This tutorial covers various ways you can create animated charts or plots using R. Animation is a very important element of data visualization. Animated charts are visually appealing and it fetches attention of audience. There are many online data visualization tools available in market which can generate animated charts but most of them are paid tools. Also problem with the online animation tools is that it asks you to upload data to their server, which is a data breach if you work on a real-world data of your client. Since R is open-source, you can download it for free and can create animated charts without moving data to server of any external server.

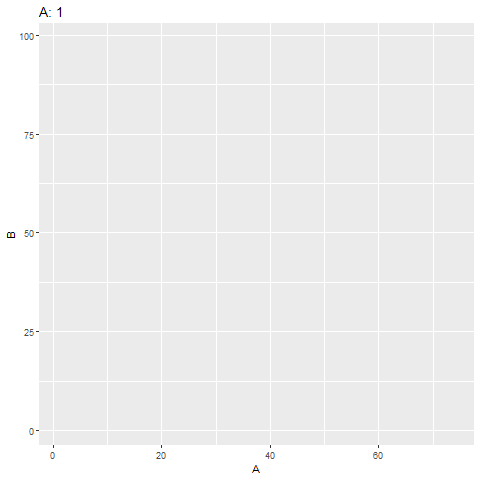
**Simple Animation in R**

Let’s create dummy data for illustration. In the program below, we are generating 3 columns containing some random observations. First column named A contains 50 observations ranging from 1 to 75. Similarly second column contains similar number of observations but range interval is different.

df = data.frame(A=sample(1:75, 50, replace=TRUE),  
 B=sample(1:100, 50, replace=TRUE),  
 stringsAsFactors = FALSE)

gganimate package is used for animation in R. It is an extension of popular package for graphics – ggplot2 package.

library(ggplot2)  
library(tidyverse)  
library(gganimate)  
library(directlabels)  
library(png)  
library(transformr)  
library(grid)  
  
ggplot(df, aes(A, B)) +  
 geom\_line() +  
 transition\_reveal(A) +  
 labs(title = 'A: {frame\_along}')

[](https://i0.wp.com/1.bp.blogspot.com/-xReGLuyxaOs/XOhAQQW1vmI/AAAAAAAAHkU/ghCR5mO27n0owu0mvbWeVe6UQzsCN68KACLcBGAs/s1600/basic_animation.gif?ssl=1)

geom\_line() is used for creating line chart. transition\_reveal(A) allows you to let data gradually appear.frame\_along gives the position that the current frame corresponds to.

**What is frame and rendering in animation?**

In animation, a frame is one of the many still images which compose the complete moving picture. Rendering is a kind of computing to output the final result. In gganimate package, it is by default 100 frames to render. You can change the number of frames under nframes= parameter in animatefunction.

p = ggplot(df, aes(A, B, group = C)) +  
 geom\_line() +  
 transition\_reveal(A) +  
 labs(title = 'A: {frame\_along}')  
  
**animate(p, nframes=40)**

**How to save animated plot in GIF format file?**

You can use anim\_save(file\_location,plot) function to export animated chart in GIF format.

anim\_save("basic\_animation.gif", p)

**Frames per Second (fps)**

It is the amount of time spend on each frame per second. You can use parameter fps in animate() function. By default, it is 10 frames per second.

animate(p, nframes=40, fps = 2)

Decreasing fps from 10 means slowing down speed of animation.

**How to stop loop in animation?**

Loop means continuously repeating animation over and over again. To end loop, you can use renderer = gifski\_renderer(loop = FALSE) option in animate function.

animate(p, renderer = gifski\_renderer(loop = FALSE))

**How to change layout of plot?**

You can change height and width of plot by mentioning the size in animate( ) function.

animate(p, fps = 10, duration = 14, width = 800, height = 400)

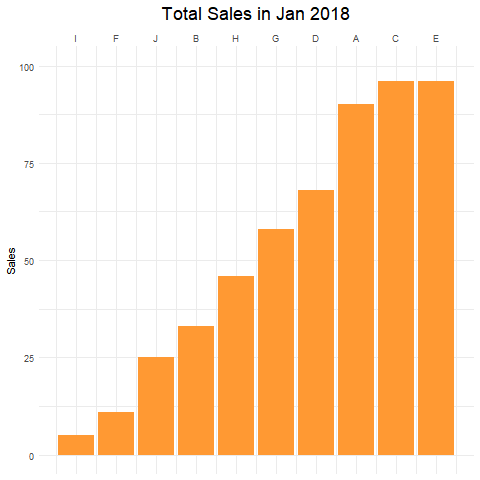
**Advanced Animation in R : Examples**

**Prepare Data for Example**  
In this example, we will create bar chart showing change in monthly sales figure of different products.

set.seed(123)  
dates = paste(rep(month.abb[1:10], each=10), 2018)  
df = data.frame(Product=rep(sample(LETTERS[1:10],10), 10),  
 Period=factor(dates, levels=unique(dates)),  
 Sales=sample(1:100,100, replace = TRUE))

**head(df)**  
 Product Period Sales order  
1 E Jan 2018 15 1  
2 H Jan 2018 34 2  
3 F Jan 2018 42 3  
4 E Jan 2018 49 4  
5 J Jan 2018 49 5  
6 C Jan 2018 60 6

