3. Plotting with ggplot2 and plotly

Jonathan Hersh, PhD (Chapman Argyros School of Business)

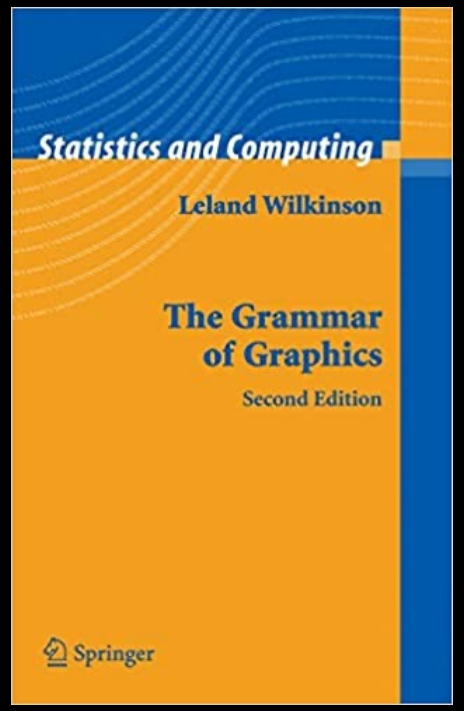
10/24/22

Language follows grammar

The boy hit the ball

S V O

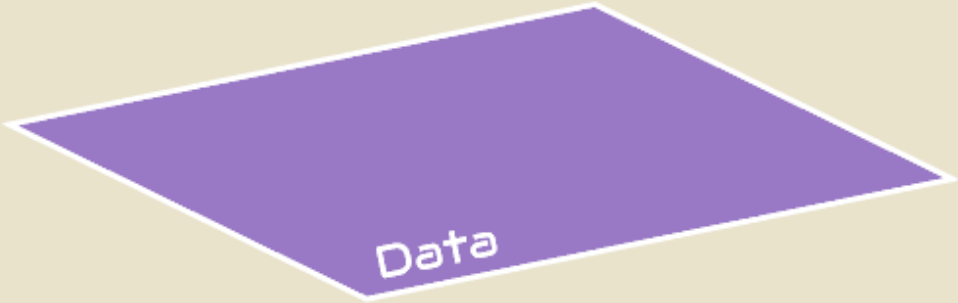
Can graphics follow a grammar?

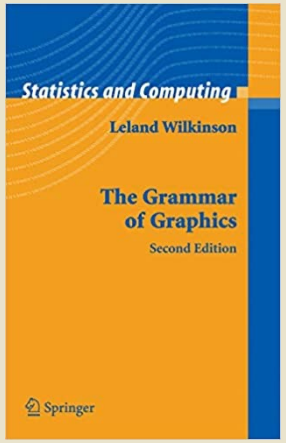


Grammar of Graphics

```r
#------------------------------- Visualising your boxplot -------------
# Before plotting if not installed install.package("ggplot2")
# Then activate ggplot2 package
library(ggplot2)

# Create new variable for plot only x and y axis. ('data' and 'aestatics' layer)
plot <- ggplot(data=new.data, aes(x=Genre, y=Gross...US))

# Create new variable with geometries layer.
q <- plot + geom_jitter(aes(fill=Studio, size=Budget...mill.),
                        shape = 21, # this will shape a border around data points.
                        colour = "Black") + # with the border color of black.
  geom_boxplot(alpha=0.7, outlier.color = NA) # places the boxplot on the data points
                                              # and removes boxplot layer outliers.

# Change axis and title if needed.
q <- q +
  xlab("Genre") +
  ylab("Gross % US") +
  ggtitle("Domestic Gross % by Genre")

# Make your plot visually attractive and readable with the 'theme' function. (Theme layer)
q + theme(axis.title = element_text(colour = "Blue", size = 14),
          axis.text = element_text(size = 12),
          legend.title = element_text(size = 12),
          legend.text = element_text(size = 10),
          plot.title = element_text(size = 14, hjust = 0.5), # 'hjust' will center your text.
          panel.background = element_rect(fill = "#E6E3C5"))
```

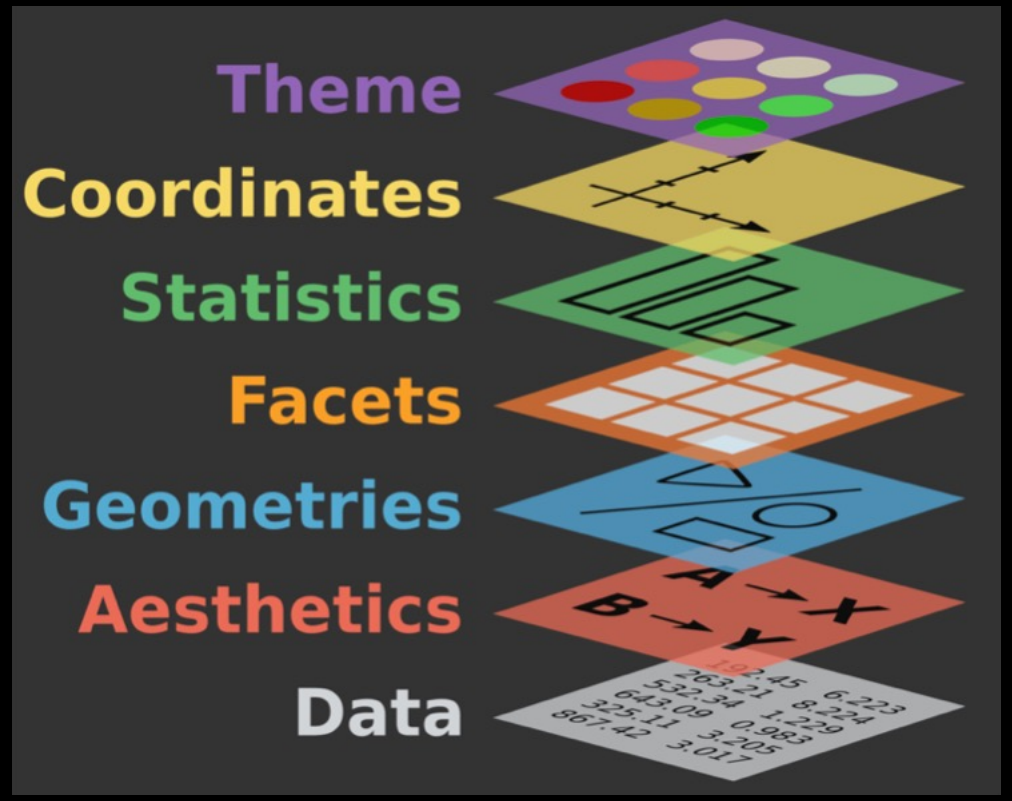
xy, 3902, 29, 9,
4756, x, 72, 633,
647, 617, 827, 3,
1, 21, 45, tyu, 6,
987, 457, 283, 8,
4, 5, 671, 34, 67,
x, 981, hu, 89, 5





