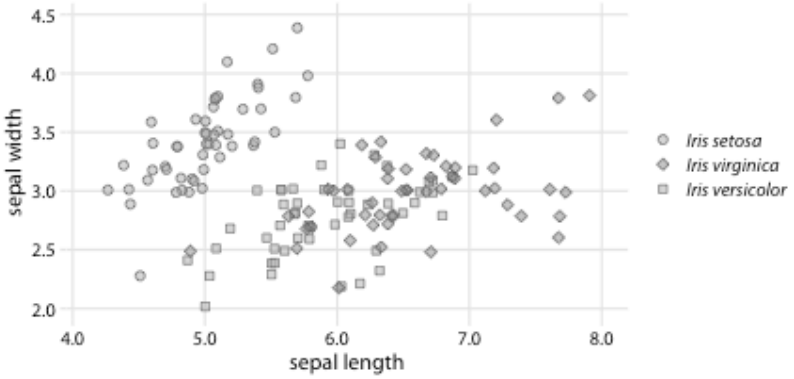
protanomaly



tritanomaly



desaturated

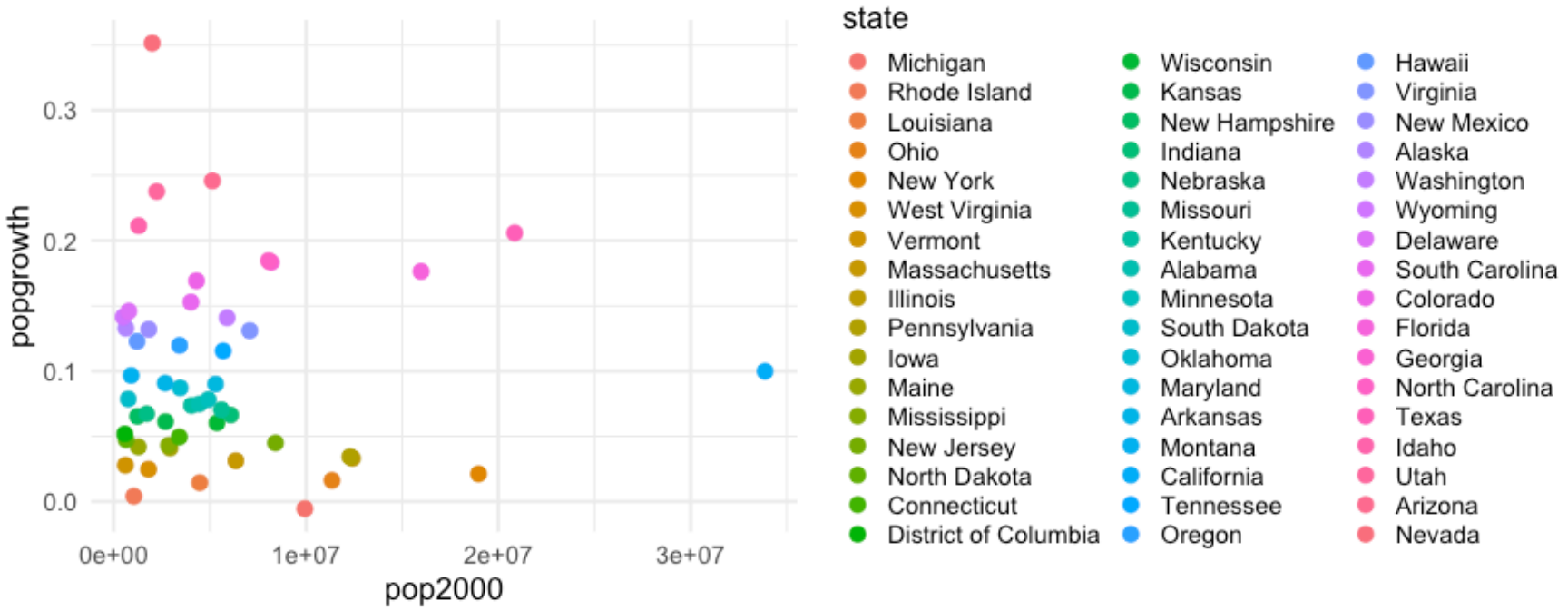


Common problems with color

Too many colors

More than 5-ish categories generally becomes too difficult to track

```
ggplot(popgrowth_df, aes(pop2000, popgrowth, color = state)) +
    geom_point()
```

