

In-class worksheet 30

May 9, 2019

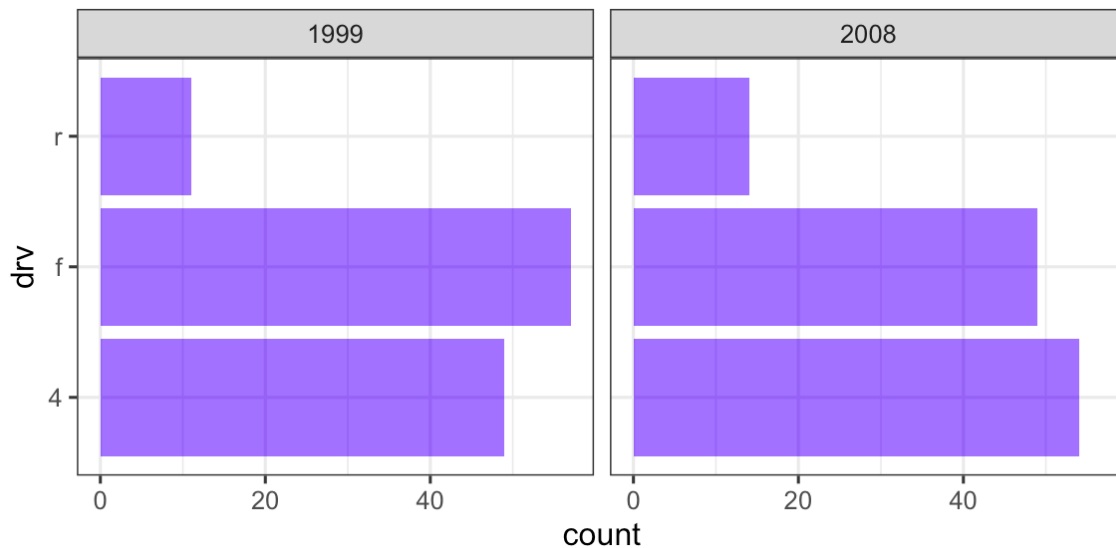
In this worksheet, we will use the libraries tidyverse, sf, and gganimate:

```
library(tidyverse)
theme_set(theme_bw(base_size=12)) # set default ggplot2 theme
library(sf)
library(gganimate)
```

1. A grammar of animation

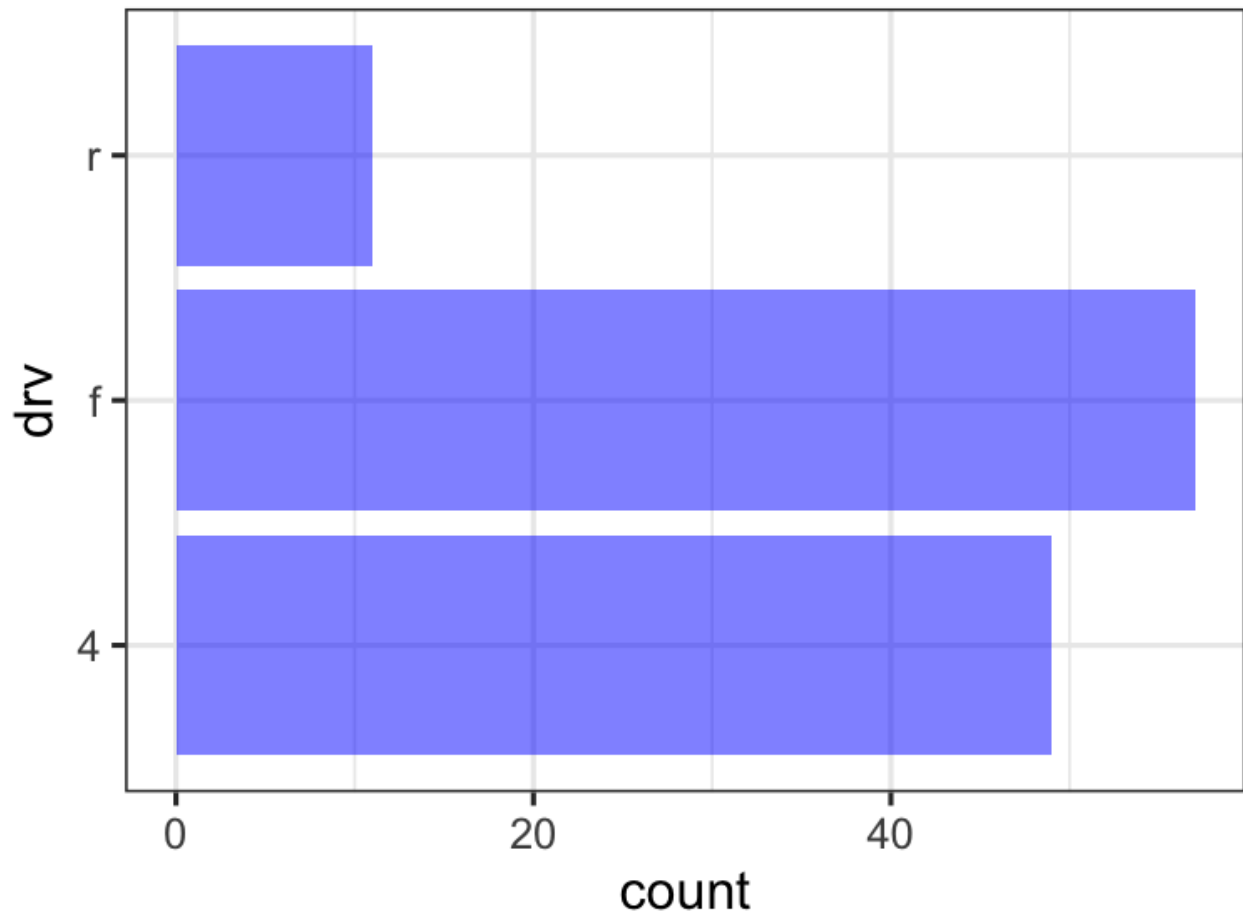
The gganimate package provides a rich grammar of animation. The key idea is that animations are similar to faceting, only that we facet in time rather than in space. For example, consider this plot, which shows the distribution of drive trains in cars from 1999 and 2008.

```
ggplot(mpg, aes(drv)) +
  geom_bar(fill = "blue", alpha = 0.5) +
  coord_flip() +
  facet_wrap(~year)
```



