The *ease_aes()* Function

The *ease_aes()* function controls the easing of aesthetics or variables in gganimate. The default, *ease_aes()*, models a linear transition between states. Other easing functions are specified using the easing function name, appended with one of three modifiers (3) :

Easing Functions

quadratic models an exponential function of exponent 2.
cubic models an exponential function of exponent 3.
quartic models an exponential function of exponent 4.
quintic models an exponential function of exponent 5.
sine models a sine function.
circular models a pi/2 circle arc.
exponential models an exponential function of base 2.
elastic models an elastic release of energy.
back models a pullback and release.
bounce models the bouncing of a ball.

Modifiers

-in applies the easing function as-is.
-out applies the easing function in reverse.
-in-out applies the first half of the transition as-is and the last half in reverse.

The formulas used to implement these options can be found here (4). They are illustrated below using animated scatter plots and bar charts.

Data File Description

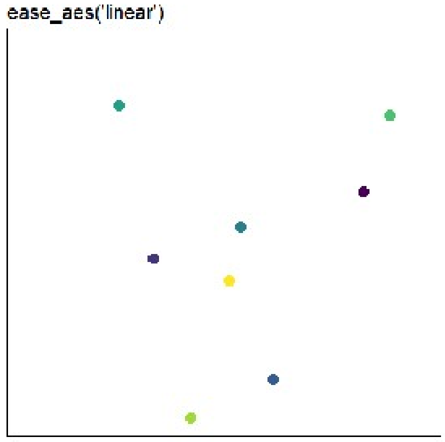
'data.frame': 40 obs. of 7 variables:
Cat : chr "A" "A" "A" "A" …
OrdInt: int 1 2 3 4 5 1 2 3 4 5 …
X : num 70.5 78.1 70.2 78.1 70.5 30.7 6.9 26.7 6.9 30.7 …
Y : num 1.4 7.6 -7.9 7.6 1.4 -7 -23.8 19.8 -23.8 -7 …
Rank : int 2 2 2 2 2 7 8 8 8 7 …
OrdDat: chr "01/01/2019" "04/01/2019" "02/01/2019" "05/01/2019" …
Ord : Date, format: "2019-01-01" "2019-04-01" "2019-02-01" "2019-05-01" …

The data used for this demo is a much abbreviated and genericized version of this (5) data set.

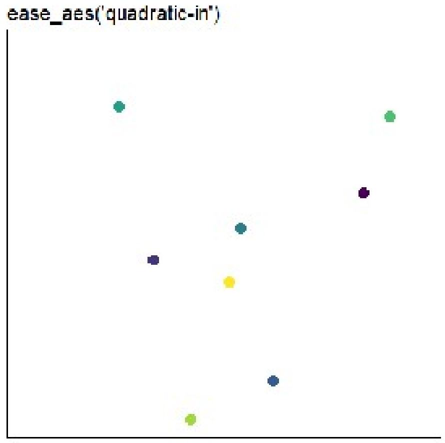
Scatter Plots

Here is the code used for the *ease_aes('cubic-in')* animated scatter plot:

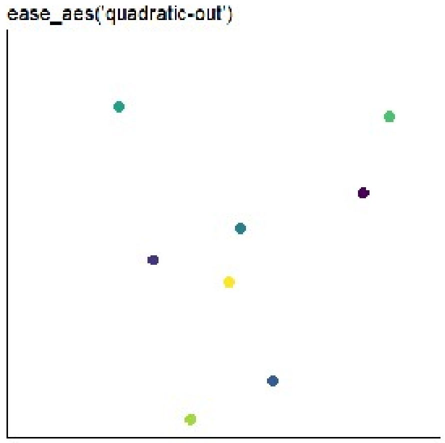
```
# load libraries
library(gganimate)
library(tidyverse)

# the data file format is indicated above
data <- read.csv('Data.csv')

# convert date to proper format
data$Ord <- as.Date(data$OrdDat, format='%m/%d/%Y')
# specify the animation length and rate
options(gganimate.nframes = 30)
options(gganimate.fps = 10)

# loop the animation
options(gganimate.end_pause = 0)
# specify the data source and the X and Y variables
ggplot(data, aes(X, Y)) +
# specify the plot format
theme(panel.background = element_rect(fill = 'white'))+
theme(axis.line = element_line()) +
theme(axis.text = element_blank())+
theme(axis.ticks = element_blank())+
theme(axis.title = element_blank()) +
theme(plot.title = element_text(size = 20)) +
theme(plot.margin = margin(25, 25, 25, 25)) +
theme(legend.position = 'none') +

# create a scatter plot
geom_point(aes(color = Cat), size = 5) +

# indicate the fill color scale
scale_fill_viridis_d(option = "D", begin = 0, end = 1) +

# apply the fill to the 'Cat' variable
aes(group = Cat) +

# animate the plot on the basis of the 'Ord' variable
transition_time(Ord) +

# the ease_aes() function
ease_aes('cubic-in') +

# title the plot
labs(title = "'ease_aes(cubic-in)'")
```



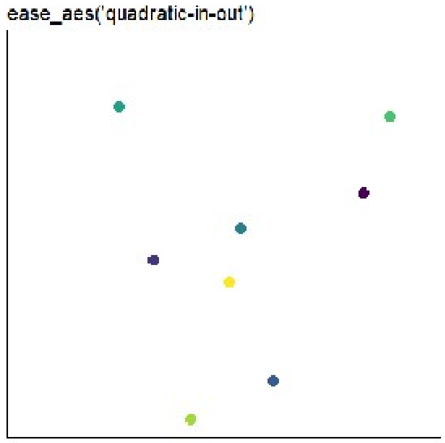
This is the default value, equivalent to ease_aes(). It is the only easing function that does not take a modifier.