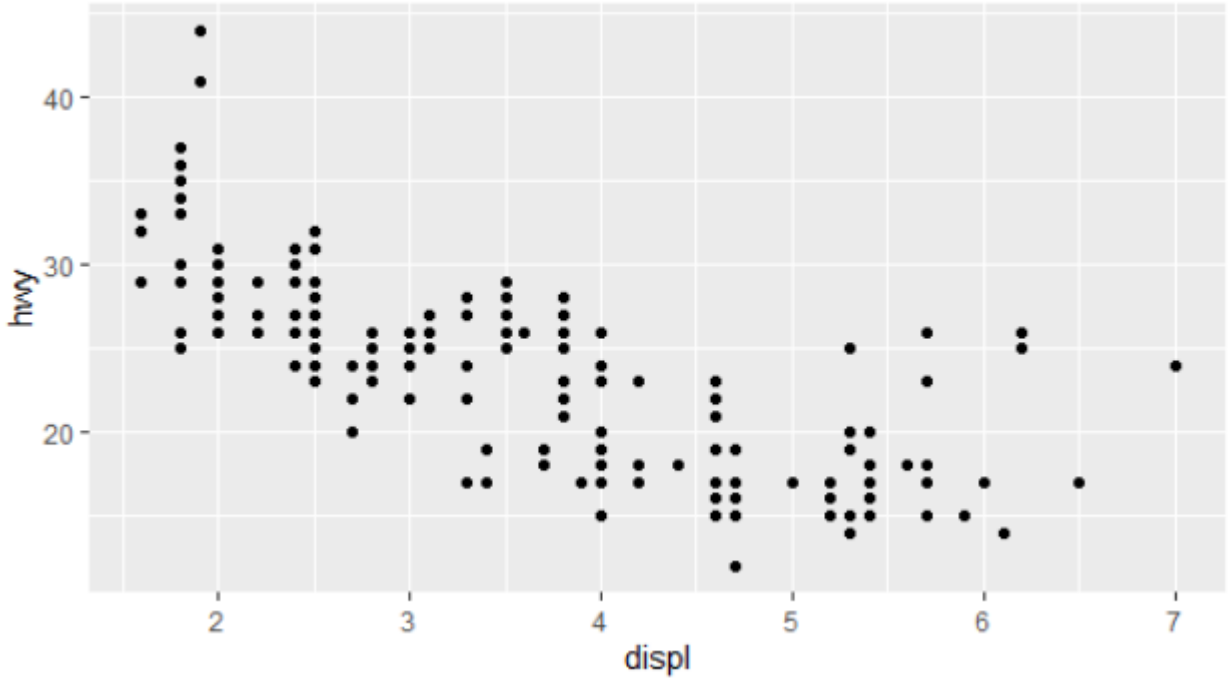
Seven grammar elements of every plot

1. **Data**: What is the data you want to visualize?
2. **Aesthetics**: What data will be on the x and y axes?
3. **Geometry**: What shapes (bars, lines, points) will you use to represent your data?
4. **Facets**: Will your data be split into multiple plots? If so, how?
5. **Statistics**: Will you use statistical summarizes on your data (e.g. smoothing lines)
6. **Coordinates**: What's the numeric plotting space?
7. **Themes**: What is the visual identity (fonts, size, colors)?



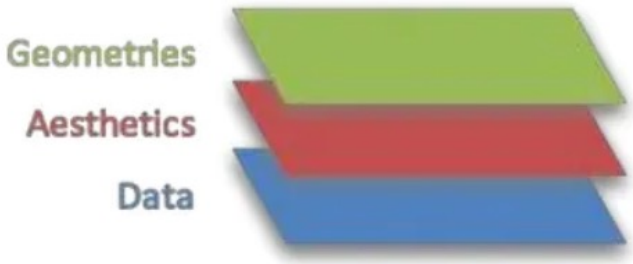
Layers 1-2-3: Data – Aesthetics - Geometries


```
library('ggplot2')
data(mpg)
ggplot(data = mpg , aes(x = displ, y = hwy)) + geom_point()
```

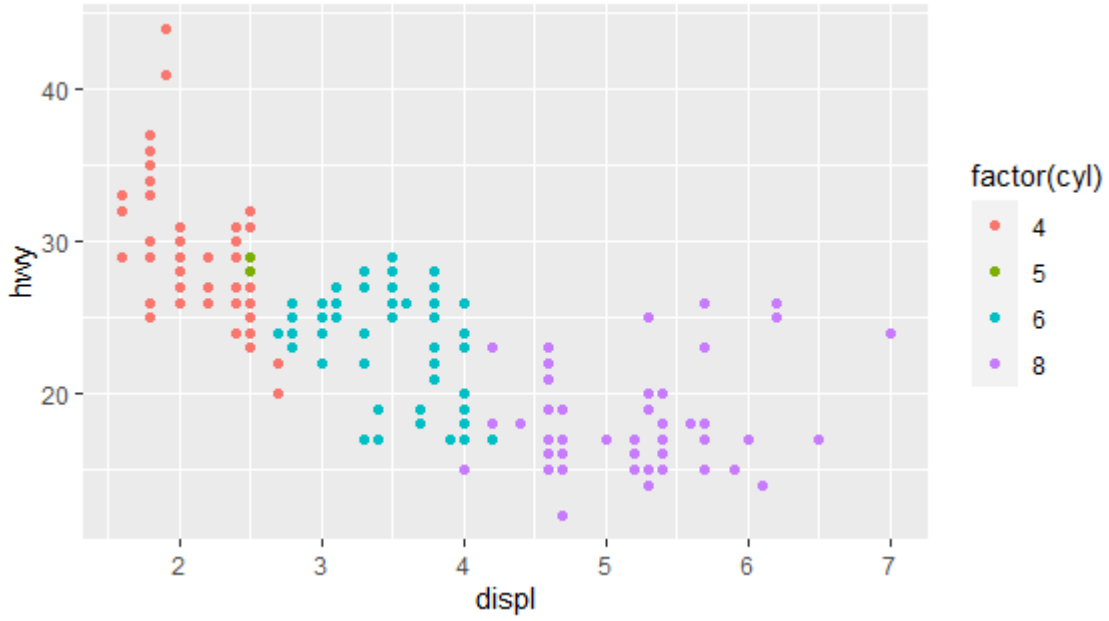


Layers 1-2-3: Data – Aesthetics - Geometries

Let's change the aesthetic by coloring points by number of cylinders

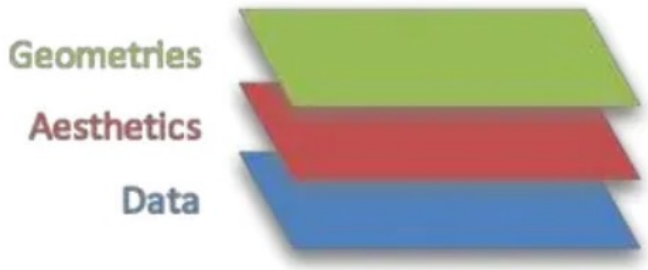



```
library('ggplot2')
data(mpg)
ggplot(data = mpg, aes(x = displ, y = hwy, color = cyl)) + geom_point()
```

