COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Data Science for Business Analytics

Lecture 5

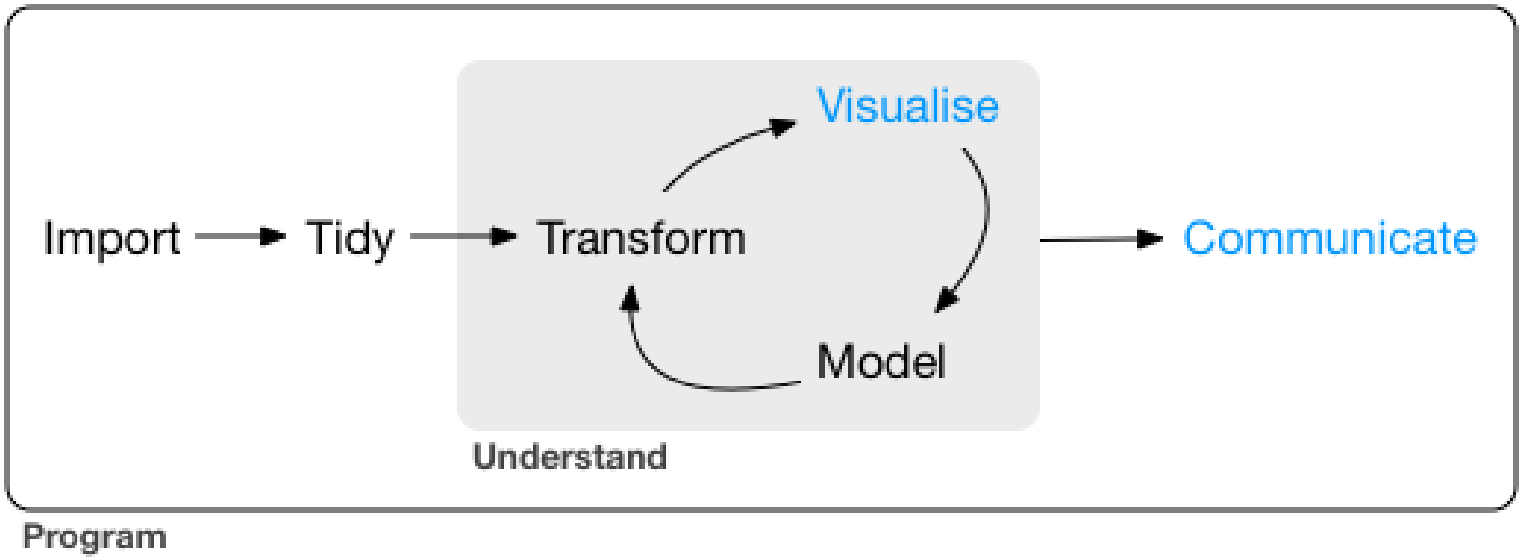
Thibault Vatter

Department of Statistics, Columbia University

03/16/2020

This morning

- Tibbles
- Strings



Most of the material (e.g., the picture above) is borrowed from

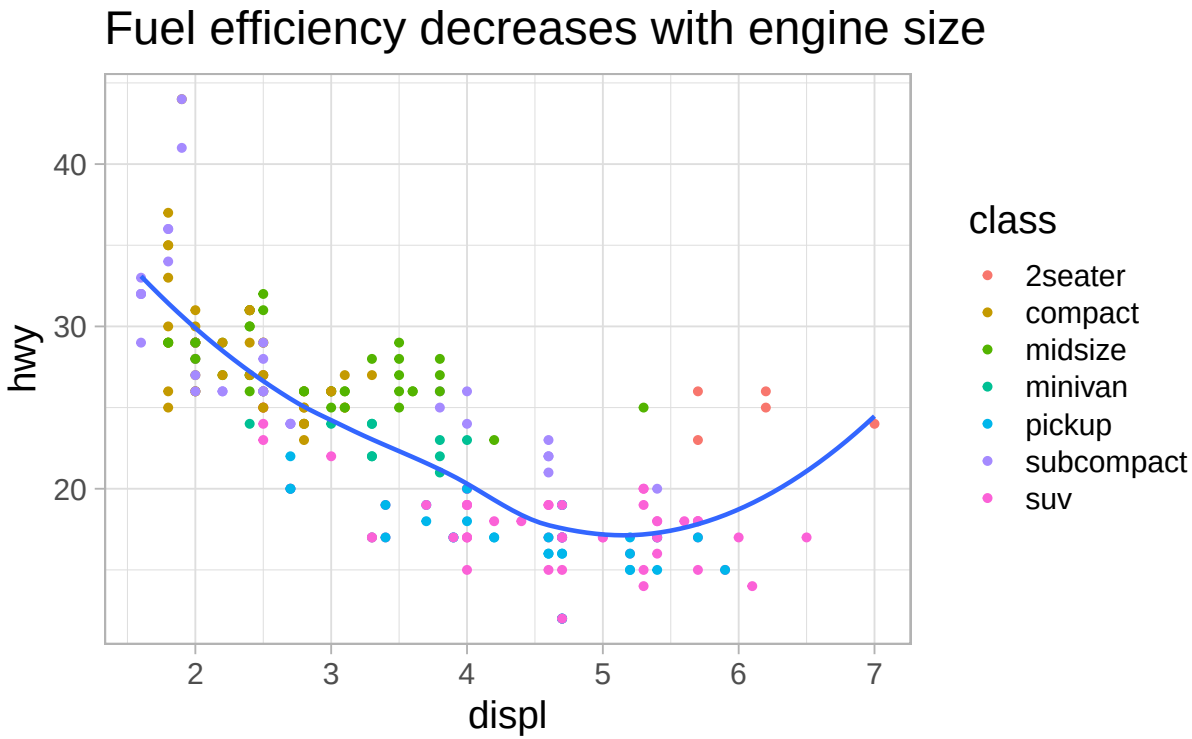
R for data science

Outline

Outline

Label


```
ggplot(mpg, aes(displ, hwy)) + geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  labs(title = "Fuel efficiency decreases with engine size")
```

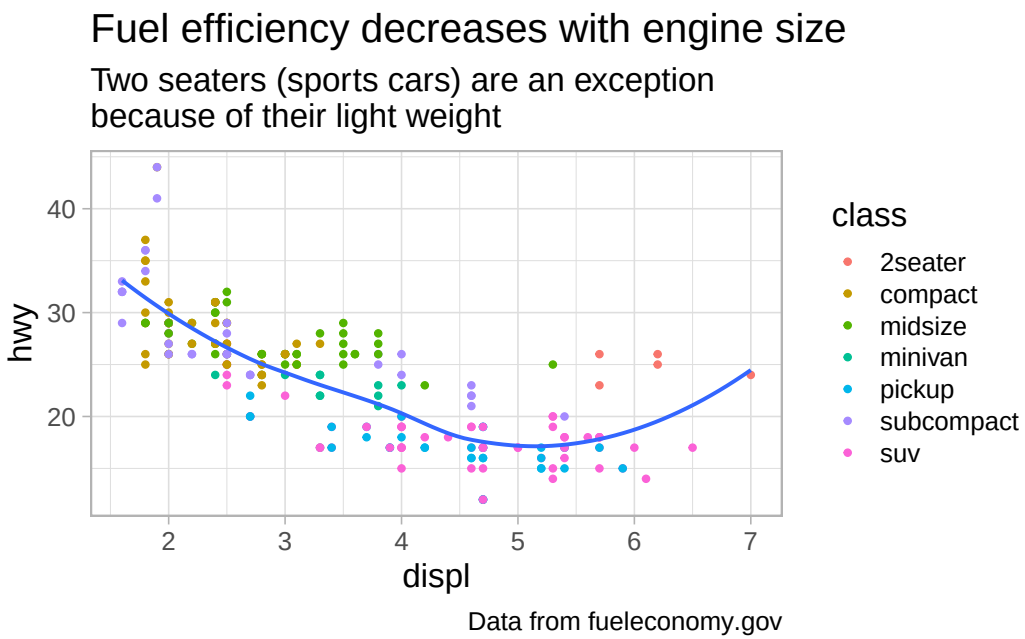


- Avoid titles that just describe what the plot is!

More text

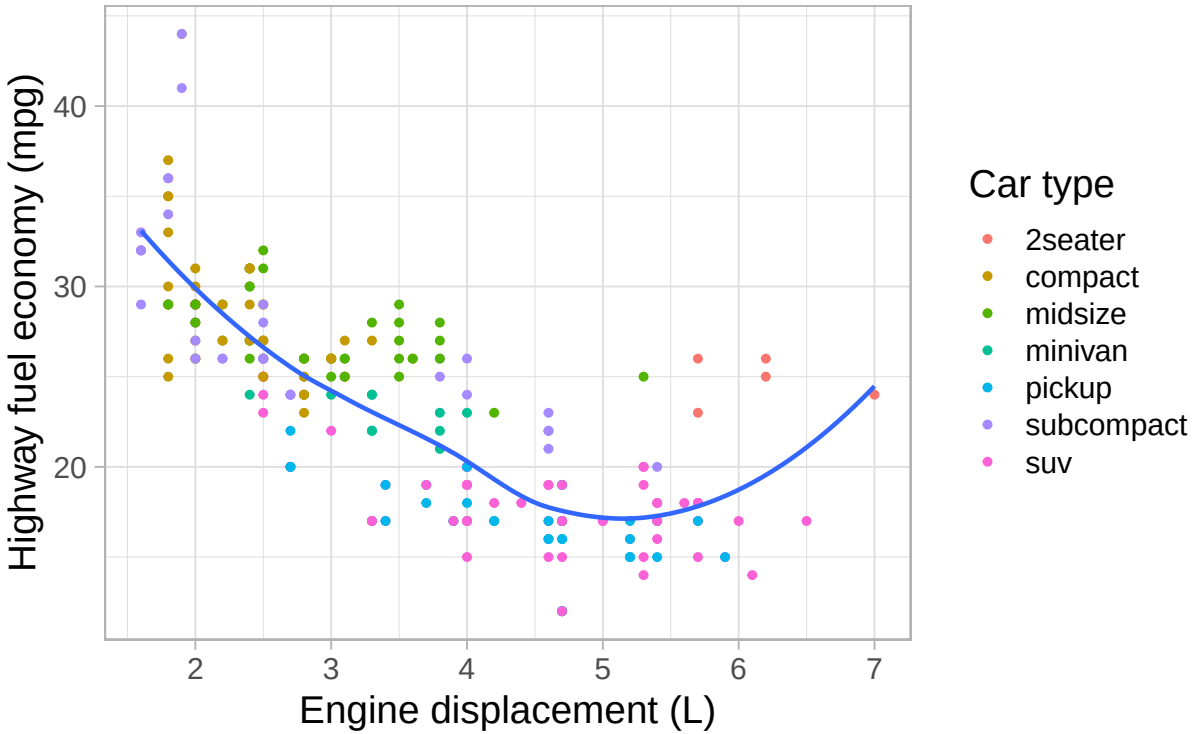
- `subtitle`: additional details beneath the title.
- `caption`: text at the bottom right of the plot.