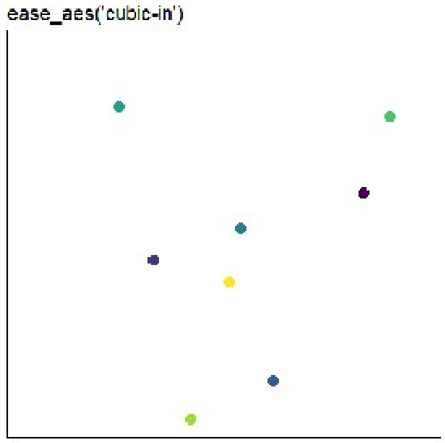
ease_aes('quadratic-in') scatter plot



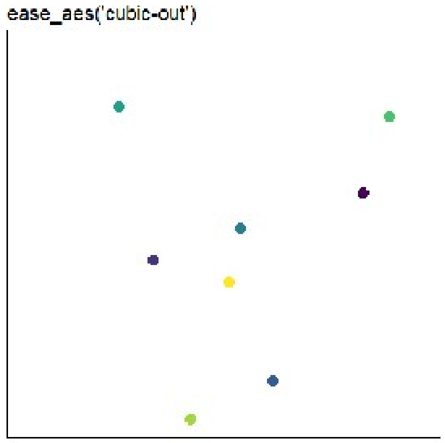
ease_aes('quadratic-out') scatter plot



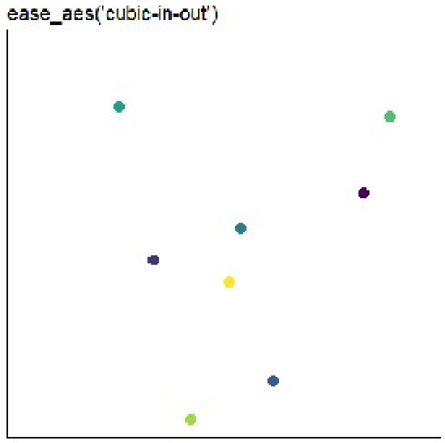
ease_aes('quadratic-in-out') scatter plot