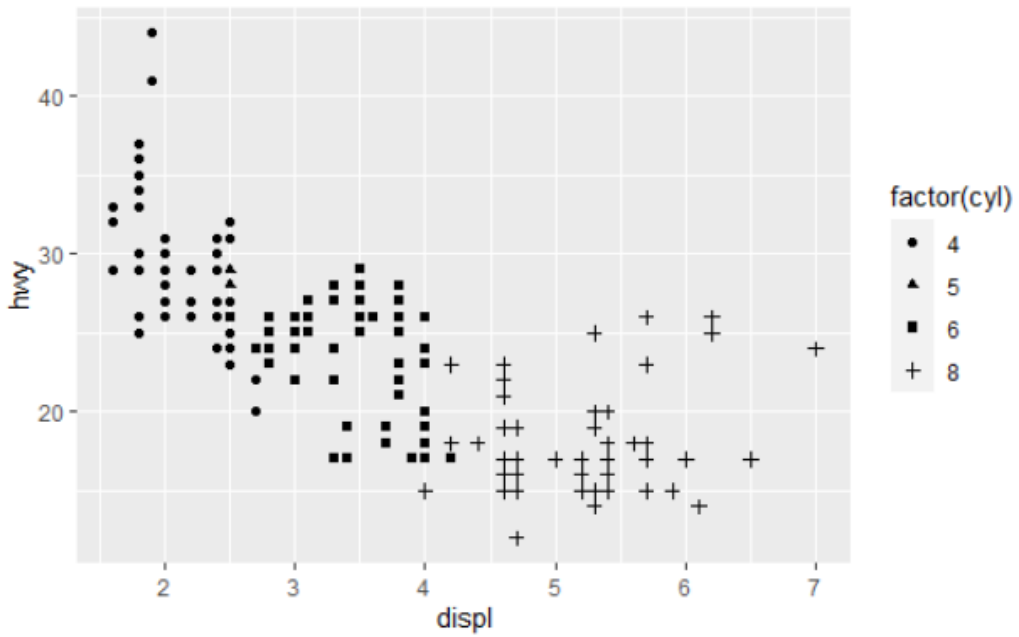


Layers 1-2-3: Data – Aesthetics - Geometries

Let's change the shape of the points




```
library('ggplot2')
data(mpg)
ggplot(data = mpg, aes(x = displ, y = hwy, shape = cyl)) + geom_point()
```



Example geometries

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin"**)**
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly()
x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)**)**
x, y, alpha, color, fill, linetype, size, weight

TWO VARIABLES
both continuous

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"**)**
x, y, alpha, color, linetype, size

e + geom_smooth(method = lm**)**
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1**)** - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500)**)**
x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv"**)**
x, y, alpha, color, group, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z)**)**
x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z)**)**
x, y, alpha, color, fill, group, linetype, size, subgroup

l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE**)**
x, y, alpha, fill

l + geom_tile(aes(fill = z)**)**
x, y, alpha, color, fill, linetype, size, width

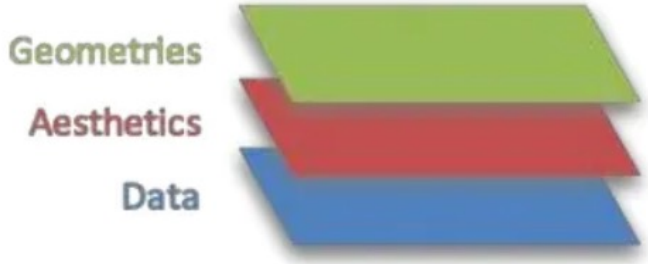
maps

```
data <- data.frame(murder = USArrests$Murder,
          state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))
```

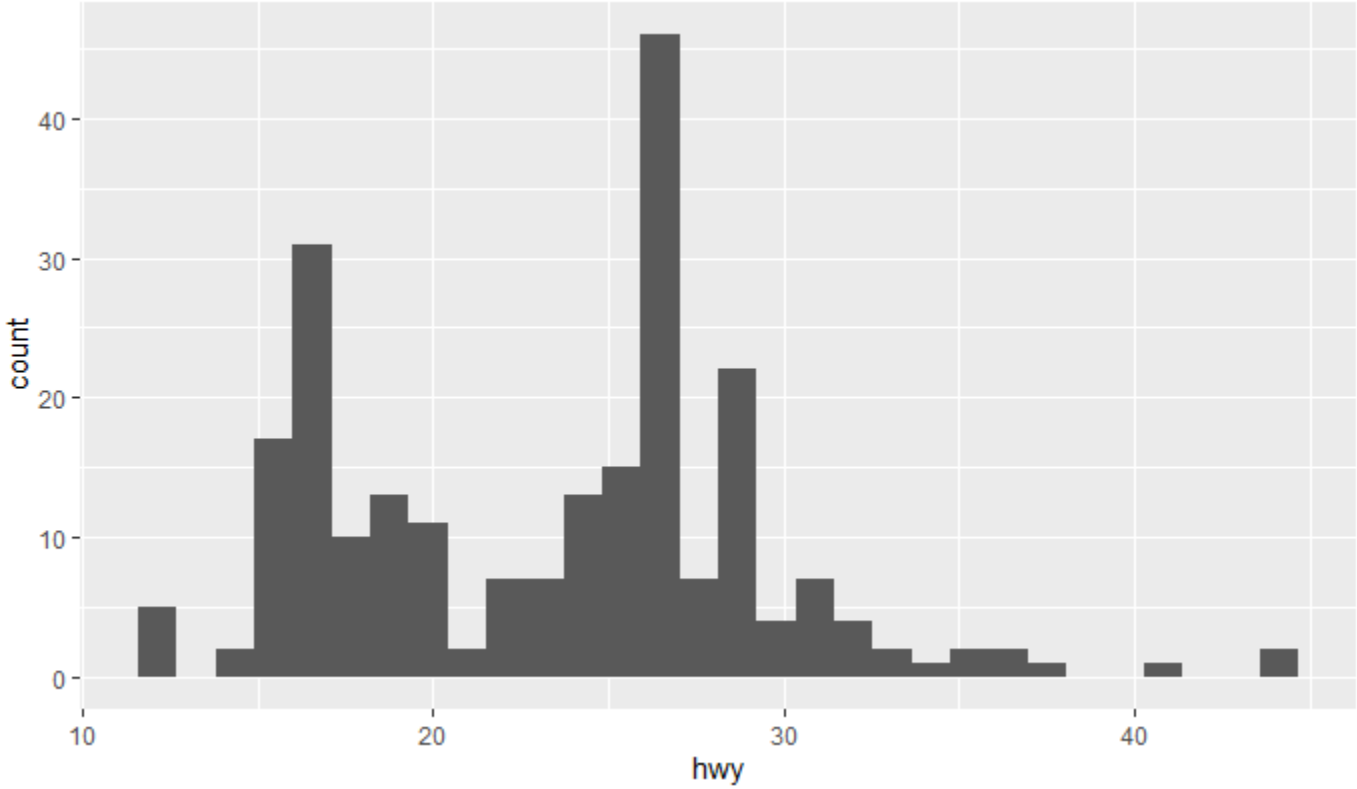
k + geom_map(aes(map_id = state), map = map**)**
+ expand_limits(x = map$long, y = map$lat**)**
map_id, alpha, color, fill, linetype, size

