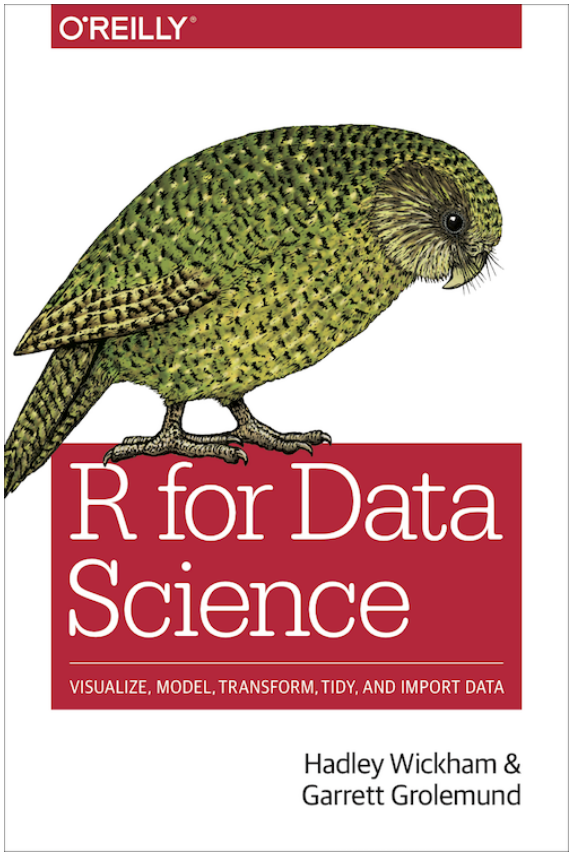
Socio-Informatics 348

Data Visualisation
Communication

Dr Lisa Martin

Department of Information Science
Stellenbosch University

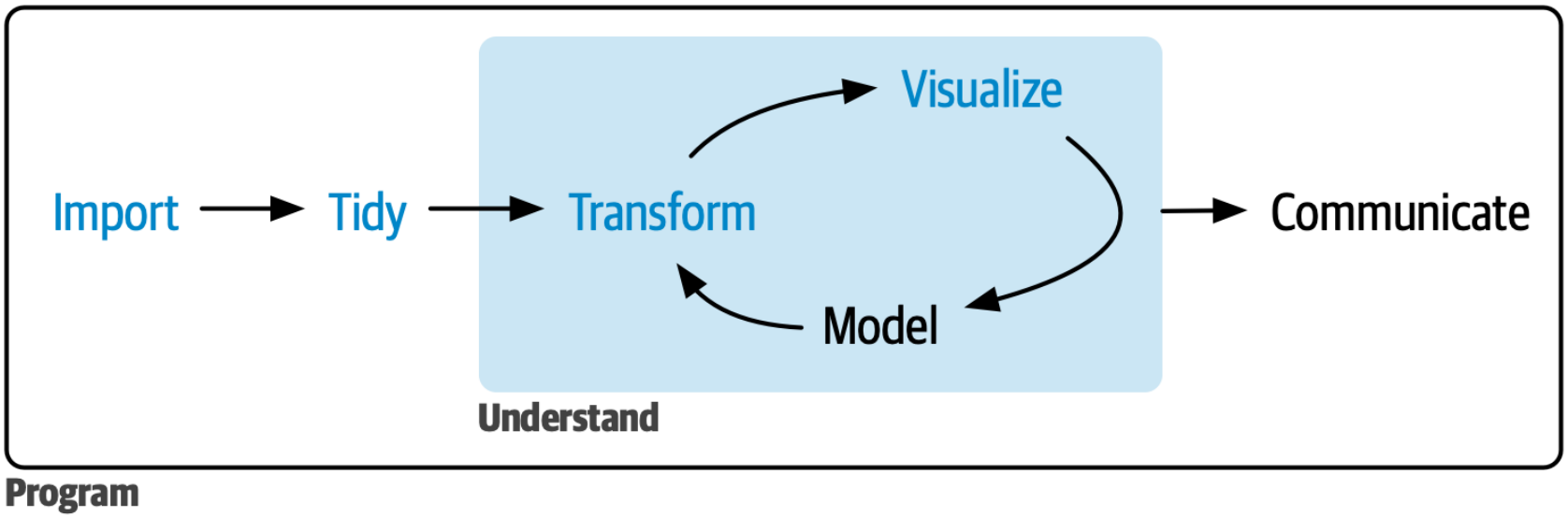
Today's Reading



R for Data Science, Chapter 11

Communication

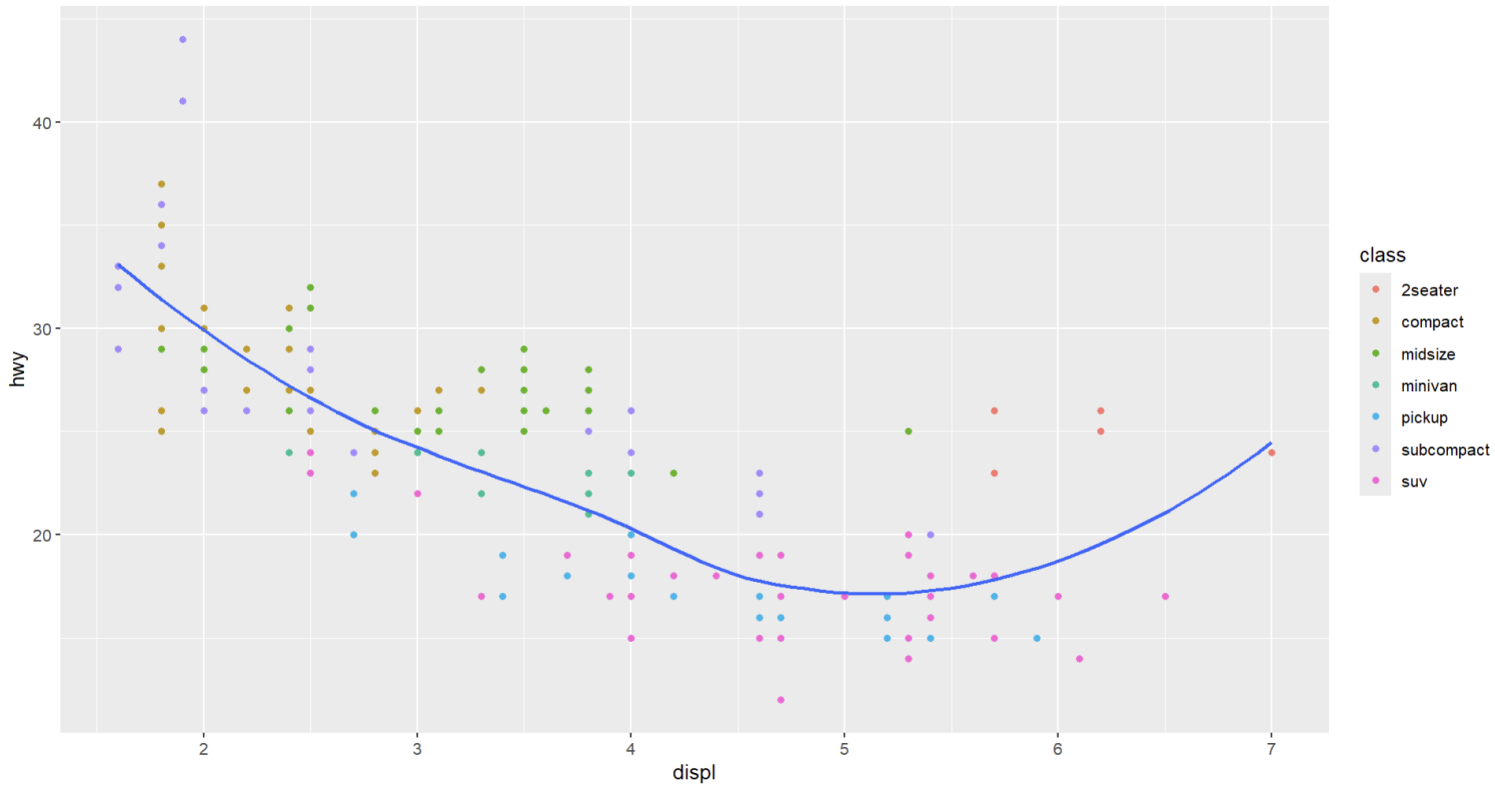
- Goal: Turn exploratory plots into explanatory graphics
- New packages to note:
  - `scales`, `ggrepel`, `patchwork`



