Multiple layers, facetting and tidying your data

Presented by Emi Tanaka
Department of Econometrics and Business Statistics

6th Dec 2021 @ Statistical Society of Australia NSW Branch | Online
Add multiple layers

ggplot(<data>, aes(<mappings>)) +
<layer>() +
<layer>() +
<layer>()

Each layer inherits mapping and data from ggplot by default.

ggplot(penguins, aes(x = species, y = bill_length_mm)) +
  geom_violin() +
  geom_boxplot() +
  geom_point()
Order of the layers matters!

Boxplot and violin plot order are switched around.

ggplot(penguins, aes(species, bill_length_mm)) +
  geom_violin() +
  geom_boxplot() +
  geom_point()

ggplot(penguins, aes(species, bill_length_mm)) +
  geom_boxplot() +
  geom_violin() +
  geom_point()
Layer-specific data and aesthetic mappings

For each layer, aesthetic and/or data can be overwritten.

ggplot(penguins, aes(species, bill_length_mm)) +
  geom_violin(aes(fill = species)) +
  geom_boxplot(data = filter(penguins, species == "Adelie")) +
  geom_point(data = filter(penguins, species == "Gentoo"),
             aes(y = bill_depth_mm))
Aesthetic or Attribute?

Not what you want

```r
ggplot(penguins) +
  geom_point(aes(body_mass_g,
                  bill_depth_mm,
                  color = "blue"))
```

What you want

```r
ggplot(penguins) +
  geom_point(aes(body_mass_g,
                  bill_depth_mm),
              color = "blue")
```

```r
ggplot(penguins) +
  geom_point(aes(body_mass_g,
                  bill_depth_mm,
                  color = I("blue")))
```
This is an obvious case of Simpson's paradox.

What if we wanted to draw the fit of a simple linear model for each group?
```r
f <- ggplot(penguins, aes(bill_length_mm, bill_depth_mm, color = species)) + geom_point()

f

f + facet_wrap(~sex)

f + facet_grid(island ~ sex)
```
facet_wrap and facet_grid

\[ g + \text{facet_wrap}( \sim \text{sex}) \]

\[ g + \text{facet_grid}(\sim \text{sex}) \]

\[ g + \text{facet_wrap}( \sim \text{sex}, \text{ncol} = 1) \]

\[ g + \text{facet_grid}(\text{sex} \sim .) \]
Data Visualization with ggplot2 :: CHEAT SHEET

Basics

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same components: a data set, a coordinate system, and geoms—visual marks that represent data points.

To display values, map variables in the data to visual properties of the geom (aesthetics) like size, color, and x and y positions.

Geoms

GRAPHICAL PRIMITIVES

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>geom_blank()</td>
<td>(initial for expanding frames)</td>
</tr>
<tr>
<td>b</td>
<td>geom_point()</td>
<td>(x, y)</td>
</tr>
<tr>
<td>c</td>
<td>geom_bar()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_histogram()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_errorbar()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>f</td>
<td>geom_wallet()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>g</td>
<td>geom_add()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>h</td>
<td>geom_text()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>i</td>
<td>geom_table()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

TWO VARIABLES

continuous x, continuous y

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_point()</td>
<td>(x, y)</td>
</tr>
<tr>
<td>c</td>
<td>geom_smooth()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_histogram()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_text()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>f</td>
<td>geom_table()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

LINE SEGMENTS

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_line()</td>
<td>x, y, alpha, color, group,linetype, size</td>
</tr>
<tr>
<td>c</td>
<td>geom_polygon()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_area()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_errorbar()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>f</td>
<td>geom_trellis()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>g</td>
<td>geom_raster()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

ONE VARIABLE

continuous

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_histogram()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>c</td>
<td>geom_density2d()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_density()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_density()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

continuous bivariate distribution

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_density2d()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>c</td>
<td>geom_density()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_density()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_density()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

visualizing error

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_bar()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>c</td>
<td>geom_histogram()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_density()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_density()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

Maps

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_point()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>c</td>
<td>geom_raster()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_polygon()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_bar()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

Two variables

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_point()</td>
<td>(x, y)</td>
</tr>
<tr>
<td>c</td>
<td>geom_line()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_polygon()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_area()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>f</td>
<td>geom_bar()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