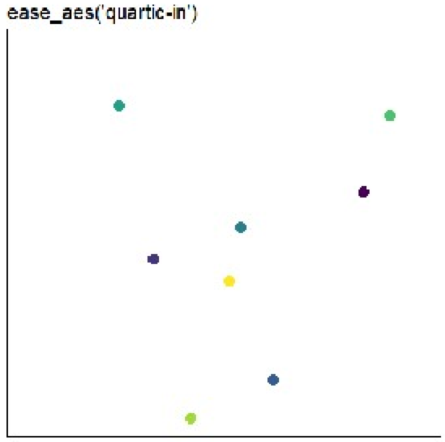
ease_aes('cubic-in') scatter plot



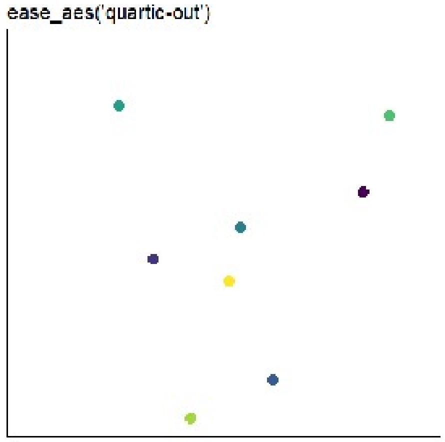
ease_aes('cubic-out') scatter plot



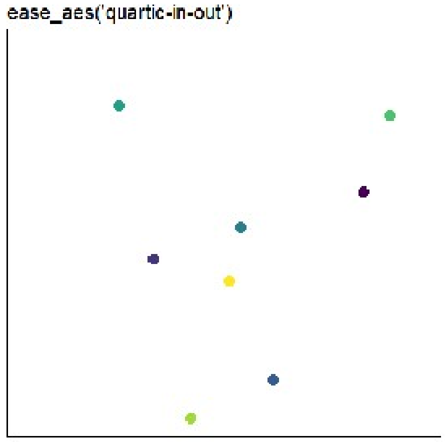
ease_aes('cubic-in-out') scatter plot



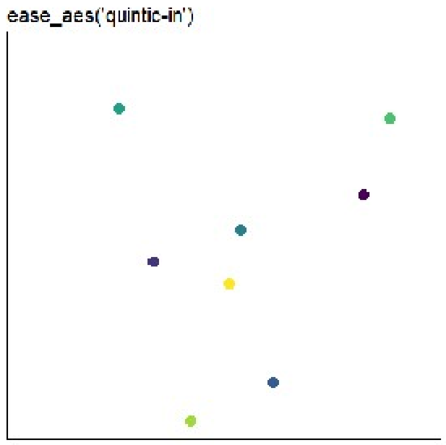
ease_aes('quartic-in') scatter plot



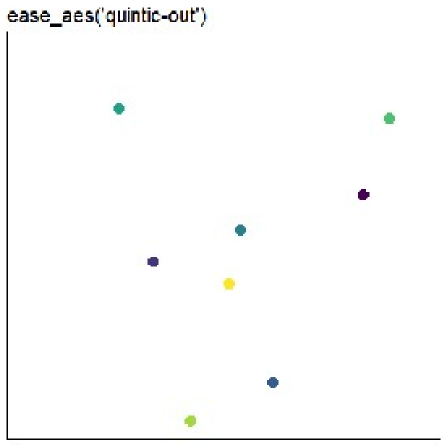
ease_aes('quartic-out') scatter plot



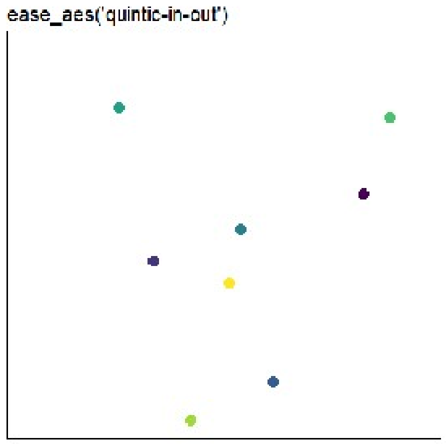
ease_aes('quartic-in-out') scatter plot



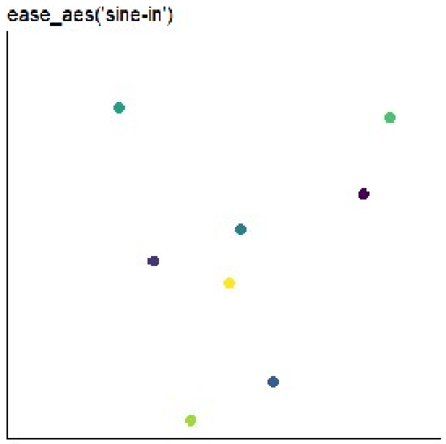
ease_aes(‘quintic-in’) scatter plot



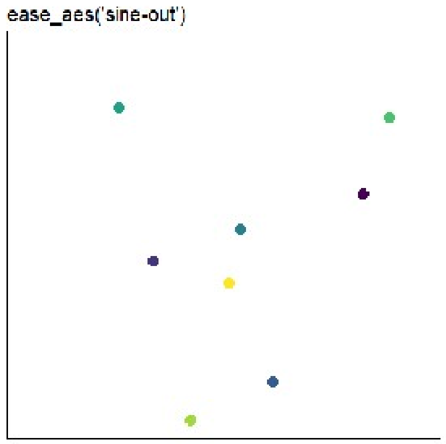
ease_aes(‘quintic-out’) scatter plot



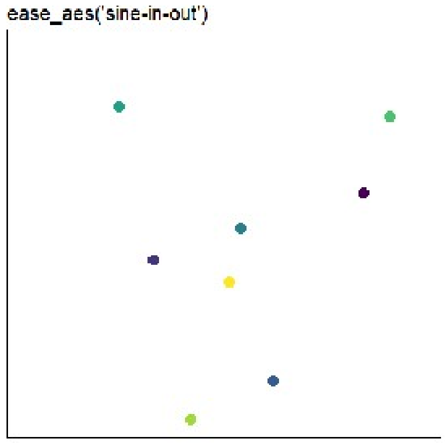
ease_aes(‘quintic-in-out’) scatter plot



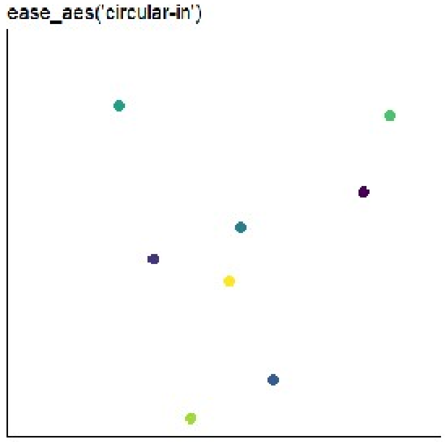
ease_aes(‘sine-in’) scatter plot



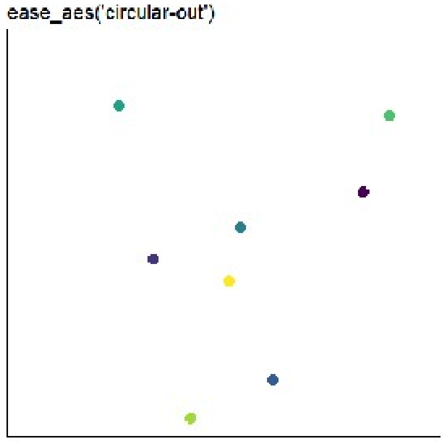
ease_aes(‘sine-out’) scatter plot



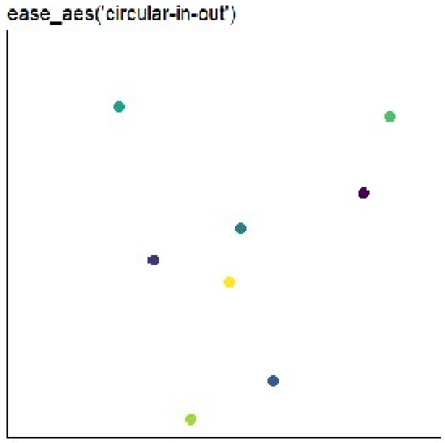
ease_aes(‘sine-in-out’) scatter plot



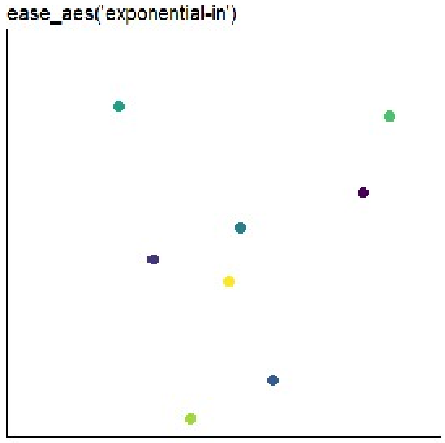
ease_aes(‘circular-in’) scatter plot



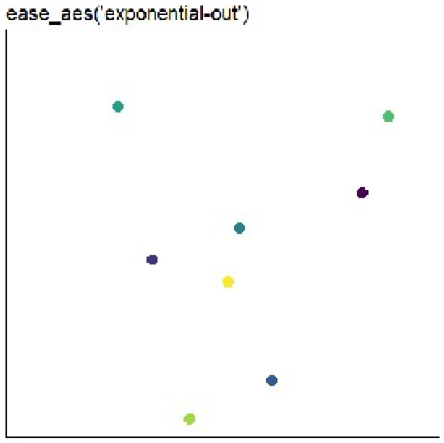
ease_aes(‘circular-out’) scatter plot



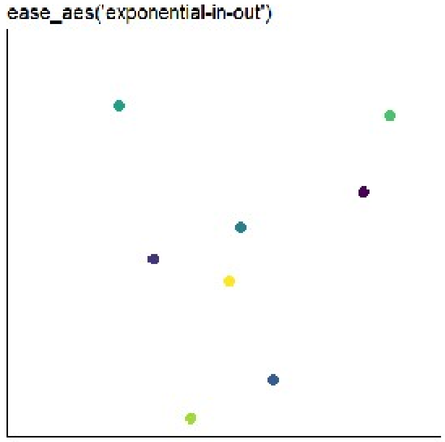
ease_aes(‘circular-in-out’) scatter plot



