Cheatography

gganimate Cheat Sheet

by Pranav V A via cheatography.com/126418/cs/24452/

Core Concepts

gganimate builds on ggplot2's grammar of graphics to provide functions for animation. You
add them to plots created with ggplot() the same way you add a geom
Main Function Groups

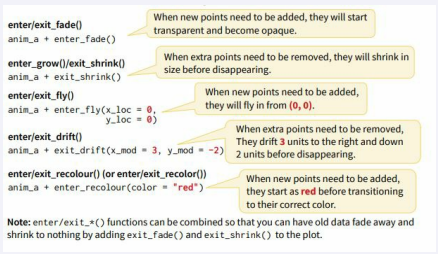
- **`transi tio n_*()`**: What variable controls
change and how?
- **`view_*()`**: Should the axes change with the
data?
- **`enter/ exi t_*()`**: How does new data get
added the plot? How does old data leave?
- **`shadow_*()`**: Should previous data be "remembered" and shown with current data?
- **`ease_aes()`**: How do you want to handle the
pace of change between transition values?

Note: you only need a `transi tio n_*()` or `view_*()` to make an animation. The other function groups enable you to add features or alter
gganimate's default settings .

Starting Plots

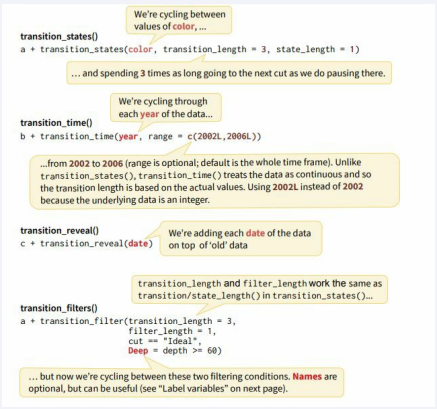
```
librar y(t idy verse)
librar y(g gan imate)
a <- ggplot (di amonds, aes(carat, price)) + geom_p oint
()
b <- ggplot (tx hou sing, aes(month, sales)) + geom_col(
)
c <- ggplot (ec ono mics, aes(date, psavert)) + geom_l i
ne()
```

enter/exit_*()



Note: `enter/ exi t_*()` functions can be combined so that you can have old data fade away and

transition_*()



Other transitions

- **`transi tio n_m anual()`**: Similar to transition_states(), but without intermediate states.
- **`transi tio n_l ayers()`**: Add layers (geoms) one at time.
- **`transi tio n_c omp one nts()`**: Transition elements independently from each other.
- **`transi tio n_e vents()`**: Each element's duration can be controlled individually.

Baseline Animation

```
anim_a <- a +
transition_states(color,
transition_length = 3,
state_length = 1)
```

Saving animations

```
anim_a + exit_shrink()`
anim_s ave ("se con d_s ave d_a nim ati on.g if ")
```

`anim_s ave()` uses gifski to render the animation as a .gif file by default. You can use the renderer argument for other output types including video files (av_renderer() or ffmeg_renderer()) or spritesheets (sprite_renderer()):

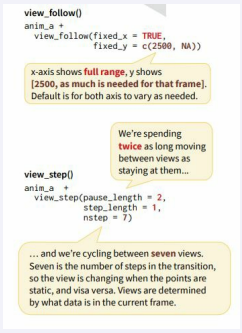
```
# requires you to have the av package installed
anim_s ave ("th ird _sa ved _an ima tio n.m p4", ren derer = av_ren der
```

view_*()

`view_z oom()`

`view_z oom()` works similarly to

view_*()

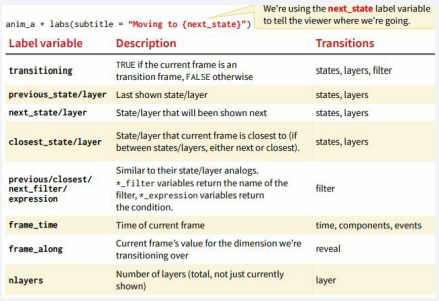


`view_z oom()`

`view_z oom()` works similarly to `view_s tep()`, except it changes the view by
zooming and panning.

Note: both `view_s tep()` and `view_z oom()` have `view_* _ma nual()` versions for setting views
directly instead of inferring it from frame data

Label variables

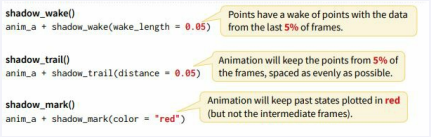


Label variable	Description	Transitions
`transitioning`	TRUE if the current frame is an transition frame, FALSE otherwise	states, layers, filter
`previous_state/layer`	Last shown state/layer	states, layers
`next_state/layer`	State/layer that will been shown next	states, layers
`closest_state/layer`	State/layer that current frame is closest to (if between states/layers, either next or closest).	states, layers
`previous/closest/ next_filter/ expression`	Similar to their state/layer analogs. `*_filter` variables return the name of the filter, `*_expression` variables return the condition.	filter
`frame_time`	Time of current frame	time, components, events
`frame_along`	Current frame's value for the dimension we're transitioning over	reveal
`nlayers`	Number of layers (total, not just currently shown)	layer

gganimate's transition_*() functions create label variables you can pass to (sub)titles and other labels
with the glue package. For example, transition_states() has next_state, which is the name of the
state the animation is transitioning towards.
Label variables are different between transitions, and details
are included in the documentation of each.

shrink to nothing by adding `exit_f ade()` and `exit_s hrink()` to the plot

shadow_*()



`view_s tep()`, except it changes the view by
zooming and panning.

Note: both `view_s tep()` and `view_z o om()` have `view_* _ma nual()` versions for setting views
directly instead of inferring it from frame data

By **Pranav V A**
cheatography.com/pranav-v-a/

Published 26th September, 2020.
Last updated 26th September, 2020.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com