We can animate this plot by replacing `facet_wrap()` with `transition_states()`. Everything else remains the same.

```
ggplot(mpg, aes(drv)) +
  geom_bar(fill = "blue", alpha = 0.5) +
  coord_flip() +
  transition_states(year) # we don't need a ~ here before year
```

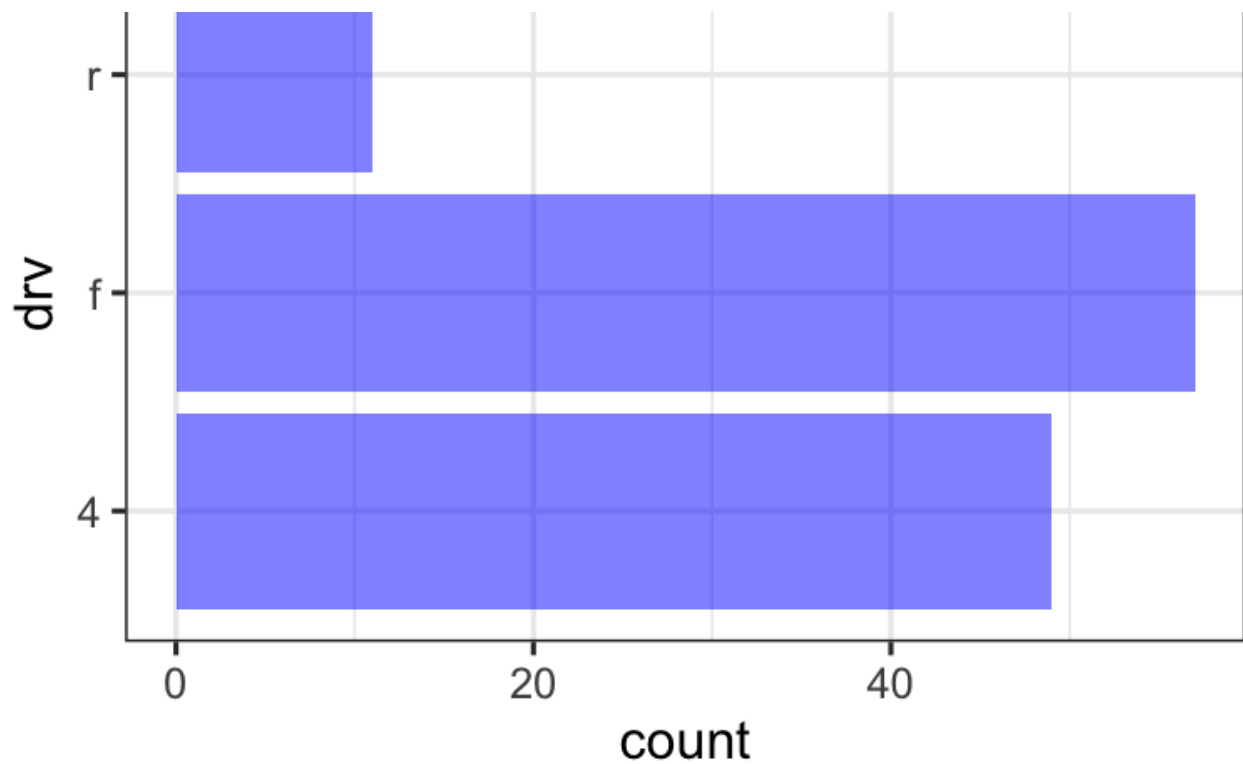


However, what is missing from this animation now is that we don't know which frames belong to which year. We can resolve this issue by adding a plot title. The string `{closest_state}` inside the title will be replaced by the value of the variable on which we are transitioning.

```
ggplot(mpg, aes(drv)) +
  geom_bar(fill = "blue", alpha = 0.5) +
  coord_flip() +
  labs(title = "Year: {closest_state}") +
  transition_states(year)
```

