Data visualization and a grammar of graphics

Data visualization

Q: What are some reasons we visualize data rather than just reporting statistics?

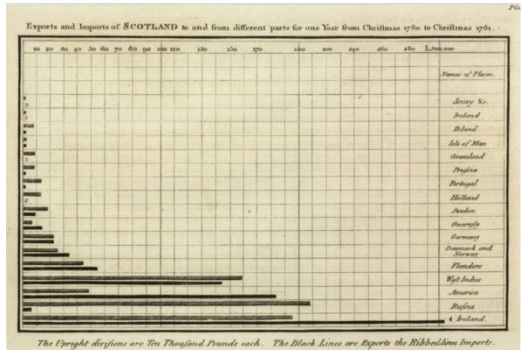
*Statistical projections which **speak to the senses without fatiguing the mind**, possess the advantage of fixing the attention on a great number of important facts.*

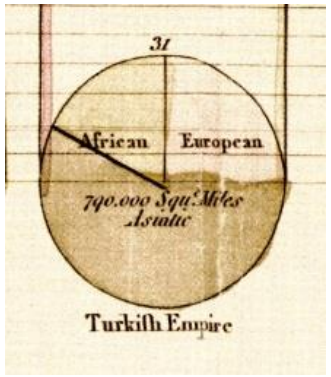
—Alexander von Humboldt, 1811

A very very brief history of data visualization

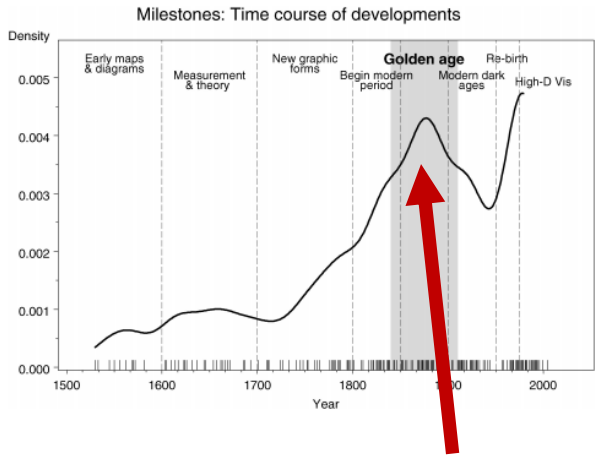
The age of modern statistical graphs began around the beginning of the 19th century

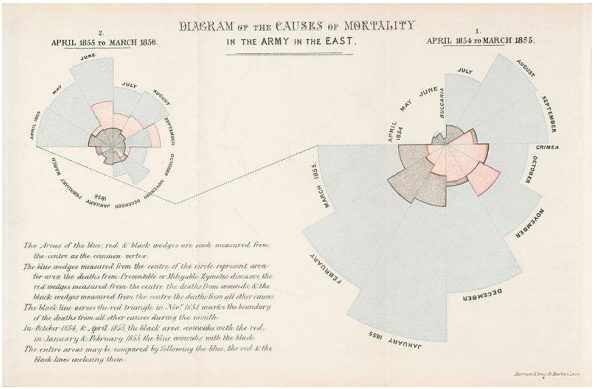
William Playfair (1759-1823)

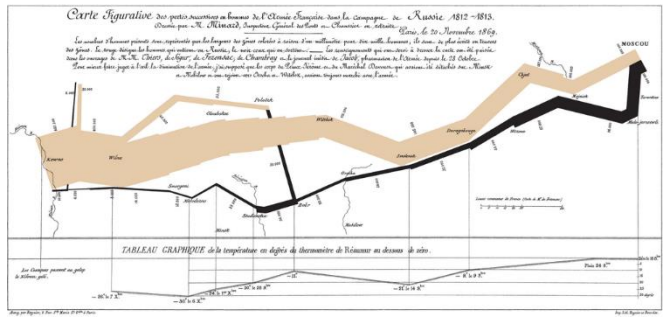


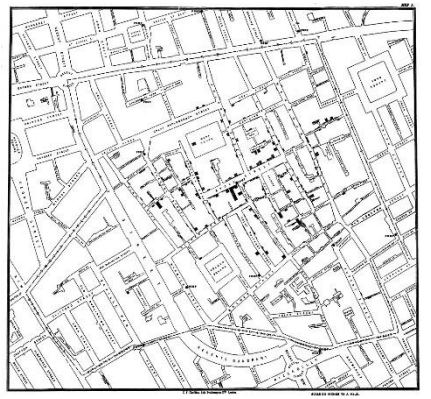


According to Friendly, statistical graphics researched its golden age between 1850-1900



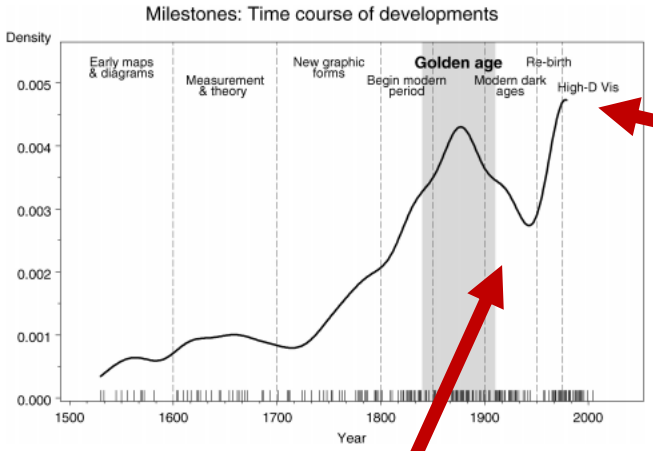


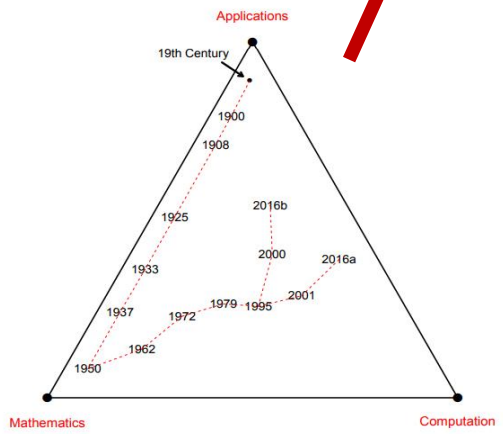


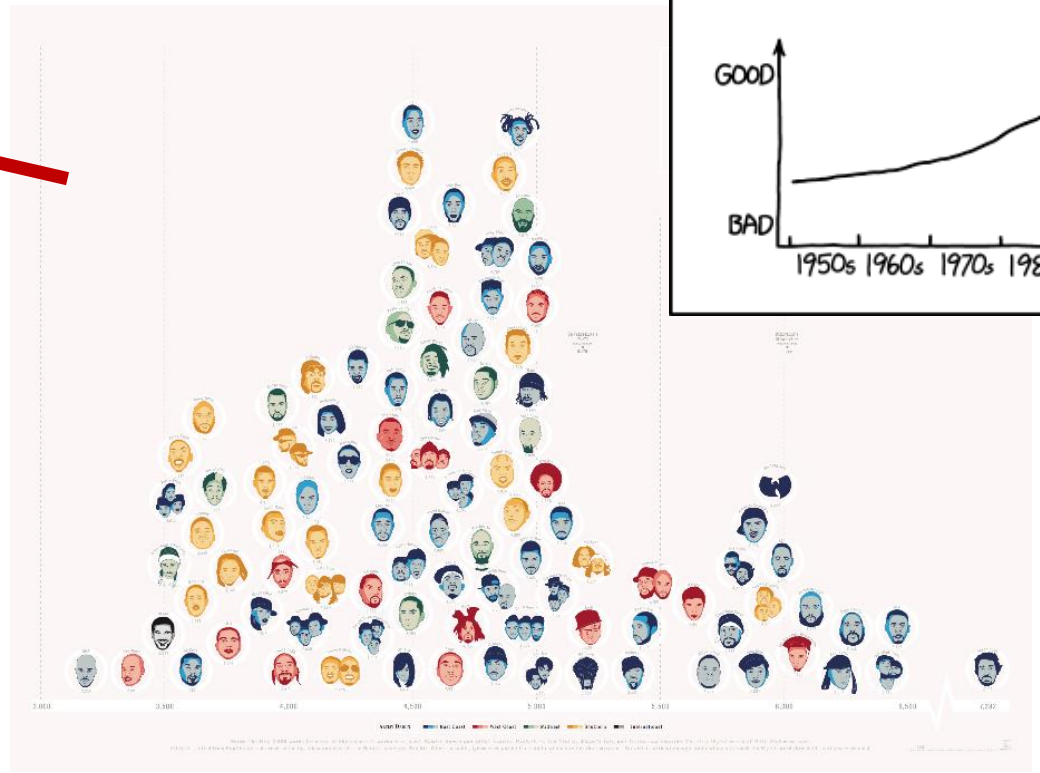


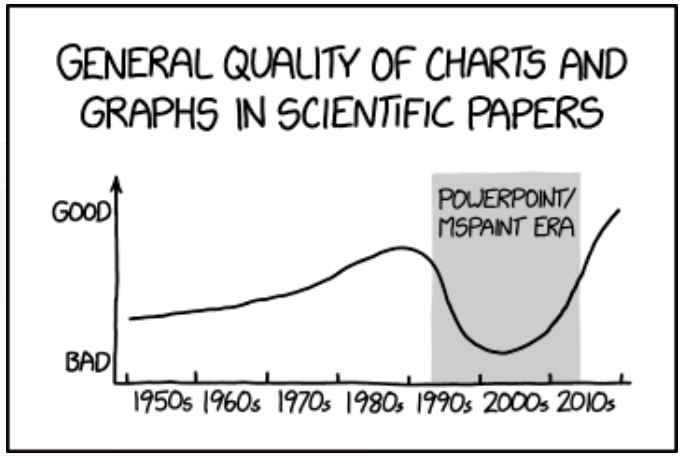
A very very brief history of data visualization