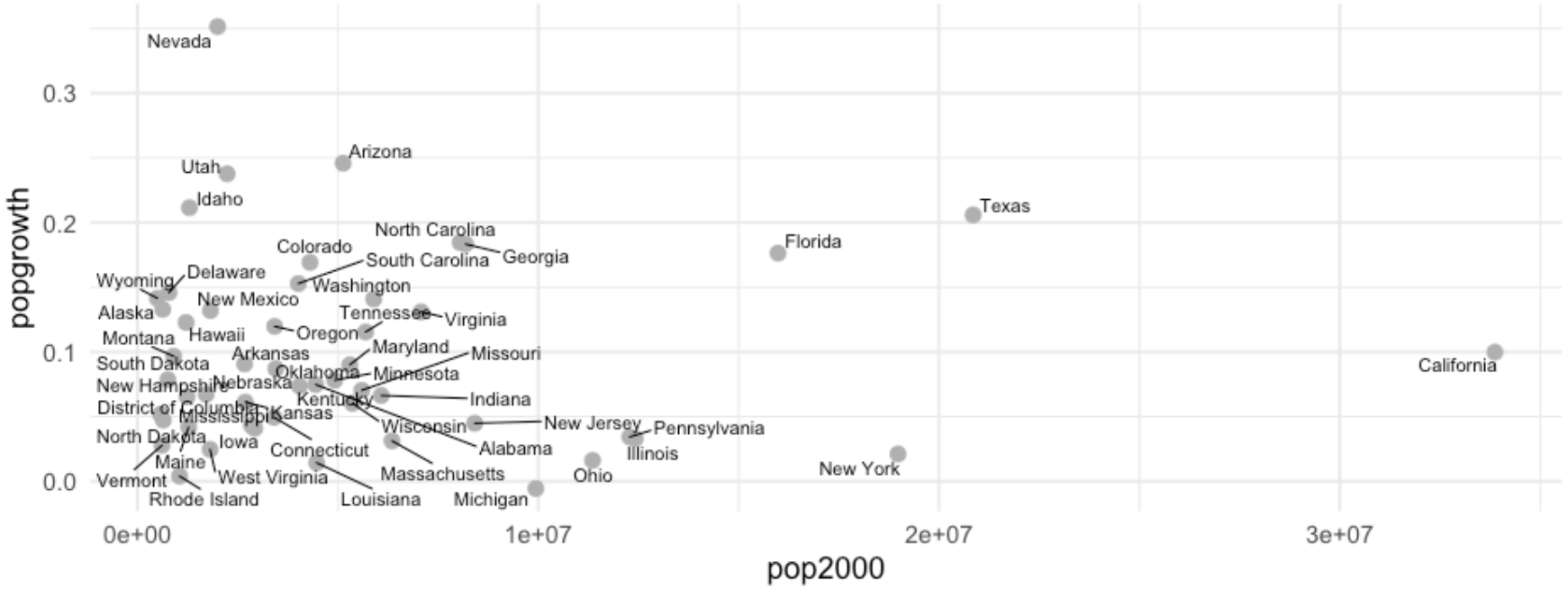


Use labels

More than 5-ish categories generally becomes too difficult to track

```
library(ggrepel)

ggplot(popgrowth_df, aes(pop2000, popgrowth)) +
    geom_point(color = "gray70") +
geom_text_repel(aes(label = state))
```