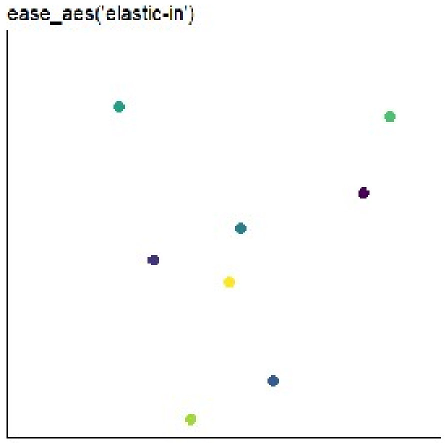
ease_aes('exponential-in') scatter plot



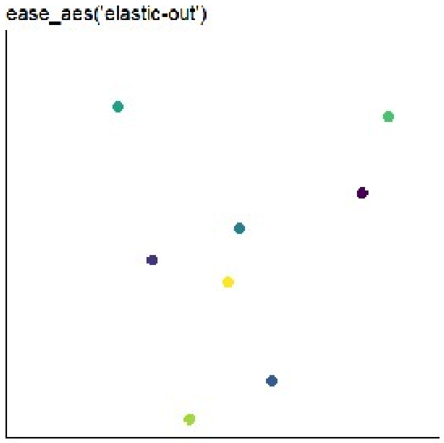
ease_aes('exponential-out') scatter plot



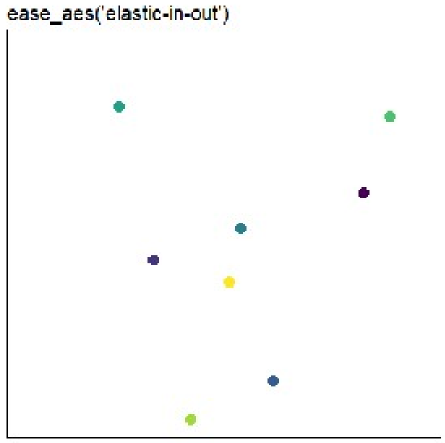
ease_aes('exponential-in-out') scatter plot