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) + geom_smooth(se = FALSE) +
  labs(title = "Fuel efficiency decreases with engine size",
       subtitle = str_wrap("Two seaters (sports cars) are an exception
                           because of their light weight", width = 45),
       caption = "Data from fueleconomy.gov")
```



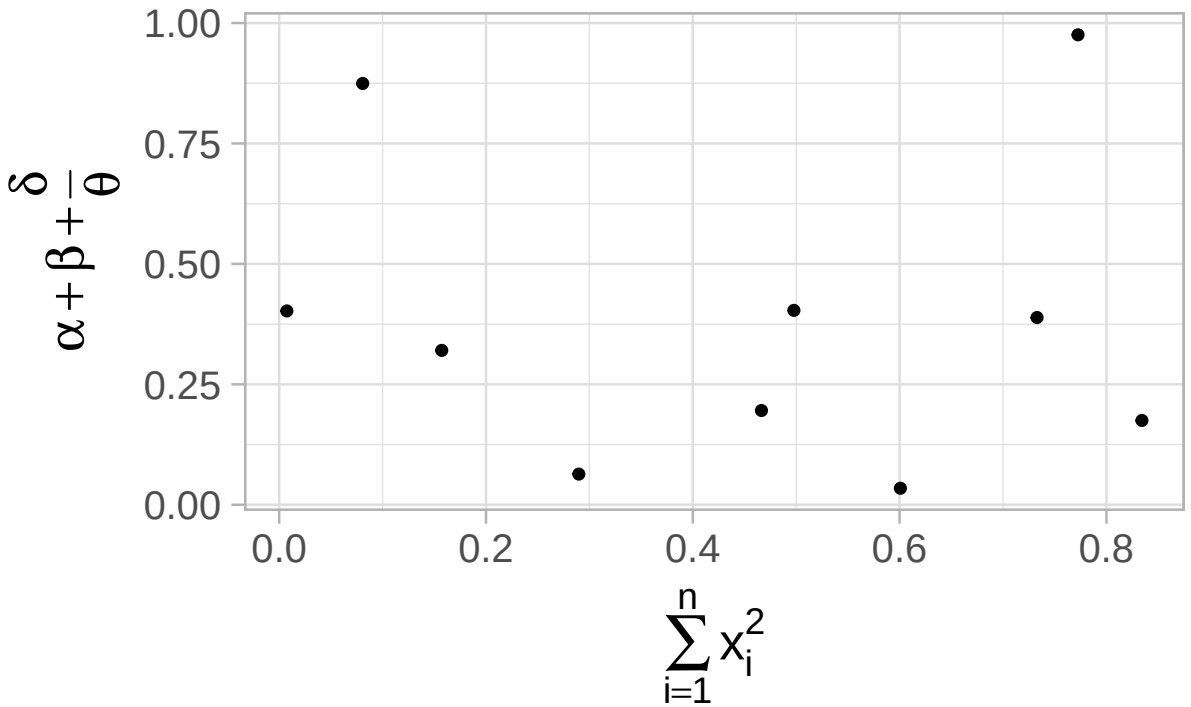
Axes


```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  labs(x = "Engine displacement (L)",
       y = "Highway fuel economy (mpg)",
       color = "Car type")
```



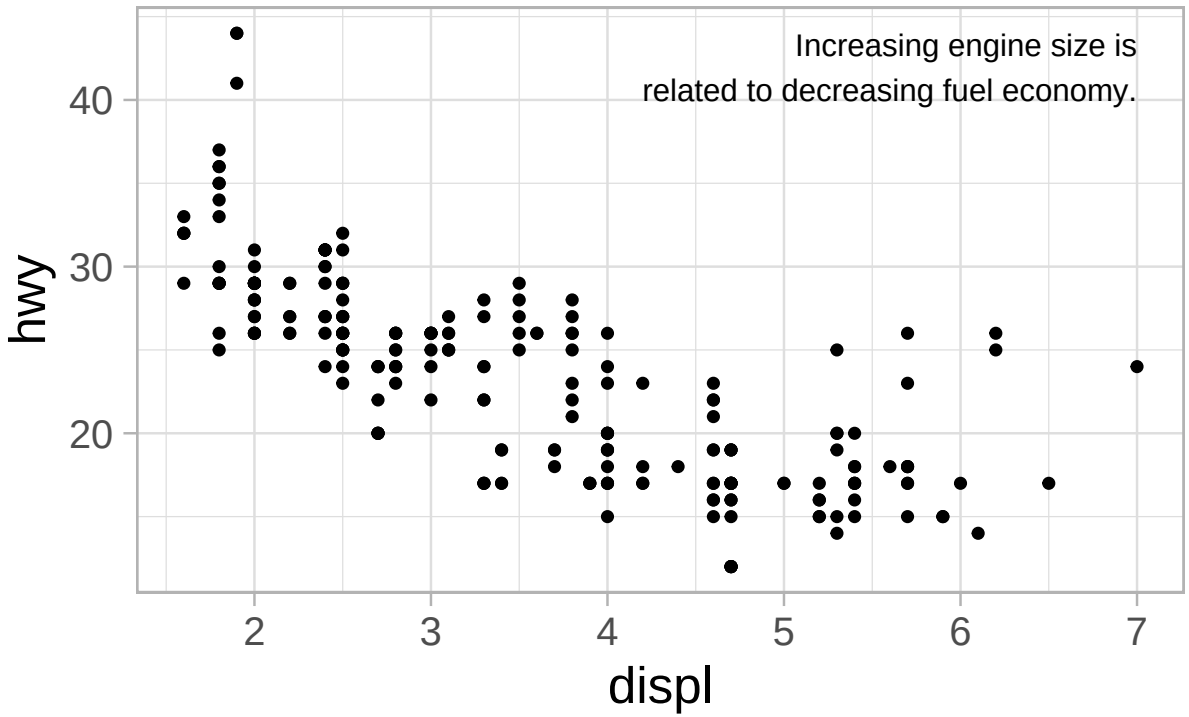
Mathematical expressions


```
df <- tibble(x = runif(10),
             y = runif(10))
ggplot(df, aes(x, y)) + geom_point() +
  labs(x = quote(sum(x[i] ^ 2, i == 1, n)),
       y = quote(alpha + beta + frac(delta, theta)))
```



To add a single label to the plot

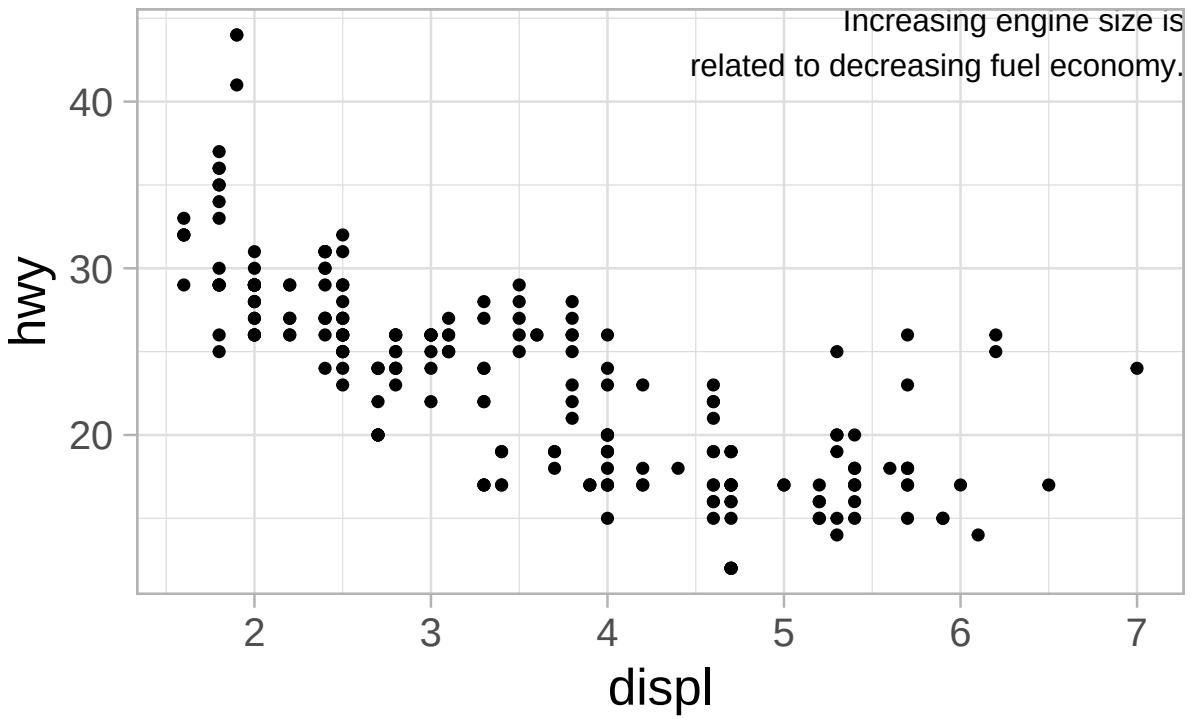

```
label <- mpg %>%
  summarize(displ = max(displ), hwy = max(hwy),
            label = "Increasing engine size is
            related to decreasing fuel economy.")