Layers 1-2-3: Data – Aesthetics - Geometries

Let's create a histogram for highway mile per gallon

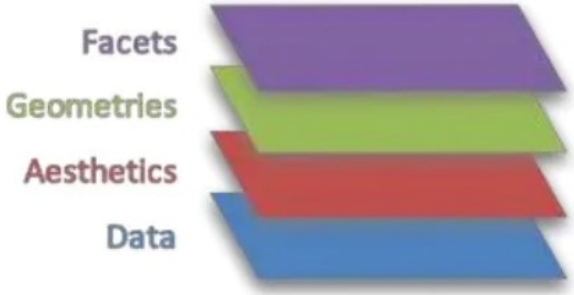


ggplot(data = mpg, aes(x = hwy)) + **geom_histogram**()

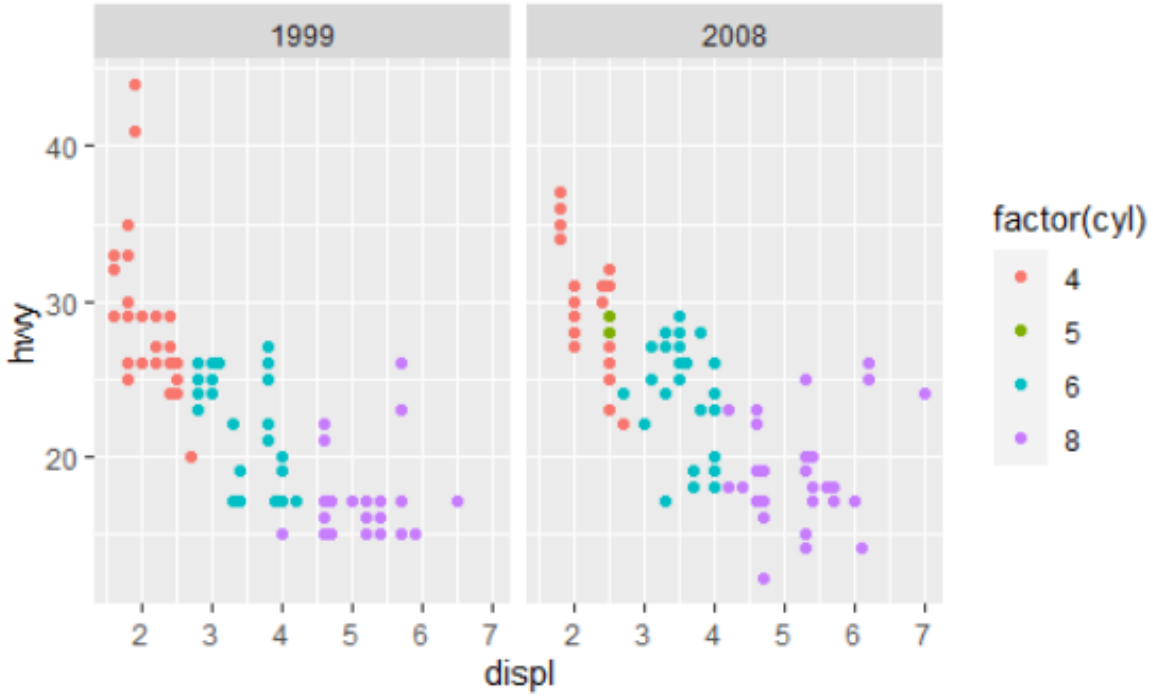


Layer 4: Facet

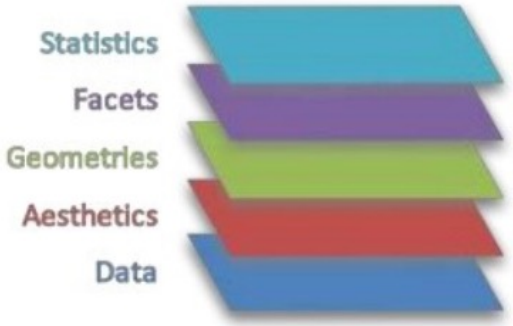
- We have data from two years
- Let's plot that data side by side




```
ggplot(data = mpg, aes(x = displ, y = hwy, shape = cyl)) + geom_point() +
        facet_wrap(~ year)
```