Year: 1999

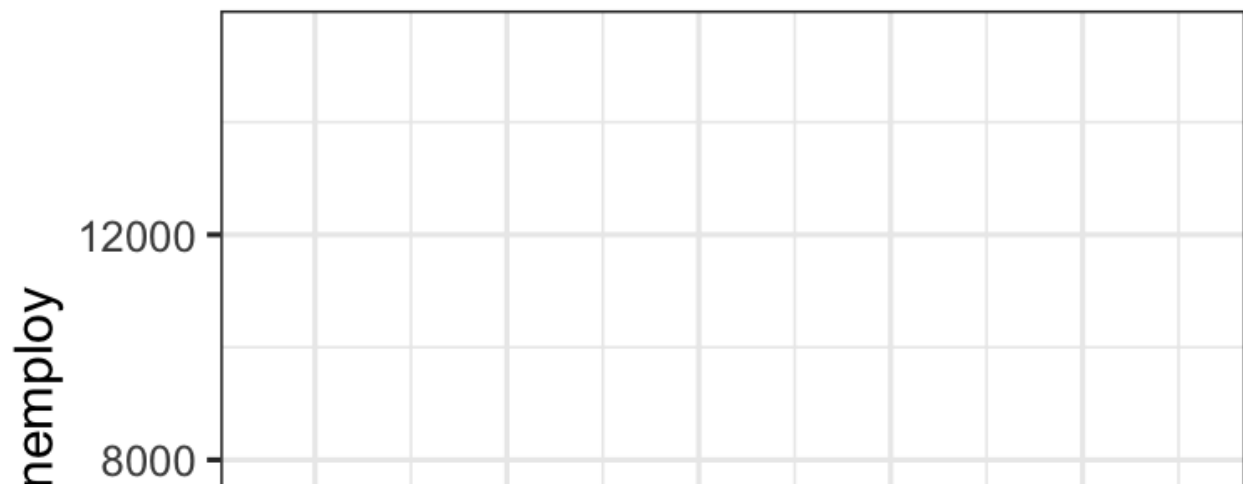


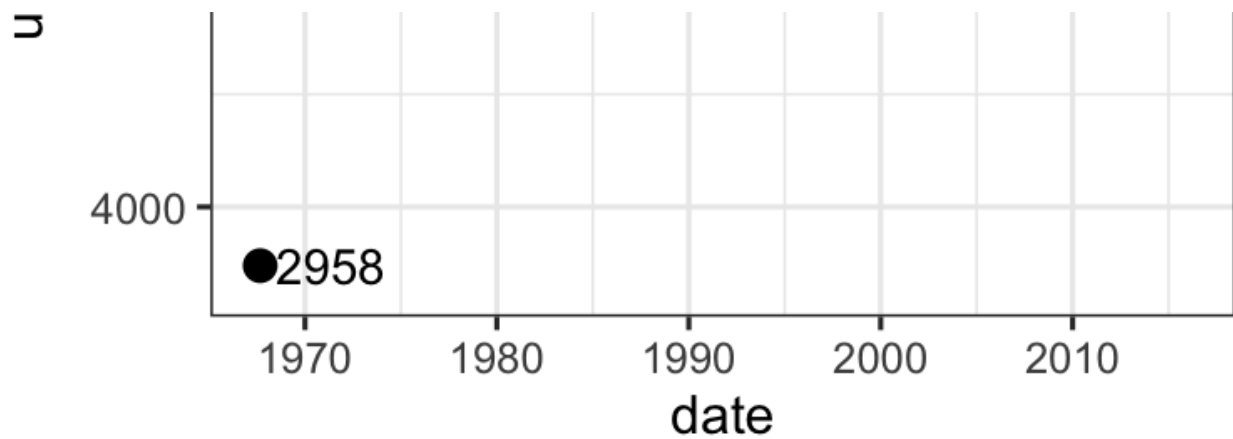


2. Changing the transition type

There are a number of different transition types. For example, `transition_reveal()` can be used to slowly reveal a line graph.