ggplot(mpg, aes(displ, hwy)) + geom_point() +
  geom_text(aes(label = label), data = label,
            vjust = "top", hjust = "right")
```



To add a single label to the plot II

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

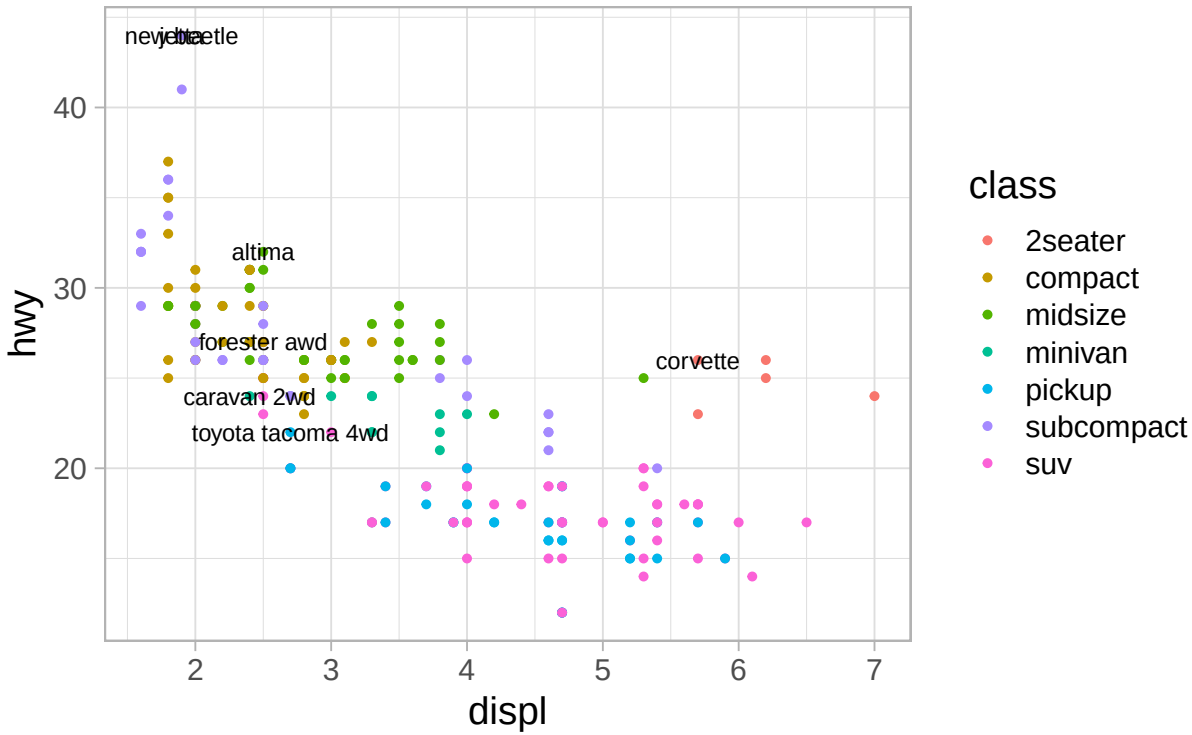
```
label <- tibble(displ = Inf, hwy = Inf,
                label = "Increasing engine size is
                related to decreasing fuel economy.")

ggplot(mpg, aes(displ, hwy)) + geom_point() +
  geom_text(aes(label = label), data = label,
            vjust = "top", hjust = "right")
```



To add a multiple labels to the plot

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

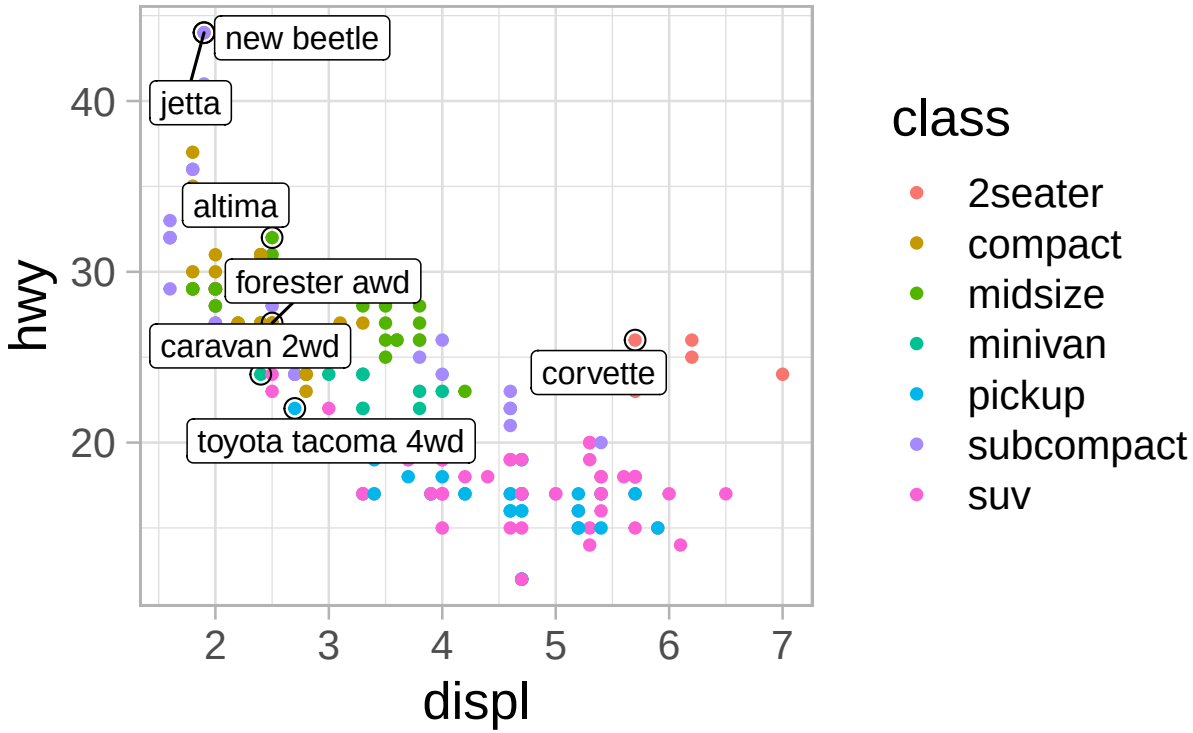
```
best_in_class <- mpg %>%
  group_by(class) %>%
  filter(row_number(desc(hwy)) == 1)

ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_text(aes(label = model), data = best_in_class)
```



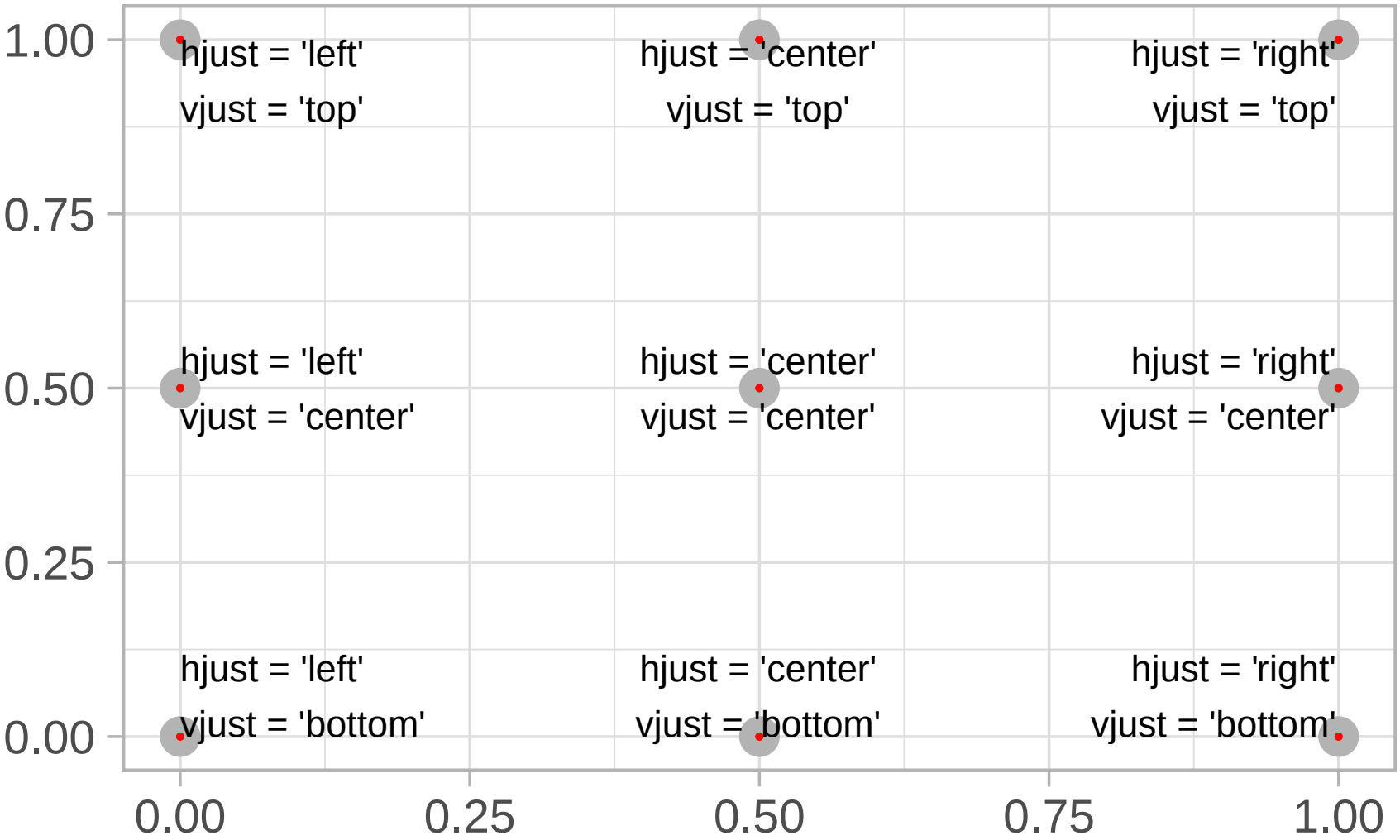
Or better

- Use the **ggrepel** package!

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_point(size = 3, shape = 1, data = best_in_class) +
  ggrepel::geom_label_repel(aes(label = model), data = best_in_class)
```



To control the alignment of the label



Geoms to help annotate your plot

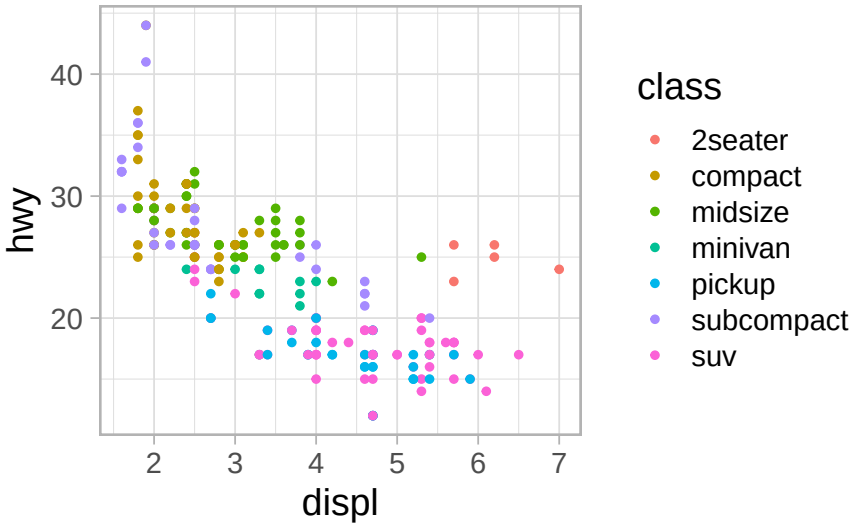
- `geom_hline()` and `geom_vline()`:
  - Add reference lines.
  - Using e.g. `size = 2` is often a good idea.
- `geom_rect()`:
  - Draw a rectangle around points of interest.
  - Boundaries defined by `xmin`, `xmax`, `ymin`, `ymax`.
- `geom_segment()` with the `arrow` argument:
  - Draw attention to a point with an arrow.
  - `x`/`xend` and `y`/`yend` define the start/end locations.
- The only limit is your imagination (and patience)!