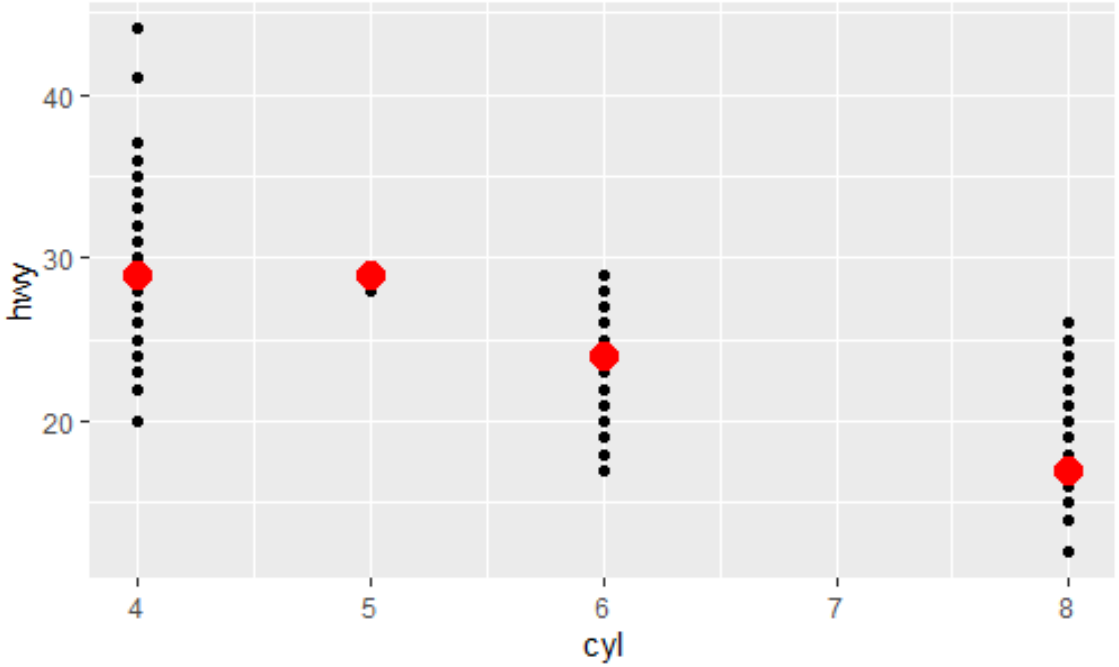


Layer 5: Statistics

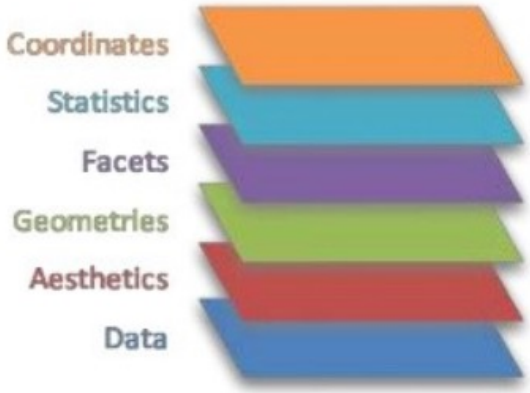


- Let's add a layer that adds the median value of highway miles per gallon for each cylinder

```
ggplot(data = mpg, aes(x = cyl, y = hwy)) +
  geom_point() +
  stat_summary(fun = "median", color = "red", size = 1)
```

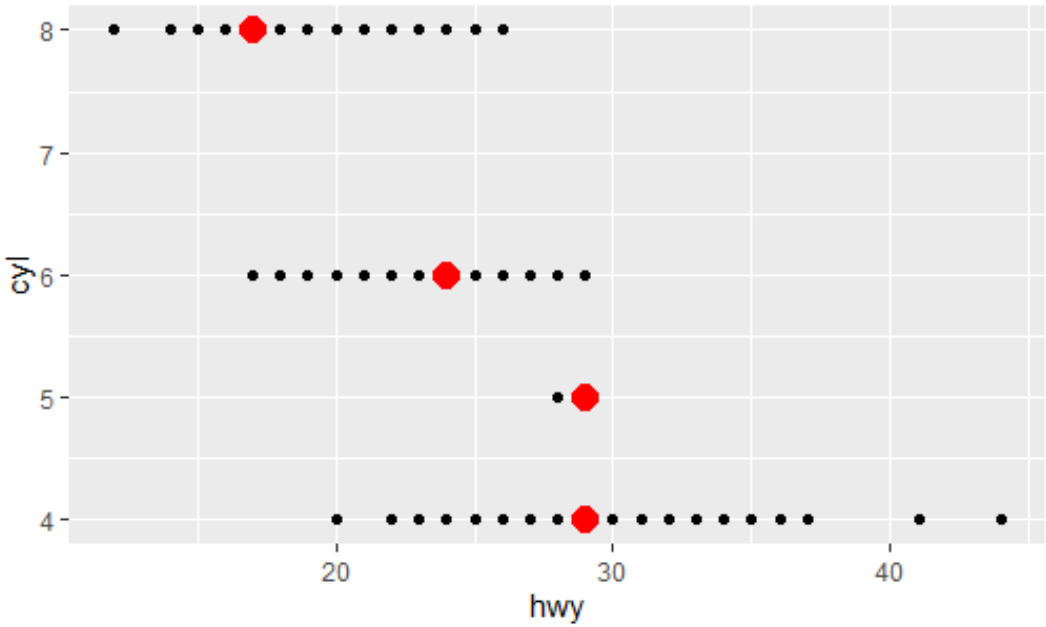


Layer 6: Coordinates

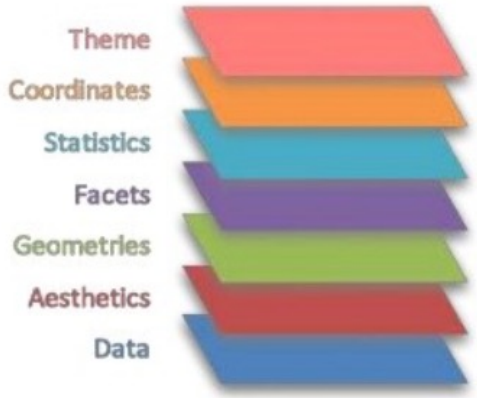


- Let's change the coordinates

```
ggplot(data = mpg, aes(x = cyl, y = hwy)) +
  geom_point() +
  stat_summary(fun = "median", color = "red") +
  coord_flip()
```