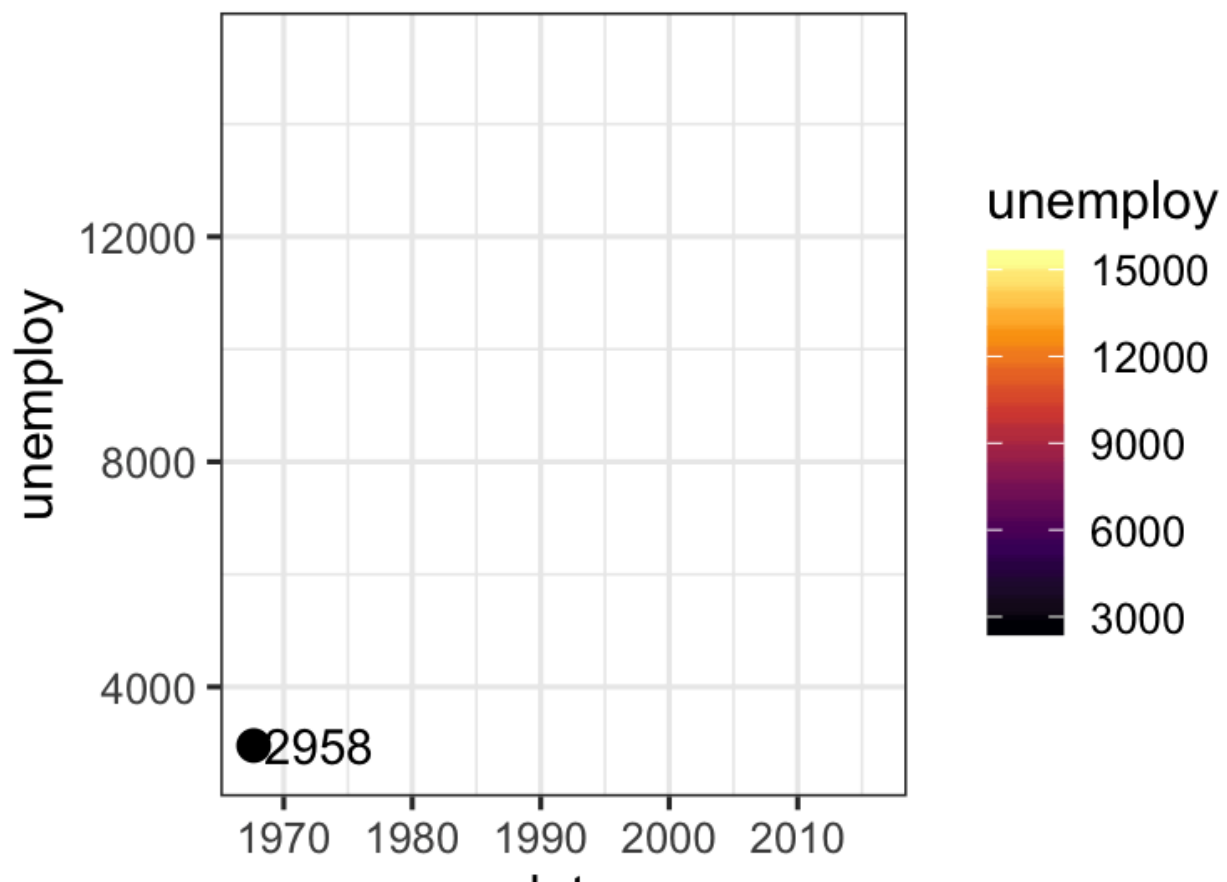
```
ggplot(economics, aes(date, unemploy)) +  
  geom_line() +  
  geom_point(size = 3) +  
  geom_text(aes(label = unemploy), hjust = 0, nudge_x = 300) +  
  transition_reveal(date)
```





Modify the above animation so the line color reflects the unemployment rate. Hint: Use `scale_color_viridis_c(option = "B")` to define the color scale.

```
ggplot(economics, aes(date, unemploy)) +
  geom_line(aes(color = unemploy)) +
  geom_point(size = 3) +
  geom_text(aes(label = unemploy), hjust = 0, nudge_x = 300) +
  scale_color_viridis_c(option = "B") +
  transition_reveal(date)
```