Outline

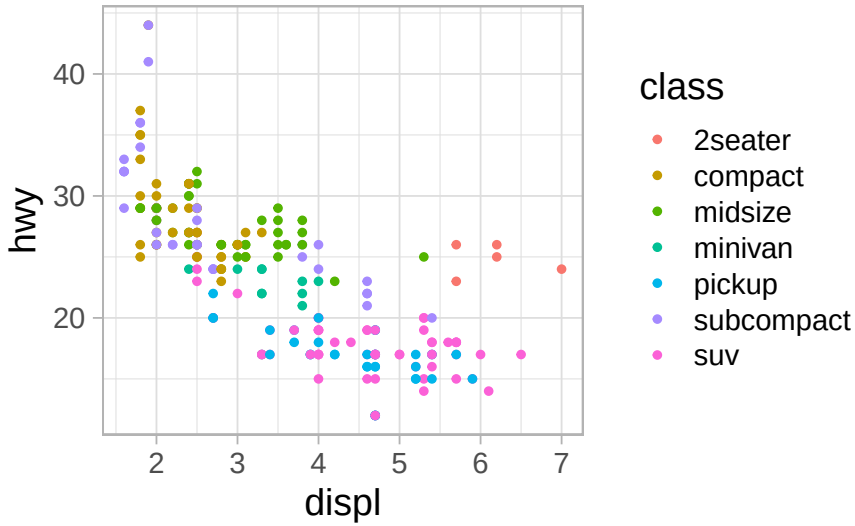
Guides and scales

- Collectively axes and legends are called **guides**:
  - Axes are used for x and y aesthetics.
  - Legends are used for everything else.
- **Scales** control mappings from data values to perceived values:

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) # +
  # scale_x_continuous() +
  # scale_y_continuous() +
  # scale_color_discrete()
```



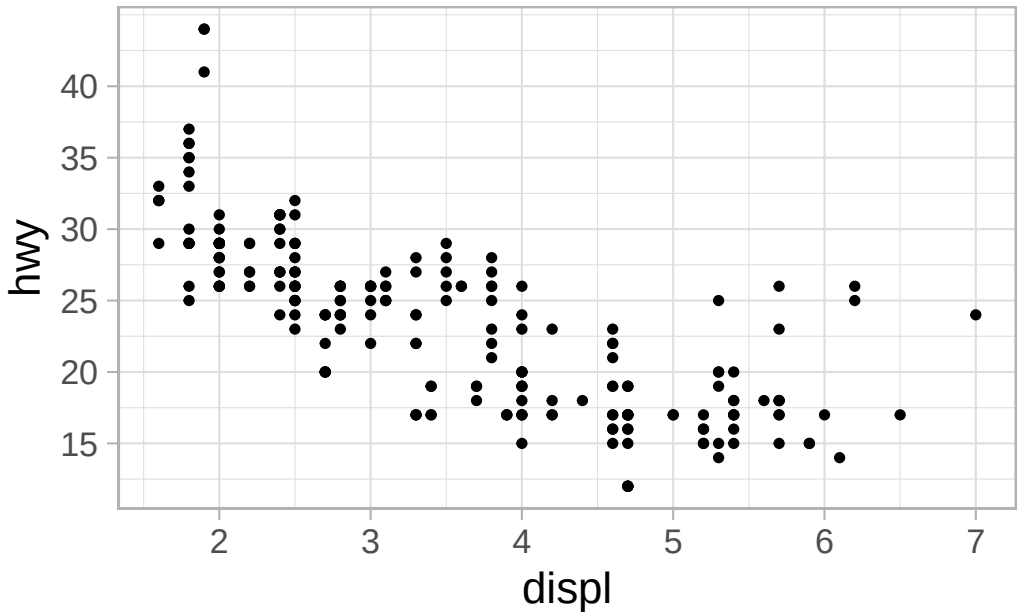

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  scale_x_continuous() +
  scale_y_continuous() +
  scale_color_discrete()
```



Axes ticks and legend keys

- To control the ticks on the axes and the keys on the legend:
  - `breaks`: ticks positions, or values associated with keys.
  - `labels`: text associated with each tick/key.
- The scales package gives you tools to override the defaults!

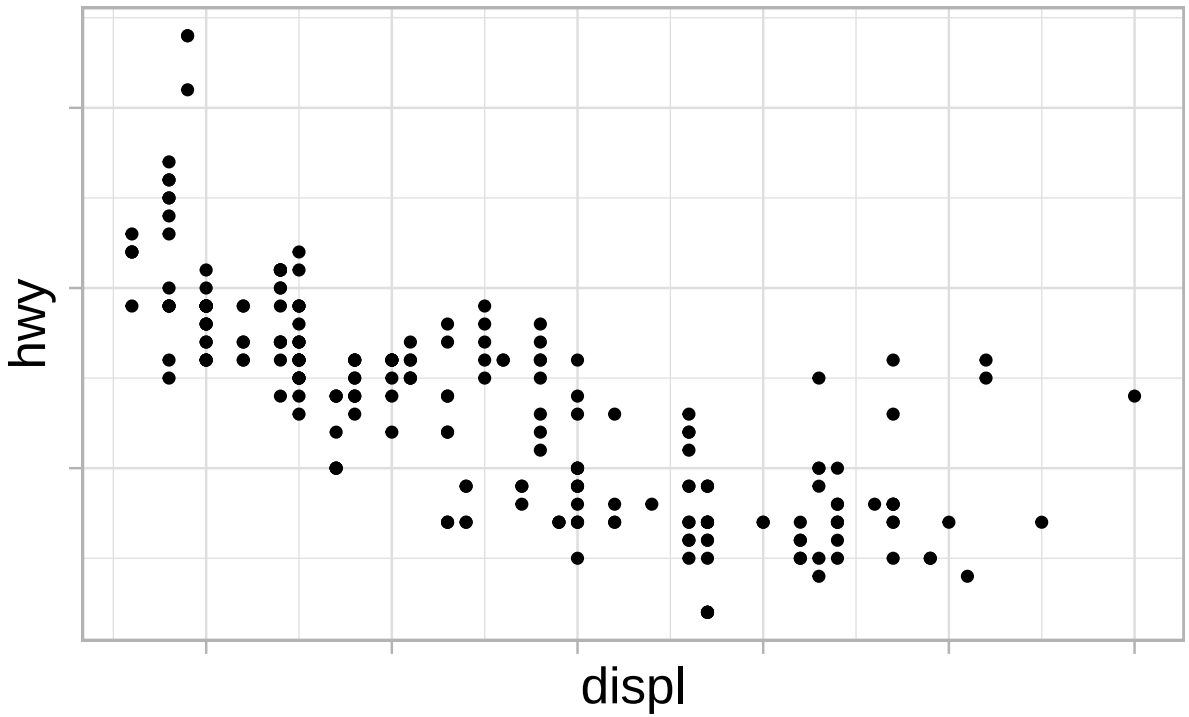
```
ggplot(mpg, aes(displ, hwy)) + geom_point() +
  scale_y_continuous(breaks = seq(15, 40, by = 5))
```



Axis ticks and legend keys II

- A useful trick for maps, or for publishing plots where you can't share the absolute numbers:

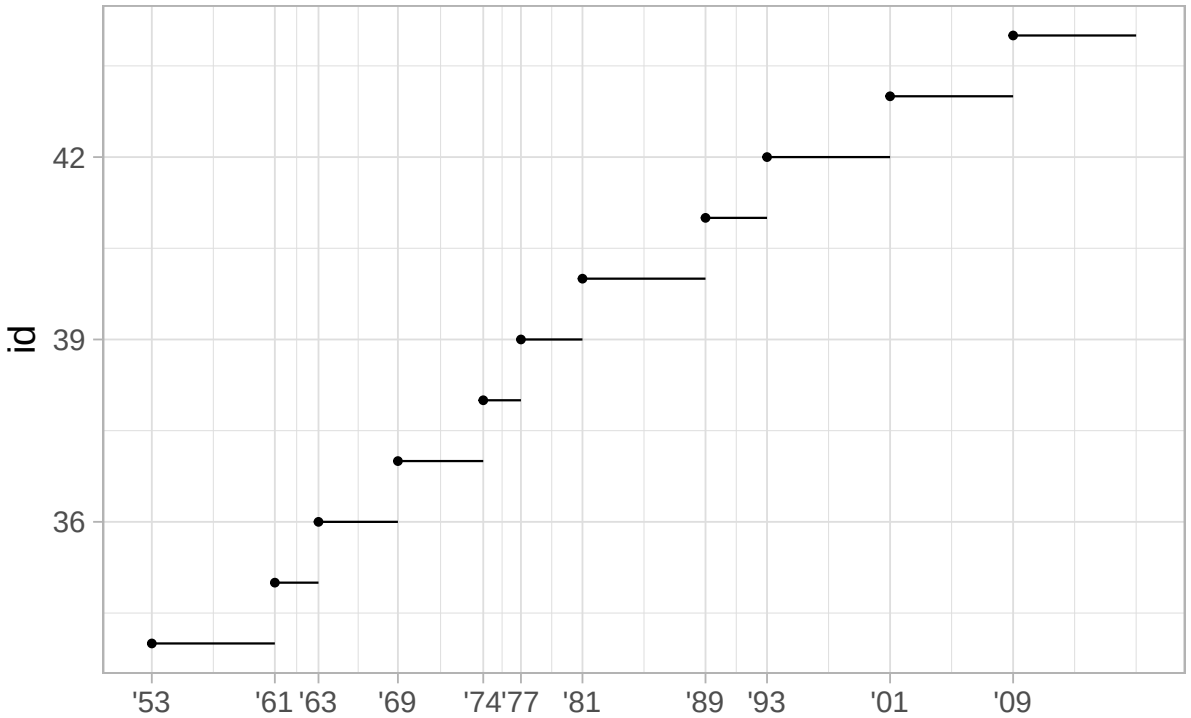
```
ggplot(mpg, aes(displ, hwy)) + geom_point() +
  scale_x_continuous(labels = NULL) +
  scale_y_continuous(labels = NULL)
```



