Chapter 04: Collective geoms

Stan Piotrowski

September 27 2021

Contents

Setup

This chapter focuses on collective geoms, or displaying multiple observations in a single object (contrast to individual geoms, which display a single observation or row as a single object).

```r
# Libraries
library(tidyverse)
library(nlme)
library(ggtext)
library(patchwork)

# Set ggplot2 theme
my_theme <- theme(
  panel.grid = element_blank(),
  panel.background = element_rect(fill = "white", color = "black"),
  axis.title = element_markdown(),
  plot.title = element_markdown(),
  plot.subtitle = element_markdown(),
  plot.caption = element_markdown()
)
```

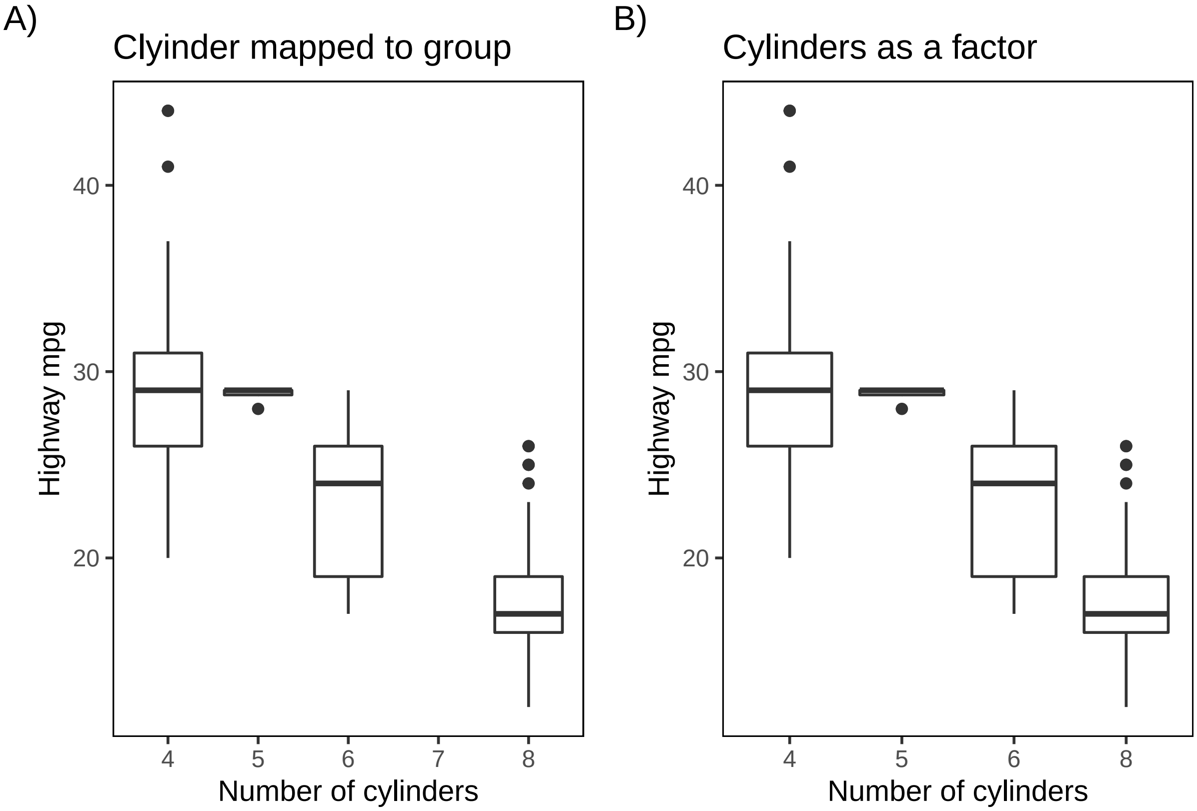
Exercises

1) *Draw a boxplot of `hwy` for each value of `cyl`, without turning `cyl` into a factor. What extra aesthetic to you need to set?*

```r
# Mapping cylinder to group
p1 <- mpg %>%
  ggplot(aes(cyl, hwy, group = cyl)) +
  geom_boxplot() +
  labs(x = "Number of cylinders",
       y = "Highway mpg",
       title = "Clyinder mapped to group") +
  my_theme

# Converting cylinder as a factor
p2 <- mpg %>%
  ggplot(aes(as.factor(cyl), hwy)) +
  geom_boxplot() +
  labs(x = "Number of cylinders",
       y = "Highway mpg",
       title = "Cylinders as a factor") +
  my_theme

p1 + p2 +
  plot_annotation(tag_levels = "A",
                  tag_suffix = ")")
```



In order to plot the highway fuel economy as a function of the number of cylinders, we need to set the `group` argument because the `cylinders` variable is continuous. By default, the `group` aesthetic is mapped to a discrete variable.