Labels

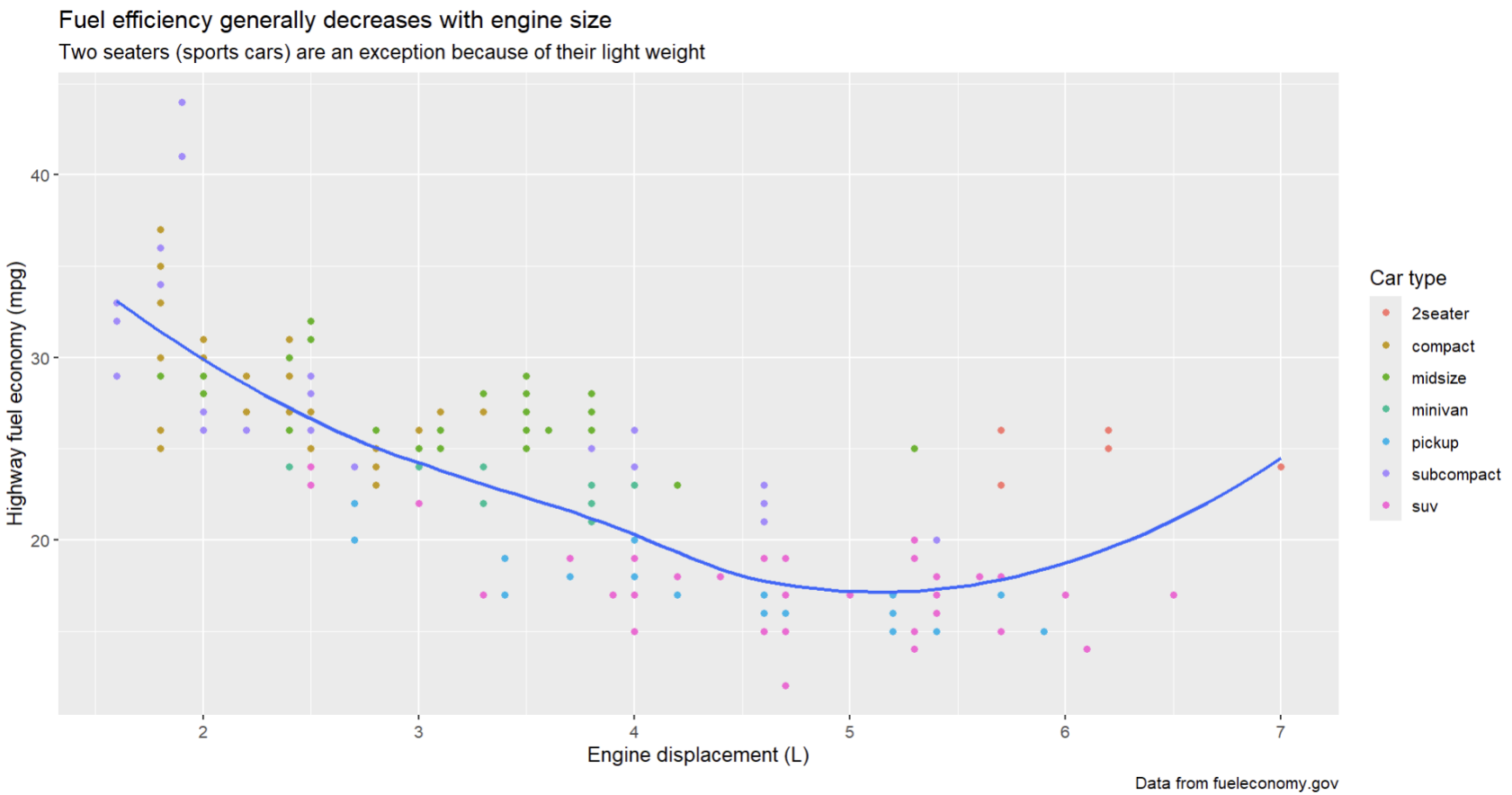
Example: Scatterplot of `displ` vs. `hwy`

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



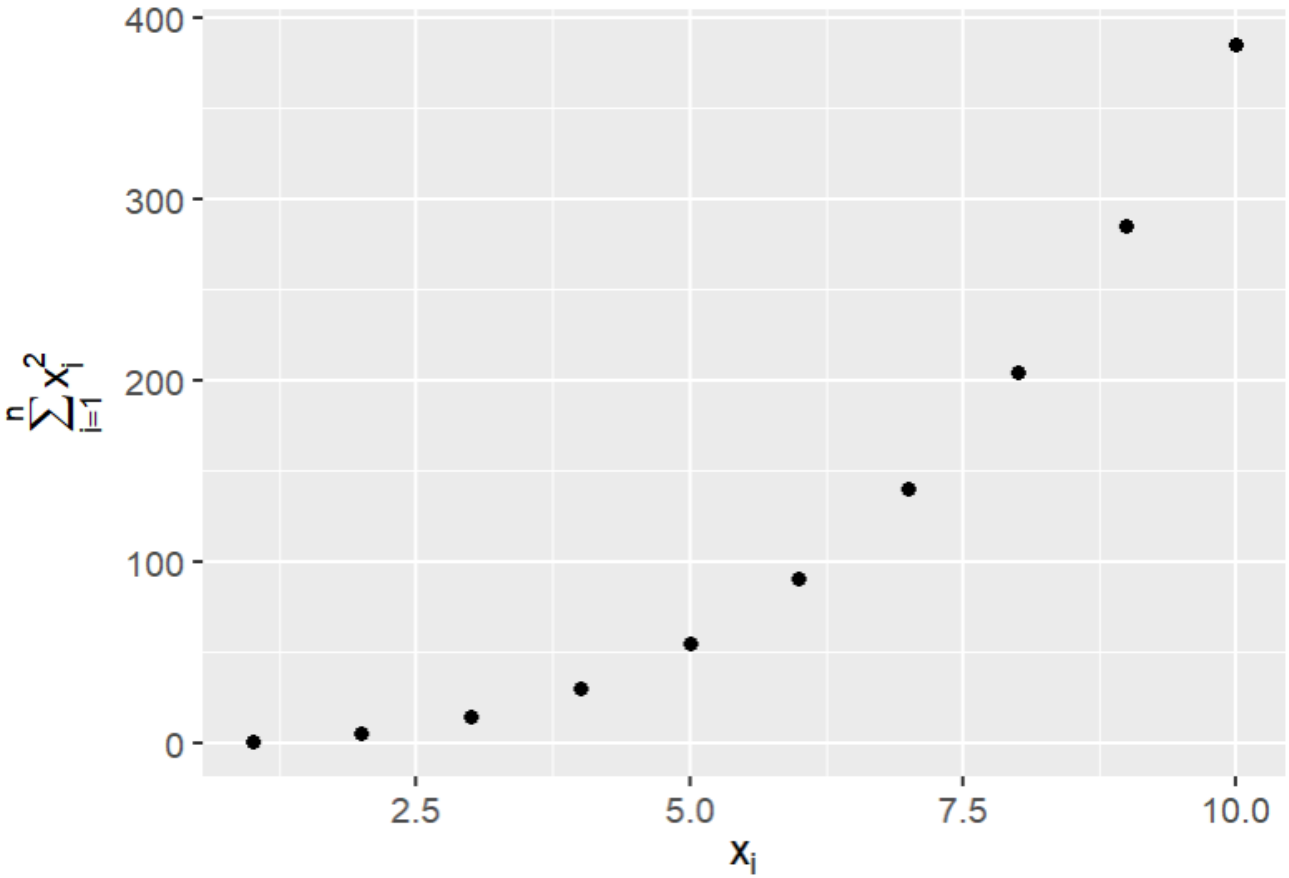
Labels

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  labs(
    x = "Engine displacement (L)",
    y = "Highway fuel economy (mpg)",
    color = "Car type",
    title = "Fuel efficiency generally decreases with engine size",
    subtitle = "Two seaters (sports cars) are an exception because of their light weight",
    caption = "Data from fueleconomy.gov"
  )
```



Labels

```
ggplot(df, aes(x, y)) +
  geom_point() +
  labs(
    x = quote(x[i]),
    y = quote(sum(x[i] ^ 2, i == 1, n))
  )
```



Annotations

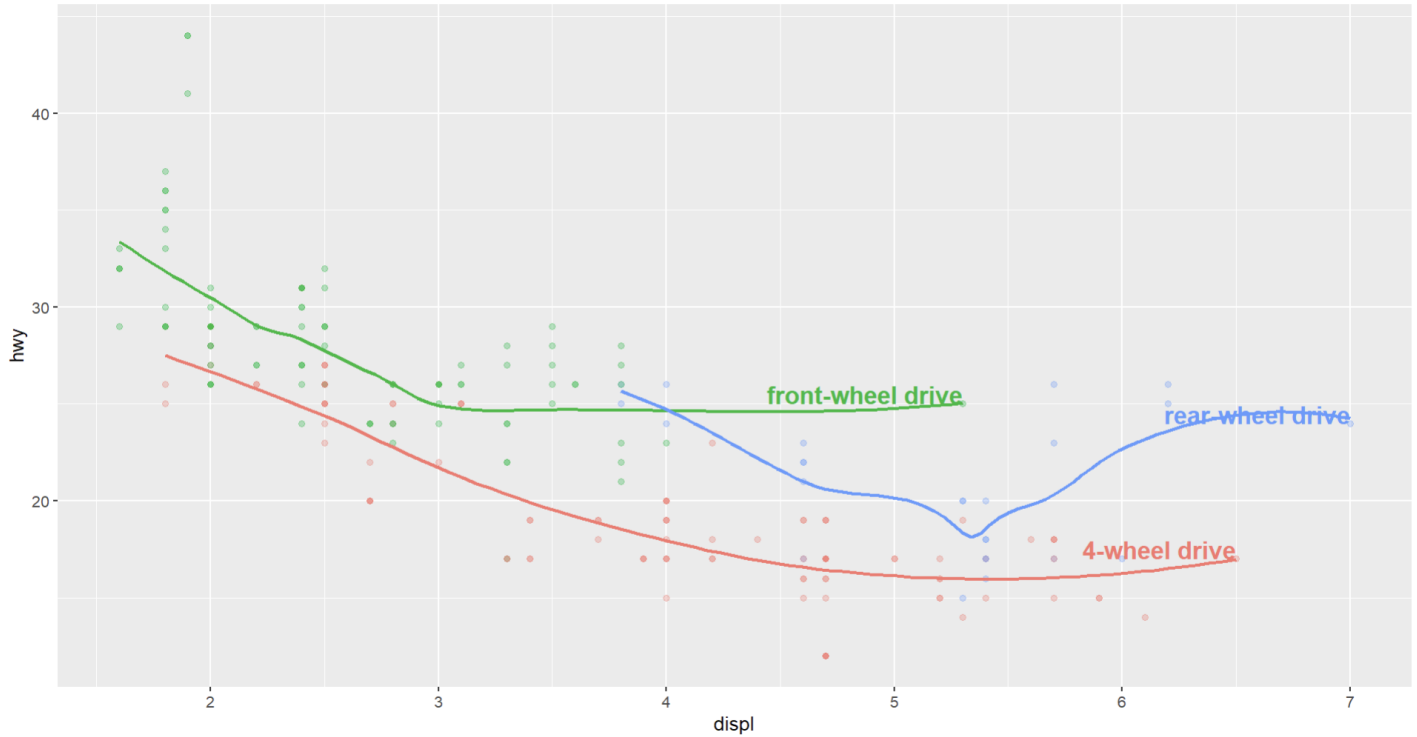
You may want to label lines on the plot instead of using the legend:

- `geom_text()` or `ggrepel::geom_text_repel()`.
- First, create the labels for your annotations:

```
label_info <- mpg |>
  group_by(drv) |>
  arrange(desc(displ)) |>
  slice_head(n = 1) |>
  mutate(
    drive_type = case_when(
      drv == "f" ~ "front-wheel drive",
      drv == "r" ~ "rear-wheel drive",
      drv == "4" ~ "4-wheel drive"
    )
  ) |>
  select(displ, hwy, drv, drive_type)

label_info
# A tibble: 3 × 4
# Groups:   drv [3]
  displ   hwy drv   drive_type
  <dbl> <int> <chr> <chr>
1   6.5    17 4     4-wheel drive
2   5.3    25 f     front-wheel drive
3   7      24 r     rear-wheel drive
> |
```

Annotations

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point(alpha = 0.3) +
  geom_smooth(se = FALSE) +
  geom_text(
    data = label_info,
    aes(x = displ, y = hwy, label = drive_type),
    fontface = "bold", size = 5, hjust = "right", vjust = "bottom"
  ) +
  theme(legend.position = "none")
```

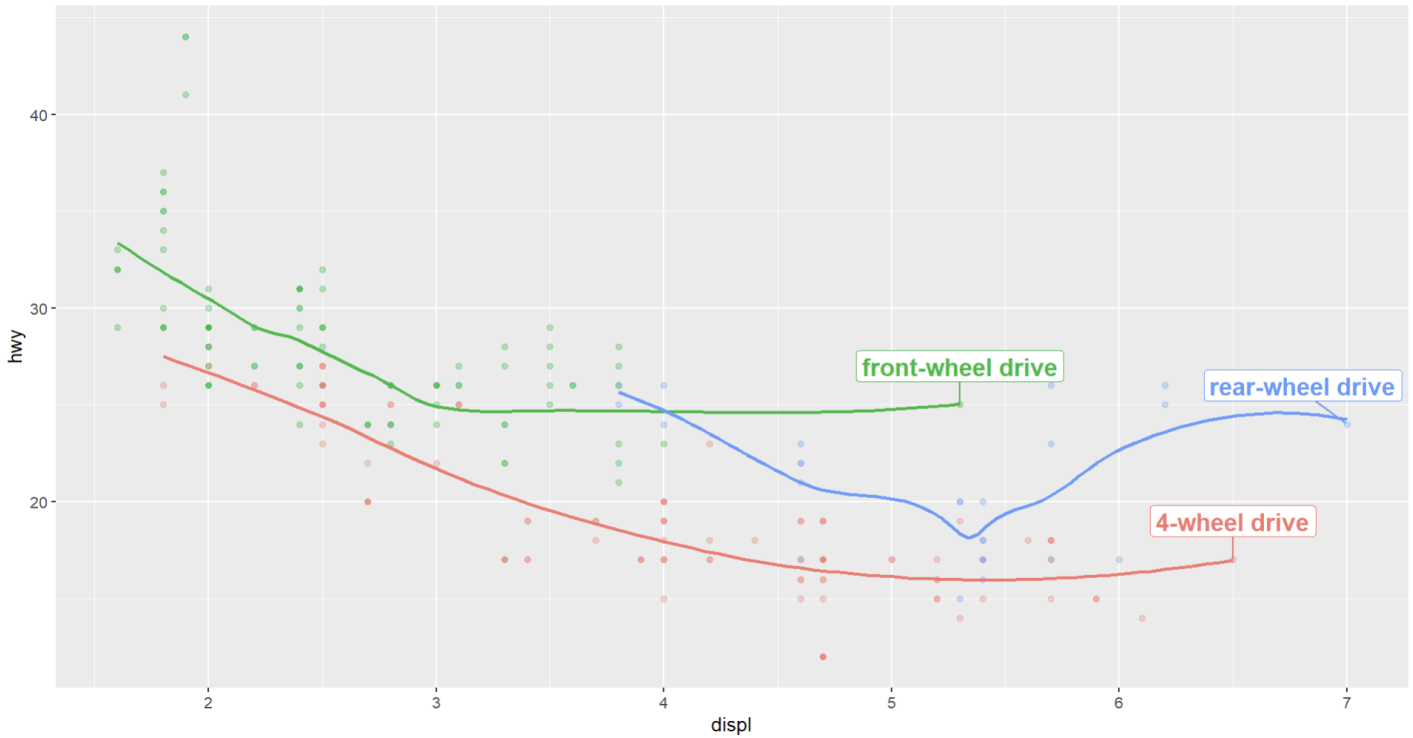


- The text overlaps with other graph elements. Can we neaten it?

Annotations

- Use `geom_label_repel()` from `ggrepel` for better label placement.

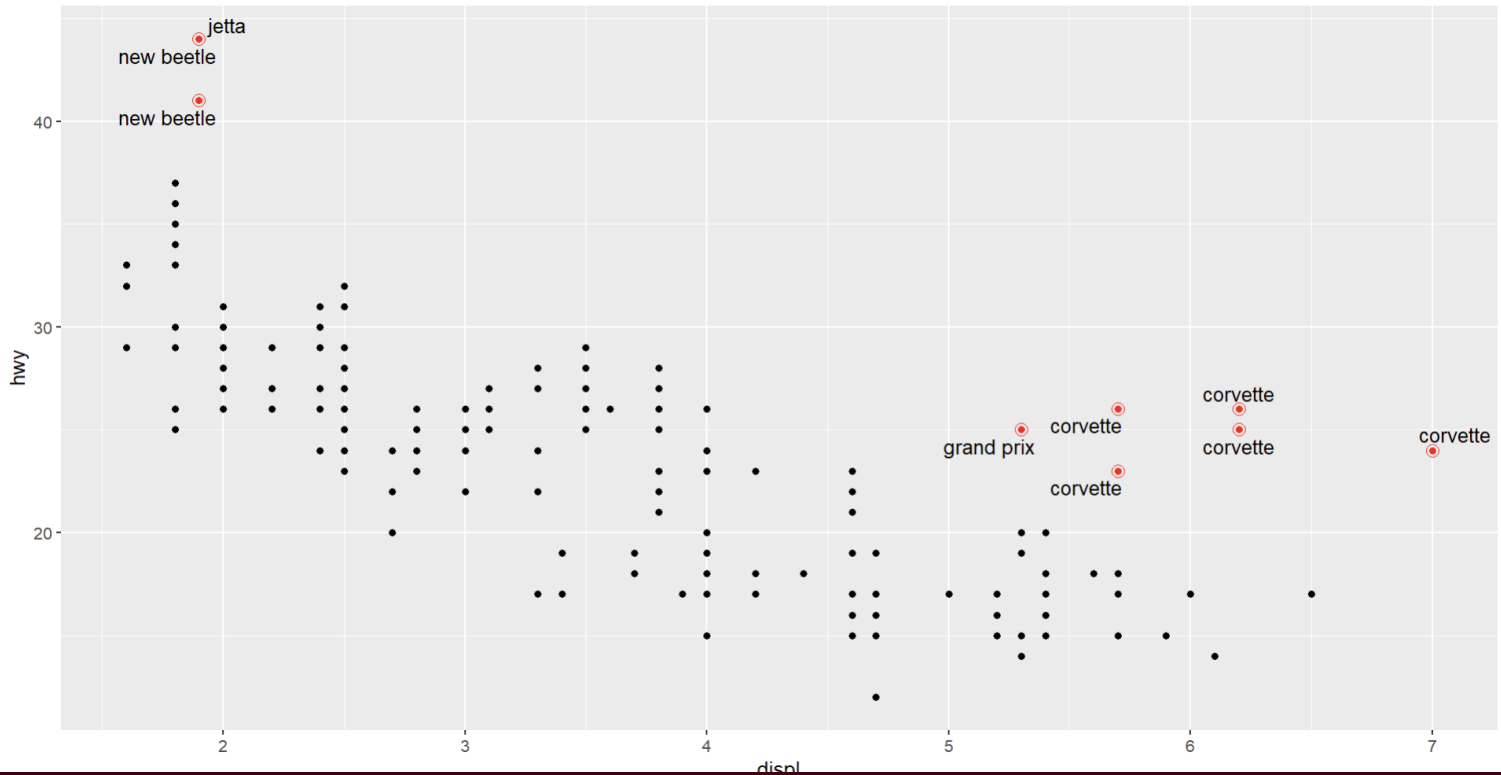
```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point(alpha = 0.3) +
  geom_smooth(se = FALSE) +
  geom_label_repel(
    data = label_info,
    aes(x = displ, y = hwy, label = drive_type),
    fontface = "bold", size = 5, nudge_y = 2
  ) +
  theme(legend.position = "none")
```