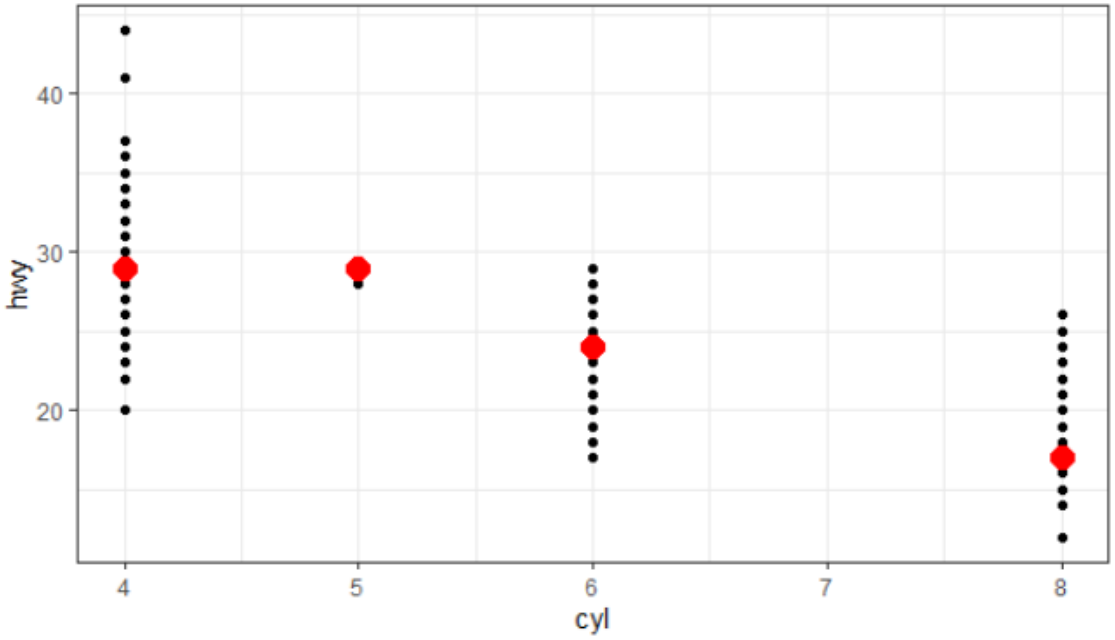


Layer 7: Themes

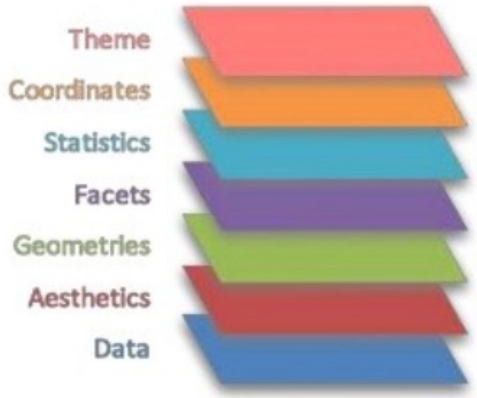


- Themes alter the font, color according to pre-determined rules

```
ggplot(data = mpg, aes(x = cyl, y = hwy)) +
  geom_point() +
  stat_summary(fun = "median", color = "red", size = 1) +
  theme_bw( ) +
```

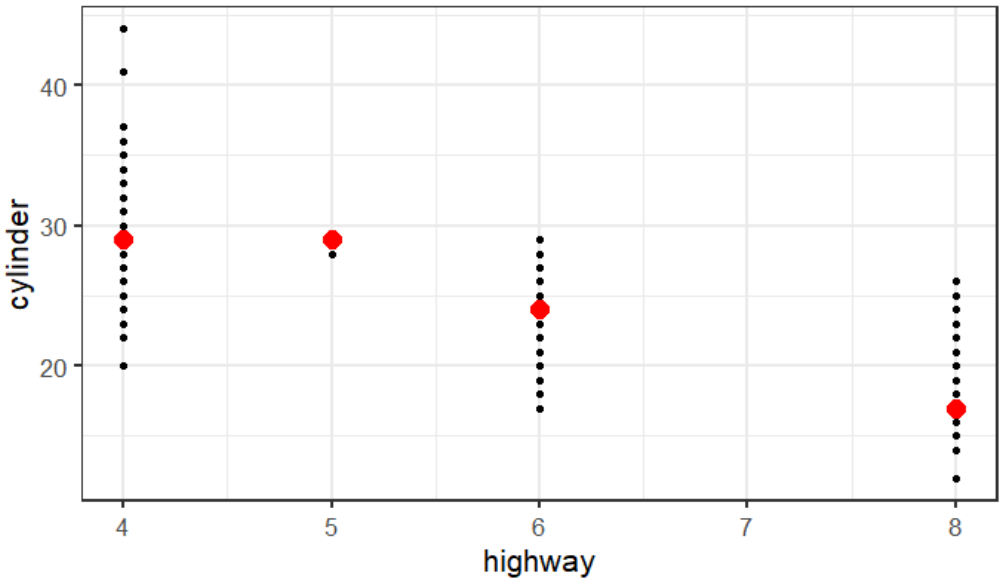


Layer 7: Axes labels and font size



- I always recommend labeling axes titles clearly
- Fonts are often too small so I recommend increasing to font size 14, 16, or 18.

```
ggplot(data = mpg, aes(x = cyl, y = hwy)) +
  geom_point() +
  stat_summary(fun = "median", color = "red") +
  theme_bw(base_size = 16) +
  labs(x = "highway", y = "cylinder")
```



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.