"Graphical dark ages" around 1950









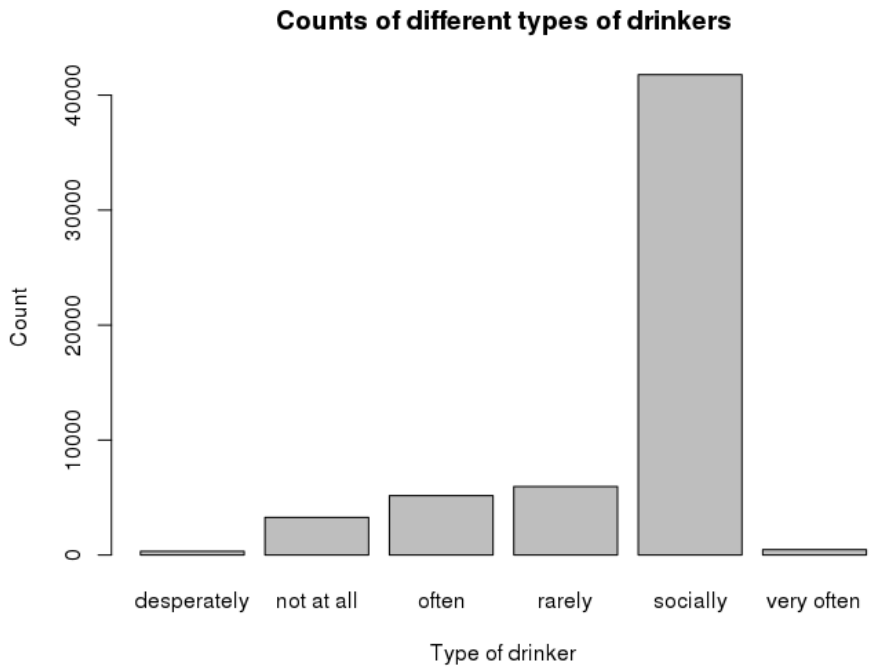
Currently undergoing a "Graphical re-birth"

Computer Age Statistical Inference, Efron and Hastie

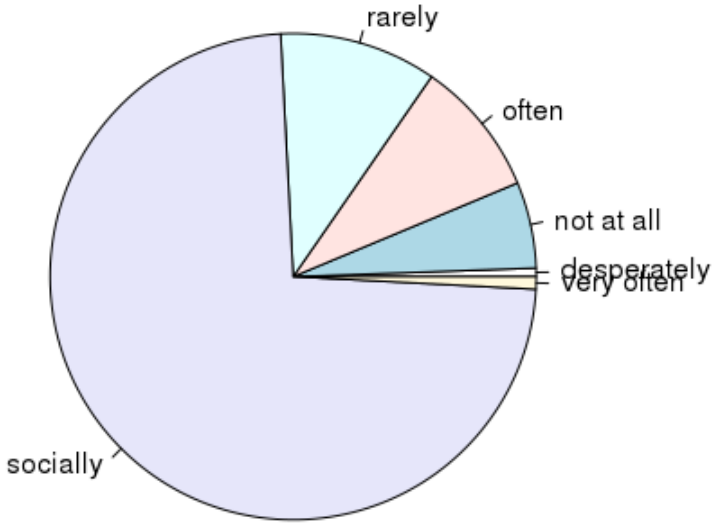
A grammar of graphics and ggplot

How have we plotted a single categorical variable?

Bar plot

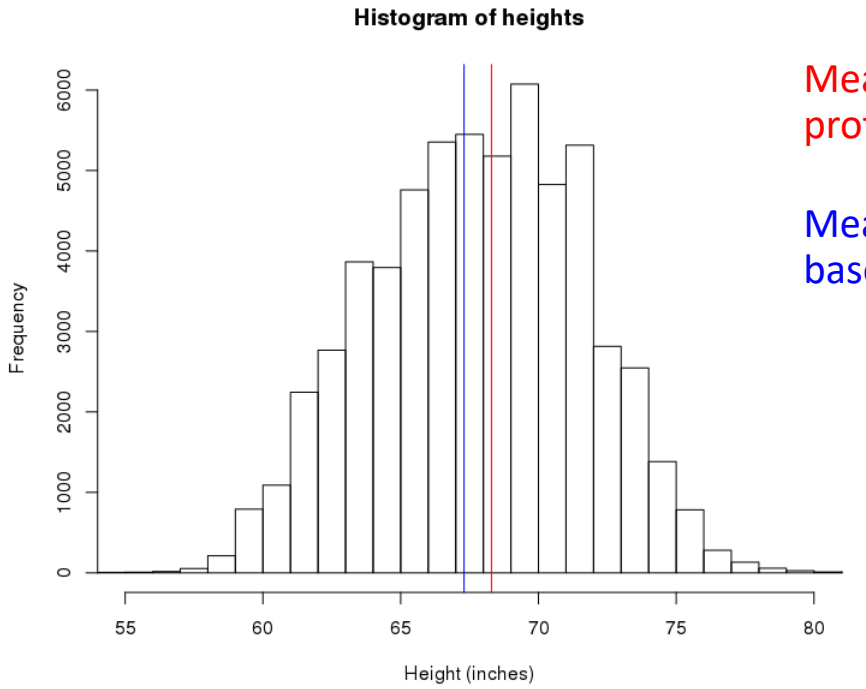


Pie chart



How have we plotted a single quantitative variable?

Histograms

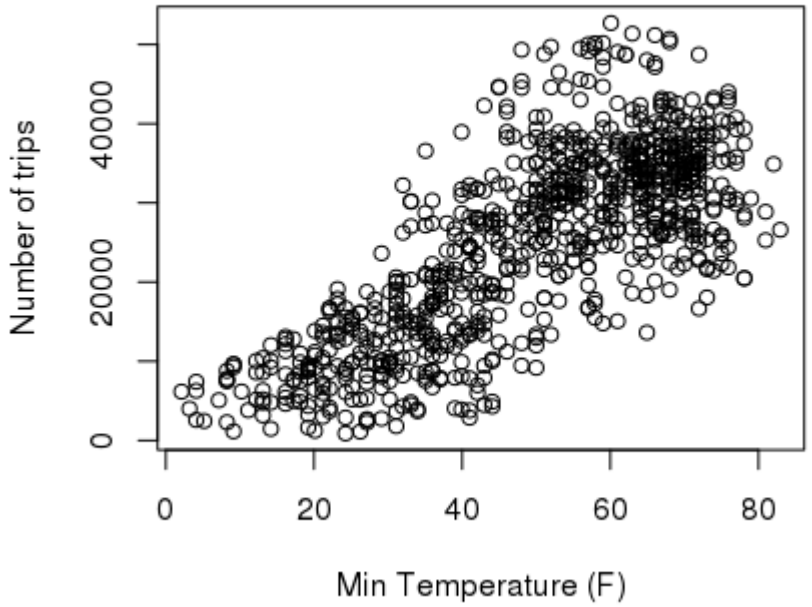


Mean height in the profiles data

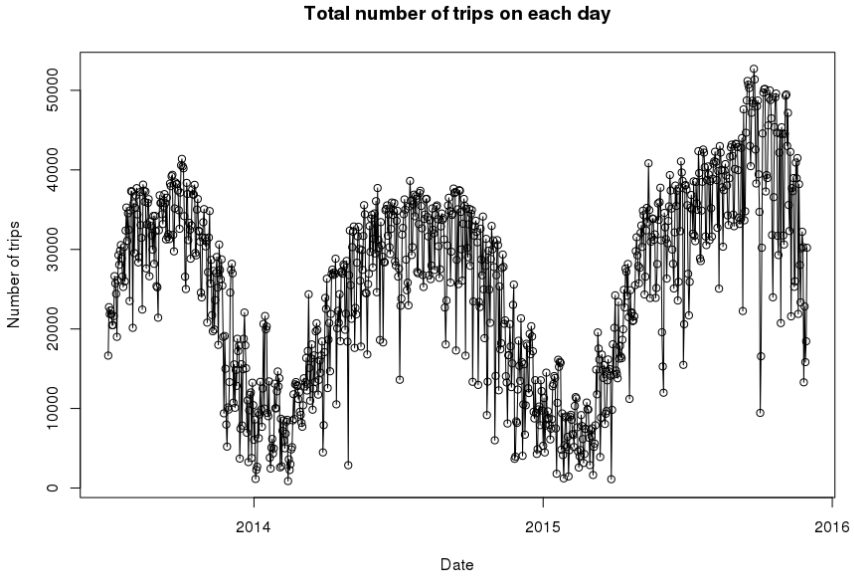
Mean height expected based on US population

How have we plotted a two quantitative variables?