Annotations

Or... You may want to highlight a subset of points on the plot:

```
potential_outliers <- mpg |>
  filter(hwy > 40 | (hwy > 20 & displ > 5))

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_text_repel(data = potential_outliers, aes(label = model)) +
  geom_point(data = potential_outliers, color = "red") +
  geom_point(
    data = potential_outliers,
    color = "red", size = 3, shape = "circle open"
  )
```



Annotations

Other useful geoms:

- `geom_hline()` for horizontal lines (`yintercept`).
- `geom_vline()` for vertical lines (`xintercept`).
- `geom_rect()` for drawing rectangles around points (`xmin, xmax, ymin, ymax`).
- `geom_segment()` for drawing line segments / arrows between points (`x, y, xend, yend`).

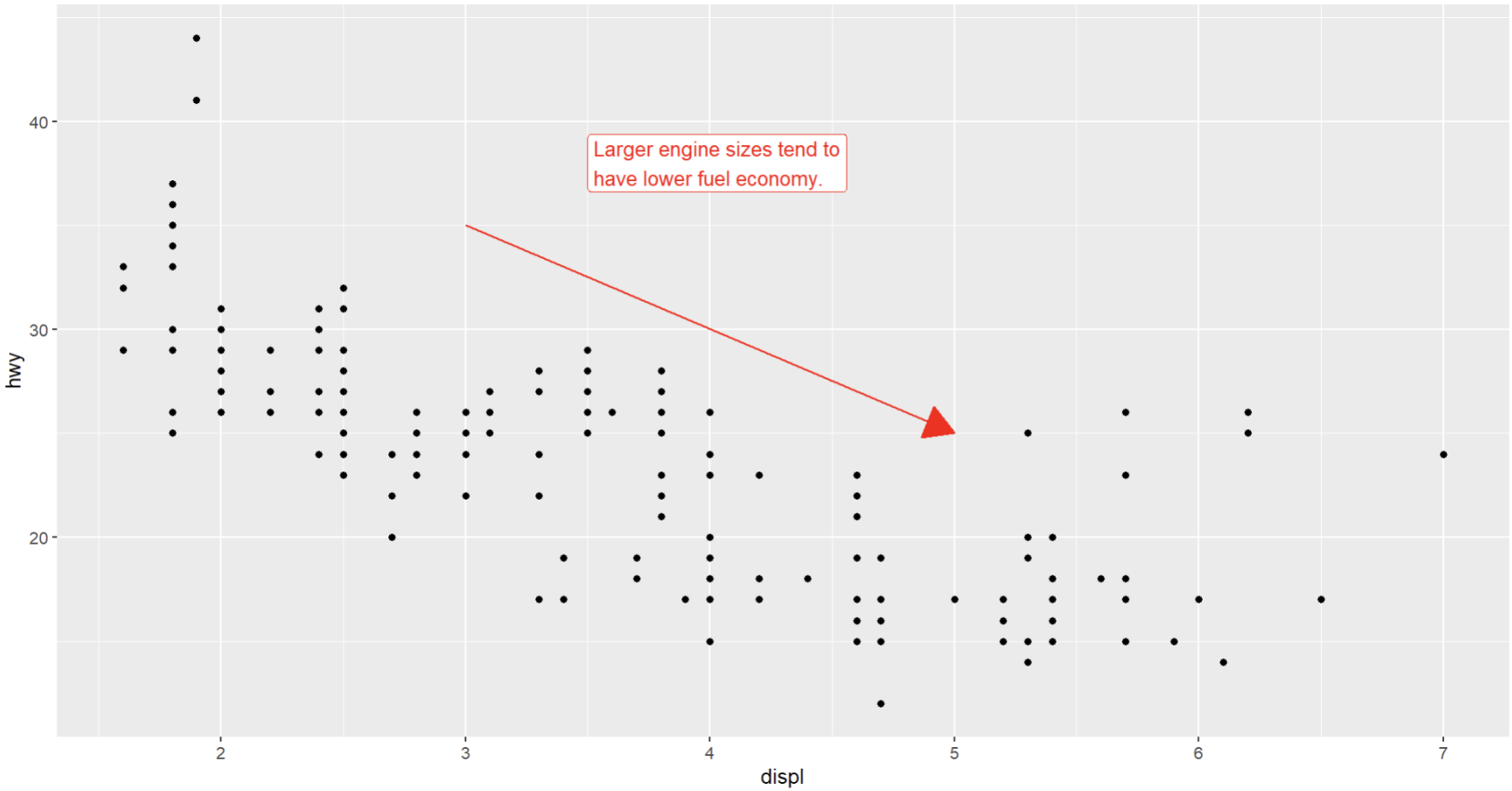
Annotations

If you only have 1 or 2 points to highlight – `annotate()`:

```
trend_text <- "Larger engine sizes tend to have lower fuel economy." |>
  str_wrap(width = 30)

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  annotate(
    geom = "label", x = 3.5, y = 38,
    label = trend_text,
    hjust = "left", color = "red"
  ) +
  annotate(
    geom = "segment",
    x = 3, y = 35, xend = 5, yend = 25, color = "red",
    arrow = arrow(type = "closed")
  )
```

Annotations

If you only have 1 or 2 points to highlight – `annotate()`:



Scales

- Improve communication through adjusting the **guides** (axes and legend).
- ggplot2 has default scales behind the scenes, but you can also adjust them yourself.
- It is a good idea to use `library(scales)` to make sure you have access to all the scale functions.
- These two are the same, because of defaults used by `ggplot2`:

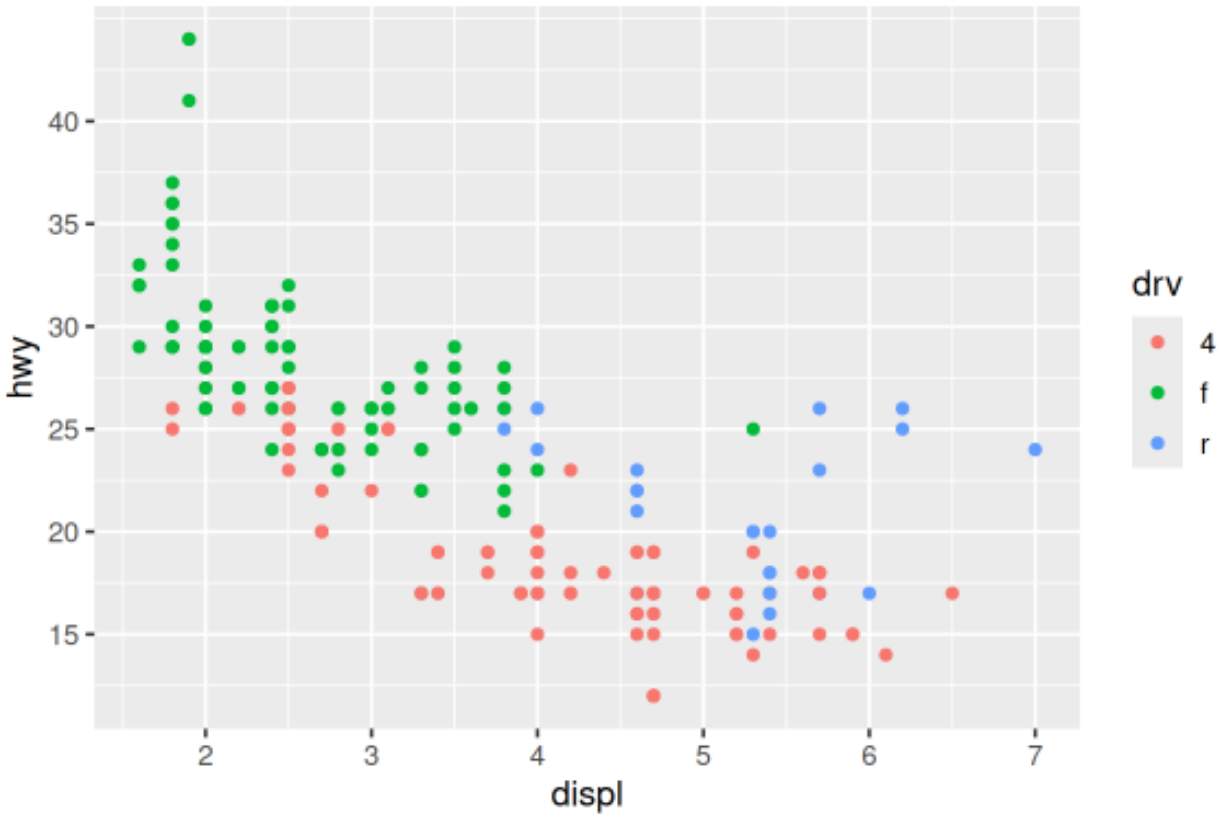
```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class))

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

Scales

Axis ticks and legend keys:

- Breaks

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

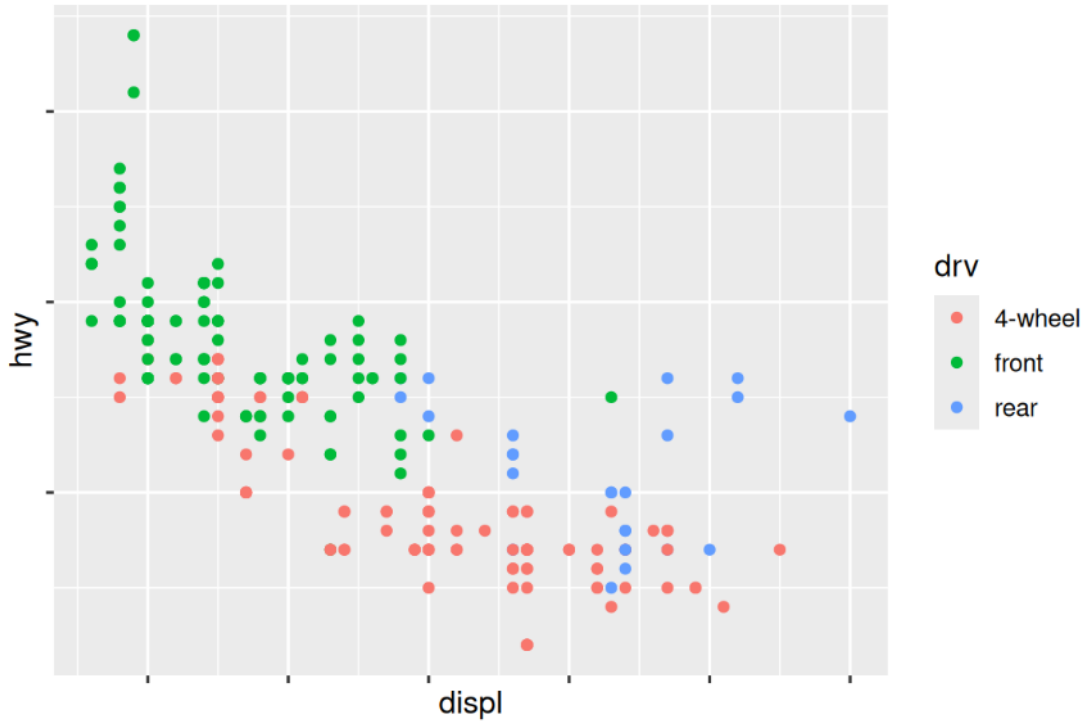


Scales

Axis ticks and legend keys:

- Labels

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  scale_x_continuous(labels = NULL) +
  scale_y_continuous(labels = NULL) +
  scale_color_discrete(labels = c("4" = "4-wheel", "f" = "front", "r" = "rear"))
```

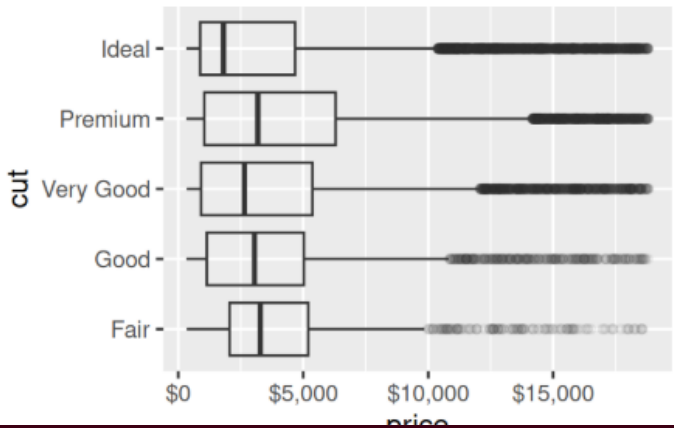


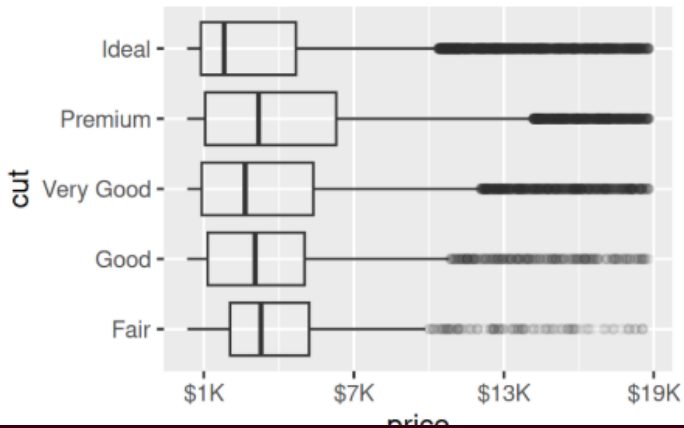
Scales

Axis ticks and legend keys:

- Labels – label formatting

```
# Left
ggplot(diamonds, aes(x = price, y = cut)) +
  geom_boxplot(alpha = 0.05) +
  scale_x_continuous(labels = label_dollar())

# Right
ggplot(diamonds, aes(x = price, y = cut)) +
  geom_boxplot(alpha = 0.05) +
  scale_x_continuous(
    labels = label_dollar(scale = 1/1000, suffix = "K"),
    breaks = seq(1000, 19000, by = 6000)
  )
```



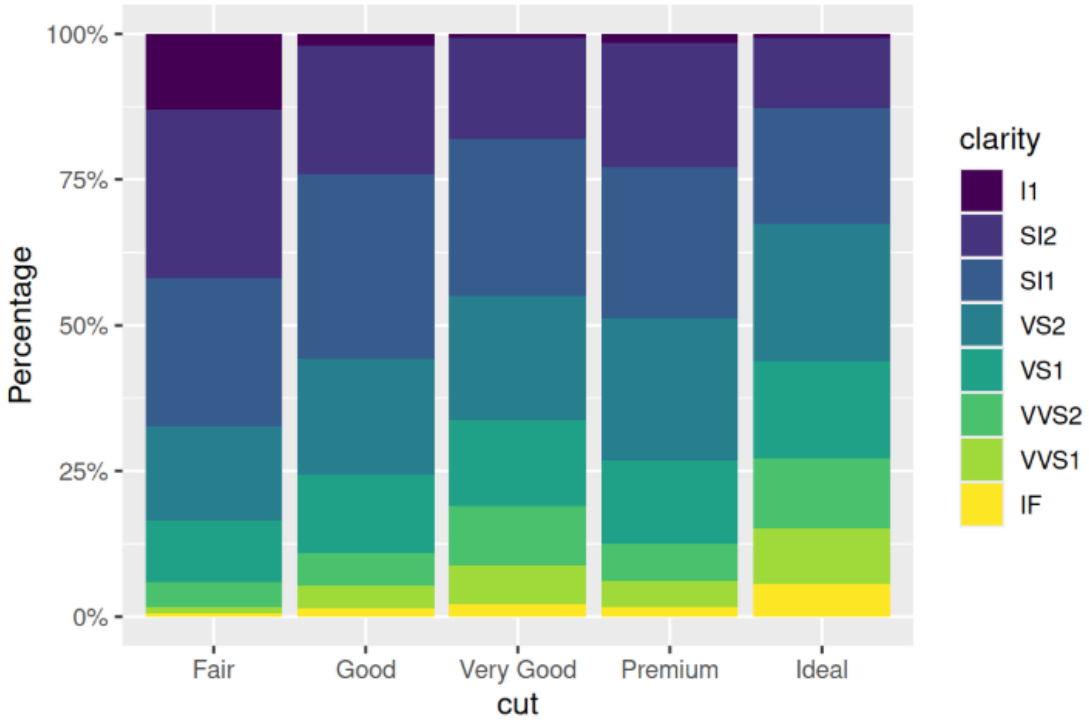


Scales

Axis ticks and legend keys:

- Labels – label formatting

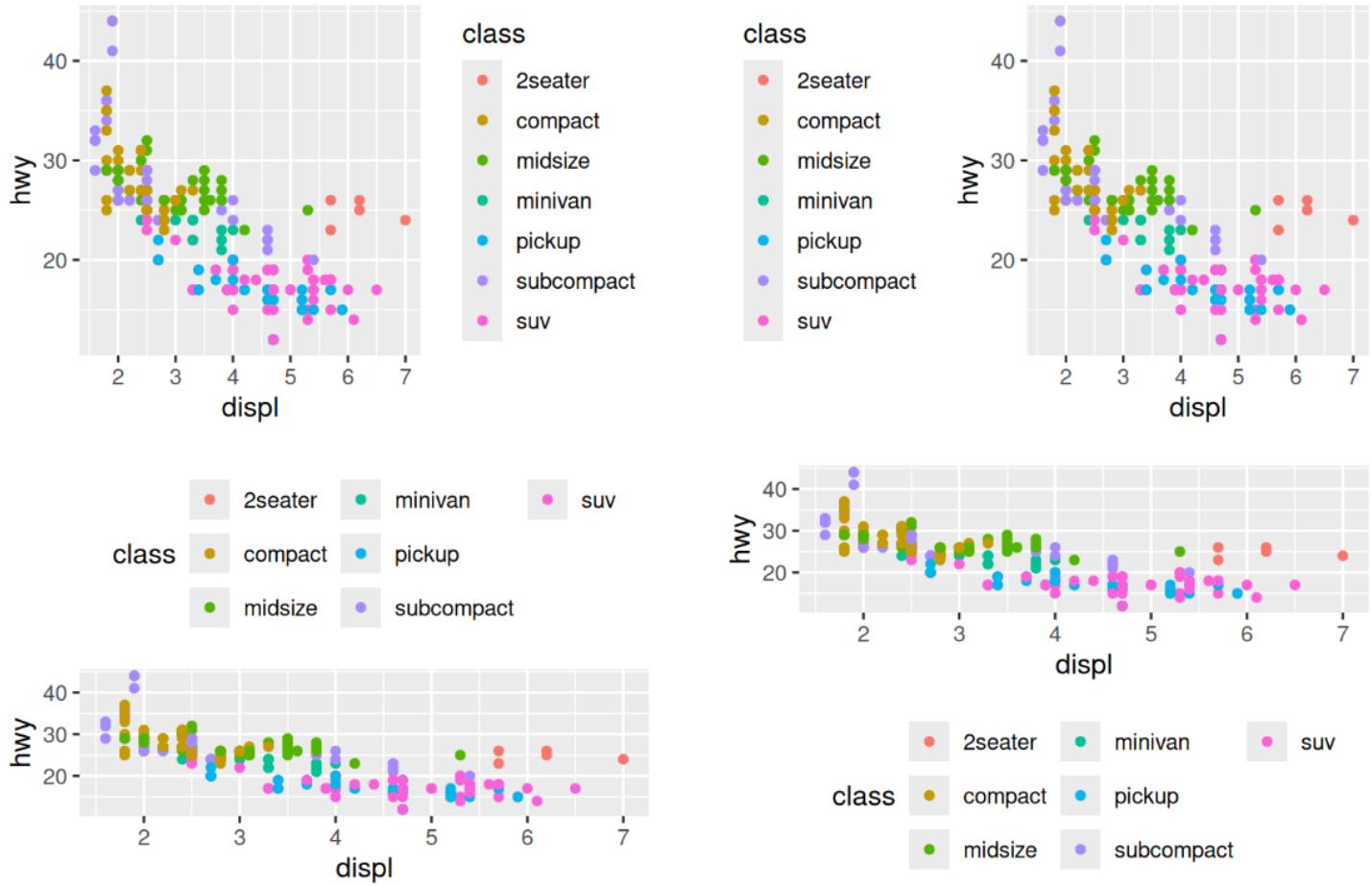
```
ggplot(diamonds, aes(x = cut, fill = clarity)) +
  geom_bar(position = "fill") +
  scale_y_continuous(name = "Percentage", labels = label_percent())
```



Scales

Legend position:

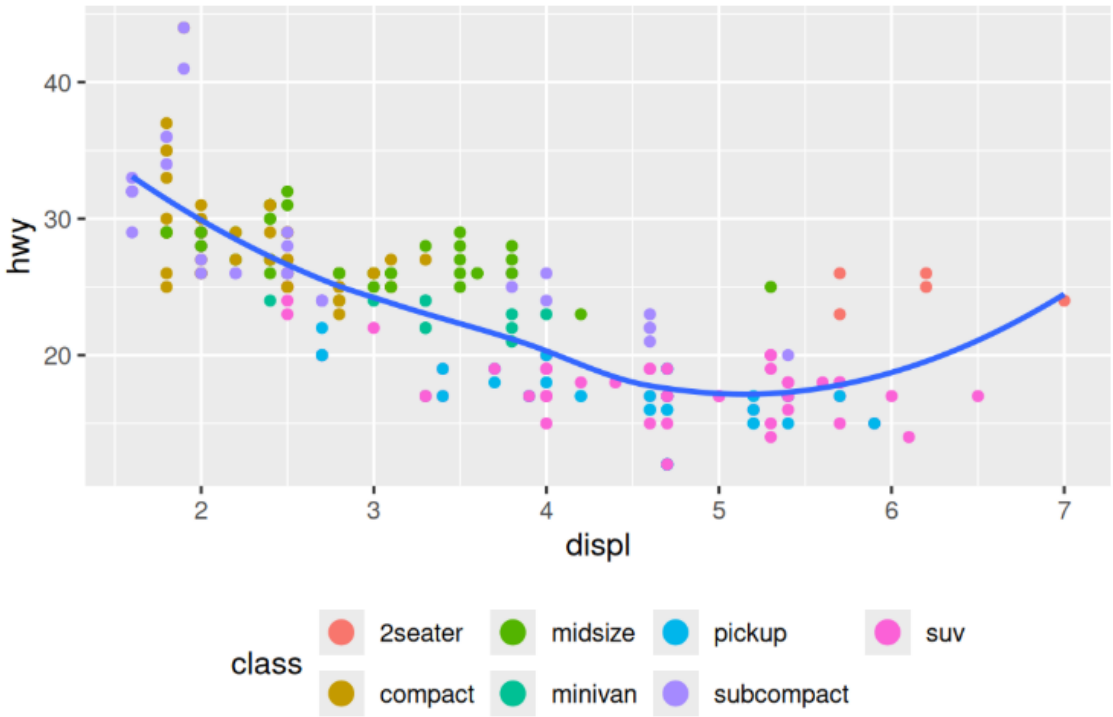
```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  theme(legend.position = "left") #right, top, bottom, none
```



Scales

Legend settings:

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  theme(legend.position = "bottom") +
  guides(color = guide_legend(nrow = 2, override.aes = list(size = 4)))
#> `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

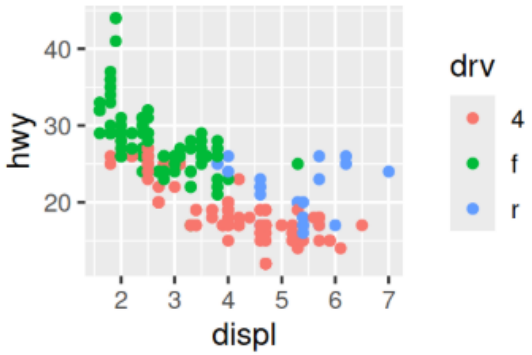


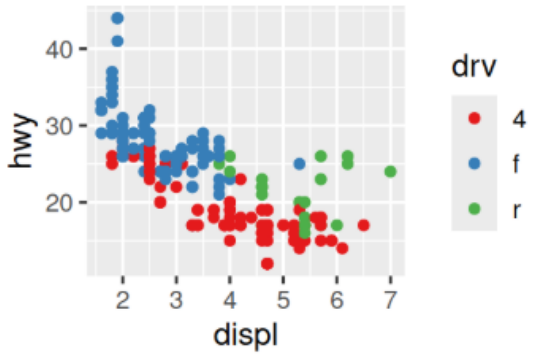
Scales

Colour Scales:

- ColorBrewer has useful palettes (taking colour blindness into account)

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv))