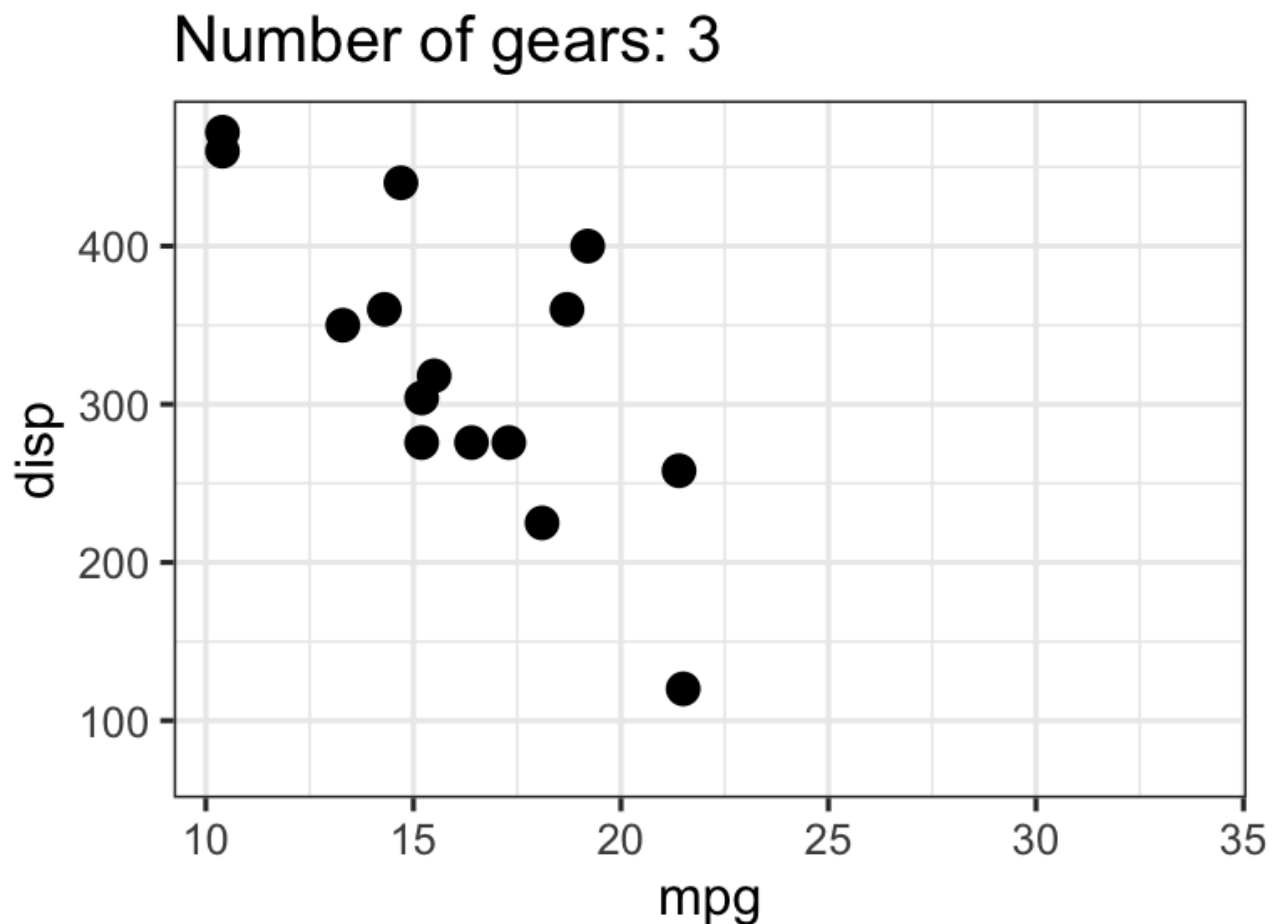


date

3. Entering and exiting data points

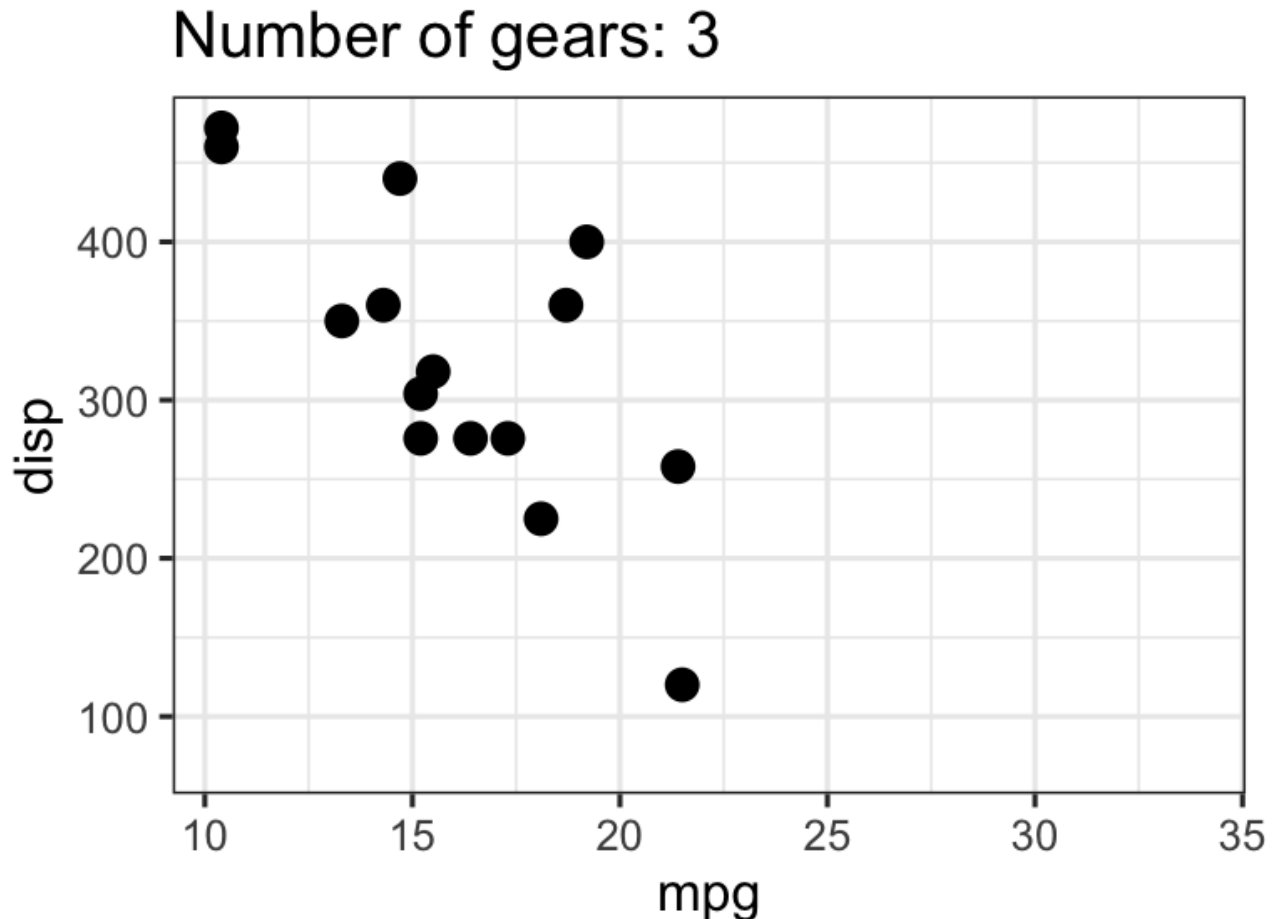
For plots that have different numbers of points in different states, we can also control how the points should appear or disappear, using the `enter_*`() and `exit_*`() functions. For example, `enter_fade()` and `exit_fade()` make the points fade in and out.

```
ggplot(mtcars, aes(mpg, disp, group = 1:nrow(mtcars))) +
  geom_point(size = 3) +
  transition_states(gear, transition_length = 2, state_length = 1) +
  labs(title = "Number of gears: {closest_state}") +
  enter_fade() +
  exit_fade()
```