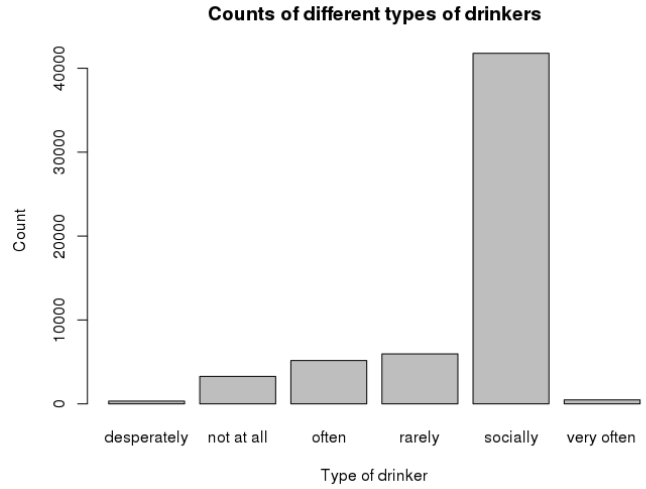
Scatter plots

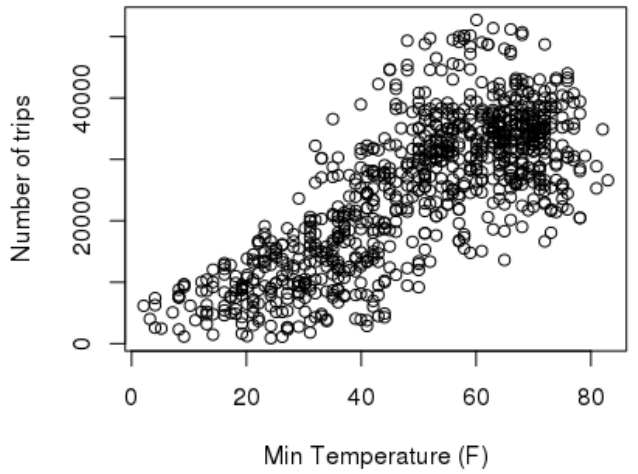


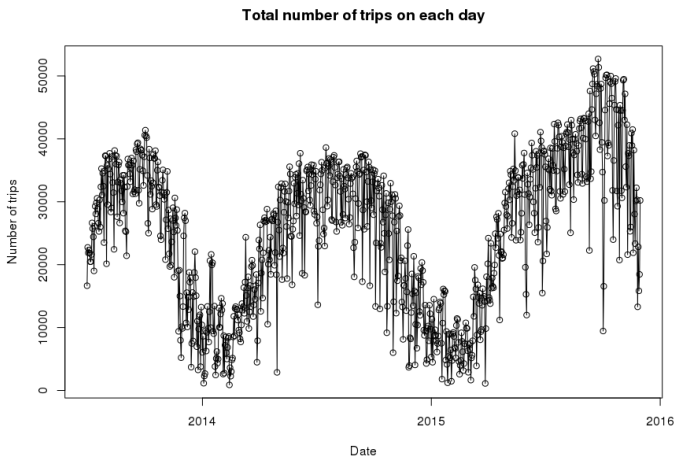
Line chart

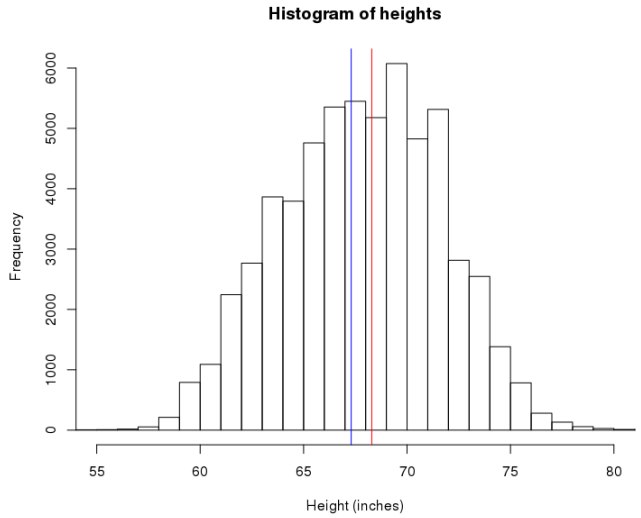


What are some similarities between these graphs?







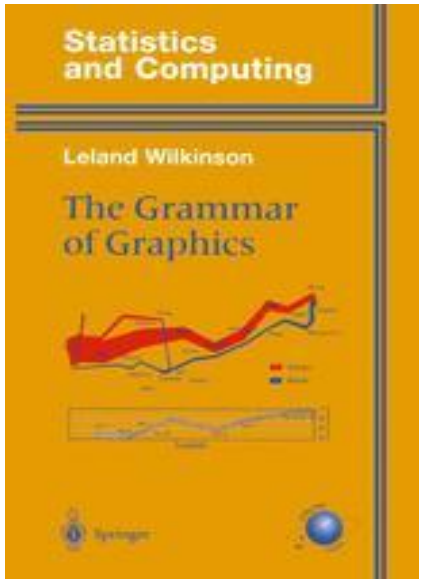


The grammar of graphics

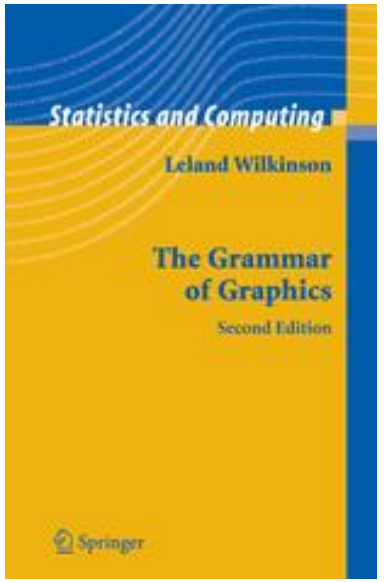
Leland Wilkinson noticed similarities between many graphs and tried to generate a ‘grammar’ that could be used to express a graph

- i.e., a list elements that can be combined together to create a graph

First edition



Second edition



Graphs are composed of…

A Frame: Coordinate system on which data is placed

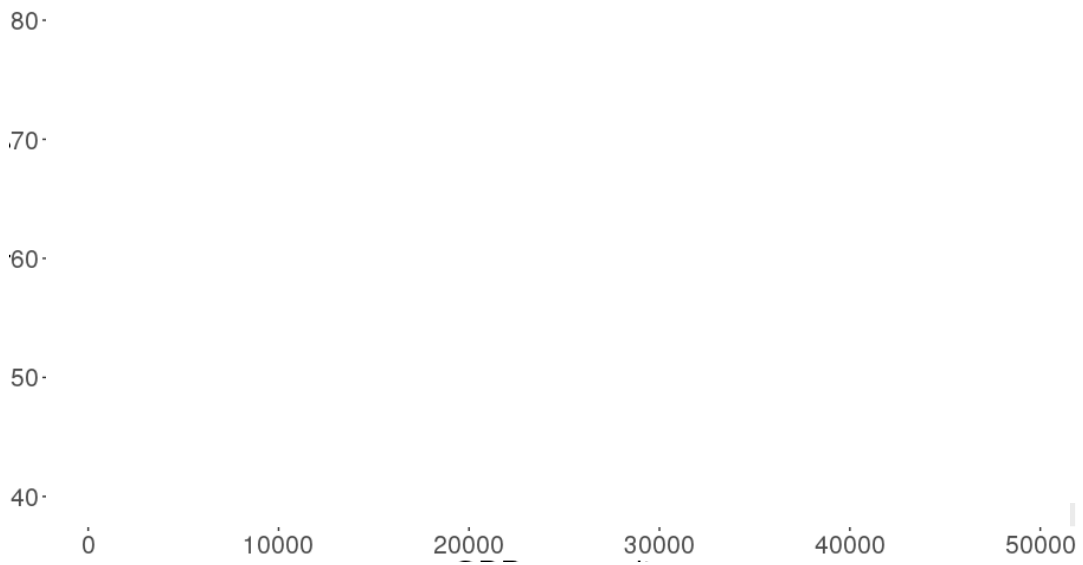
- E.g., Cartesian coordinate system, polar coordinates, etc.

Glyphs: basic graphic unit representing cases or statistics

- Contains visual properties (aesthetics) such as: shape, color, size, etc.
- Need to specify how properties of the data are **mapped** onto these aesthetics

Scales and guides: shows how to interpret axes and other properties of the glyphs

- i.e., gives information about how the data values were mapped into glyph properties



Plots can also contain…

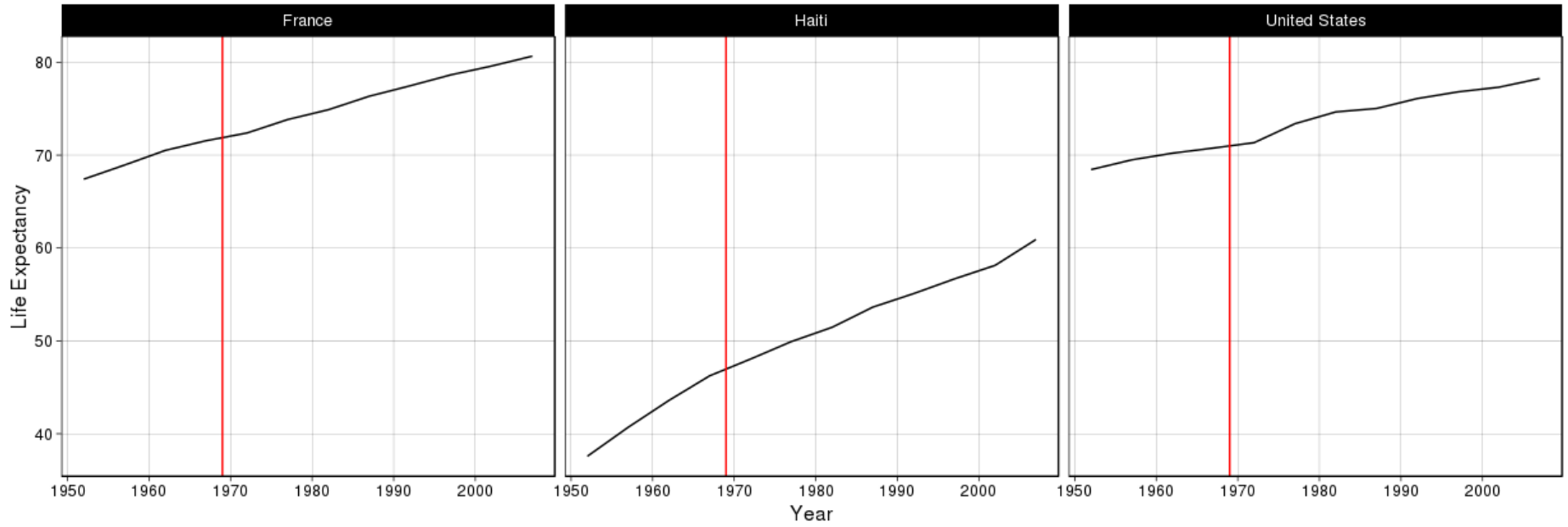
Facets: allows for multiple side-by-side graphs based on a categorical variable