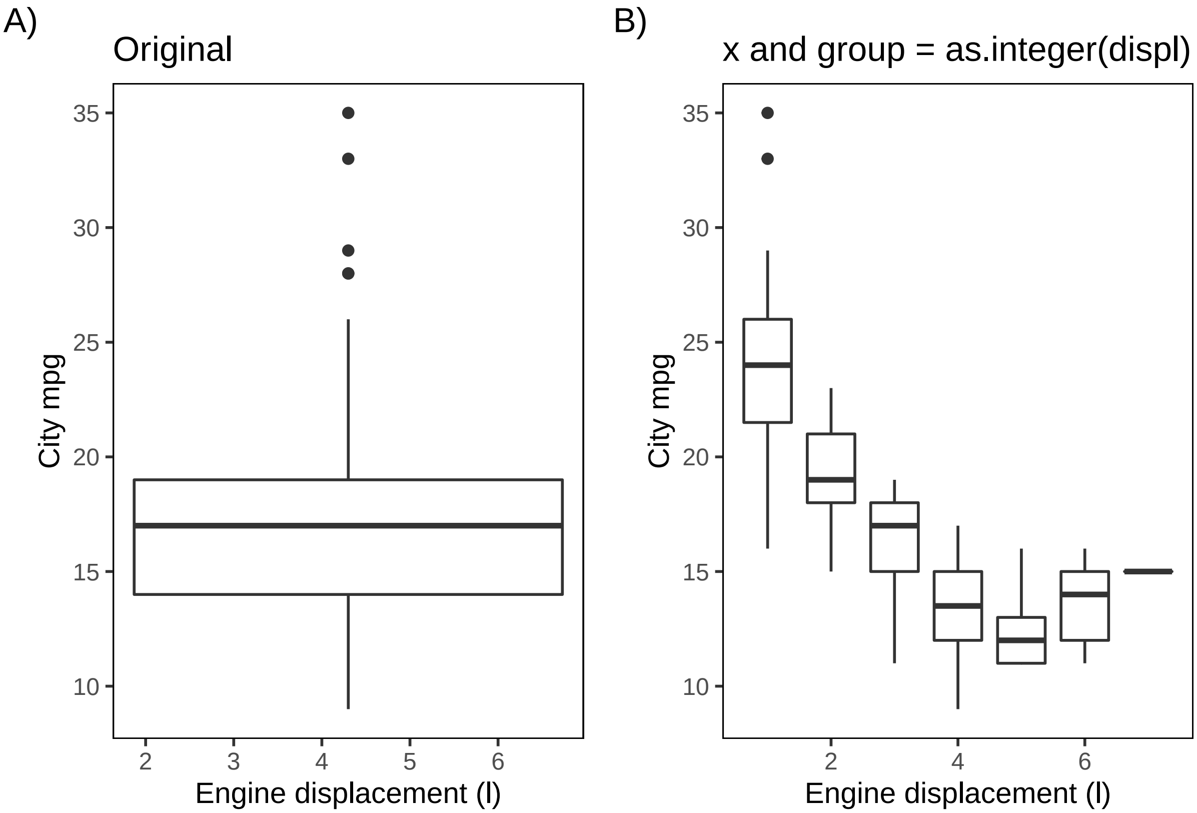
2) *Modify the following plot so that you get one boxplot per integer value of `displ`.*

```
# Original plot
p1 <- mpg %>%
  ggplot(aes(displ, cty)) +
  geom_boxplot() +
  labs(x = "Engine displacement (l)",
       y = "City mpg",
       title = "Original") +
  my_theme

# Mpa displ to group aesthetic
p2 <- mpg %>%
  ggplot(aes(as.integer(displ), cty, group = as.integer(displ))) +
  geom_boxplot() +
```

```r
  labs(x = "Engine displacement (l)",
       y = "City mpg",
       title = "x and group = as.integer(displ)") +
  my_theme

p1 + p2 +
  plot_annotation(tag_levels = "A",
                  tag_suffix = ")")
```