Breaks and labels for date/datetime

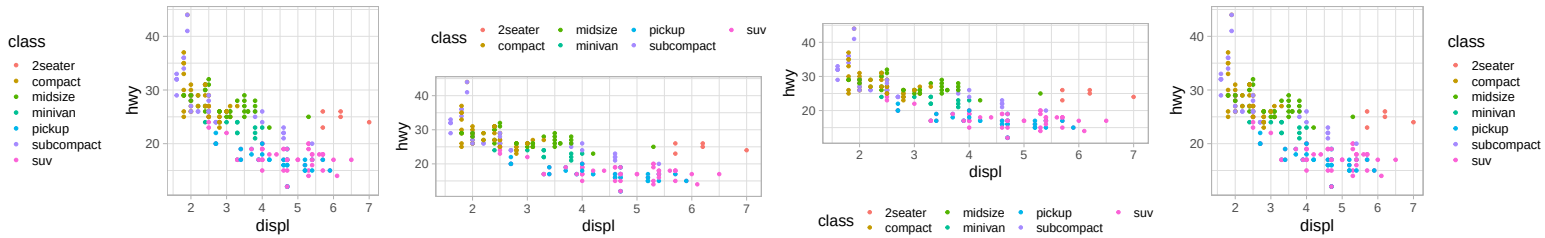
- date_labels: a format as in ?readr::parse_datetime().
- date_breaks: a string like "2 days" or "1 month".

```
presidential %>% mutate(id = 33 + row_number()) %>%
  ggplot(aes(start, id)) + geom_point() +
    geom_segment(aes(xend = end, yend = id)) +
    scale_x_date(NULL, breaks = presidential$start, date_labels = "'%y")
```



Legend layout


```
base <- ggplot(mpg, aes(displ, hwy)) + geom_point(aes(color = class))

base + theme(legend.position = "left")
base + theme(legend.position = "top")
base + theme(legend.position = "bottom")
base + theme(legend.position = "right") # the default
```

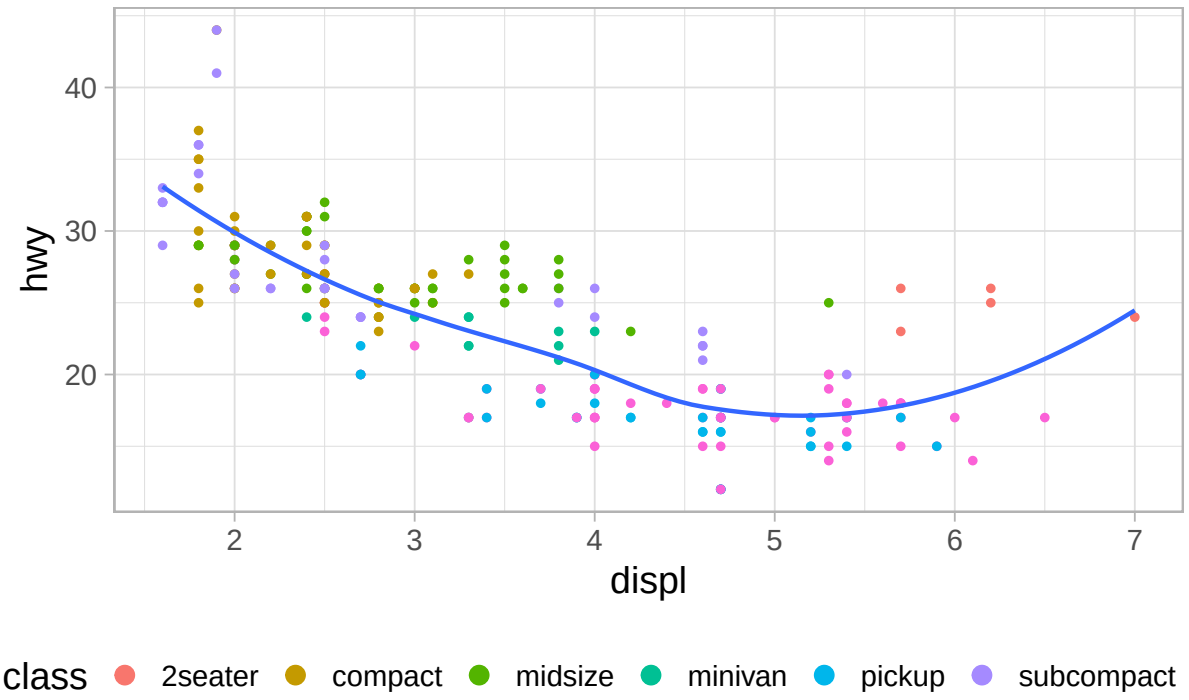


- `legend.position = "none"` suppresses the display!

To control individual legends

- Use guides(), guide_legend() or guide_colorbar():

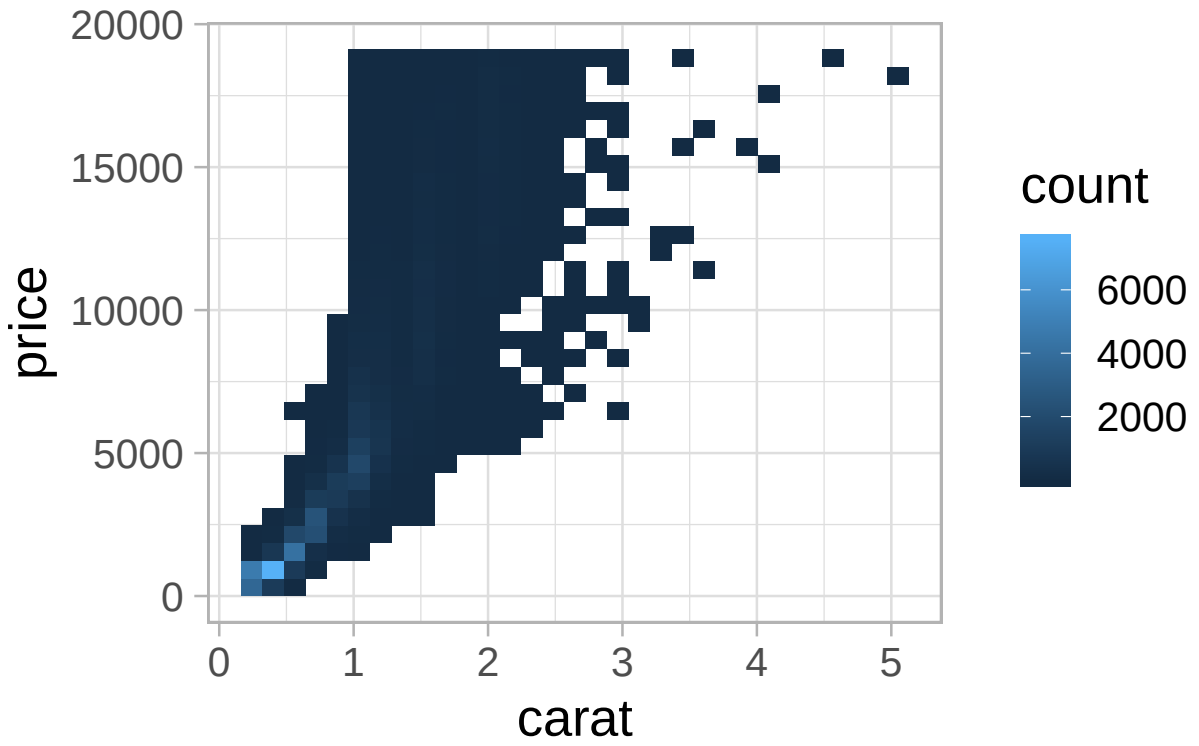
```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  theme(legend.position = "bottom") +
  guides(color = guide_legend(nrow = 1, override.aes = list(size = 4)))
```



How could we improve the scale?

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

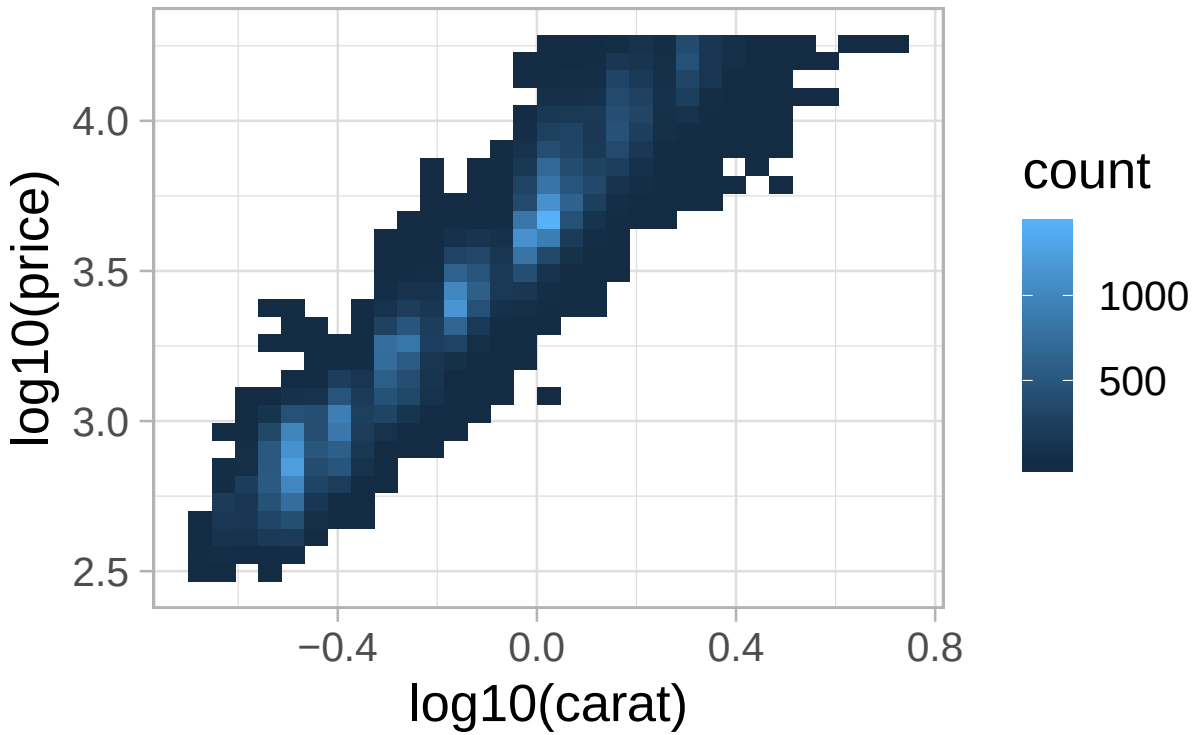
```
ggplot(diamonds, aes(carat, price)) +
  geom_bin2d()
```