data + geom (x = F, y = A) + coordinate system = plot

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

data + geom (x = F, y = A, color = F, size = A) + coordinate system = plot

Complete the template below to build a graph.

```
ggplot (data = <DATA>) +
<GEOM_FUNCTION>(mapping = aes( <MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

required

Not required, sensible defaults supplied

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

last_plot() Returns the last plot.

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Aes

Common aesthetic values.

color and **fill** - string ("red", "#RRGGBB")

linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotdash", 5 = "longdash", 6 = "twodash")

lineend - string ("round", "butt", or "square")

linejoin - string ("round", "mitre", or "bevel")

size - integer (line width in mm)

shape - integer/shape name or a single character ("a")

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + geom_blank() and **a + expand_limits()**
Ensure limits include values across all plots.

b + geom_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + geom_path(lineend = "butt", linejoin = "round", linemitre = 1)
x, y, alpha, color, group, linetype, size

a + geom_polygon(aes(alpha = 50)) - x, y, alpha, color, fill, group, subgroup, linetype, size

b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))

b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

c + geom_freqpoly()
x, y, alpha, color, group, linetype, size

c + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy))
x, y, alpha, color, fill, linetype, size, weight

discrete

d <- ggplot(mpg, aes(fl))

d + geom_bar()
x, alpha, color, fill, linetype, size, weight

TWO VARIABLES

both continuous

e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_point()
x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile()
x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl")
x, y, alpha, color, linetype, size

e + geom_smooth(method = lm)
x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

one discrete, one continuous

f <- ggplot(mpg, aes(class, hwy))

f + geom_col()
x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot()
x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill, group

f + geom_violin(scale = "area")
x, y, alpha, color, fill, group, linetype, size, weight

both discrete

g <- ggplot(diamonds, aes(cut, color))

g + geom_count()
x, y, alpha, color, fill, shape, size, stroke

e + geom_jitter(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + geom_bin2d(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + geom_density_2d()
x, y, alpha, color, group, linetype, size

h + geom_hex()
x, y, alpha, color, fill, size

continuous function

i <- ggplot(economics, aes(date, unemploy))

i + geom_area()
x, y, alpha, color, fill, linetype, size

i + geom_line()
x, y, alpha, color, group, linetype, size

i + geom_step(direction = "hv")
x, y, alpha, color, group, linetype, size

visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))

j + geom_crossbar(fatten = 2) - x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + geom_errorbar() - x, ymax, ymin, alpha, color, group, linetype, size, width
Also **geom_errorbarh()**.

j + geom_linerange()
x, ymin, ymax, alpha, color, group, linetype, size

j + geom_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))

k + geom_map(aes(map_id = state), map = map) **+ expand_limits**(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size

THREE VARIABLES

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)); l <- ggplot(seals, aes(long, lat))

l + geom_contour(aes(z = z))
x, y, z, alpha, color, group, linetype, size, weight

l + geom_contour_filled(aes(fill = z))
x, y, alpha, color, fill, group, linetype, size, subgroup

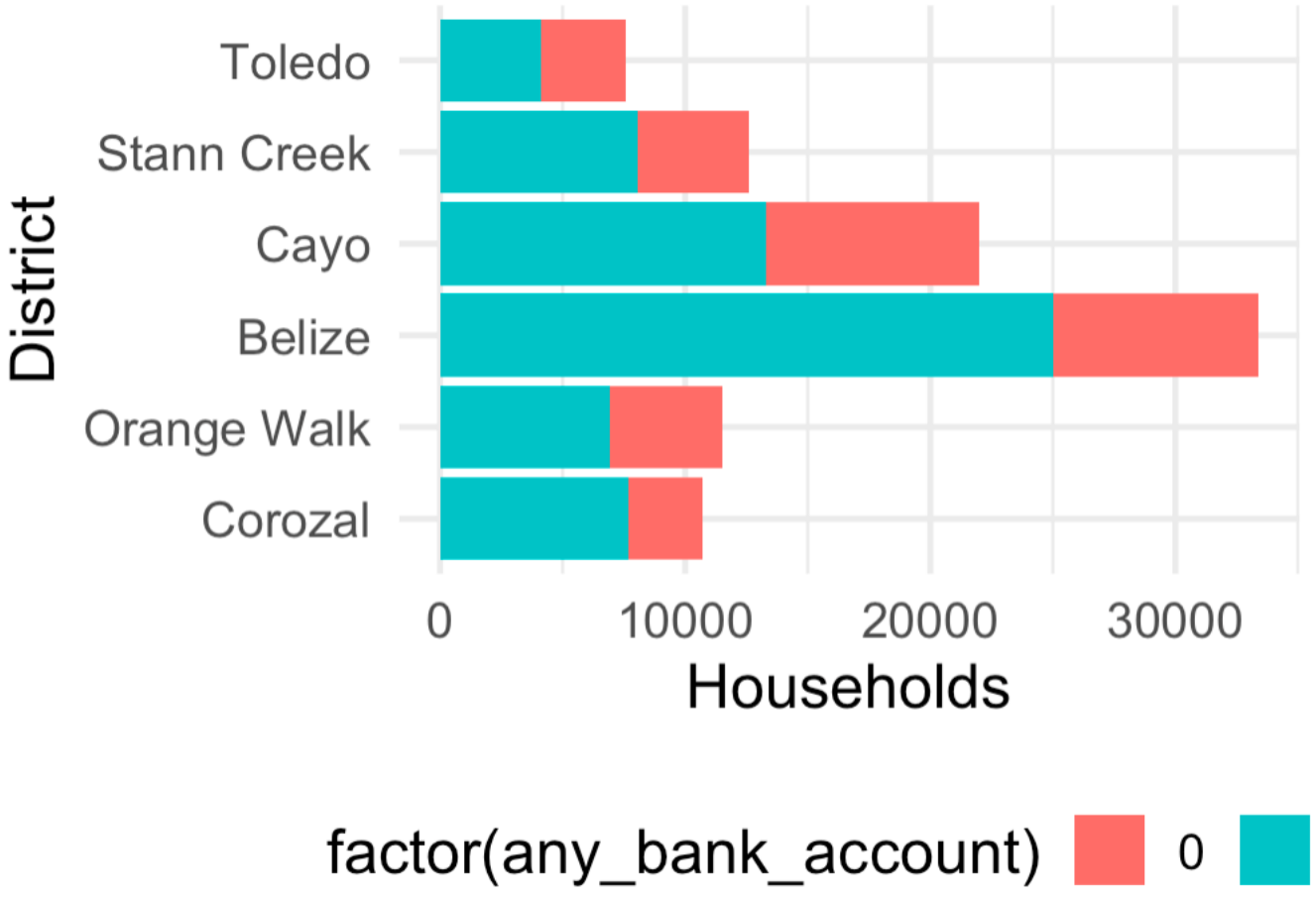
l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)
x, y, alpha, fill

l + geom_tile(aes(fill = z))
x, y, alpha, color, fill, linetype, size, width

https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf

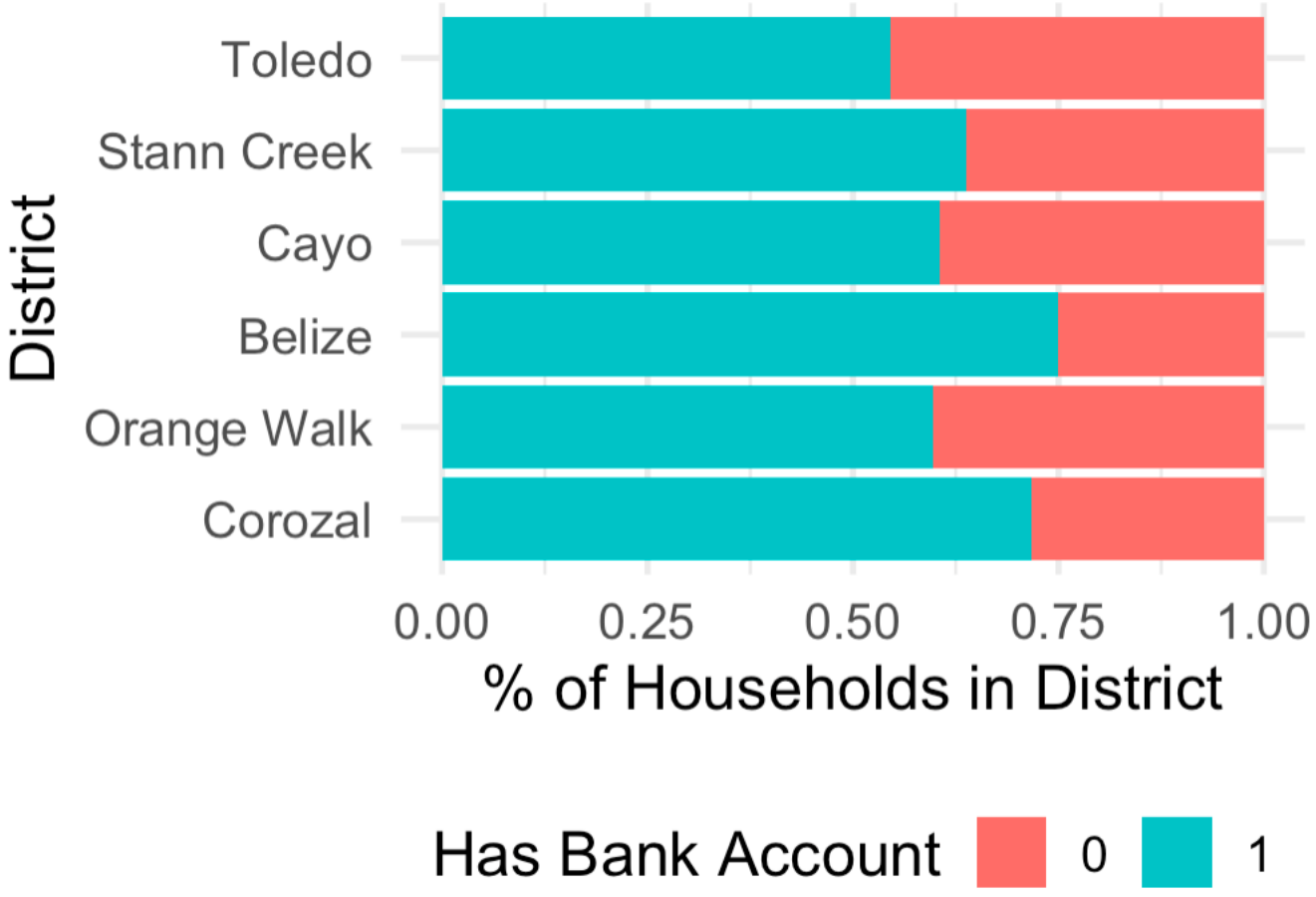
Frequency Bar Charts With Survey Data