3) *When illustrating the difference between mapping continuous and discrete colours to a line, the discrete example needed aes(group = 1). Why? What happens if that is omitted? What's the difference between aes(group = 1) and aes(group = 2)? Why?*

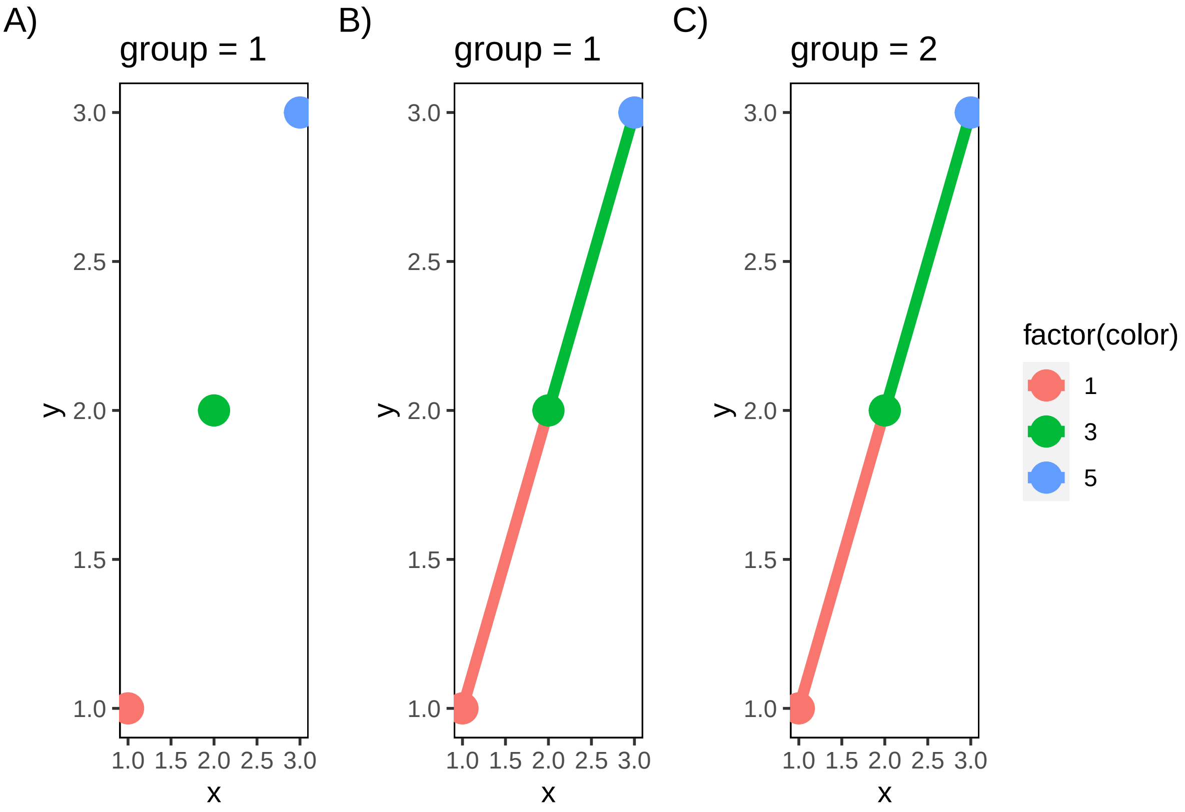
```r
# Create dummy data frame
df <- data.frame(
  x = 1:3,
  y = 1:3,
  color = c(1, 3, 5)
)

# No group
p1 <- df %>%
  ggplot(aes(x, y, color = factor(color))) +
  geom_line(size = 2) +
  geom_point(size = 5) +
  labs(title = "group = 1") +
  my_theme
```

```
# group = 1
p2 <- df %>%
  ggplot(aes(x, y, color = factor(color))) +
  geom_line(aes(group = 1), size = 2) +
  geom_point(size = 5) +
  labs(title = "group = 1") +
  my_theme

# group = 2
p3 <- df %>%
  ggplot(aes(x, y, color = factor(color))) +
  geom_line(aes(group = 2), size = 2) +
  geom_point(size = 5) +
  labs(title = "group = 2") +
  my_theme

p1 + p2 + p3 +
  plot_annotation(tag_levels = "A", tag_suffix = ")") +
  plot_layout(guides = "collect")
```