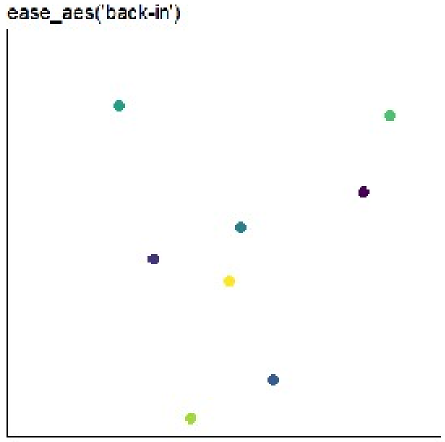
ease_aes('elastic-in') scatter plot



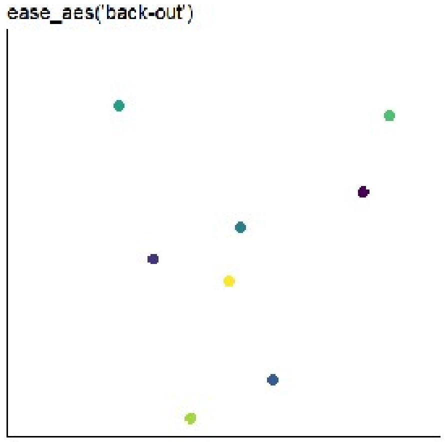
ease_aes('elastic-out') scatter plot



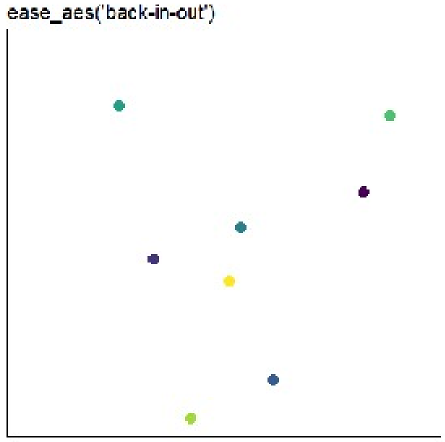
ease_aes('elastic-in-out') scatter plot



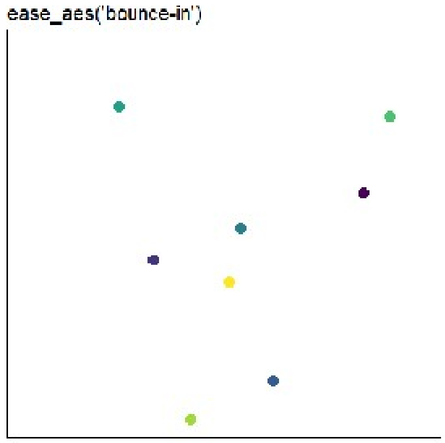
ease_aes('back-in') scatter plot



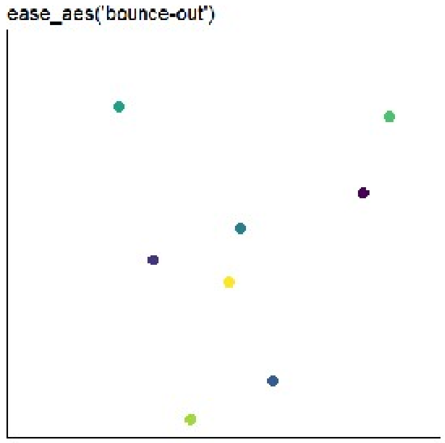
ease_aes('back-out') scatter plot



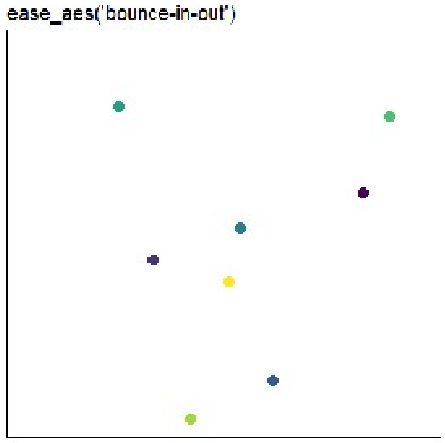
ease_aes('back-in-out') scatter plot



ease_aes('bounce-in') scatter plot



ease_aes('bounce-out') scatter plot

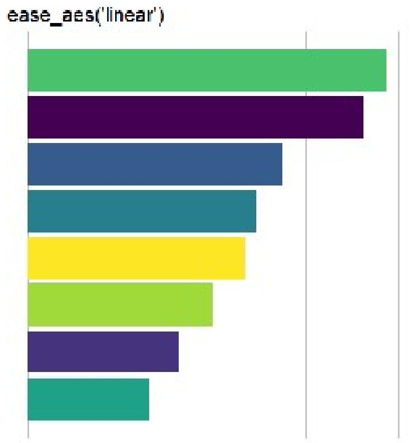


ease_aes('bounce-in-out') scatter plot

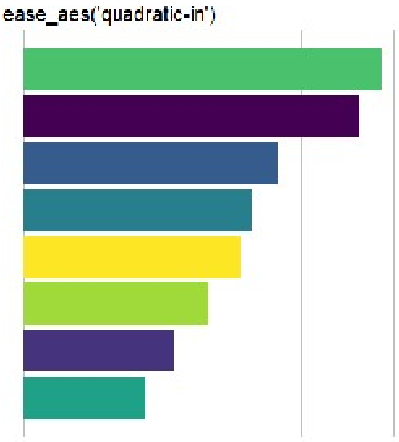
Bar Charts

Here is the code used for the *ease_aes('cubic-in')* animated bar chart:

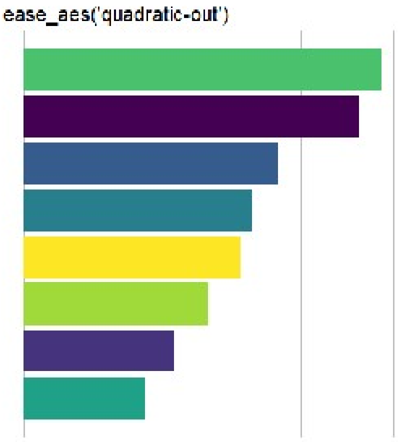
```
# load libraries
library(gganimate)
library(tidyverse)

# the data file format is indicated above
data <- read.csv('Data.csv')

# convert date to proper format
data$Ord <- as.Date(data$OrdDat, format='%m/%d/%Y')

# specify the animation length and rate
options(gganimate.nframes = 30)
options(gganimate.fps = 10)

# loop the animation
options(gganimate.end_pause = 0)

# specify the data source
ggplot(data) +

# specify the plot format
    theme(panel.background = element_rect(fill = 'white'))+
    theme(panel.grid.major.x  = element_line(color='gray'))+
    theme(axis.text = element_blank())+
    theme(axis.ticks = element_blank())+
    theme(axis.title = element_blank()) +
    theme(plot.title = element_text(size = 20)) +
    theme(plot.margin = margin(25, 25, 25, 25)) +
    theme(legend.position = 'none') +

# specify the x and y plot limits
        aes(xmin = 0, xmax=X+2) +
        aes(ymin = Rank-.45,
        ymax = Rank+.45,
        y = Rank) +

# create a bar chart
    geom_rect() +

# indicate the fill color scale
    scale_fill_viridis_d(option = "D", begin = 0, end = 1) +

# place larger values at the top
    scale_y_reverse() +

# apply the fill to the 'Cat' variable
    aes(fill = Cat) +

# animate the plot on the basis of the 'Ord' variable
    transition_time(Ord) +

# the ease_aes() function
    ease_aes('cubic-in') +

# title the plot
    labs(title = "'ease_aes(cubic-in)'")
```



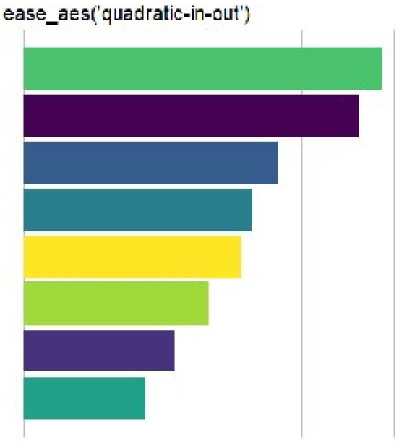
This is the default value, equivalent to *ease_aes()*. It is the only easing function that does not take a modifier.