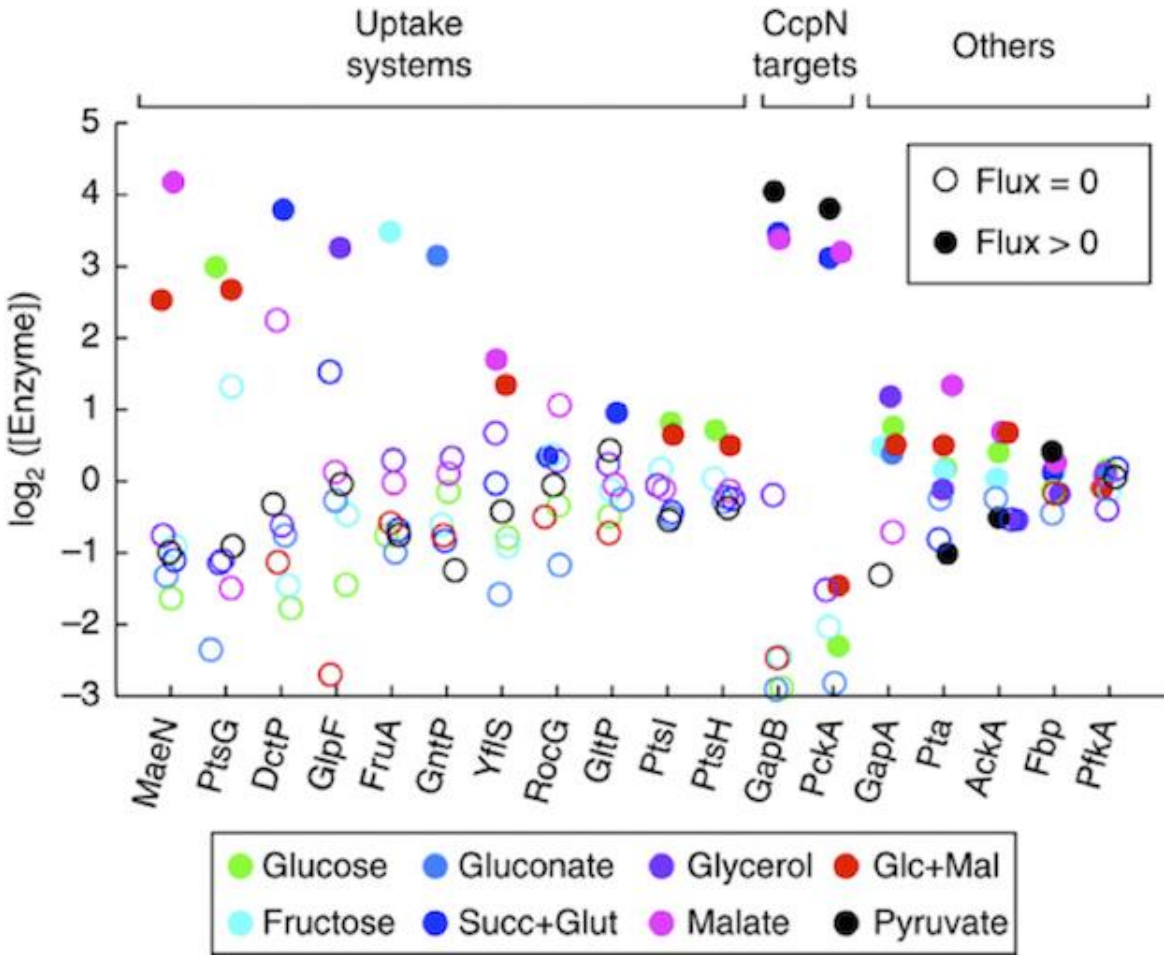
- Makes it easier to compare different conditions

Layers: allows for more than one types of data to be mapped onto the same figure

Theme: contains finer points of display

- E.g., font size, background color, etc.

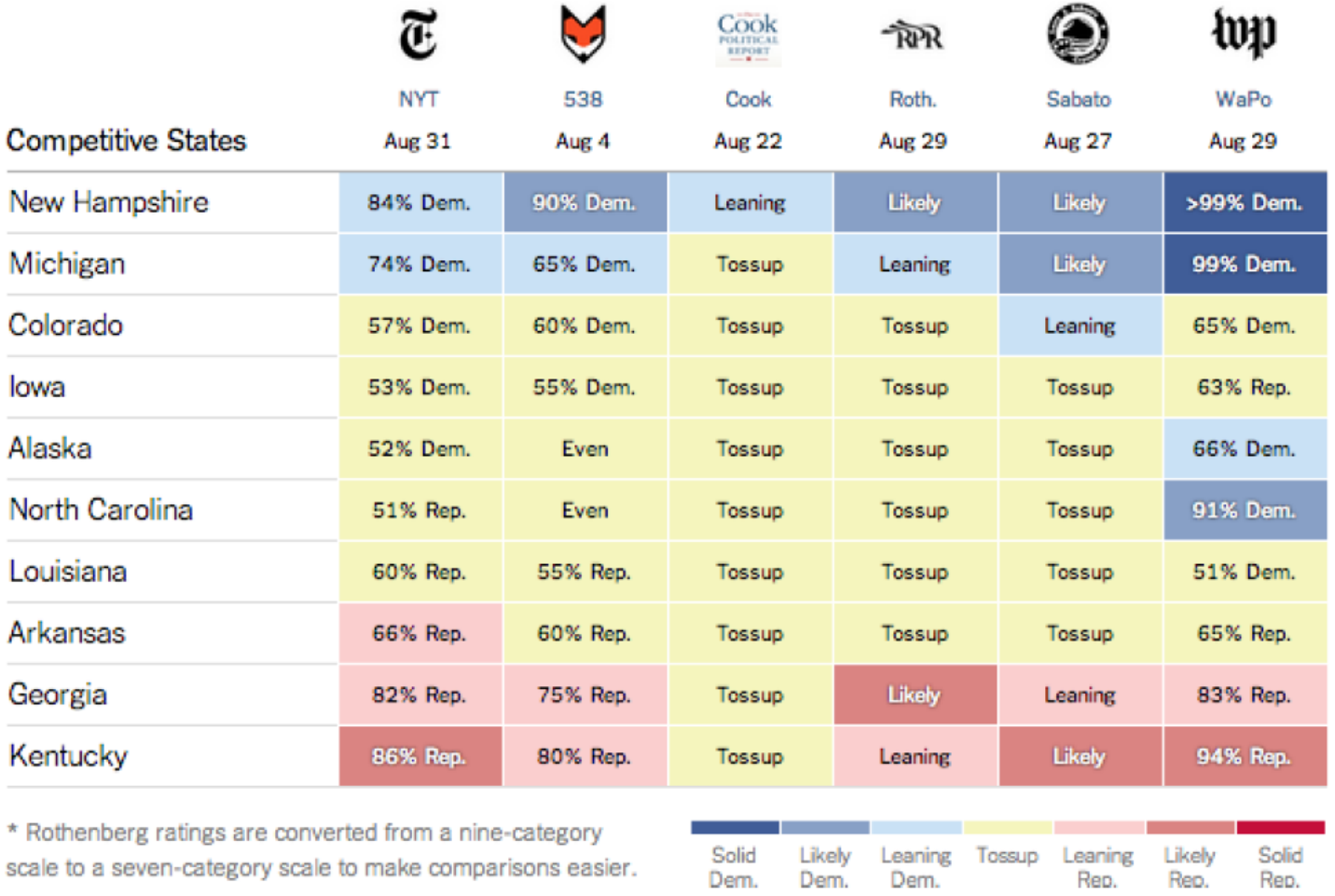




<u>The variables are:</u>

1. Log enzyme concentration
   - -3 to 5
2. Gene
   - MaeN, PtsG, …
3. Target
   - CcpN, Uptake,…
4. Flux
   - Zero or positive
5. Molecule:
   - Glocose, Fructose, …

What are the mappings between each variable and visual attribute?



Competitive States	NYT Aug 31	538 Aug 4	Cook Aug 22	Roth. Aug 29	Sabato Aug 27	WaPo Aug 29
New Hampshire	84% Dem.	90% Dem.	Leaning	Likely	Likely	>99% Dem.
Michigan	74% Dem.	65% Dem.	Tossup	Leaning	Likely	99% Dem.
Colorado	57% Dem.	60% Dem.	Tossup	Tossup	Leaning	65% Dem.
Iowa	53% Dem.	55% Dem.	Tossup	Tossup	Tossup	63% Rep.
Alaska	52% Dem.	Even	Tossup	Tossup	Tossup	66% Dem.
North Carolina	51% Rep.	Even	Tossup	Tossup	Tossup	91% Dem.
Louisiana	60% Rep.	55% Rep.	Tossup	Tossup	Tossup	51% Dem.
Arkansas	66% Rep.	60% Rep.	Tossup	Tossup	Tossup	65% Rep.
Georgia	82% Rep.	75% Rep.	Tossup	Likely	Leaning	83% Rep.
Kentucky	86% Rep.	80% Rep.	Tossup	Leaning	Likely	94% Rep.