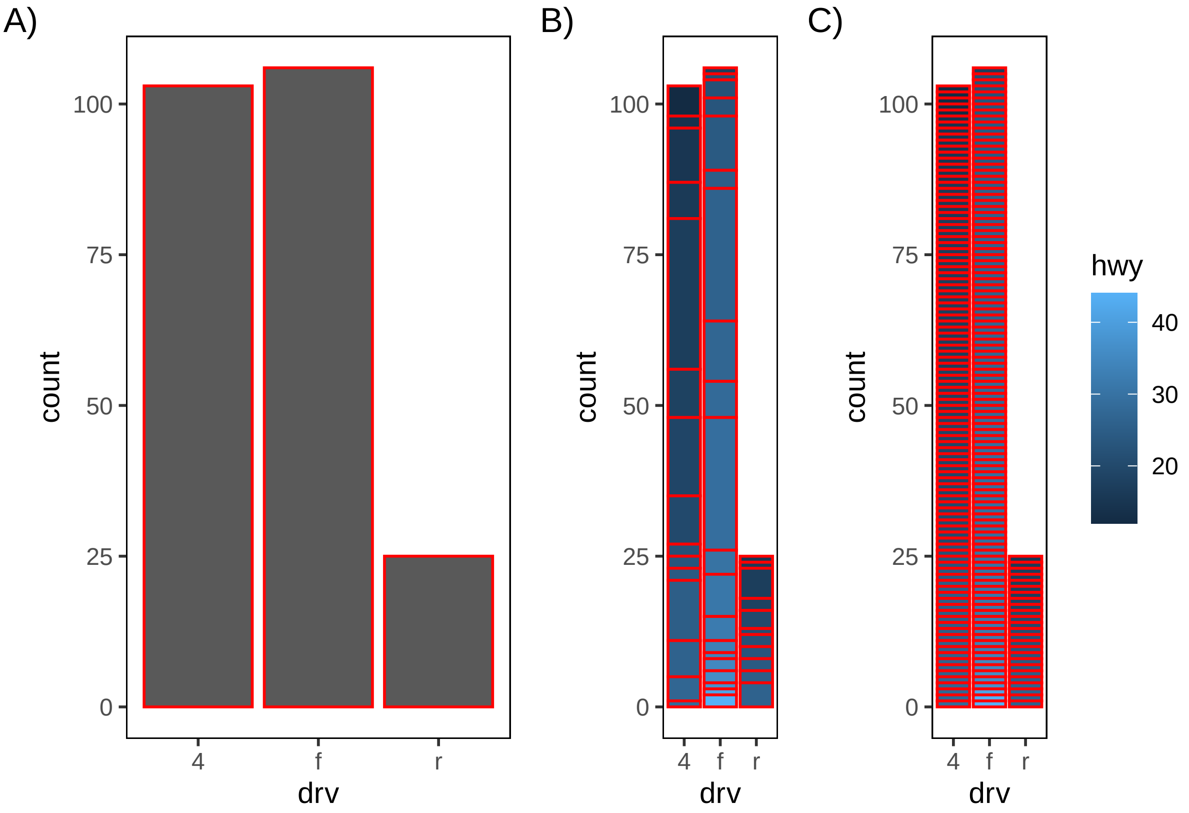
4) *How many bars are in each of the following plots?*

```
p1 <- mpg %>%
  ggplot(aes(drv)) +
  geom_bar(color = "red") +
  my_theme

p2 <- mpg %>%
  ggplot(aes(drv, fill = hwy, group = hwy)) +
  geom_bar(color = "red") +
  my_theme

p3 <- mpg %>%
  arrange(hwy) %>%
  mutate(id = seq_along(hwy)) %>%
  ggplot(aes(drv, fill = hwy, group = id)) +
  geom_bar(color = "red") +
  my_theme

p1 +
  (p2 + p3 + plot_layout(guides = "collect")) +
  plot_annotation(tag_levels = "A", tag_suffix = ")")
```



In plot A), there are only three bars because the group aesthetic maps to the discrete values of the drv variable. In plots B) and C), however, the three values of drv are still presented on the x-axis, but now the group aesthetic is mapped to continuous variables: in plot B), group maps to the hwy value for each observation in each class of drv, with the same hwy values binned into a single black; in plot C), group maps to the id variable for each observation in each class of drv. In plot C), the fill aesthetic is still mapped to the hwy variable, but the data are now grouped by id, so there is one bar for each id value.