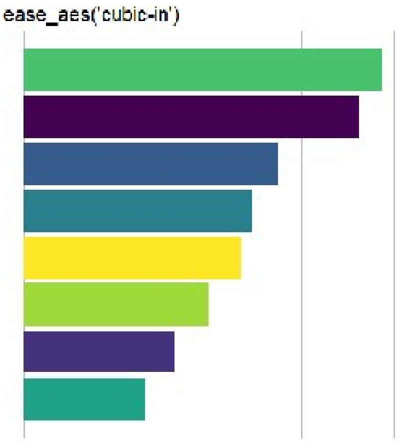
ease_aes(‘quadratic-in’) bar chart



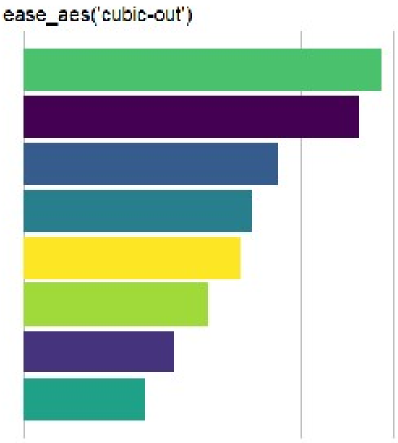
ease_aes(‘quadratic-out’) bar chart



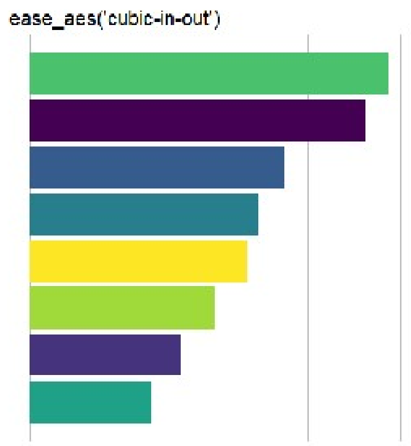
ease_aes(‘quadratic-in-out’) bar chart



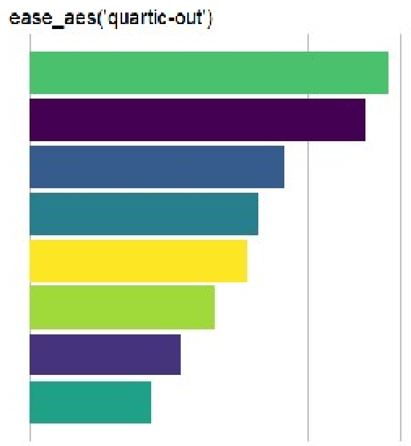
ease_aes(‘cubic-in’) bar chart



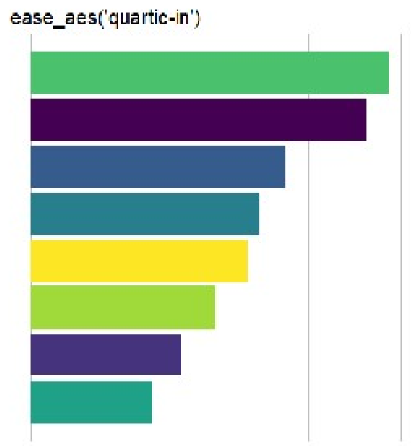
ease_aes(‘cubic-out’) bar chart



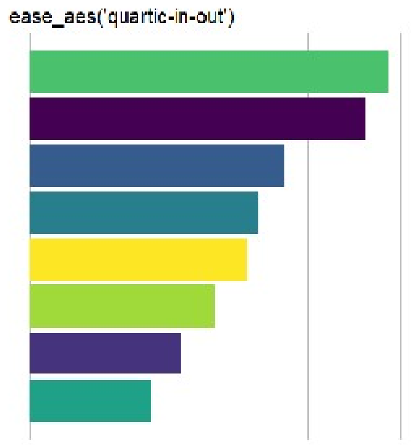
ease_aes('cubic-in-out') bar chart



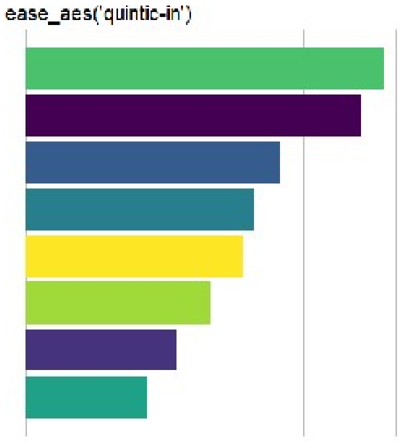
ease_aes('quartic-out') bar chart



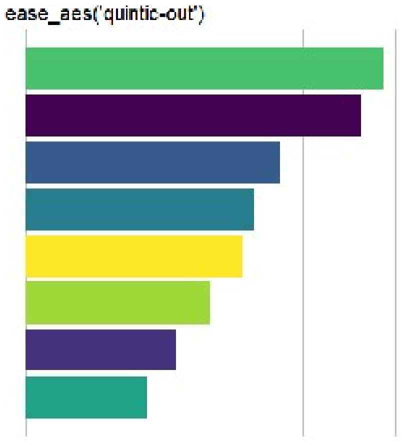
ease_aes('quartic-in') bar chart



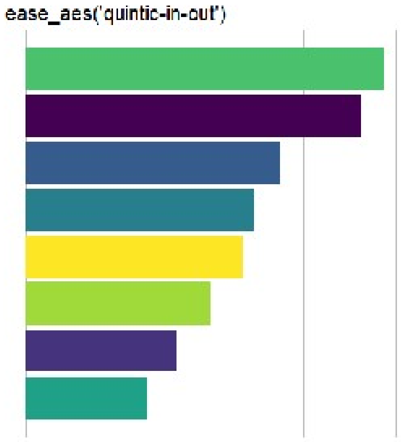
ease_aes('quartic-in-out') bar chart



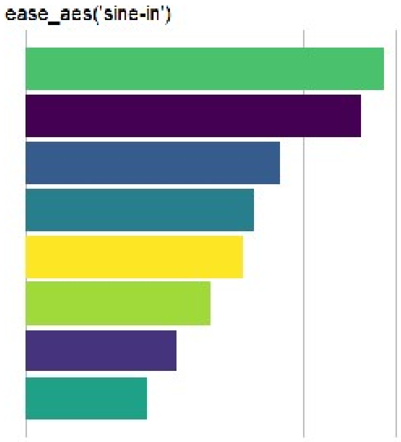