```
ggplot(LFS_2019) +
  geom_bar(aes(x = DISTRICT_STR,
               fill = factor(any_bank_account),
               weight = Weight)) +
  coord_flip() +
  xlab("District") +
  ylab("Households") +
  theme_minimal(base_size = 14) +
  theme(legend.position="bottom")
```

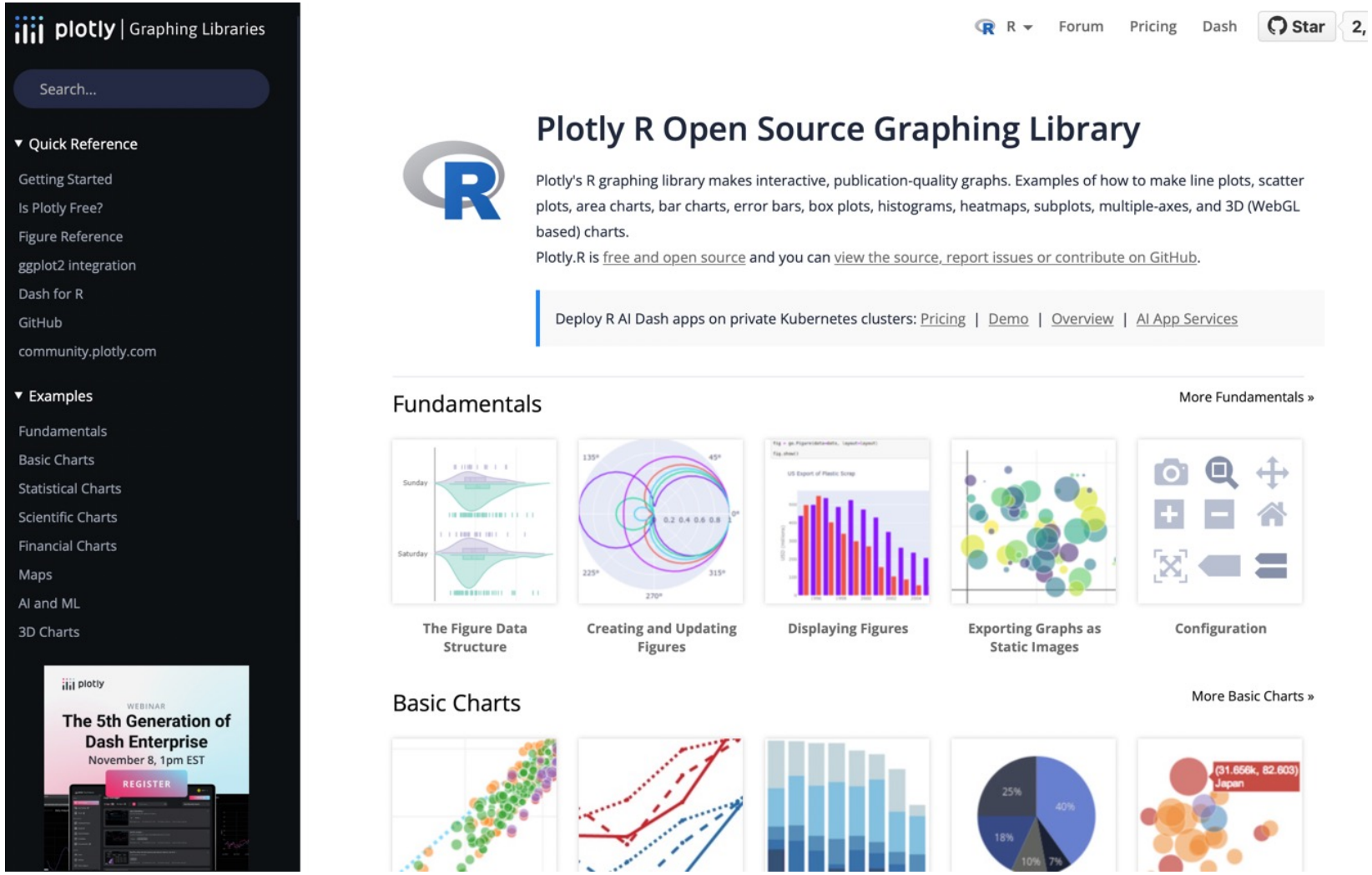


Bar Charts With Survey Data

```
ggplot(LFS_2019, aes(x = DISTRICT_STR,
                     fill = factor(any_bank_account),
                      weight = Weight)) +
  geom_bar(position = "fill") +
  coord_flip() +
  xlab("District") +
  ylab("% of Households in District") +
  theme_minimal(base_size = 14) +
  theme(legend.position="bottom") +
  labs(fill = "Has Bank Account")
```



Interactive Plots with Plotly



https://plotly.com/r/

```r
# ----------------------------------------------
# Lab Exercises
# ----------------------------------------------

# 1. Produce a bar chart of the fraction of households in each district
#    that has borrowed formally.

# 2. Save the plot using the the function ggsave()
```