Three variables

<table>
<thead>
<tr>
<th>geom</th>
<th>function</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>geom_point()</td>
<td>(x, y)</td>
</tr>
<tr>
<td>c</td>
<td>geom_line()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>d</td>
<td>geom_polygon()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>e</td>
<td>geom_area()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
<tr>
<td>f</td>
<td>geom_bar()</td>
<td>x, y, alpha, color, group, linetype, size</td>
</tr>
</tbody>
</table>

RStudio > Help > Cheatsheets

- R4DS Community Slack
- Twitter with hashtag #rstats
- RStudio Community
- Stackoverflow
Tidying your data

Weight gain in pigs for different treatments

The crampton.pig is from the agridat 📦

```r
library(agridat)
glimpse(crampton.pig)
```

### Rows: 50
### Columns: 5

- **$ treatment**: T1, T1, T1, T1, T1, T1, T1, T1, T1, T1, T2, T2, T2, T2, T2,
- **$ rep**: R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R1, R2, R3, R4, R5
- **$ weight1**: 30, 21, 21, 33, 27, 24, 20, 29, 28, 26, 26, 24, 20, 35, 25,
- **$ feed**: 674, 628, 661, 694, 713, 585, 575, 638, 632, 637, 699, 626,
- **$ weight2**: 195, 177, 180, 200, 197, 170, 150, 180, 192, 184, 194, 204,

weight1 is initial weight and weight2 is final weight


Crampton and Hopkins (1934). The Use of the Method of Partial Regression in the Analysis of Comparative Feeding Trial Data, Part II. *The Journal of Nutrition* 8 113-123.
What are the mappings to get the above graph?

```r
ggplot(crampton.pig, aes(x = ???, y = ???)) +
  geom_point() +
  geom_line() +
  facet_grid(. ~ treatment)
```
The x-axis is the time when pig was weighed
- The y-axis is the weight
- The facetting is by treatment

How I wrangled this data

```r
pig_df <- crampton.pig %>%
  mutate(id = paste0("pig", 1:n())) %>%
  pivot_longer(c(weight1, weight2),
    names_to = "when",
    values_to = "weight") %>%
  mutate(when = factor(when,
    levels = c("weight1", "weight2"),
    labels = c("initial", "final")))
```

(note: teaching wrangling is not part of this workshop, please see here if you want to learn more)
Putting it all together

```r
ggplot(pig_df, aes(when, weight)) +  # tidying your data for plotting
  geom_point(size = 3) +  # attribute not aesthetic
  geom_line(aes(group = id)) +  # grouping
  facet_grid(. ~ treatment) +  # facetting
  labs(x = "")  # we'll learn this in the last session
```
Meaningfully order categorical variables

```r
ggplot(crampton.pig, 
aes(treatment, weight2 - weight1)) + 
geom_point(size = 3)
```

- Treatments are ordered alphabetically by default
- It's better to order categorical variables meaningfully

```
library(forcats) # for easy factor manipulation
crampton.pig2 <- crampton.pig %>%
  mutate(
    treatment = fct_reorder(treatment, 
      weight2 - weight1, 
      mean))
ggplot(crampton.pig2, 
aes(treatment, weight2 - weight1)) + 
geom_point(size = 3)
```
Plot you may want:

One way to do this:

```r
fig <- ggplot(crampton.pig2,
  aes(treatment,
       weight2 - weight1)) +
  geom_point(size = 3) +
  stat_summary(fun.data = mean_se,
                geom = "pointrange",
                fatten = 2,
                color = "#027EB6", size = 3) +
  stat_summary(fun = mean, geom = "line",
              group = 1, color = "#027EB6",
              size = 2)
```
Plotting annotations

```r
fig + geom_text(data = data.frame(treatment = 4.5, weight2 = 185, weight1 = 0),
                label = "Treatment\n means",
                size = 3,
                color = "#027EB6",
                fontface = "bold")
```

But it might be just easier to:

```r
fig + annotate("text",
               x = 4.5, y = 185,
               label = "Treatment\n means",
               size = 3,
               color = "#027EB6",
               fontface = "bold")
```
Open part1-exercise-02.Rmd

15:00
devtools::session_info()

```r
## ─ Session info
### hash: baby bottle, beach with umbrella, superhero: medium skin tone
###
### setting  value
### version  R version 4.1.2 (2021-11-01)
### os       macOS Big Sur 10.16
### system   x86_64, darwin17.0
### ui       X11
### language (EN)
### collate  en_AU.UTF-8
### ctype    en_AU.UTF-8
### tz       Australia/Melbourne
### date     2021-12-06
```

These slides are licensed under

[CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)