Todd Soo, Annie Kelly, Nathan Mermilliod

What is gganimate?

- Extends the grammar of graphics as implemented by ggplot2 to include the description of animation
  - Provides new grammar classes added to the plot object in order to customize how should change with time
- Allows user to expand range of graphics by allowing animations.
  - Can be added to plot to demonstrate how data can change with time.
- Inputs and outputs different classes of data.
- Works by loading a series of .png's, which are loaded into a gif.
- Data sets used before work with gganimate. mtcars can be used to generate animated box plots, iris can be used to generate animated scatter plots, and gapminder can be used to map GDP and life expectancy

Different functions in gganimate

transition_*() : how the data should be spread out and how it relates to itself across time.

view_*() : how the positional scales should change along the animation.

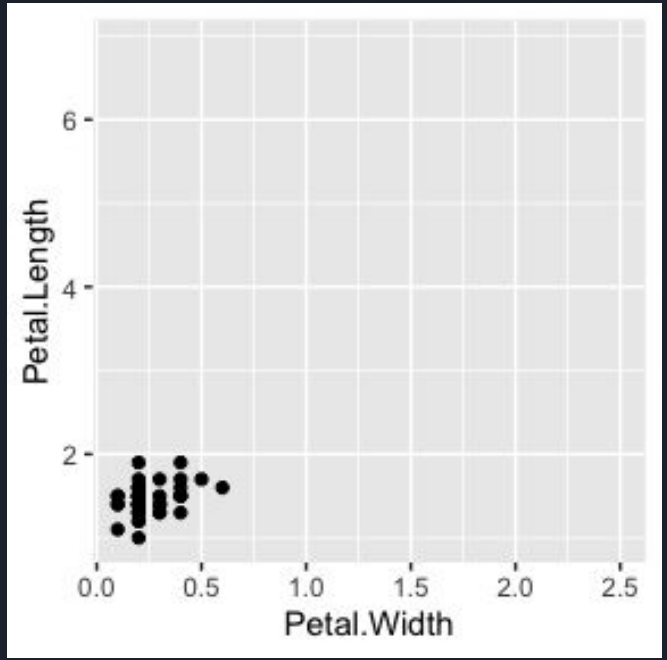
shadow_*() : how data from other points in time should be presented in the given point in time.

enter_*()/exit_*() : how new data should appear and how old data should disappear during the course of the animation.

ease_aes() :how different aesthetics should be eased during transitions.

Easing

- In the transition between states, or the animation of these, transition_states() uses a concept called easing.
- With ease_aes(), you are able to define things like velocity of transition to make the animation look smoother or faster.



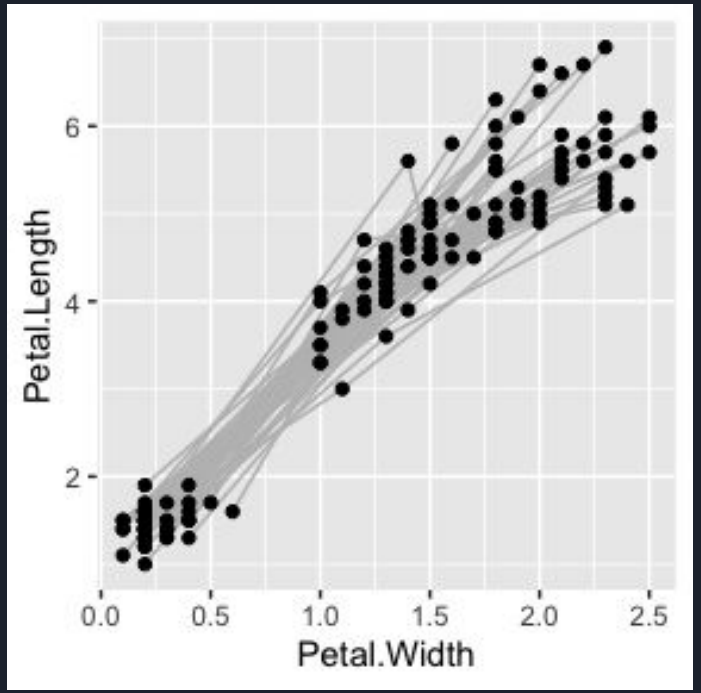
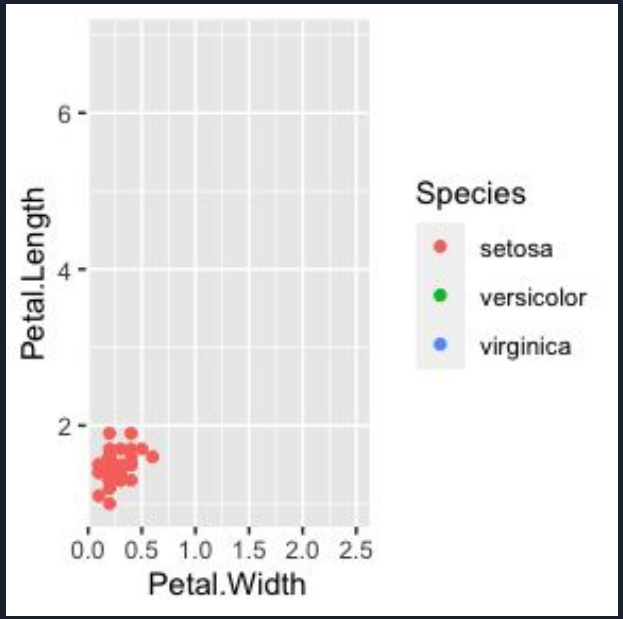

```
Anim +

ease_aes('cubic-in-out') #slow start and end for smoother look
```