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  scale_color_brewer(palette = "Set1")
```



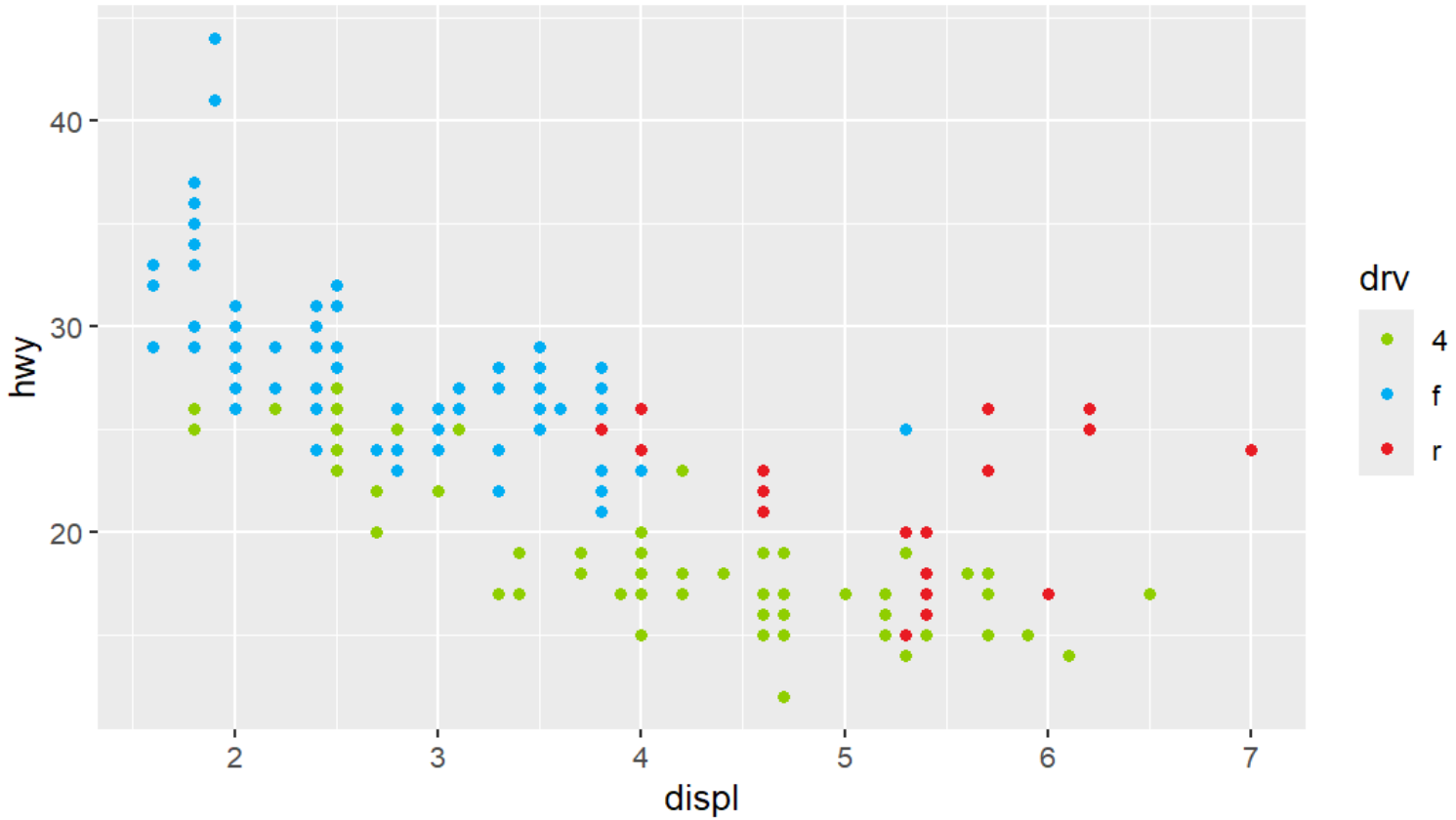


Scales

Colour Scales:

- Manually set colours

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  scale_color_manual(values = c(`4` = "#8FCE00", r = "#E81B23", f = "#00AEF3"))
```



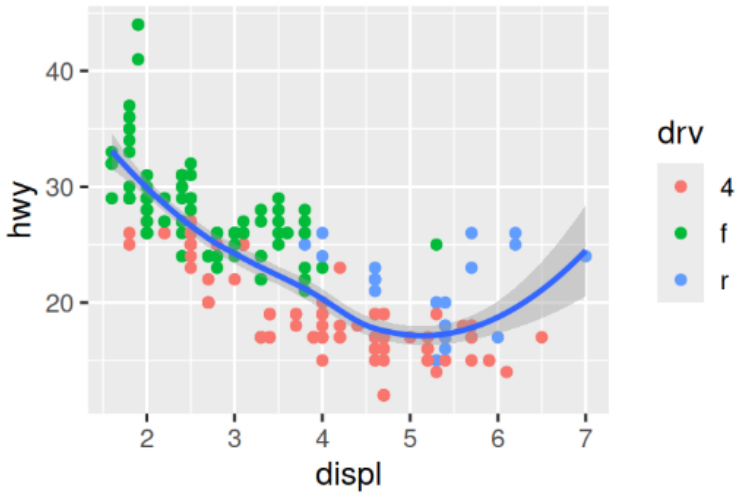
Zooming

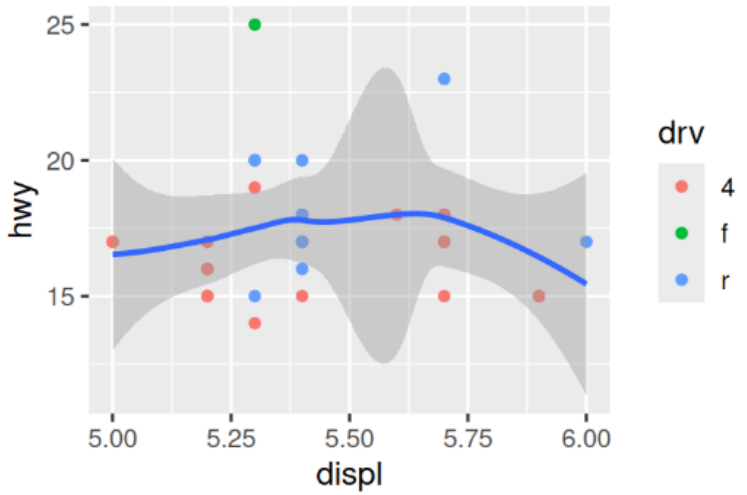
Three ways to control the plot limits:

- Adjust/filter the input data
- Use `xlim()` and `ylim()` in `coord_cartesian()`
- Setting limits in the scale functions:
  `scale_x_continuous(limits = c(...))`

Zooming

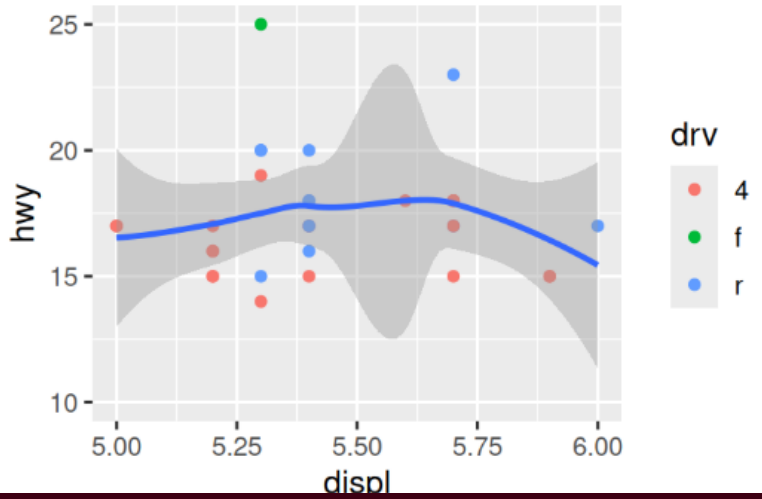
```
# Left
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth()

# Right
mpg |>
  filter(displ >= 5 & displ <= 6 & hwy >= 10 & hwy <= 25) |>
  ggplot(aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth()
```

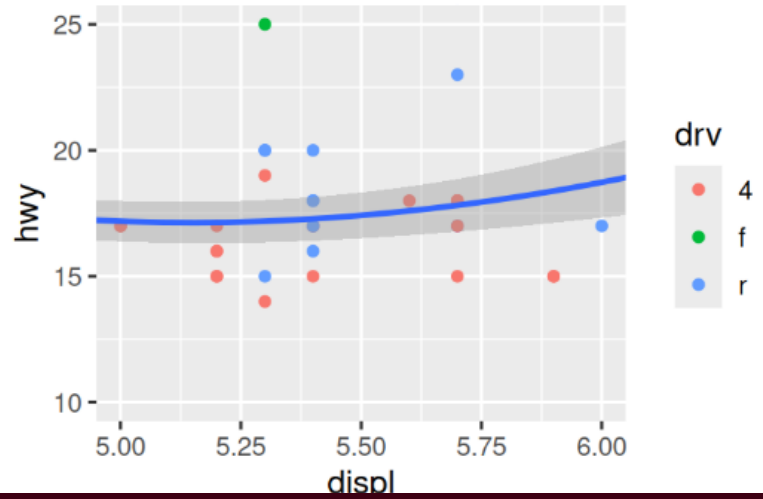




Zooming

```
# Left
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth() +
  scale_x_continuous(limits = c(5, 6)) +
  scale_y_continuous(limits = c(10, 25))

# Right
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth() +
  coord_cartesian(xlim = c(5, 6), ylim = c(10, 25))
```