Log-transform the variables

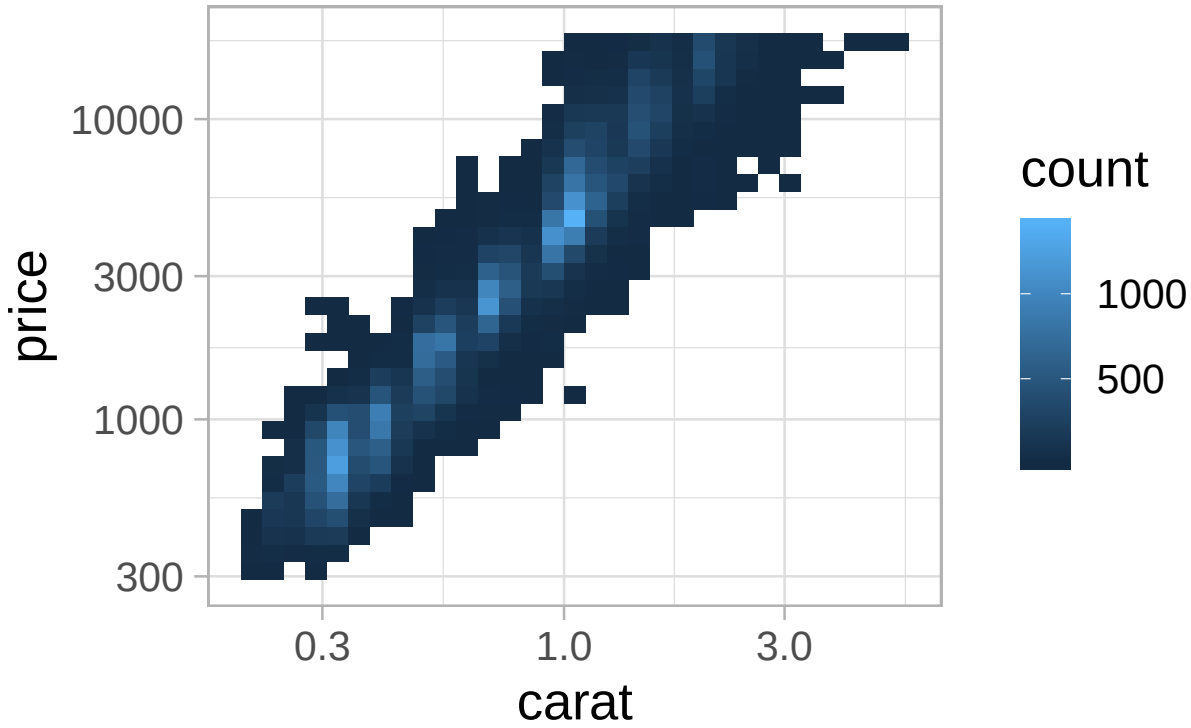
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

```
ggplot(diamonds, aes(log10(carat), log10(price))) +
  geom_bin2d()
```



. . . or simply replace the scale

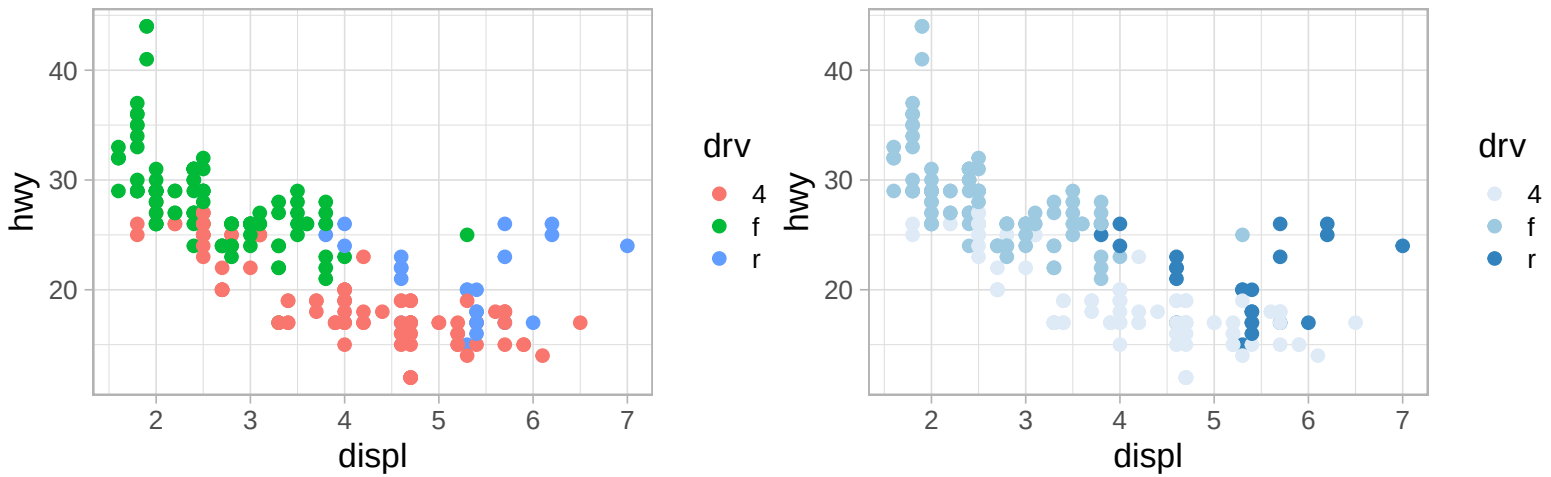

```
ggplot(diamonds, aes(carat, price)) +
  geom_bin2d() +
  scale_x_log10() +
  scale_y_log10()
```



Outline

Replacing color scales

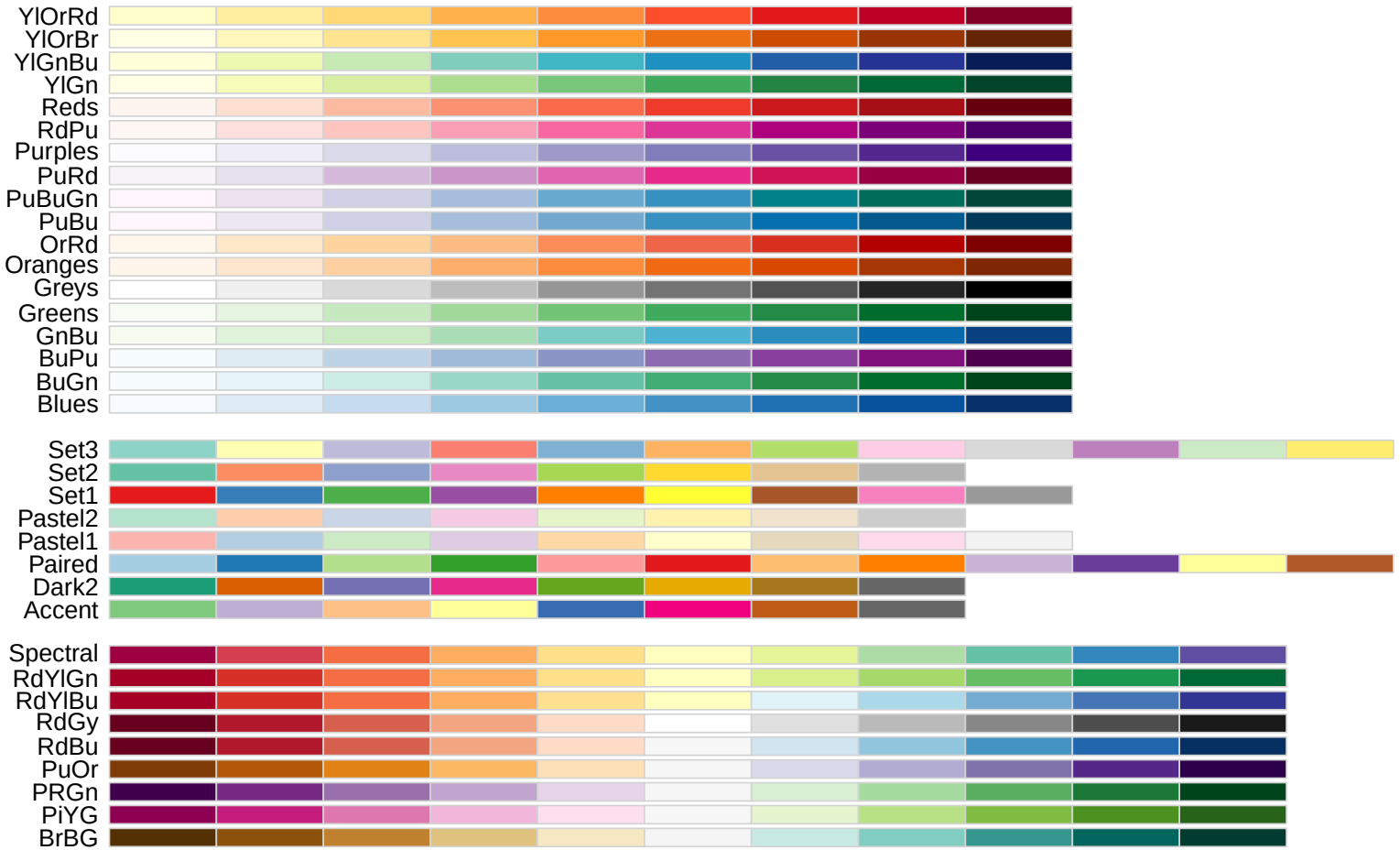

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = drv), size = 3)

ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = drv), size = 3) +
  scale_color_brewer(palette = "Blues")
```



- Color scales come in two variety:
  - `scale_color_x()` for the `color` aesthetics (available in UK/US spellings).
  - `scale_fill_x()` for the `fill` aesthetics.

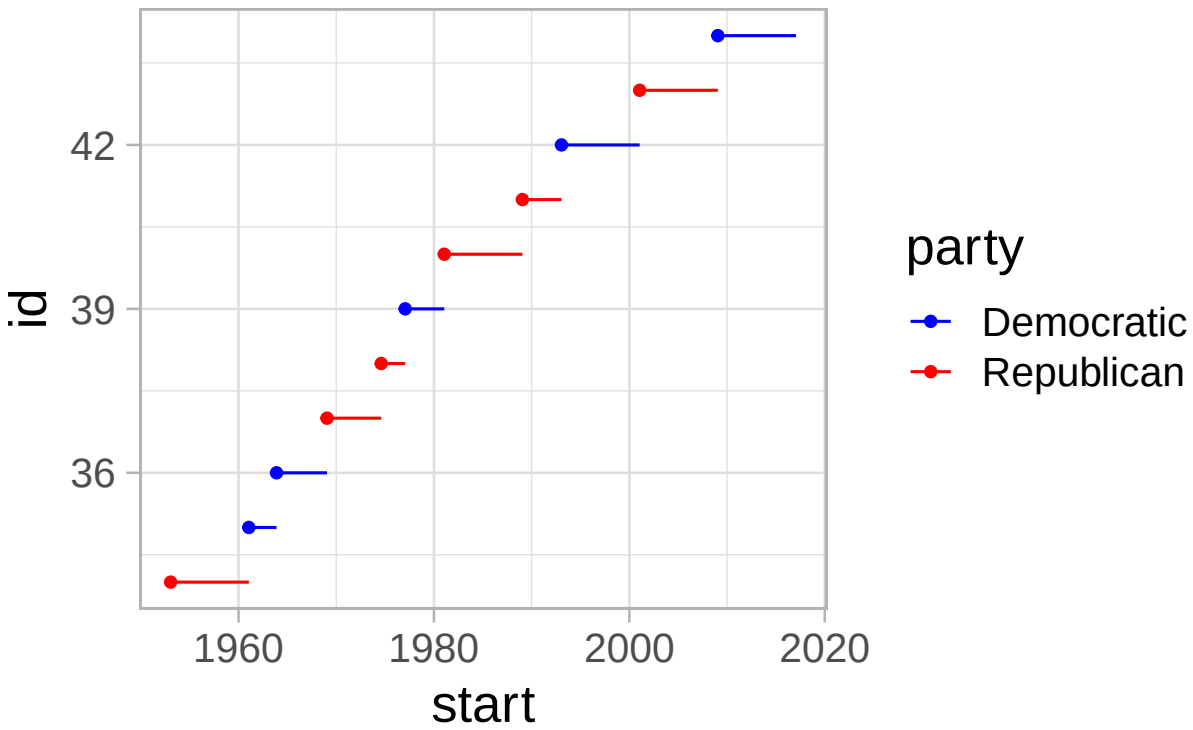
The ColorBrewer scales

- Documented online at http://colorbrewer2.org/
- Available via the **RColorBrewer** package.



Using manually defined mappings

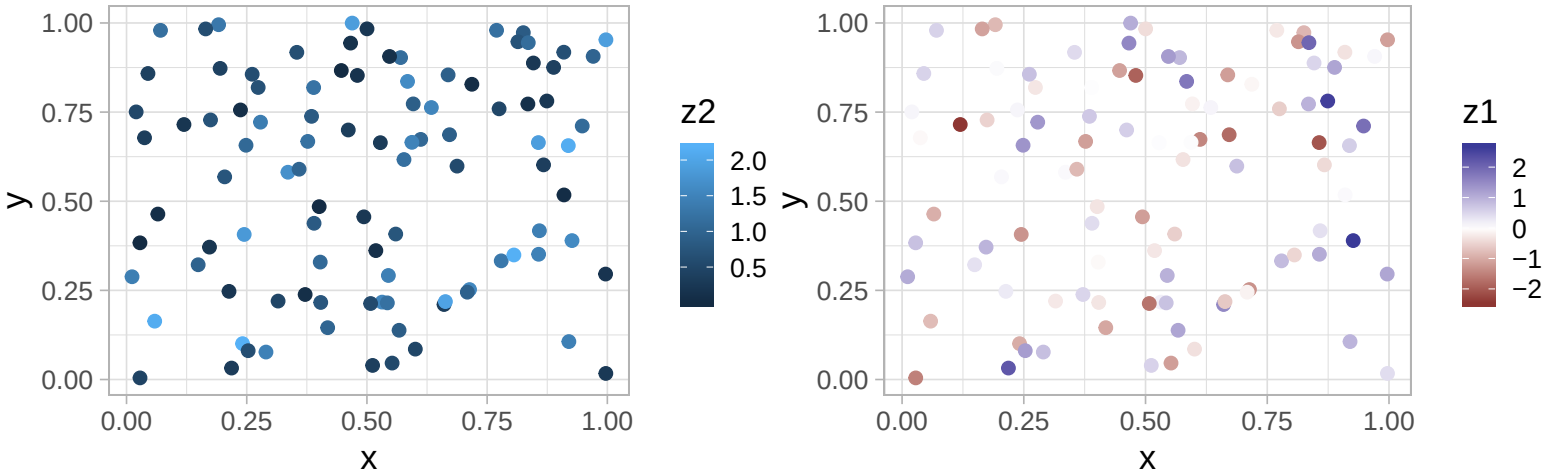

```
presidential %>%
  mutate(id = 33 + row_number()) %>%
  ggplot(aes(start, id, color = party)) +
    geom_point() +
    geom_segment(aes(xend = end, yend = id)) +
    scale_color_manual(values = c(Republican = "red",
                                  Democratic = "blue"))
```



Continuous vs diverging color scales

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

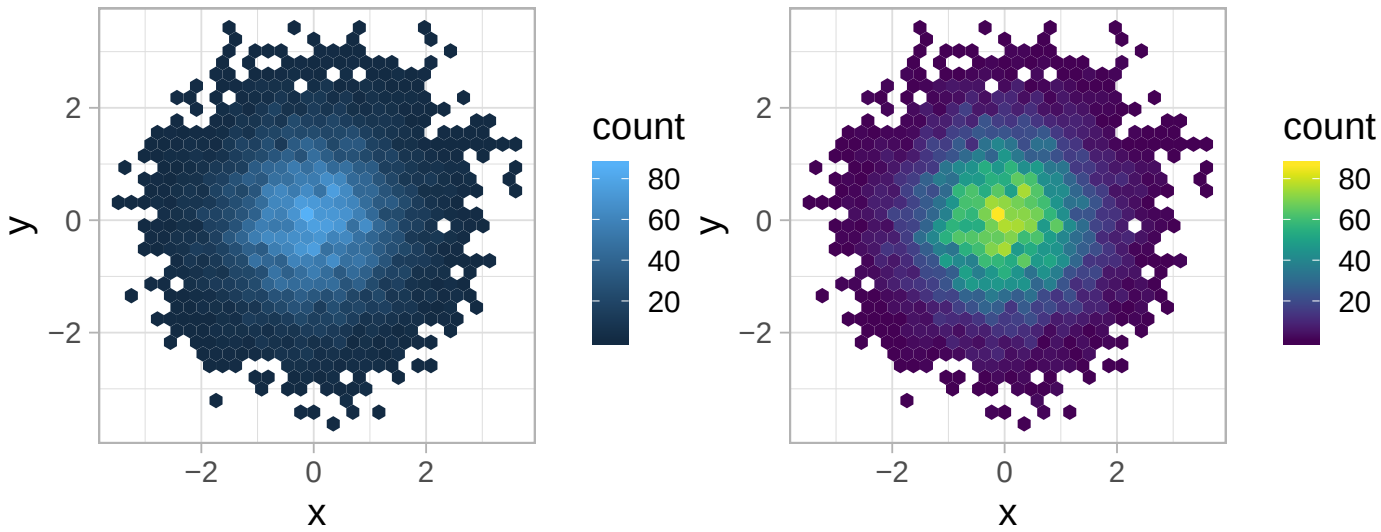
```
df <- data.frame(x = runif(100), y = runif(100),
                 z1 = rnorm(100), z2 = abs(rnorm(100)))

ggplot(df, aes(x, y)) +
  geom_point(aes(color = z2), size = 3)

ggplot(df, aes(x, y)) +
  geom_point(aes(color = z1), size = 3) +
  scale_color_gradient2()
```



A continuous analog of ColorBrewer

- The `viridis` package!

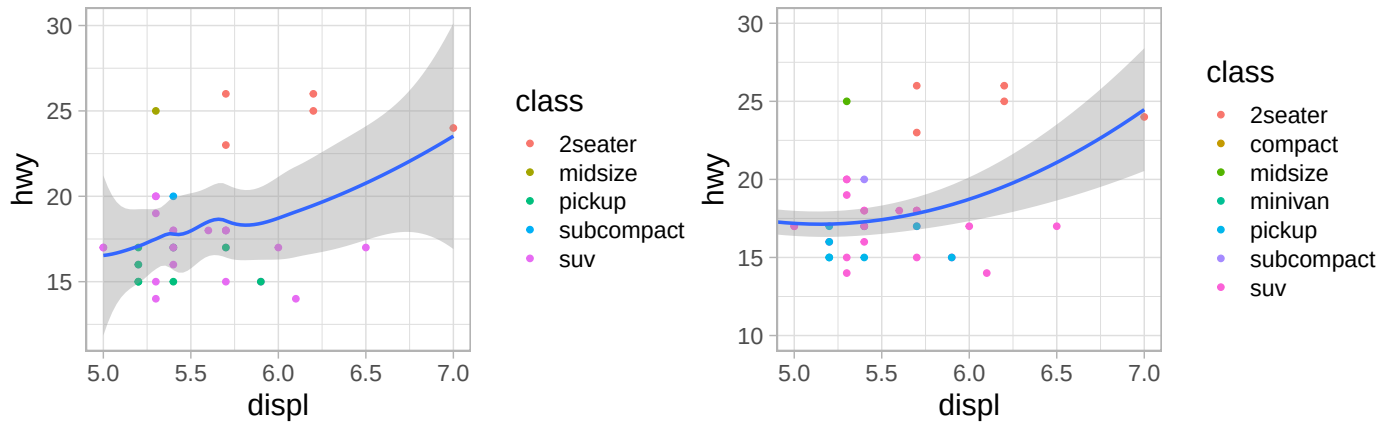
```
df <- tibble(x = rnorm(10000), y = rnorm(10000))

ggplot(df, aes(x, y)) +
  geom_hex() +
  coord_fixed()