Our Code

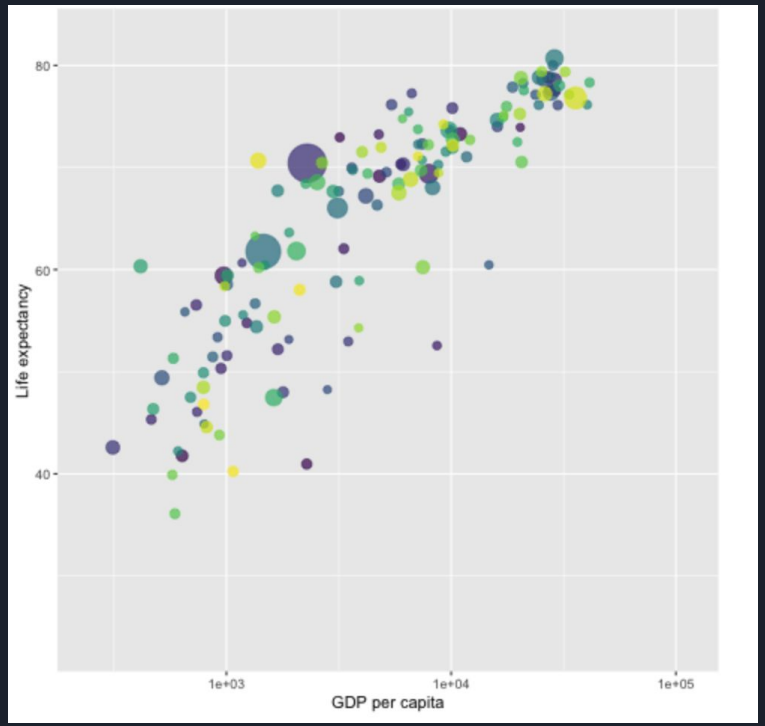
- This is the code we wrote to demonstrate easing in gganimate.
- Turns a graph from a static plot to an animation

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

#Now using gganimate...

anim <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length)) +
  geom_point(aes(colour = Species, group = 1L)) + #adding some color...
  transition_states(Species,
                    transition_length = 2, #this changes the length of the transition in seconds
                    state_length = 1) #this changes how long the plot stays still between transitions
anim +
  ease_aes('cubic-in-out') #this makes the plot have smoother transitions


# Animated GIF - This code is necessary to load the gif. The prior code generates .png files.
animate(anim, renderer =  gifski_renderer())
anim_save(last_animation(), filename = "plot1.gif")

anim

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

#another option to animate using fade and shrink

anim <- ggplot(iris, aes(x = Petal.Width, y = Petal.Length)) +
  geom_point(aes(colour = Species), size = 2) +
  transition_states(Species,
                    transition_length = 2,
                    state_length = 1) +

  enter_fade() + #this causes the code to enter from a fade
  exit_shrink() #this causes the code the exit by shrinking out of the plot
  ease_aes('cubic-in-out') #this makes the plot have smoother transitions

animate(anim, renderer =  gifski_renderer())
anim_save(last_animation(), filename = "plot1.gif")
```

This is important to render as a gif! Otherwise it just creates .png files! (part of package "gifski")
```

Object Permanence

- We are trying to take data points and move them across a line graph to connect observations of the same species (each species representing a line).
- You can use the group Aesthetics to distinguish between different observations.
- Each line represents a species, and Aesthetics helps R recognize that.




```
ggplot(iris, aes(x = Petal.Width, y = Petal.Length)) +
  geom_line(aes(group = rep(1:50, 3)), colour = 'grey') +
  geom_point()
```

Enter and Exit

- You can display the same data using enter and exit functions.
- This makes the different groups of data appear and disappear.




```
anim +
 enter_fade() + #this causes the code to enter from a fade
  exit_shrink() #this causes the code the exit by shrinking out of the plot
```

Gapminder

- A more complicated example showing how gganimate can make useful information plots showing data over time.

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(gapminder)
library(ggplot2)
library(gganimate)

file_renderer(dir = ".", prefix = "gganim_plot", overwrite = FALSE)
data(gapminder)
p <- ggplot(
gapminder,
aes(x = gdpPercap, y=lifeExp, size = pop, colour = country)
) +
geom_point(show.legend = FALSE, alpha = 0.7) +
scale_color_viridis_d() +
scale_size(range = c(2, 12)) +
scale_x_log10() +
labs(x = "GDP per capita", y = "Life expectancy")
p + transition_time(year)
file_renderer(dir = ".", prefix = "gganim_plot", overwrite = FALSE)
animate(p +transition_time(year), renderer =  gifski_renderer())
anim_save(last_animation(), filename = "plot2.gif")
```



Taken from User:Technocrat
```

References

- https://community.rstudio.com/t/warning-message-file-renderer-failed-to-copy-frames-to-the-destination-directory/45261/2
- https://gganimate.com/
- https://exts.ggplot2.tidyverse.org/gallery/