ease_aes('quintic-in') bar chart



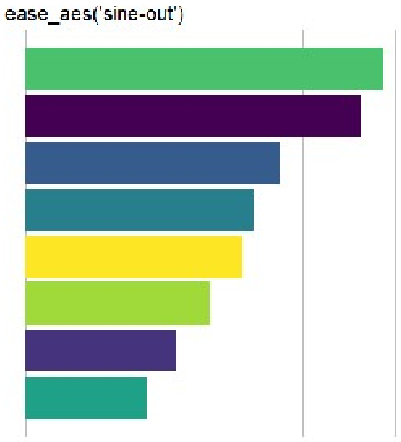
ease_aes('quintic-out') bar chart



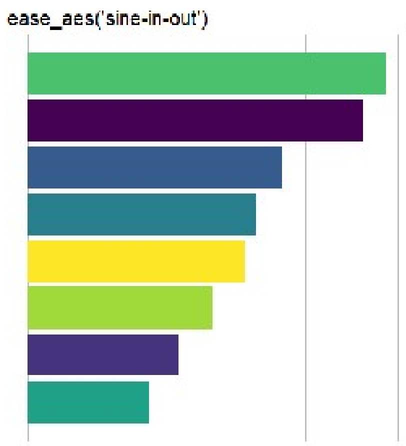
ease_aes('quintic-in-out') bar chart



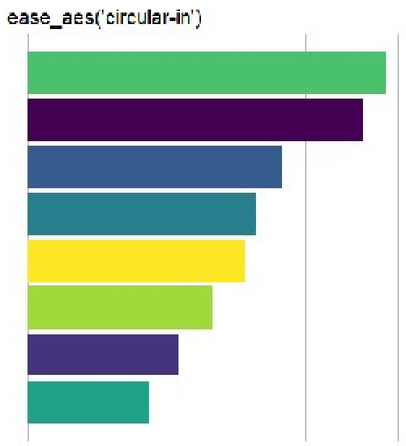
ease_aes('sine-in') bar chart



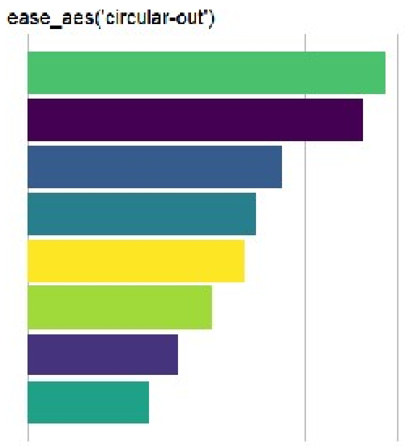
ease_aes('sine-out') bar chart



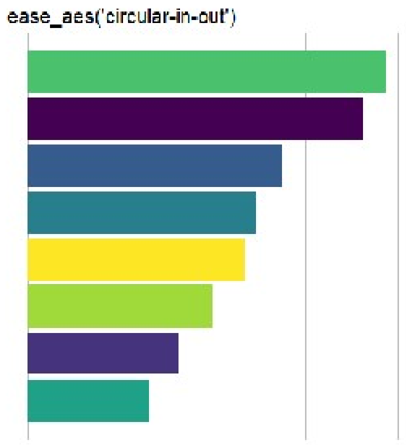
ease_aes('sine-in-out') bar chart



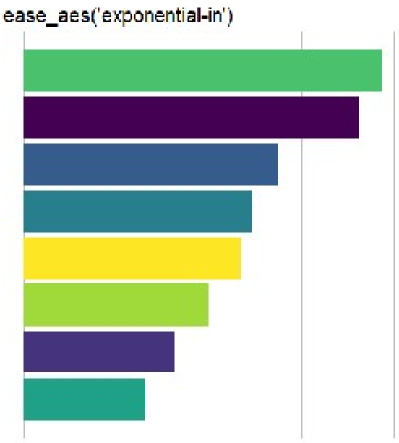
ease_aes('circular-in') bar chart



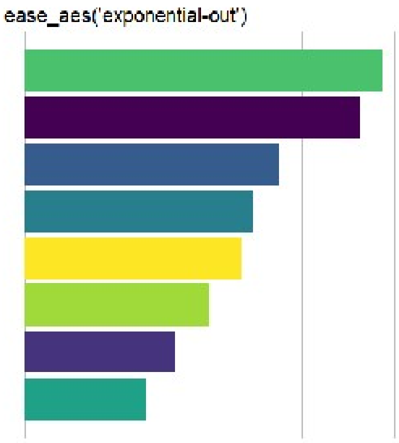
ease_aes('circular-out') bar chart



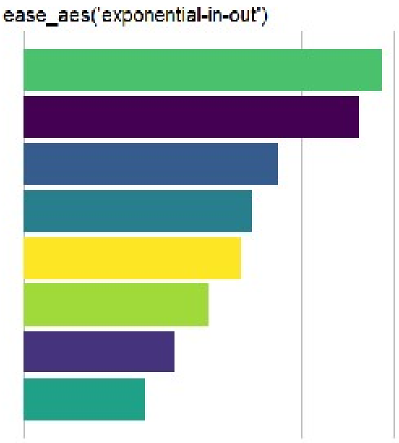
ease_aes('circular-in-out') bar chart



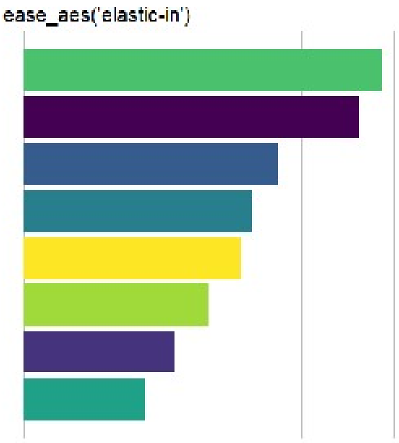
ease_aes(‘exponential-in’) bar chart