# Ranking by Period and Sales  
df = df %>%   
 arrange(Period, Sales) %>%   
 mutate(order = 1:n())  
  
# Animation  
p = df %>%   
 ggplot(aes(order, Sales)) +  
 geom\_bar(stat = "identity", fill = "#ff9933") +  
 labs(title='Total Sales in {closest\_state}', x=NULL) +  
 theme(plot.title = element\_text(hjust = 0.5, size = 18)) +  
 scale\_x\_continuous(breaks=df$order, labels=df$Product, position = "top") +  
 transition\_states(Period, transition\_length = 1, state\_length = 2) +  
 view\_follow(fixed\_y=TRUE) +  
 ease\_aes('cubic-in-out')  
  
animate(p, nframes=50, fps=4)  
anim\_save("bar\_animation.gif", p)

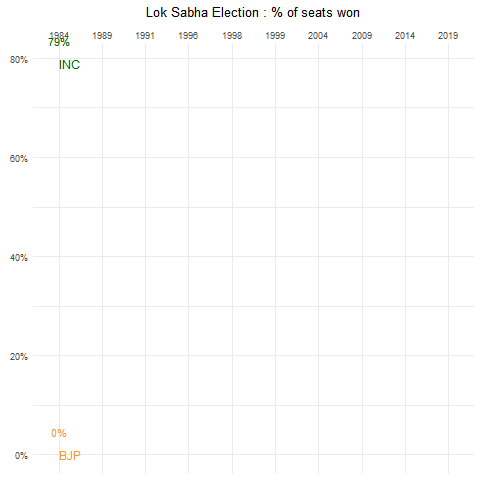
[](https://i0.wp.com/1.bp.blogspot.com/-CywNk_NXUdU/XOh44MDBC3I/AAAAAAAAHks/ffOgjEjSnW8RgQNVXbP_jU736uWXfstIwCLcBGAs/s1600/bar_animation.gif?ssl=1)

**Detailed Explanation**

1. transition\_states() animates plot by categorical or discrete variable. “States” are the animation sequences which plays. When a state transition is triggered, there will be a new state whose animation sequence will then run. In this case, state is Period column. state\_length refers to relative length of the pause at the states. transition\_length refers to relative length of the transition.
2. view\_follow(fixed\_y=TRUE) means y-axis would be fixed when animation is running.
3. ease\_aes( ) refers to motion in animation that starts quickly and then decelerates. Or vice-versa.
4. You can set theme using theme\_set(theme\_minimal())

**Indian General Election (1984 to 2019) Study**

Recently BJP secured majority in Lok Sabha Election. In 1984, they contested first time in Lok Sabha Election. INC (Indian National Congress) used to be the biggest political party in India a decade ago. Here we will see the trend analysis on “% of seats won by these two parties) from 1984 to 2019. Source of Data : Election Commission of India

[](https://i0.wp.com/1.bp.blogspot.com/-_IdhdAE6Osc/XOh8M8qhSbI/AAAAAAAAHk4/t1fUII8U104pb74fa0kgHNKjJ_QwJI20wCLcBGAs/s1600/election.gif?ssl=1)

library(ggplot2)  
library(tidyverse)  
library(gganimate)  
library(directlabels)  
library(png)  
library(transformr)  
library(grid)  
  
# Read Data  
df = read.table(text =   
 " Year Perc\_Seats Party  
 1984 0.79 INC  
 1989 0.38 INC  
 1991 0.45 INC  
 1996 0.27 INC  
 1998 0.27 INC  
 1999 0.22 INC  
 2004 0.28 INC  
 2009 0.4 INC  
 2014 0.09 INC  
 2019 0.1 INC  
 1984 0 BJP  
 1989 0.17 BJP  
 1991 0.23 BJP  
 1996 0.31 BJP  
 1998 0.35 BJP  
 1999 0.35 BJP  
 2004 0.27 BJP  
 2009 0.23 BJP  
 2014 0.52 BJP  
 2019 0.56 BJP  
 ", header=TRUE)  
  
# Set Theme  
theme\_set(theme\_minimal())  
  
# Plot and animate  
p =   
 ggplot(data = df, aes(x= factor(Year), y=Perc\_Seats, group=Party, colour=Party)) +  
 geom\_line(size=2, show.legend = FALSE) +  
 scale\_color\_manual(values=c("#ff9933", "#006400")) +  
 scale\_x\_discrete(position = "top") +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1)) +  
 labs(title = 'Lok Sabha Election : % of seats won',   
 x = NULL, y = NULL) +  
 geom\_text(aes(label=scales::percent(Perc\_Seats, accuracy = 1),  
 vjust= -2), show.legend = FALSE) +  
 theme(plot.title = element\_text(hjust = 0.5)) +  
 geom\_dl(aes(label=Party), method="last.points") +  
 transition\_reveal(Year) +  
 coord\_cartesian(clip = 'off') +   
 ease\_aes('cubic-in-out')  
  
animate(p, fps = 10, width = 800, height = 400)  
anim\_save("election.gif", p)

**How to save animated plot as video**

Make sure ffmpeg is installed on your system before using the code below. It is available for download for all the operating systems.

animate(nations\_plot, renderer = ffmpeg\_renderer(), width = 800, height = 450)  
anim\_save("nations.mp4")