

Tutorial-11

Dr. Kate Saunders and Krisanat Anukarnsakulchularp

Table of contents

Interactive and Animated visualisations	1
Learning Objectives	1
Preparation	1
Motivation	2
Exercise 1	2
Exercise 2	6
Exercise 3	7
Finishing Up	7

Interactive and Animated visualisations

Learning Objectives

- Practice creating interactive visualisations
- Practice creating animated visualisations

Preparation

- Load the `gapminder` package. We will be using this data.
- Also download the `big-tech-stock-price.csv` data from Moodle.
- Be sure to have installed the `plotly` and `gganimate` packages

Motivation

Interactive and animated visualisations can improve the audience's understanding of the data visualisation and its key messages.

Both interactivity and animation help show details on demand, reducing the overall cognitive load, and providing options for a more personalised view.

In this tutorial, we'll practice creating interactive and animated visualisations.

Exercise 1

- Load the plotly package
- Turn the Apple stock price plot from last lecture into an interactive visualisation.
- Change the tooltip, add new information or change the information shown.

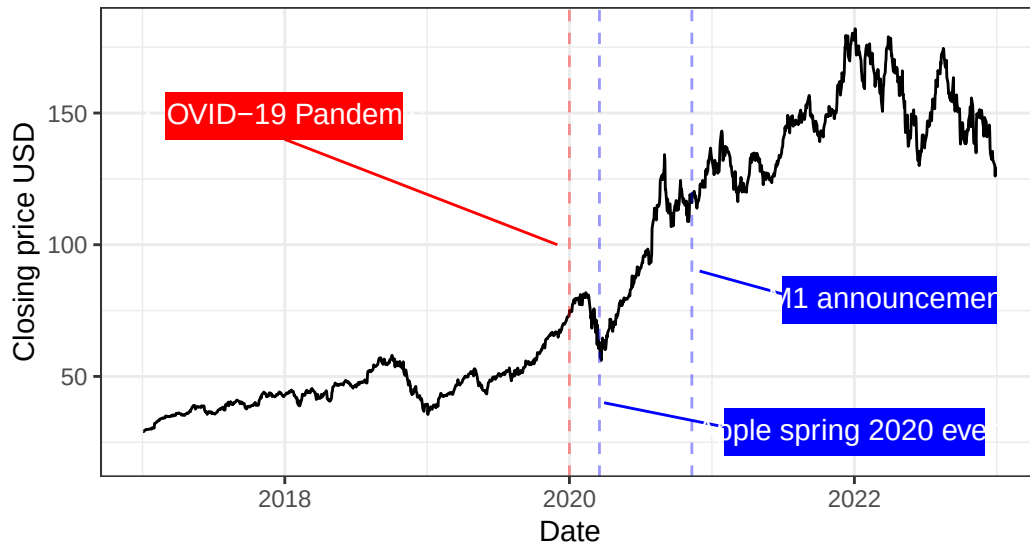
Code to reproduce the plot

```
library(tidyverse)

# you may need to change your data directory
stock <- read_csv("../data/big-tech-stock-price.csv") |>
  mutate(date = ymd(date)) |>
  filter(stock_symbol == "AAPL",
         year(date) > 2016)

apple_stock_plot <- ggplot(stock, aes(x = date, y = close)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date("2020-01-01")), linetype = 2, color = "red", a
  geom_vline(xintercept = as.numeric(as.Date("2020-03-18")), linetype = 2, color = "blue",
  geom_vline(xintercept = as.numeric(as.Date("2020-11-10")), linetype = 2, color = "blue",
  # spring
  annotate("rect", xmin = as.Date("2021-02-01"), xmax = as.Date("2022-12-01"), ymin = 20, y
  annotate("segment", x = as.Date("2020-03-30"), xend = as.Date("2021-03-01"), y = 40, yend
  annotate("text", x = as.Date("2022-01-01"), y = 30, label = "Apple spring 2020 event", co
  # m1
  annotate("rect", xmin = as.Date("2021-07-01"), xmax = as.Date("2023-01-01"), ymin = 70, y
  annotate("segment", x = as.Date("2020-11-30"), xend = as.Date("2021-08-01"), y = 90, yend
  annotate("text", x = as.Date("2022-04-01"), y = 80, label = "M1 announcement", color = "v
  # covid
  annotate("rect", xmin = as.Date("2017-03-01"), xmax = as.Date("2018-11-01"), ymin = 140,
  annotate("segment", x = as.Date("2018-01-01"), xend = as.Date("2019-12-01"), y = 140, yend
  annotate("text", x = as.Date("2018-01-01"), y = 150, label = "COVID-19 Pandemic", color =
  labs(x = "Date", y = "Closing price USD", title = "Apple Inc stock price during the pande
  theme_bw() +
  theme(aspect.ratio = 0.5,
        plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
```

Apple Inc stock price during the pandemic



Solutions

I decided to add the difference between the open and close price to my tooltip. To do this:

- I added a new variable `price_change` using `mutate`
- I only show 2 significant figures for this variable using `signif` so it will display nicely
- I add the `price_change` variable to my aesthetic mapping so I can use it in the tooltip

```
apple_stock_plot <- stock |>
  mutate(price_change = signif(close - open), 2) |>
  ggplot(aes(x = date, y = close, price_change = price_change)) +
  geom_line() +
  geom_vline(xintercept = as.numeric(as.Date("2020-01-01")), linetype = 2, color = "red", )
  geom_vline(xintercept = as.numeric(as.Date("2020-03-18")), linetype = 2, color = "blue", )
  geom_vline(xintercept = as.numeric(as.Date("2020-11-10")), linetype = 2, color = "blue", )
  # spring
  annotate("rect", xmin = as.Date("2021-02-01"), xmax = as.Date("2022-12-01"), ymin = 20, ymax = 40, fill = "red", )
  annotate("segment", x = as.Date("2020-03-30"), xend = as.Date("2021-03-01"), y = 40, yend = 20, color = "red", )
  annotate("text", x = as.Date("2022-01-01"), y = 30, label = "Apple spring 2020 event", color = "red", )
  # m1
  annotate("rect", xmin = as.Date("2021-07-01"), xmax = as.Date("2023-01-01"), ymin = 70, ymax = 90, fill = "blue", )
  annotate("segment", x = as.Date("2020-11-30"), xend = as.Date("2021-08-01"), y = 90, yend = 70, color = "blue", )
  annotate("text", x = as.Date("2022-04-01"), y = 80, label = "M1 announcement", color = "blue", )
  # covid
  annotate("rect", xmin = as.Date("2017-03-01"), xmax = as.Date("2018-11-01"), ymin = 140, ymax = 150, fill = "red", )
  annotate("segment", x = as.Date("2018-01-01"), xend = as.Date("2019-12-01"), y = 140, yend = 150, color = "red", )
  annotate("text", x = as.Date("2018-01-01"), y = 150, label = "COVID-19 Pandemic", color = "red", )
  labs(x = "Date", y = "Closing price USD", title = "Apple Inc stock price during the pandemic")
  theme_bw() +
  theme(aspect.ratio = 0.5,
        plot.title = element_text(size = 16, face = "bold", hjust = 0.5)
  )
```

Then using plotly I can make this an interactive plot.

```
library(plotly)

ggplotly(apple_stock_plot, tooltip = "price_change")
```

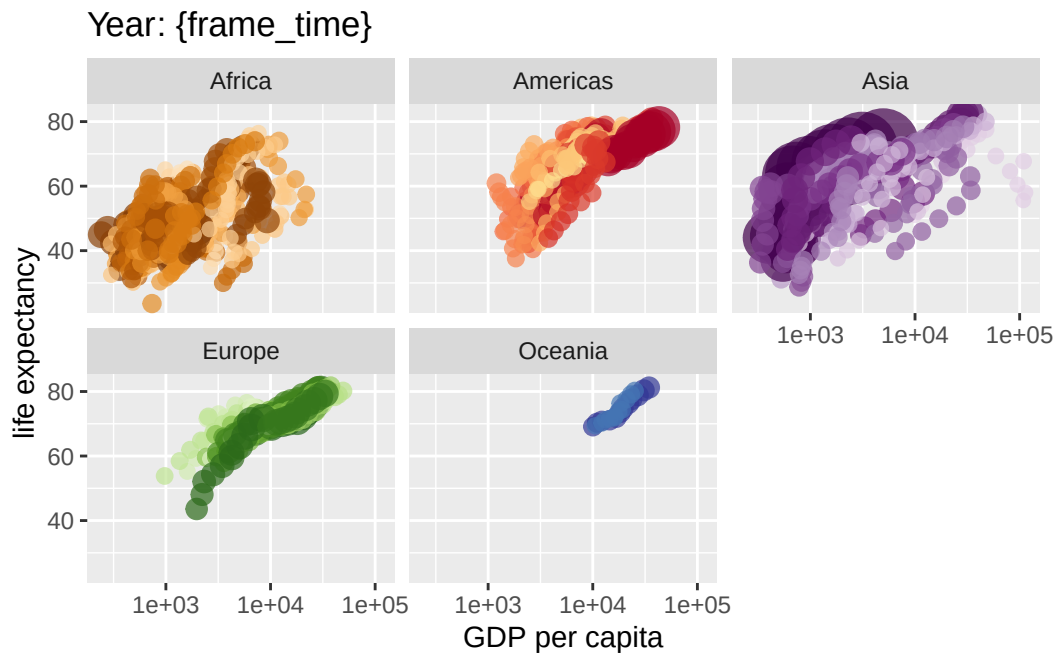
! Important

ggplotly and label were incompatible until recently (2025).

If everything is in your R is updated your labels should display. If not they may not appear.

The teaching team has not yet identified the exact packages/updates needed to get the labels to display. So don't worry if your labels don't display. The point of this exercise is for you to alter the tooltip.

Exercise 2



Using the code provided fill in the missing blanks ??? to turn the above plot into an animation that shows how the `gdpPercap` changes through time.

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

p <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  ???(alpha = 0.7) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_???() +
  ???(~continent) +
  theme(legend.position = 'none') +
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
  ???

p
```

Solutions

```
library(gapminder)
library(gganimate)

p <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  facet_wrap(~continent) +
  theme(legend.position = 'none') +
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
  transition_time(year) +
  ease_aes('linear')

p
```

Exercise 3

Enhance one of your visualisations from Assignment 1 using interactivity or animation.

Remember, be careful using these tools. Think about how interactivity and animation can be used to enhance your key messages! We want to avoid creating overly complex visualisations.

Finishing Up

- Make sure you understand how to turn a ggplot object into an interactive graphic
- Makes sure you understand the grammar of animation (`transition_*()`)
- Think about when is an appropriate time to create interactive and animated visualisations.
- Lastly, we don't need complex visualisations, sometimes, the simpler, the better.

Material developed by Dr. Kate Saunders and Krisanat Anukarnsakulchularp.

© Copyright 2024 Monash University