* Rothenberg ratings are converted from a nine-category scale to a seven-category scale to make comparisons easier.

Solid Dem. | Likely Dem. | Leaning Dem. | Tossup | Leaning Rep. | Likely Rep. | Solid Rep.

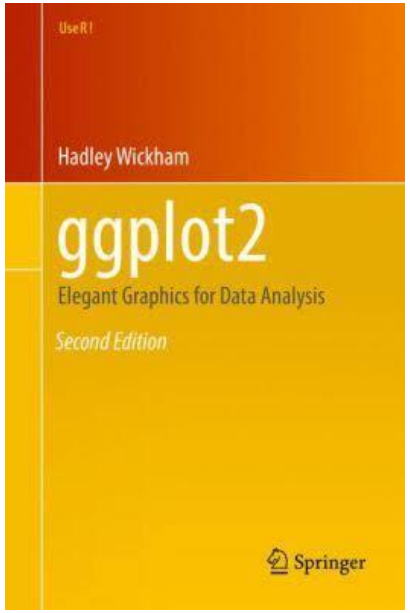
1. What variables define the frame?
2. What is the glyphs and the mapping from data to glyph?
3. What sets the order for the vertical variable?

ggplot

ggplot2 is an R package that implements the grammar of graphics

- It builds up graphics by starting with a frame, adding glyphs, etc.

```
# load the ggplot2 library
> library('ggplot2')
```



Get the book on GitHub

Example data: mtcars

ROAD TEST By Jim Brokaw

THE LUXURY CARS

**Imperial Palace
Fortress Fleetwood
Castle Continental**

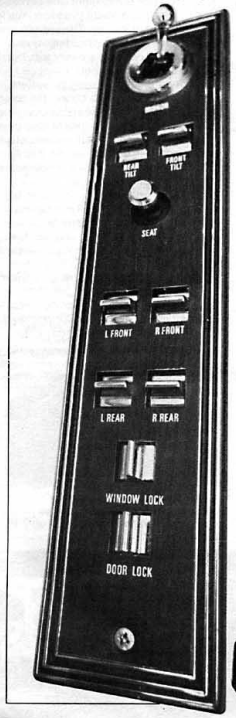
Crippled by the fuel flap and sniggered at lewdly by those smug Mercedes owners, today it seems that these great mastadons are dismissed as symbols of an ancient aristocracy whose strata was marked by expanse of wheelbase, and the heavenly quantity of gross cubic mass able to be shouldered by four beleagured tires. As the sole surviving heirs of princely Packards, dynosaurean Duesenbergs and the Brobdingnagian Bugattis, these marvels of grand proportion should be headed for the Smithsonian Institute by way of the mucky La Brea tar pits.

But are they?

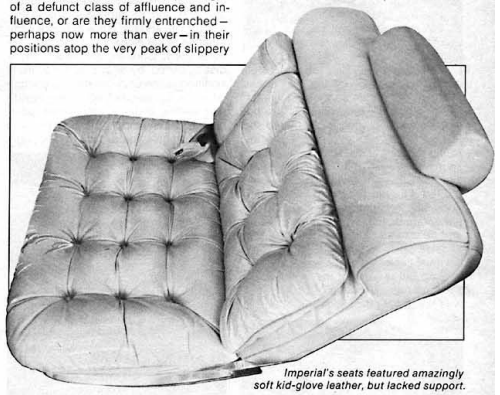
Are these behemoths simply vestiges of a defunct class of affluence and influence, or are they firmly entrenched—perhaps now more than ever—in their positions atop the very peak of slippery Mount Status, whose crags we scale daily, whether we wish to acknowledge that fact or not?

Ah, but welcome to the current state of American affairs: beef prices manipulated by withholding the animals from market; endless rounds of strikes by unions whose members are engaged in serving the public; gasoline supplies that rise and fall in mysterious coincidence with rising prices; dry rot, manipulation and partisan hatcheteering in government. They leave us little to believe in, and less to trust.

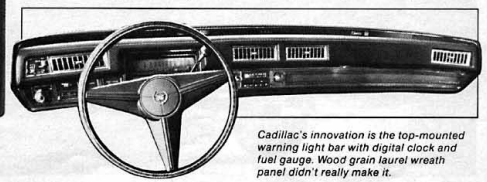
The very qualities that appear to condemn these sail-less luxury liners will very likely ensure their perpetuation. In days past, the block-long Cadillac and shiny Lincoln with paint jobs three feet deep flaunted a socio-economic position we could never hope to achieve. They did, however, constantly remind us that such positions, such wealth, such power, did, in fact, exist. The gliding specter of the shiny Imperial eagle stirred within a few heretical souls the bold idea that if such positions of power and wealth existed, there must be some means of attainment. More than a few of the haughty, distant drivers of the velvet tanks clawed their way up from the very pavement they now whisper over to



Lincoln's placement of seat controls on arm rest panel is less desireable than lower side-of seat location of Cad and Imperial.

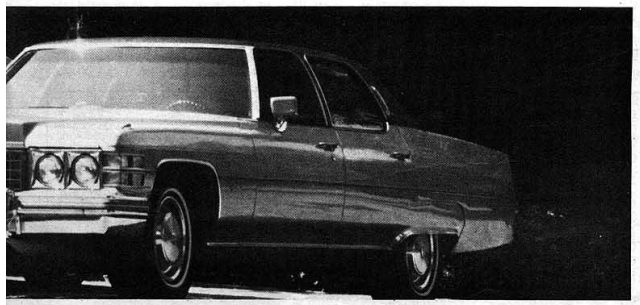


Imperial's seats featured amazingly soft kid-glove leather, but lacked support.



Cadillac's innovation is the top-mounted warning light bar with digital clock and fuel gauge. Wood grain laurel wreath panel didn't really make it.

PERFORMANCE	CADILLAC	LINCOLN	IMPERIAL
Acceleration			
0-30 mph	4.30	3.97	4.2
0-50 mph	8.49	8.00	9.15
0-60 mph	12.00	9.50	12.1
Standing Start 1/4-mile			
Mph	77.05	77.65	80.28
Elapsed time	17.98	17.82	17.42
Passing speeds			
40-60 mph	6.58	5.9	7.1
50-70 mph	7.00	6.8	6.8
Stopping distance			
From 30 mph	32'1"	31'4"	27'5"
From 60 mph	182'7"	153'10"	129'3"
Gas mileage range	10.43	10.42	14.7
Width – in.	79.8	80.0	79.7
Front Track – in.	63.5	64.3	64
Rear Track – in.	63.3	64.3	63.7
Wheelbase – in	133.0	127.0	124.0
Overall length – in.	233.7	232.6	231.1
Height – in.	55.6	55.4	54.7
Curb Weight – lbs.	5,250	5,425	5,345
Fuel Capacity – gals.	27	22.5	25
Oil Capacity – qts.	4 (1)	4 (1)	4 (1)
Storage Capacity – cu. ft.	19.27	20.9	20+
Base Price	$9,312	$7,637	$7,062
Price as tested	$11,435	$9,452	$8,737
Engine:	OHV V-8	OHV V-8	OHV V-8
Bore & Stroke – ins.	4.3x4.06	4.36x3.85	4.32x3.75
Displacement – cu. in.	472	460	440
HP @ RPM	205 @ 3600	215 @ 4000	230 @ 4000
Torque: lbs.-ft. @ rpm	365 @ 2000	350 @ 2600	350 @ 3200
Compression Ratio	8.25:1	NA	8.2:1
Carburetion	4V	4V	4V
Transmission	Auto. Turbo Hydra-Matic	Auto. Select Shift	Auto. Torqueflite
Final Drive Ratio	2.93	3.00	3.23 (?)
Steering Type	Recirculating Ball & Nut Power	Recirculating Ball & Nut With Integral Power Unit	Recirculating Ball Power
Steering Ratio	17.8-9.0	21.6 To 1	18.9:1
Turning Diameter (curb-to-curb-ft.)	(Wall To Wall) 24.54'	46.7'	44.69'
Wheel Turns (lock-to-lock)	2.83	3.99	3.5
Tire Size	LR78X15 Steel Belted Radials	LR78X15 Steel Belted Radials	LR78X15 Steel Belted Radial Ply
Brakes	Power Disc/Drum	Power Disc/Drum	Power Disc/Disc
Front Suspension	Coils/Shocks Front Diagonal Tie Struts Stabilizer	Coils/Shocks Axial Strut Stabilizer	Torsion Bar Shocks Stabilizer
Rear Suspension	4 Link, Coils/ Shocks	Three Link, Rubber Cushioned Pivots Coils/Shocks	Leaf Springs Shocks
Body/Frame Construction	Perimeter Frame	Body On Perimeter Frame	Unitized Construction



mtcars data frame

How can you determine what variables are in a data frame?

```
> View(mtcars)      # only works in Rstudio, not in Markdown
> glimpse(mtcars)
> ? mtcars      # this data frame as a code book
```

[, 1]	mpg	Miles/(US) gallon
[, 2]	cyl	Number of cylinders
[, 4]	hp	Gross horsepower
[, 6]	wt	Weight (1000 lbs)
[, 9]	am	Transmission (0 = automatic, 1 = manual)

Do cars that weigh more use more fuel?

Question: do cars that weigh more use more fuel?

What variables in the mtcars data frame are of interest?