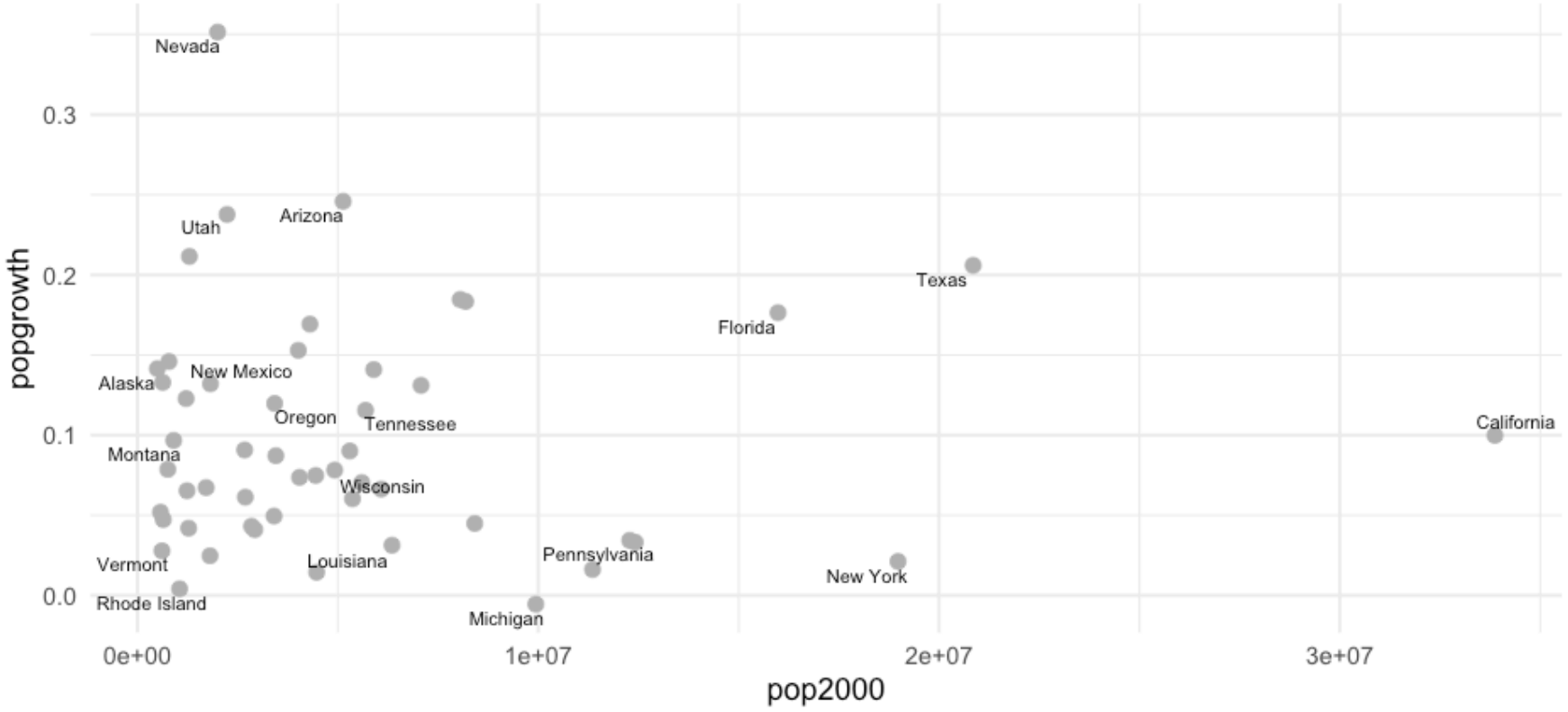
Better

Get a subset

```
to_label <- c("Alaska", "Arizona", "California", "Florida", "Wisconsin",
              "Louisiana", "Nevada", "Michigan", "Montana", "New Mexico",
              "Pennsylvania", "New York", "Oregon", "Rhode Island",
              "Tennessee", "Texas", "Utah", "Vermont")
subset_states <- popgrowth_df %>%
    filter(state %in% to_label)
```

```
library(ggrepel)

ggplot(popgrowth_df, aes(pop2000, popgrowth)) +
    geom_point(color = "gray70") +
geom_text_repel(aes(label = state), data = subset_states)
```



(still lots more cleaning up we could do here...)