Note that in the above animation, we have given each data point a different value for the group aesthetic. This tells `gganimate` that the points in one state are not related to the points in another state. If we don't set a group aesthetic, we get a mix of points that move and points that fade.

```
ggplot(mtcars, aes(mpg, disp)) +
  geom_point(size = 3) +
  transition_states(gear, transition_length = 2, state_length = 1) +
  labs(title = "Number of gears: {closest_state}") +
  enter_fade() +
  exit_fade()
```

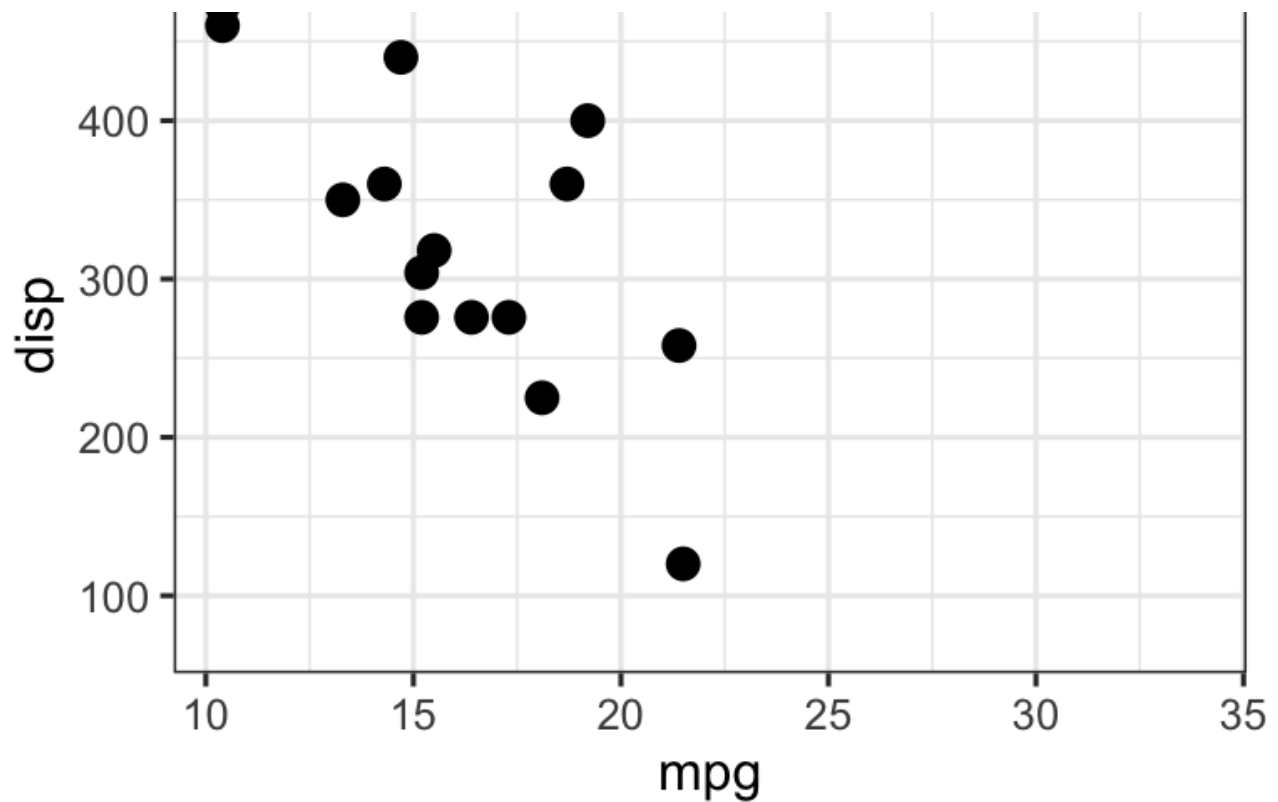


Now try different enter and exit effects. For example, try `enter_grow()` and `exit_fly()`.

```
ggplot(mtcars, aes(mpg, disp, group = 1:nrow(mtcars))) +
  geom_point(size = 3) +
  transition_states(gear, transition_length = 2, state_length = 1) +
  labs(title = "Number of gears: {closest_state}") +
  enter_grow() +
  exit_fly(x_loc = 50)
```