- mpg
- wt

We can create a scatter plot using base graphics…

```
> plot(mtcars$wt, mtcars$mpg)
```

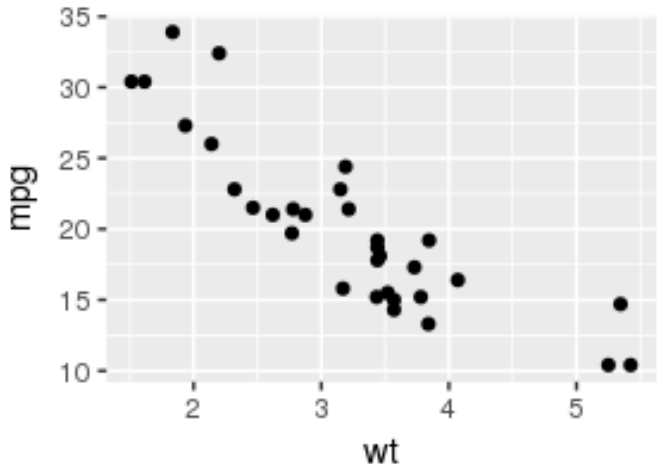
Creating a scatter plot in ggplot

Data frame to be used

Aesthetic mapping

```
> ggplot(data = mtcars, mapping = aes(x = wt, y = mpg)) +
    geom_point()
```

Adds a layer with glyphs



	wt	cyl	hp	mpg	disp
Mazda RX4	2.620	6	110	21.0	160.0
Mazda RX4 Wag	2.875	6	110	21.0	160.0
Datsun 710	2.320	4	93	22.8	108.0
Hornet 4 Drive	3.215	6	110	21.4	258.0
Hornet Sportabout	3.440	8	175	18.7	360.0

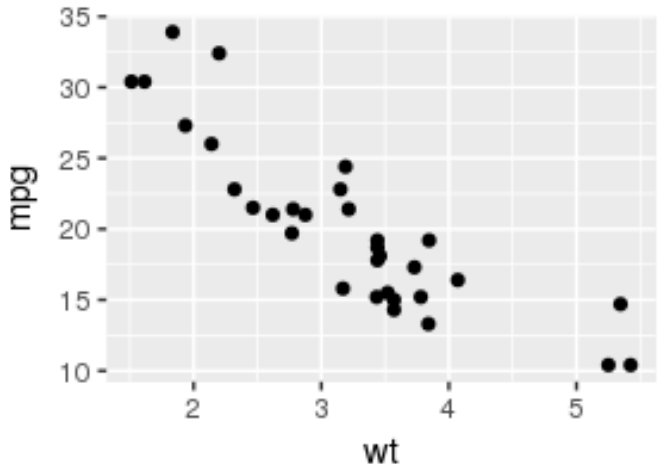
Creating a scatter plot in ggplot

Data frame to be used

Aesthetic mapping

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
    geom_point()
```

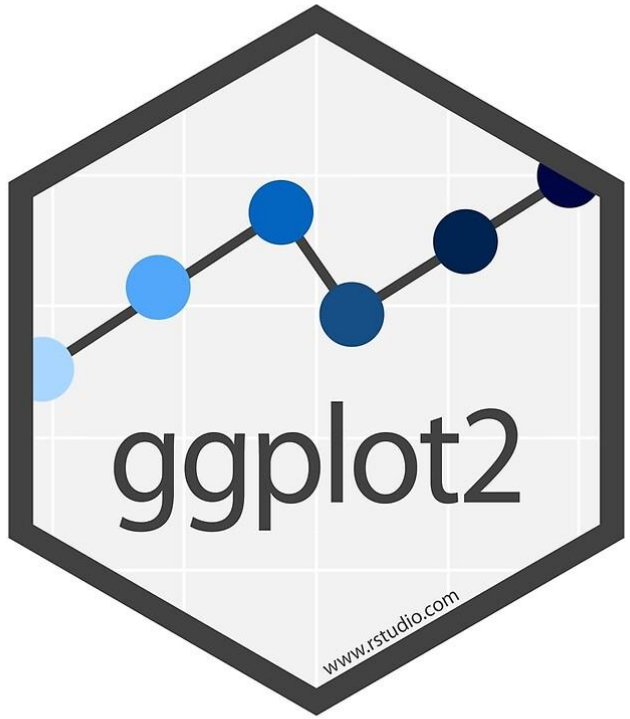
Adds a layer with glyphs



	wt	cyl	hp	mpg	disp
Mazda RX4	2.620	6	110	21.0	160.0
Mazda RX4 Wag	2.875	6	110	21.0	160.0
Datsun 710	2.320	4	93	22.8	108.0
Hornet 4 Drive	3.215	6	110	21.4	258.0
Hornet Sportabout	3.440	8	175	18.7	360.0

A lot more that ggplot can do!

- More aesthetic mapping
- Multiple glyphs/layers
- Axis labels
- Facets
- Visual themes
- Different coordinate systems
- Etc.



[The R Graph Gallery]

Let’s try the rest in R!

Adding labels to plots – version 1

We can add labels to the plots using the xlab("label1") and ylab("label2") functions

Add labels to your last plot

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point()  +
        xlab("Weight") +
        ylab("Miles per Gallon")
```

Adding labels to plots – version 2

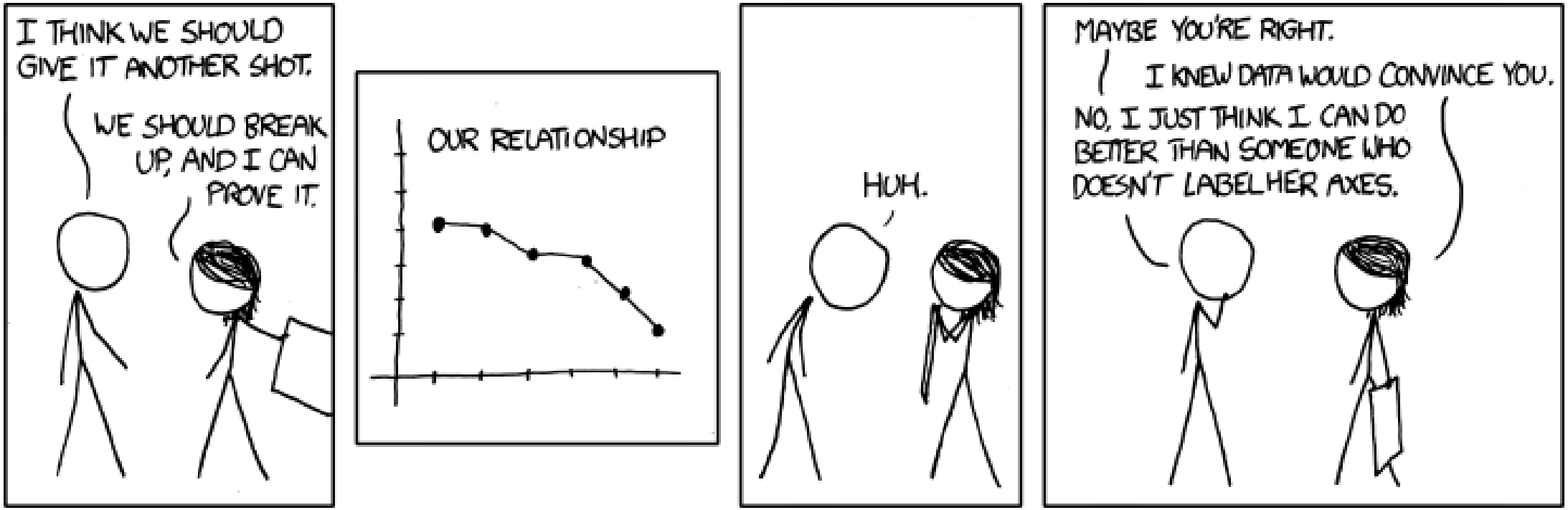
We can add labels to the plots using the lab() functions

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point()  +
        labs(x = "Weight",
             y = "Miles per Gallon",
             title = "MPG as a function of weight",
             subtitle = "ggplot is cool")
```

Adding text annotations

We can text annotations using the annotate(“text”, x = , y = , label =) function

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point()  +
        annotate("text", x = 4, y = 30, label = "AWESOME!")
```



If you don’t want an ex, label you axes!

More aesthetic mappings

Let's look at the relationship between weight, miles per gallon and transmission type on the same graph by plotting… (?)

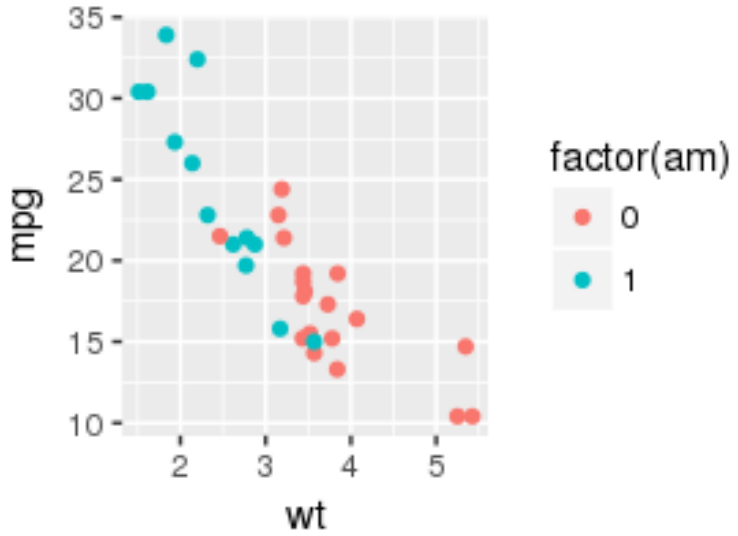
```
> ggplot(mtcars, aes(x = wt, y = mpg, col = am)) +
        geom_point()

# It is better if we make am a categorical variable
> ggplot(mtcars, aes(x = wt, y = mpg, col = factor(am))) +  geom_point()
```

Notice the guides!!!