Themes

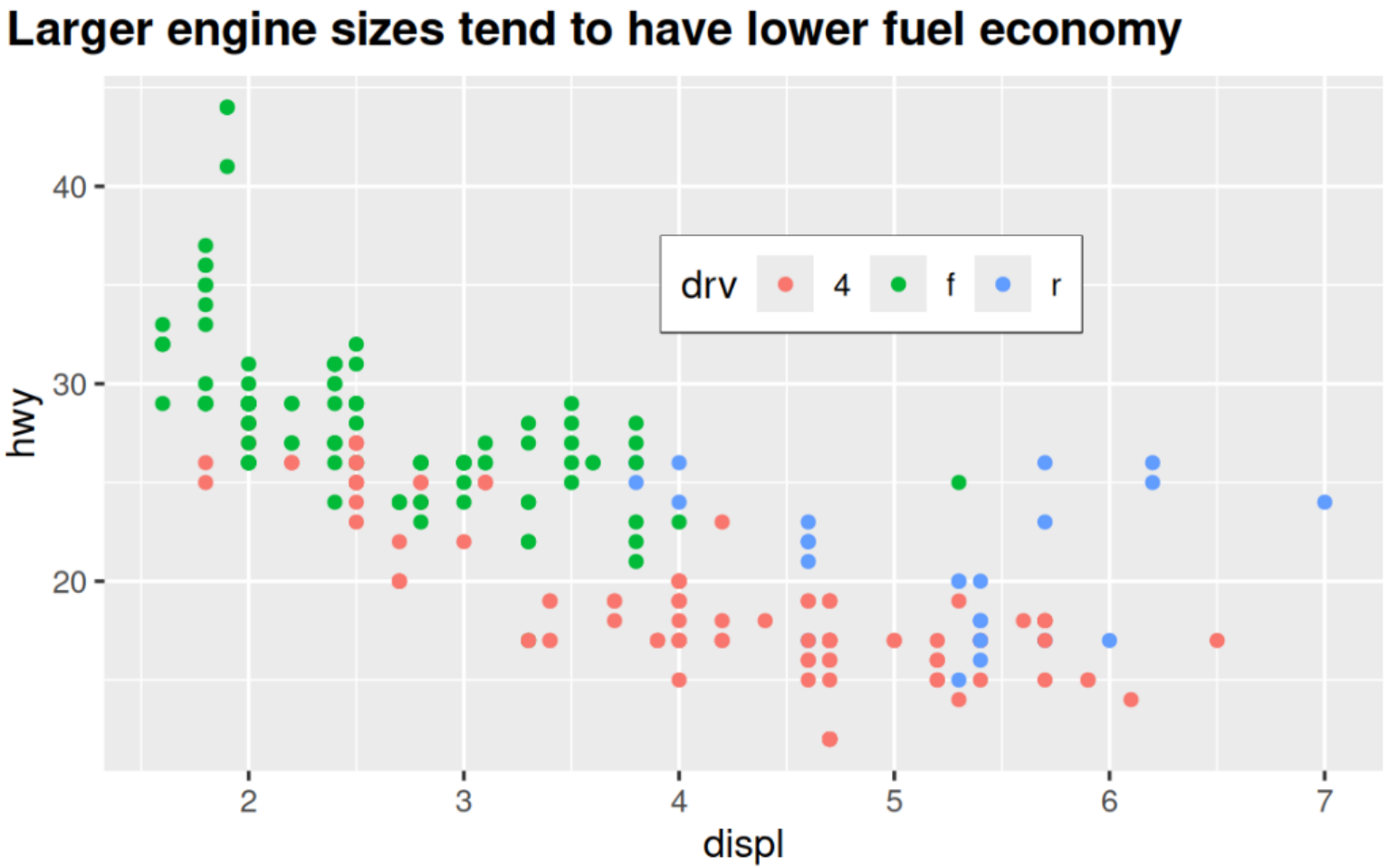
- Apply themes: e.g., `theme_minimal()`, `theme_classic()`
- See R4DS textbook for list of eight themes built into `ggplot2`.
- Customize individual theme elements: fonts, grid lines, etc.

Themes

- Customize individual theme elements: fonts, grid lines, etc.

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  labs(
    title = "Larger engine sizes tend to have lower fuel economy",
    caption = "Source: https://fueleconomy.gov."
  ) +
  theme(
    legend.position = c(0.6, 0.7),
    legend.direction = "horizontal",
    legend.box.background = element_rect(color = "black"),
    plot.title = element_text(face = "bold"),
    plot.title.position = "plot",
    plot.caption.position = "plot",
    plot.caption = element_text(hjust = 0)
  )
```

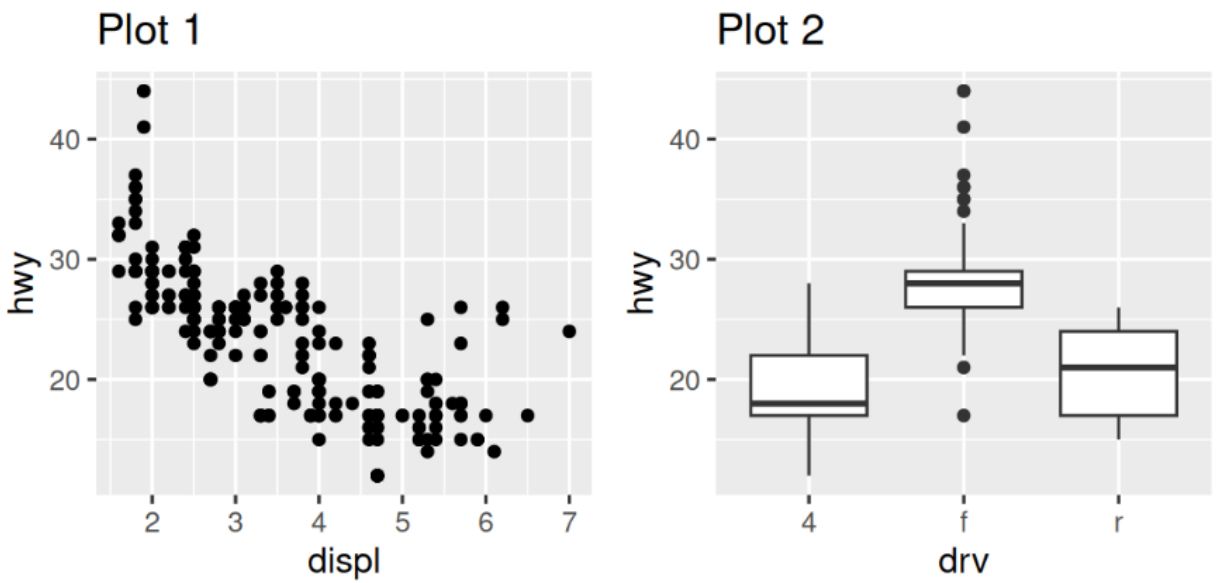
Themes


Larger engine sizes tend to have lower fuel economy
drv
4
f
r
hwy
40
30
20
displ
2
3
4
5
6
7
Source: https://fueleconomy.gov.

Layout

Use `patchwork` to arrange plots:

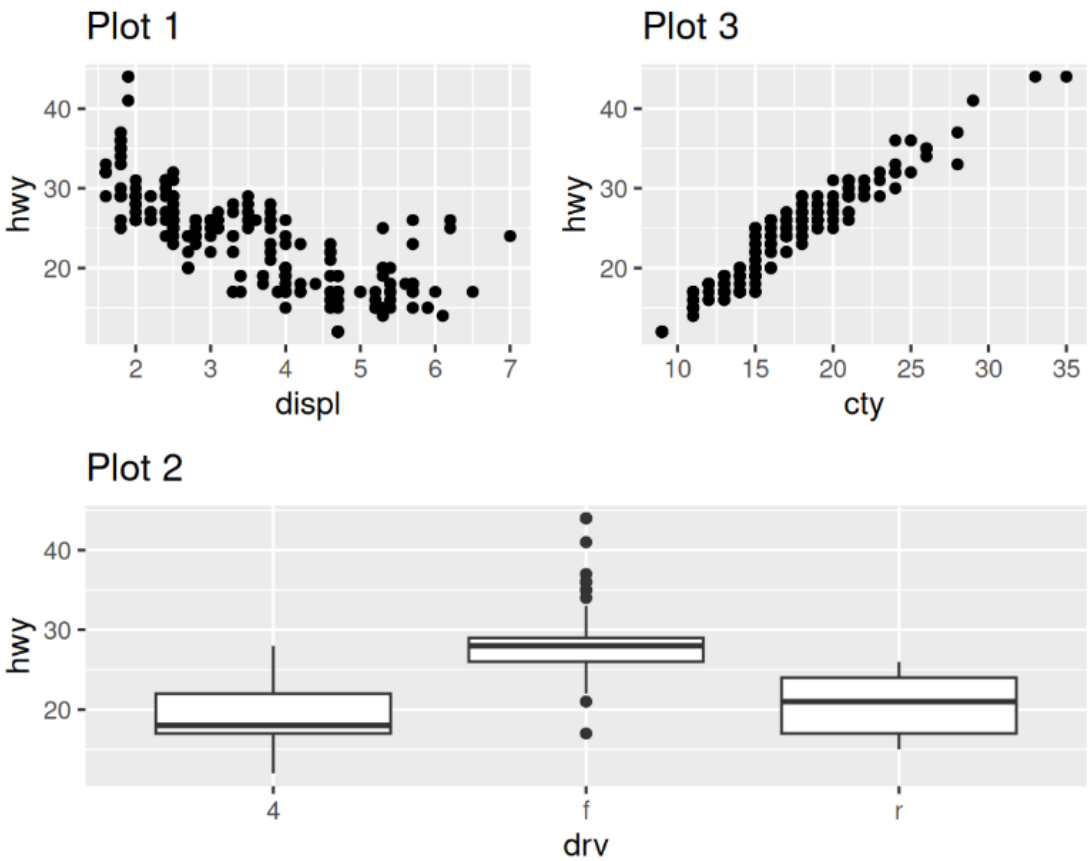
```
p1 <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  labs(title = "Plot 1")
p2 <- ggplot(mpg, aes(x = drv, y = hwy)) +
  geom_boxplot() +
  labs(title = "Plot 2")
p1 + p2
```



Layout

Use `patchwork` to arrange plots:

```
p3 <- ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  labs(title = "Plot 3")
(p1 | p3) / p2
```



More info

- Comprehensive guide to `ggplot2`: https://ggplot2-book.org/