Number of gears: 3

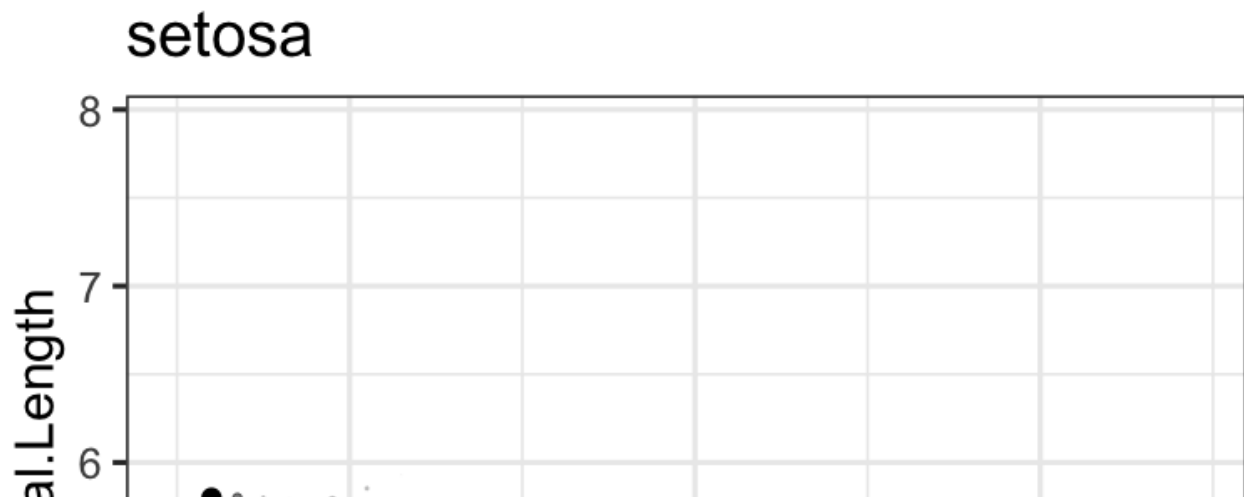


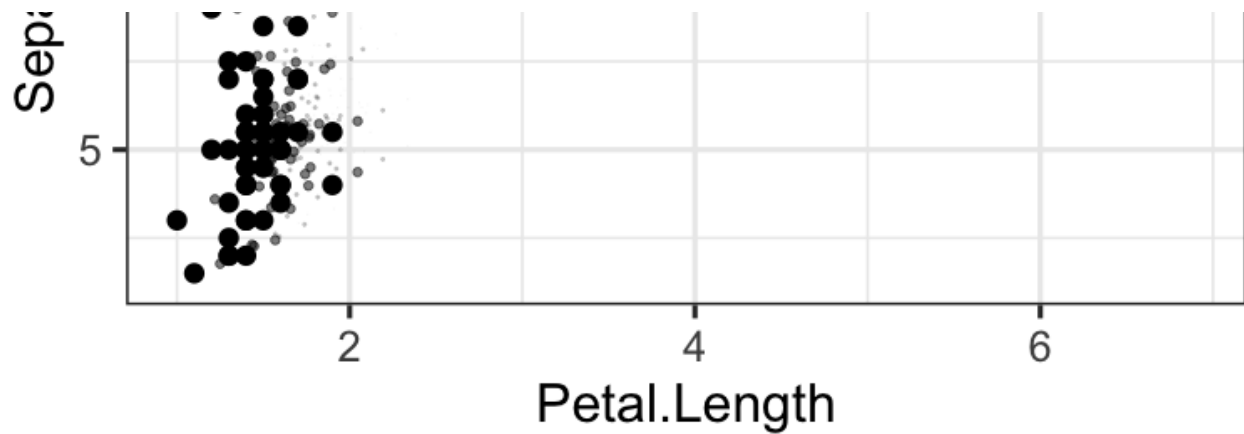


4. Shadow wakes

We can also create shadow wakes to visually indicate movement.

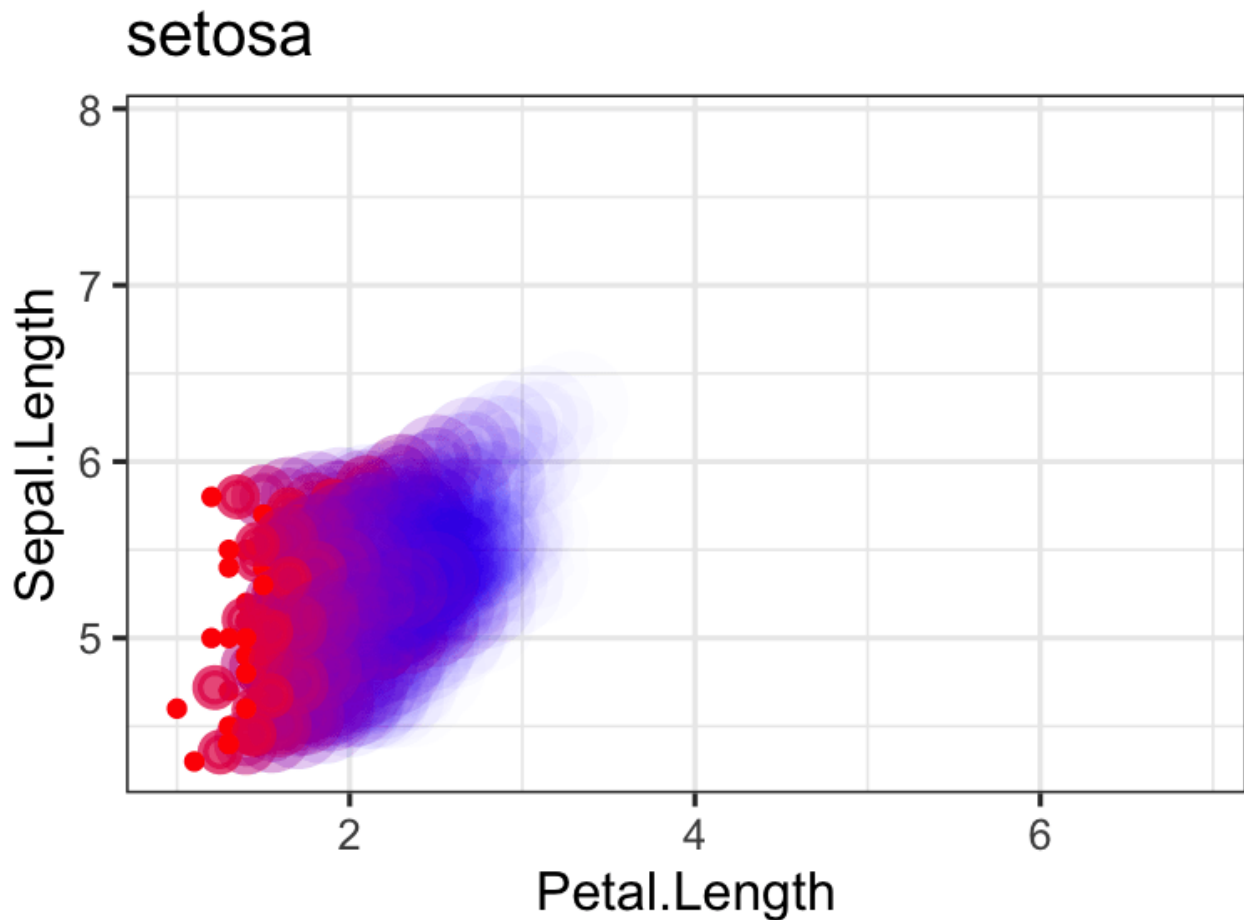
```
ggplot(iris, aes(Petal.Length, Sepal.Length)) +
  geom_point() +
  labs(title = "{closest_state}") +
  transition_states(Species, transition_length = 4, state_length = 1) +
  shadow_wake(wake_length = 0.05)
```