Try mapping am on to shape using:

1. *shape = am*
2. size using: *size = am*

Which is better to use color or shape or size?



Attributes vs. Aesthetics

Setting **aesthetics** map a variable to a glyph property

Setting **attributes** set a glyph property to a fixed value

```
# setting a aesthetic
> ggplot(mtcars) +
        geom_point(aes(x = wt, y = mpg, col = factor(am)))

# setting an attribute
> ggplot(mtcars) +
        geom_point(aes(x = wt, y = mpg), col = "red")
```

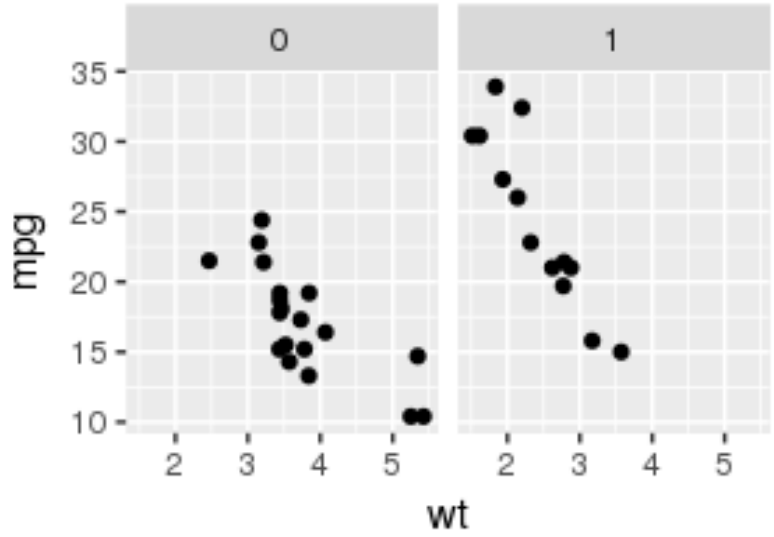
Outside the aesthetic mapping!

Facets

Beyond comparing variables based on aesthetics you can compare categorical variables by splitting a plot into subplots (called facets) using facet_wrap

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +   geom_point() +
        facet_wrap(~am)
```

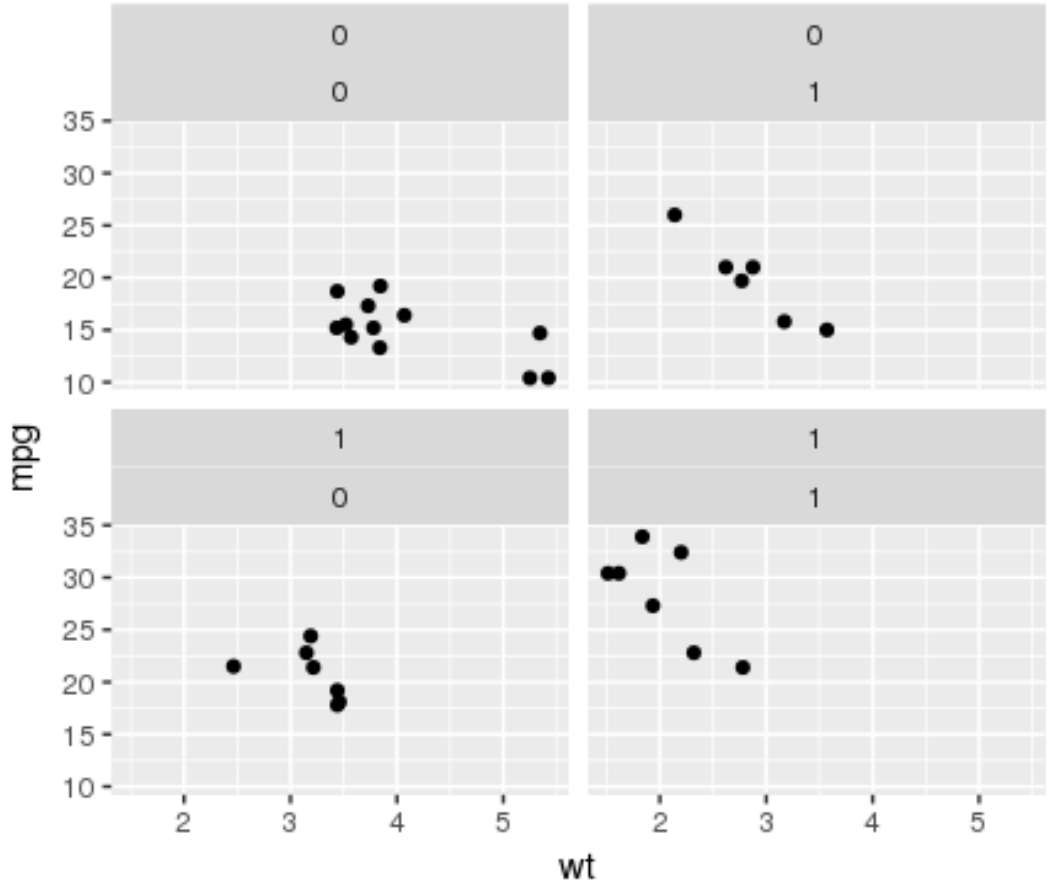
What do facets make it easy to see on this graph?



Facets along two dimensions

One can also do facets in two dimensions

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point() +
        facet_wrap(vs ~am)
```



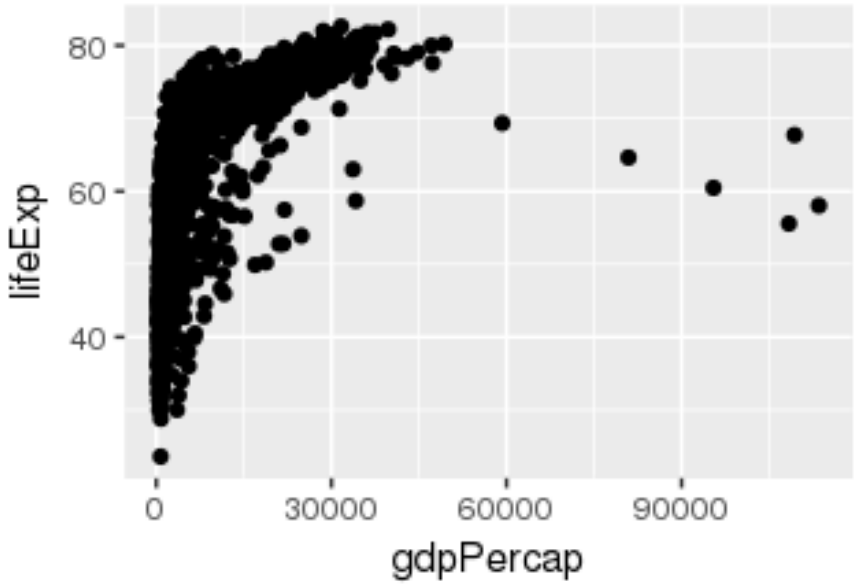
Overplotting

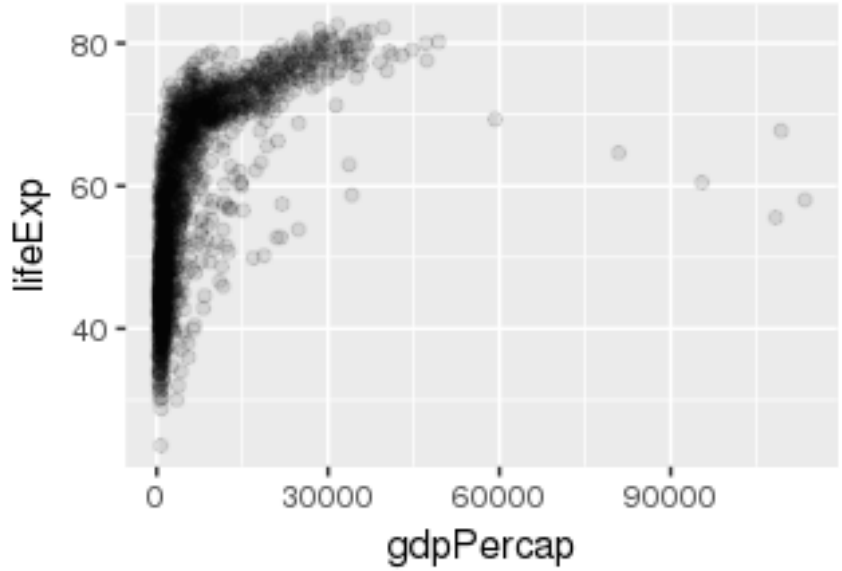
Sometimes points overlap making it hard to estimate the number of points at a particular range of values

We can control the transparency of points by changing their alpha values

```
# compare these two plots
> ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
        geom_point()

> ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
        geom_point(alpha = .1)
```