Rainbow palette

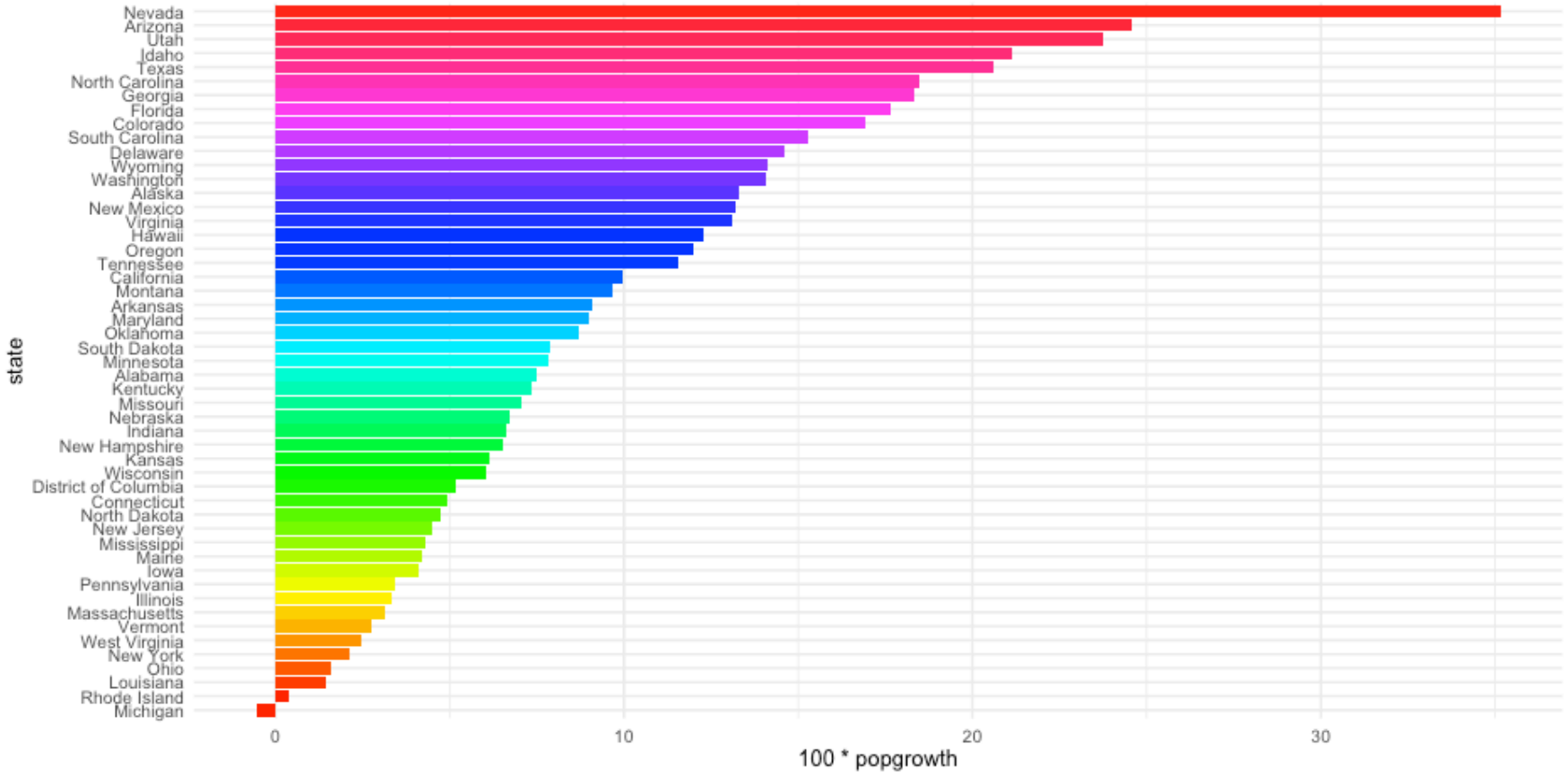
```
rainbow(3)
```

```
## [1] "#FF0000FF" "#00FF00FF" "#0000FFFF"
```

```
rainbow(7)
```

```
## [1] "#FF0000FF" "#FFDB00FF" "#49FF00FF" "#00FF92FF" "#0092FFFF" "#4900FFFF"
## [7] "#FF00DBFF"
```

```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = state)) +
  scale_fill_manual(values = rainbow(51)) +
  coord_flip() +
  guides(fill = "none")
```