ggplot(df, aes(x, y)) +
  geom_hex() +
  coord_fixed() +
  viridis::scale_fill_viridis()
```



Zooming

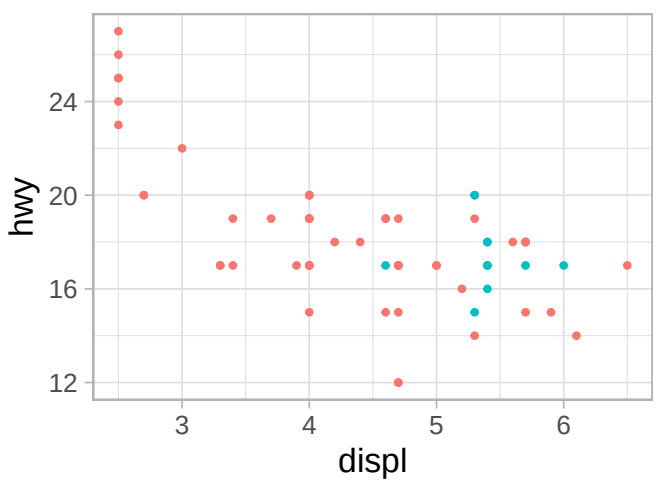
- Three methods:
  - Adjust what data are plotted.
  - Set `xlim` and `ylim` in `coord_cartesian()`.
  - Set the limits in each scale.

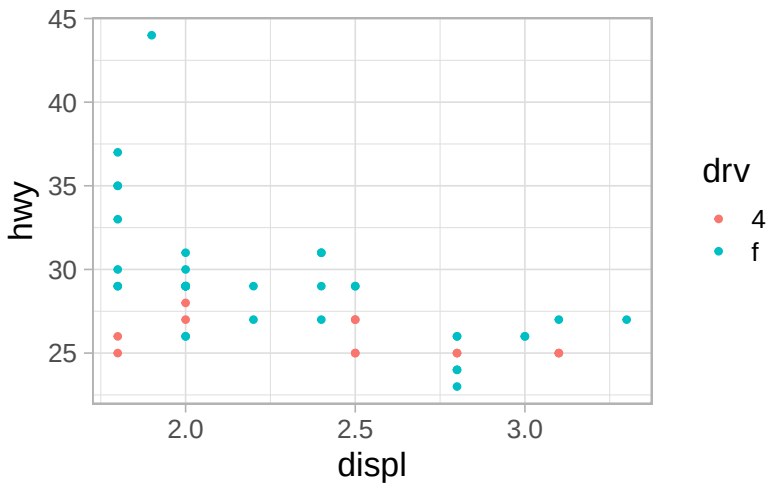
```
mpg %>%
  filter(displ >= 5, displ <= 7, hwy >= 10, hwy <= 30) %>%
  ggplot(aes(displ, hwy)) +
    geom_point(aes(color = class)) + geom_smooth()

ggplot(mpg, mapping = aes(displ, hwy)) +
  geom_point(aes(color = class)) + geom_smooth() +
  coord_cartesian(xlim = c(5, 7), ylim = c(10, 30))
```



Zooming cont'd


```
suv <- mpg %>%
  filter(class == "suv")
compact <- mpg %>%
  filter(class == "compact")

ggplot(suv, aes(displ, hwy, color = drv)) +
  geom_point()
ggplot(compact, aes(displ, hwy, color = drv)) +
  geom_point()
```

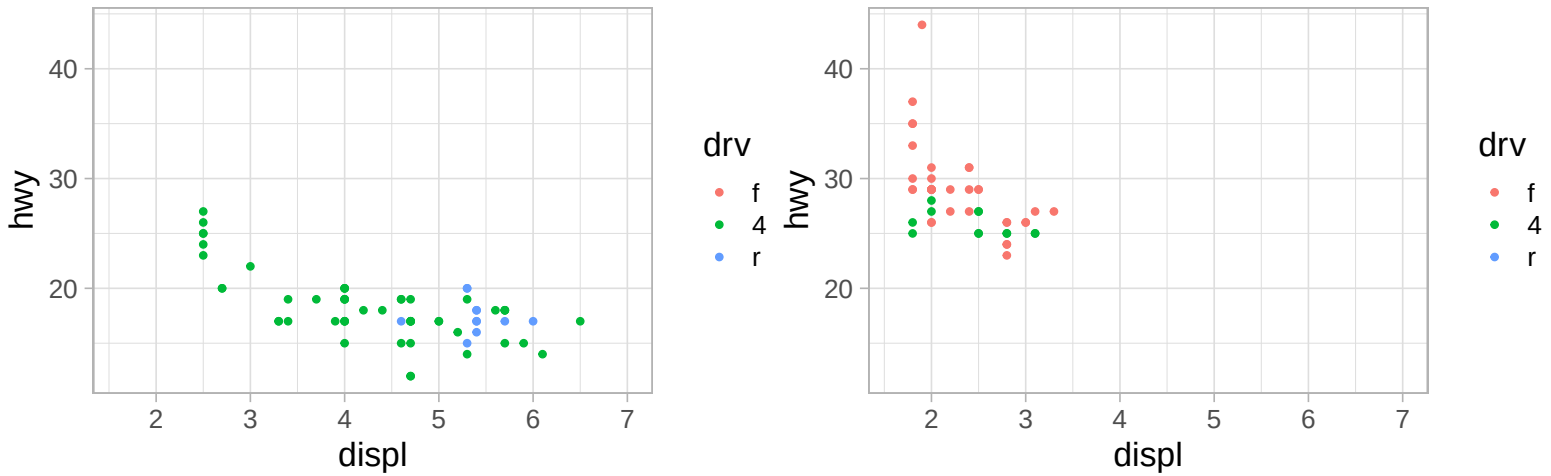




Share scales across multiple plots

- Training the scales with the `limits` of the full data:

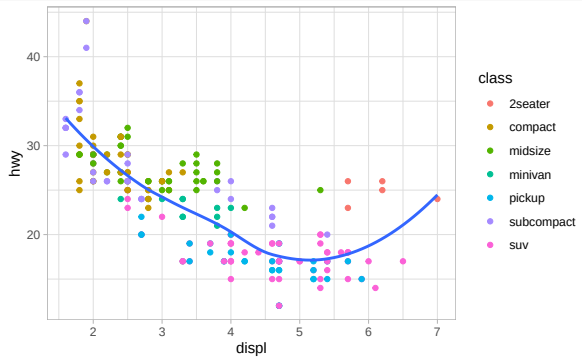
```
x_scale <- scale_x_continuous(limits = range(mpg$displ))
y_scale <- scale_y_continuous(limits = range(mpg$hwy))
col_scale <- scale_color_discrete(limits = unique(mpg$drv))

ggplot(suv, aes(displ, hwy, color = drv)) + geom_point() +
  x_scale + y_scale + col_scale
ggplot(compact, aes(displ, hwy, color = drv)) + geom_point() +
  x_scale + y_scale + col_scale
```



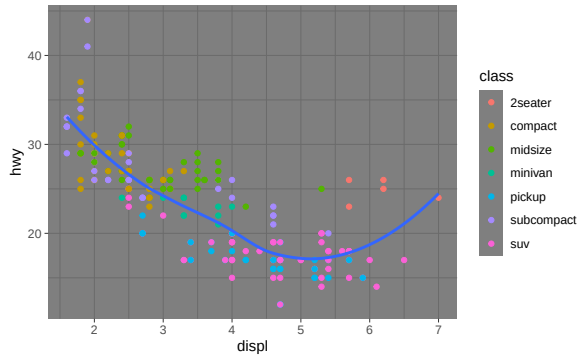
Themes

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

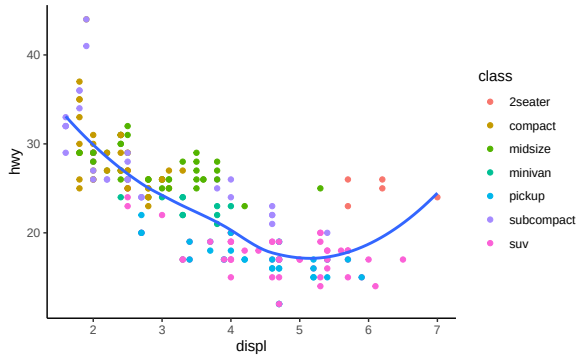
```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  theme_light()
```



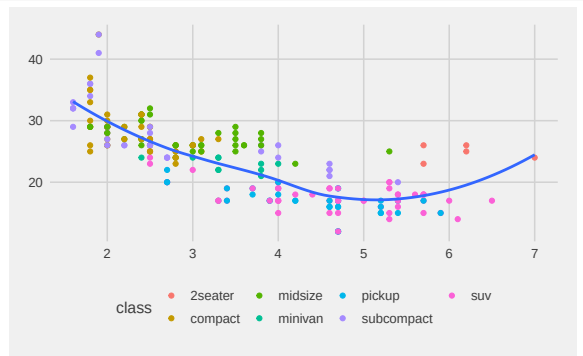

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  theme_dark()
```



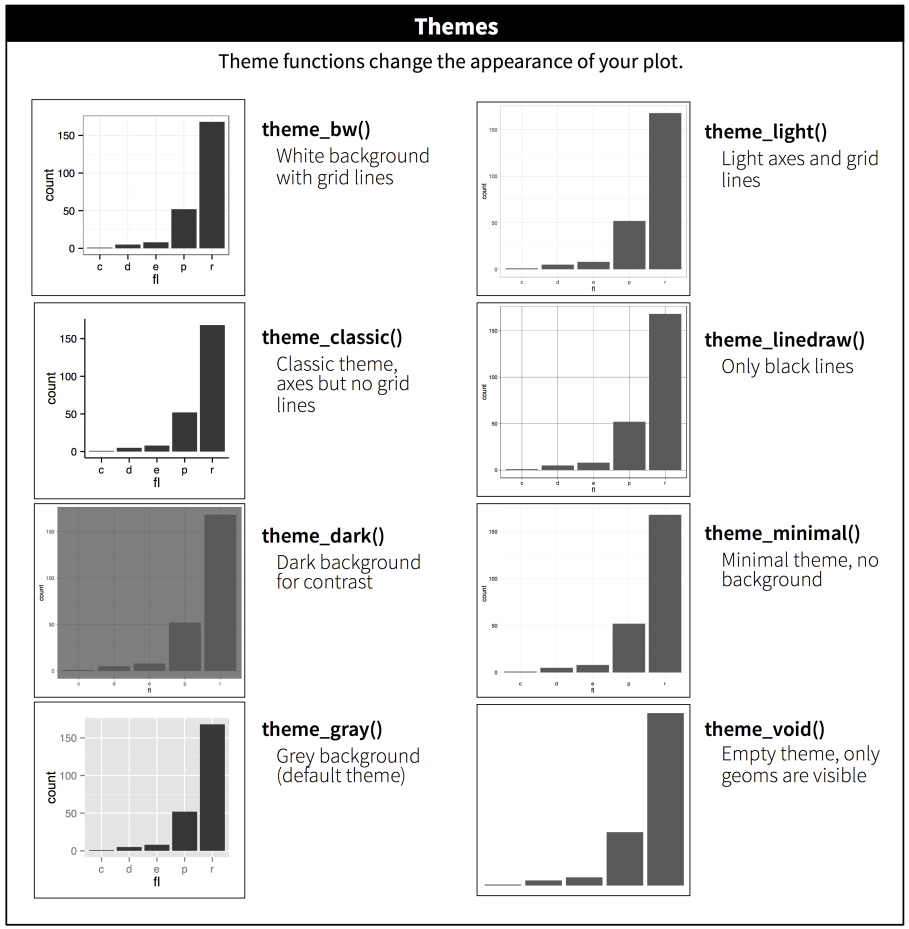

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  theme_classic()
```




```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  ggthemes::theme_fivethirtyeight()
```



ggplot2 default themes



- More in add-on packages like ggthemes!