Now modify the parameters of the `shadow_wake()` command to generate a different wake effect. Hint: Look at the documentation of `shadow_wake()` for inspiration.

```
ggplot(iris, aes(Petal.Length, Sepal.Length)) +  
  geom_point(color = "red") +  
  labs(title = "{closest_state}") +  
  transition_states(Species, transition_length = 4, state_length = 1) +  
  shadow_wake(wake_length = 0.1, alpha = 0, size = 5, colour = "blue")
```

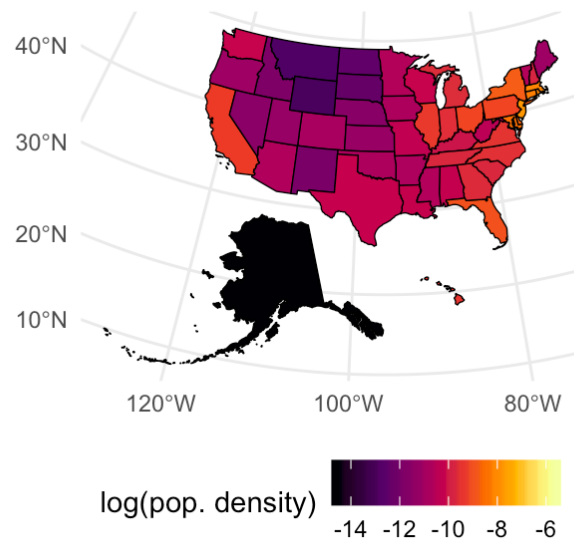


5. If this was easy

The map data frames used in last class contain not only a map of all US states with median income data but also a cartogram version of this map where each state has been reshaped so its area is proportional to the number of people living there.

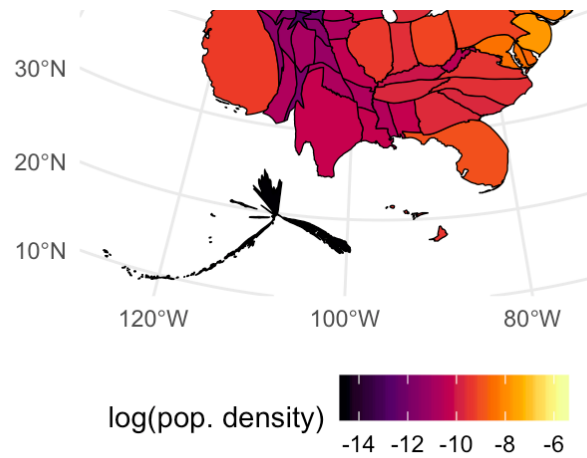
```
load(url("https://wilkelab.org/classes/SDS348/data_sets/US_income.RData"))

ggplot(US_income, aes(fill = log(as.numeric(popdens)))) +
  geom_sf(color = "black", size = 0.25) +
  scale_fill_viridis_c(
    option = "B",
    name = "log(pop. density)"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



```
ggplot(US_income_cartogram, aes(fill = log(as.numeric(popdens)))) +
  geom_sf(color = "black", size = 0.25) +
  scale_fill_viridis_c(
    option = "B",
    name = "log(pop. density)"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```





Make an animation that switches between these two maps.

```
# To animate between these two maps, we need to add a variable
# indicating state to each data frame. Then we can bind the two together and an
imate.
US_income$state <- "normal"
US_income_cartogram$state <- "cartogram"

US_combined <- rbind(US_income, US_income_cartogram)

ggplot(US_combined, aes(fill = log(as.numeric(popdens)))) +
  geom_sf(color = "black", size = 0.25) +
  transition_states(state) +
  scale_fill_viridis_c(
    option = "B",
    name = "log(pop. density)"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