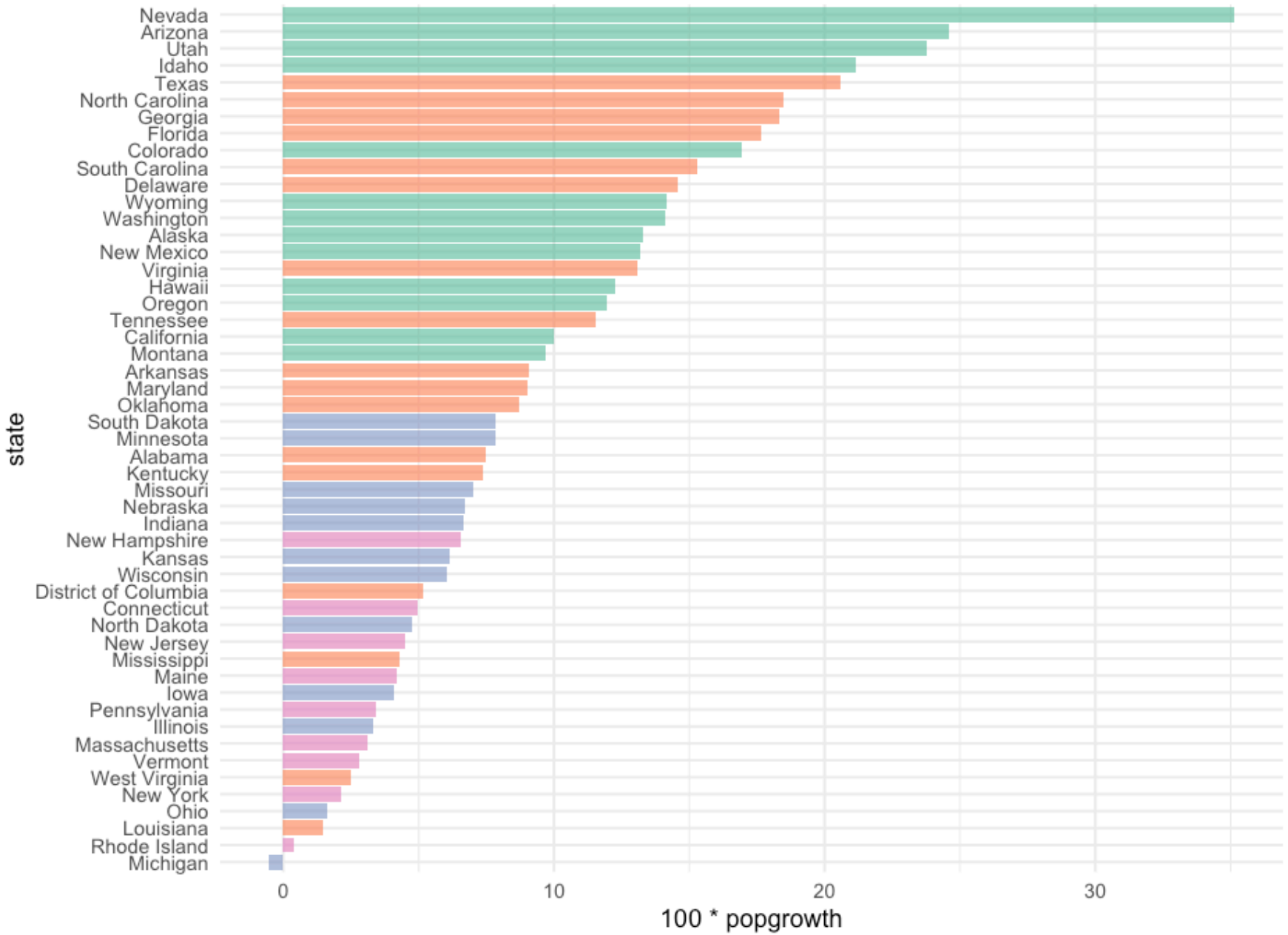


Last few note on palettes

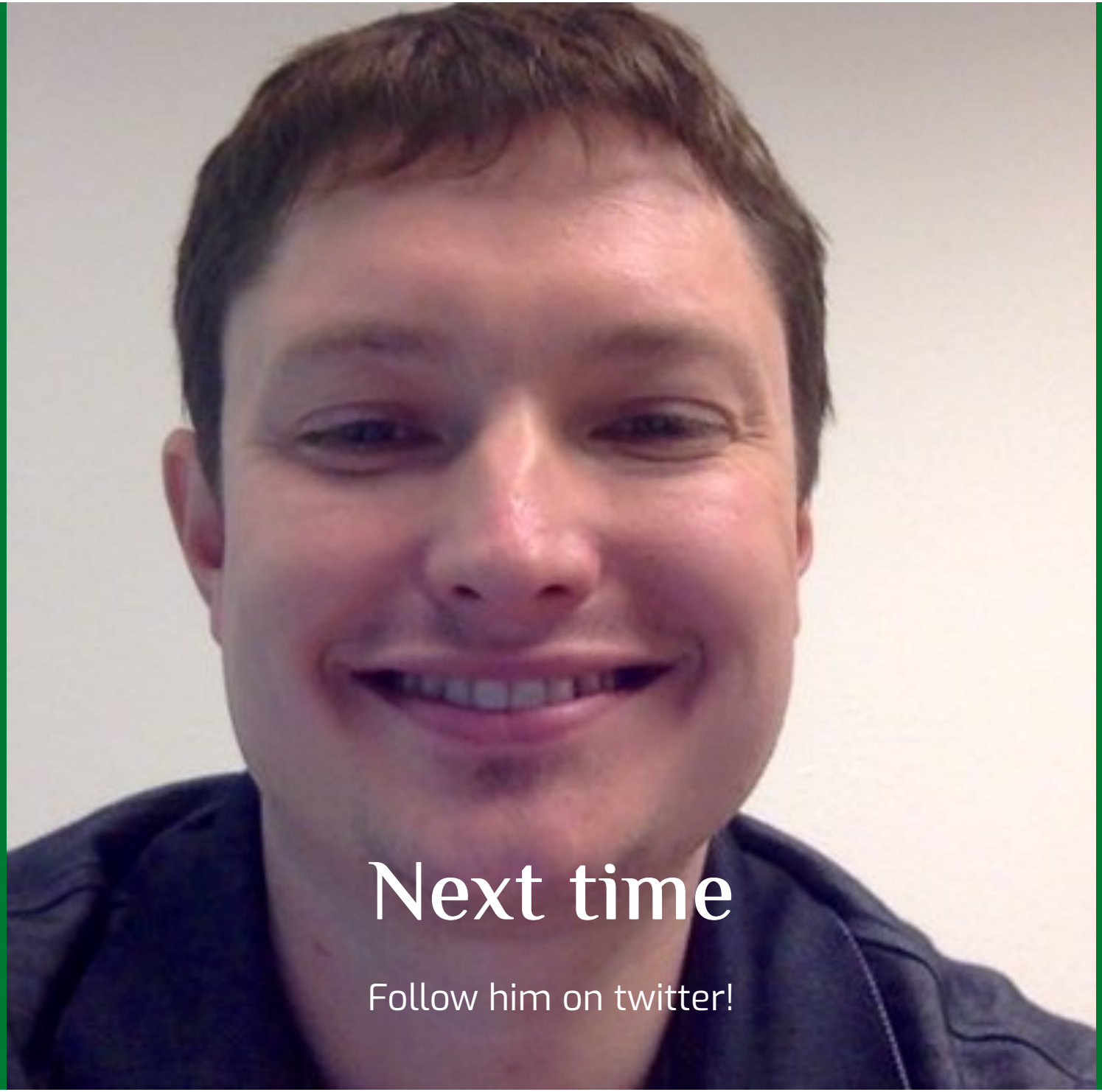
- Do some research, find what you like **and** what tends to work well
- Check for colorblindness
- Look into http://colorbrewer2.org/
    - `scale_color_brewer()` and `scale_fill_brewer()` ship with ggplot2

e.g.,

```
ggplot(popgrowth_df,
       aes(x = state,
           y = 100*popgrowth)) +
  geom_col(aes(fill = region),
           alpha = 0.7) +
  scale_fill_brewer(palette = "Set2") +
  coord_flip() +
  guides(fill = "none")
```


Nevada
Arizona
Utah
Idaho
Texas
North Carolina
Georgia
Florida
Colorado
South Carolina
Delaware
Wyoming
Washington
Alaska
New Mexico
Virginia
Hawaii
Oregon
Tennessee
California
Montana
Arkansas
Maryland
Oklahoma
South Dakota
Minnesota
Alabama
Kentucky
Missouri
Nebraska
Indiana
New Hampshire
Kansas
Wisconsin
District of Columbia
Connecticut
North Dakota
New Jersey
Mississippi
Maine
Iowa
Pennsylvania
Illinois
Massachusetts
Vermont
West Virginia
New York
Ohio
Louisiana
Rhode Island
Michigan
state
0
10
20
30
100 * popgrowth

Paleteer package



Next time

Follow him on twitter!