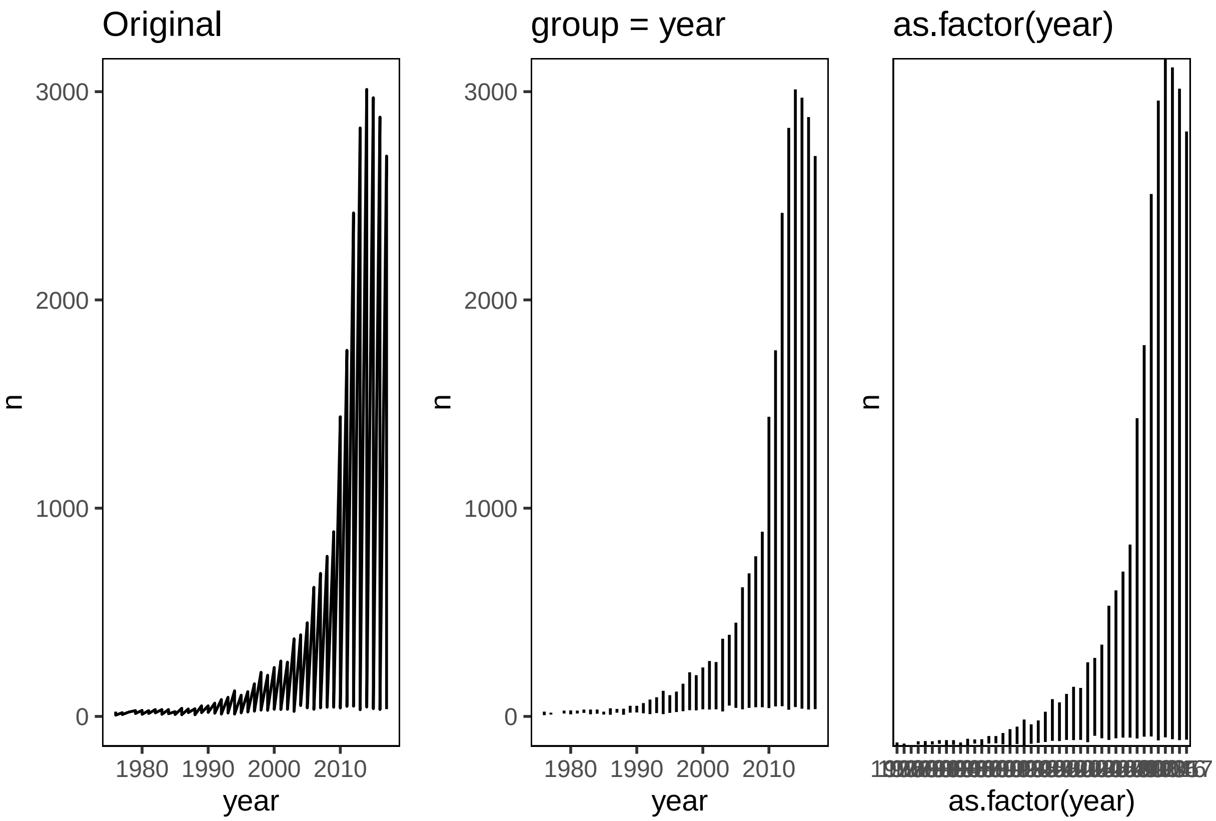
5) *Install the babynames package. It contains data about the popularity of babynames in the US. Run the following code and fix the resulting graph. Why does this graph make me unhappy?*

```
library(babynames)
p1 <- babynames %>%
  filter(name == "Hadley" & year > 1975) %>%
  ggplot(aes(year, n)) +
  geom_line() +
  labs(title = "Original") +
  my_theme

p2 <- babynames %>%
  filter(name == "Hadley" & year > 1975) %>%
  ggplot(aes(year, n, group = year)) +
  geom_line() +
  labs(title = "group = year") +
  my_theme

p3 <- babynames %>%
  filter(name == "Hadley" & year > 1975) %>%
  ggplot(aes(as.factor(year), n)) +
  geom_line() +
  scale_y_discrete(label = c("1980", "1990", "2000", "2010")) +
  labs(title = "as.factor(year)") +
  my_theme

p1 + p2 + p3
```



Original

group = year

as.factor(year)

The first plot uses the default `group` aesthetic mapping, which essentially creates a `geom_path()`, where there is a single connected line across all years. This is the characteristic sawtooth plot we saw earlier in the chapter. On the other hand, if we map the `group` aesthetic to the `year` explicitly, there is a single line drawn for each year.