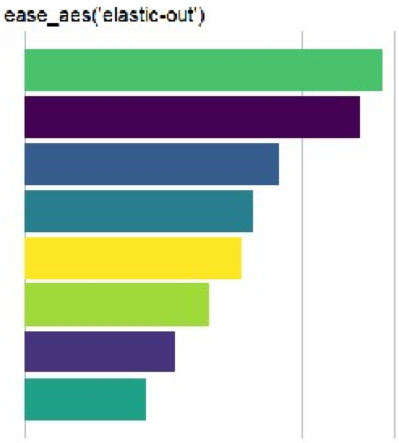
ease_aes(‘exponential-out’) bar chart



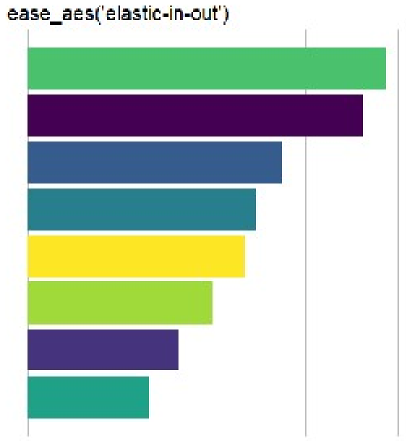
ease_aes(‘exponential-in-out’) bar chart



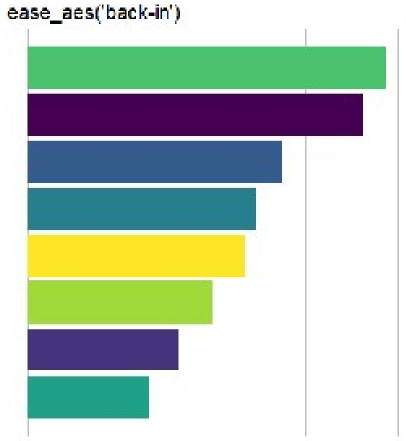
ease_aes(‘elastic-in’) bar chart



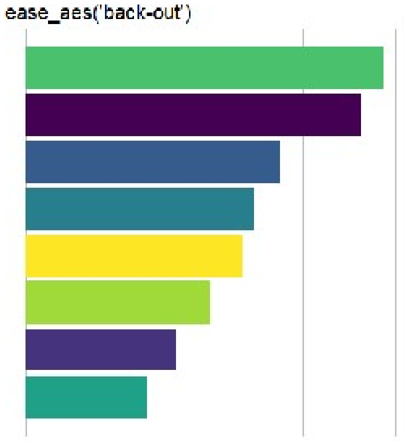
ease_aes(‘elastic-out’) bar chart



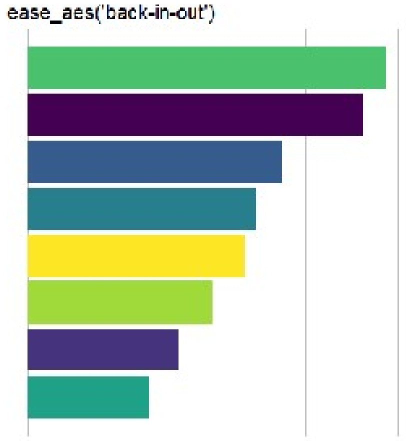
ease_aes('elastic-in-out') bar chart



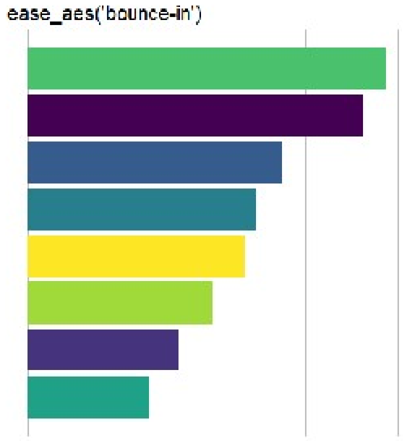
ease_aes('back-in') bar chart



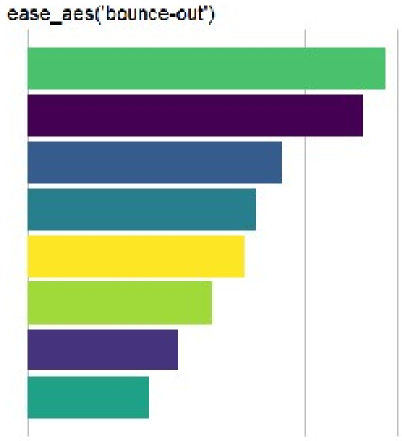
ease_aes('back-out') bar chart



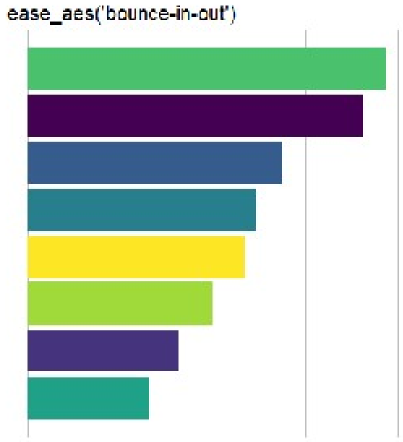
ease_aes('back-in-out') bar chart



ease_aes(‘bounce-in’) bar chart



ease_aes(‘bounce-out’) bar chart



ease_aes(‘bounce-in-out’) bar chart