Overplotting





Scales

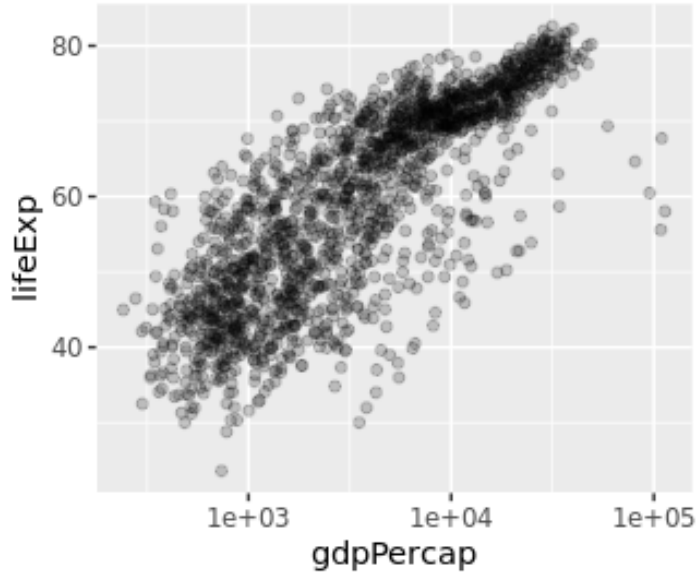
We can change the scale underlying each aesthetic visual feature

- We use functions that start with scale_ to do this

For example, we can change the x scale from linear to logarithmic using:

- scale_x_continuous(trans='log10')

```
> ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
                geom_point(alpha = .2) +
                scale_x_continuous(trans='log10')
```



Scales

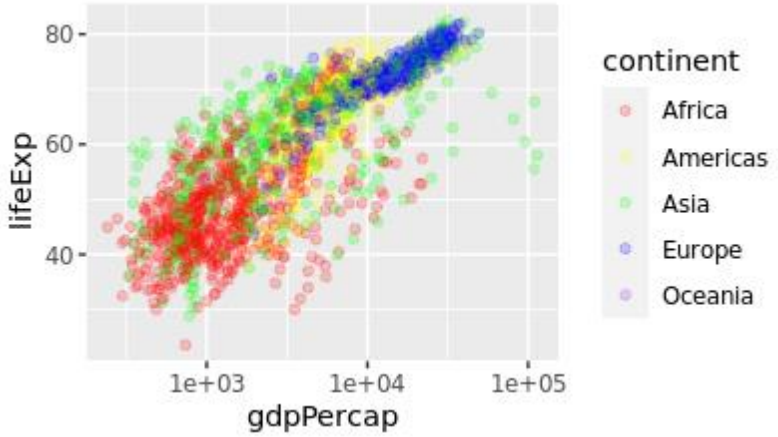
We can change the scale underlying each aesthetic visual feature

- We use functions that start with scale_ to do this

We can change the color scale using:

- scale_color_manual()

```
> ggplot(gapminder, aes(x = gdpPercap, y = lifeExp,
                        col =continent)) +
    geom_point(alpha = .2) +
    scale_color_manual(values = c("red", "yellow",
                                  "green", "blue", "purple"))
```



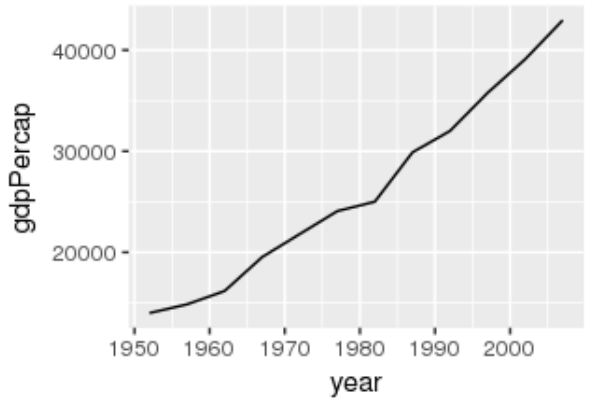
Geometries: line plot

So far we've only created scatter plots, but we can use different geoms to create other types of plots

Create a plot that shows the GDP in the United States as a function of the year using the geom *geom_line()*

- Hint: filter the gapminder data first…

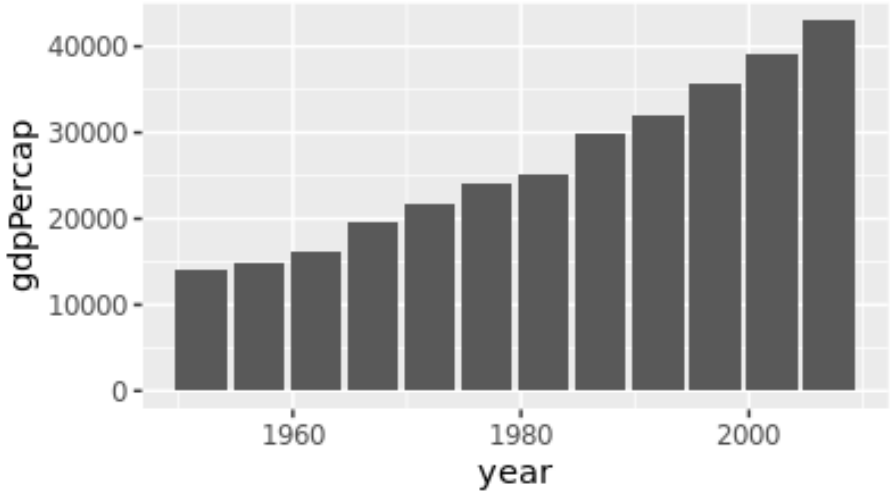
```
> gapminder %>% filter(country == 'United States') %>%
      ggplot(aes(x = year, y = gdpPercap)) +
      geom_line()
```



Geometries: columns

Create a plot that shows the GDP in the United States as a function of the year as columns geom *geom_col()*

```
> gapminder %>% filter(country == 'United States') %>%
      ggplot(aes(x = year, y = gdpPercap)) +
      geom_col()
```

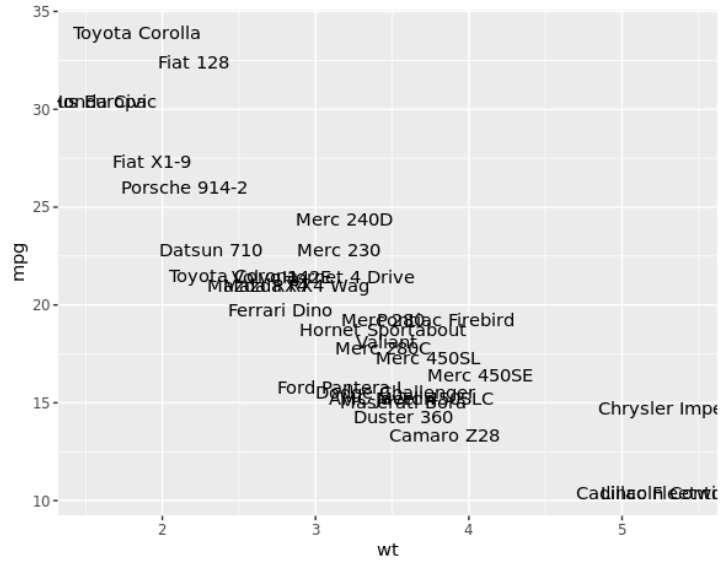


Geometries: text

Create can also use text as a geom using *geom_text(aes(label =))*

We will first add the row names as a column to our data frame using tibble::rownames_to_column()

```
> mtcars %>%
      tibble::rownames_to_column() %>%
      ggplot(aes(x = wt, y = mpg)) +
            geom_text(aes(label = rowname))
```



Geometries: histograms

We can also make histograms using the geom_histogram() function.

Plot a histogram of the weights of cars

```
> ggplot(mtcars, aes(x = wt)) +
        geom_histogram()
```

Note the histogram geom only has an x aesthetic, and does not have a y aesthetic value.

Geometries: boxplot

There are many other geom as well, including geom_boxplot()

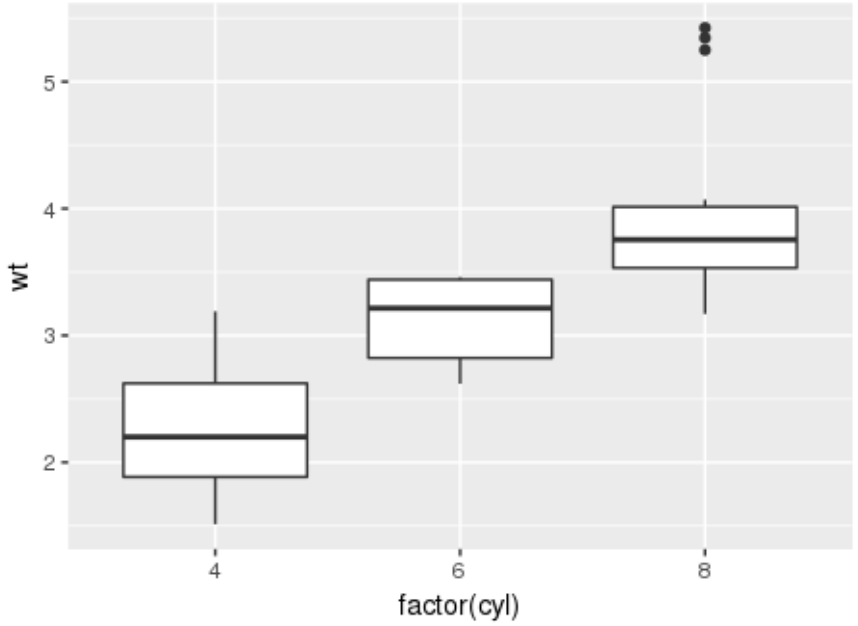
Plot a boxplot of the weights of cars

```
> ggplot(mtcars, aes(x = "", y = wt)) +
        geom_boxplot()
```

Side-by-side boxplots

Often it is useful to compare boxplots across different groups

```
> ggplot(mtcars, aes(x = factor(cyl), y = wt)) +
        geom_boxplot()
```



Violin and Joy plots

Violin and Joy plots are other ways to view distributions of data

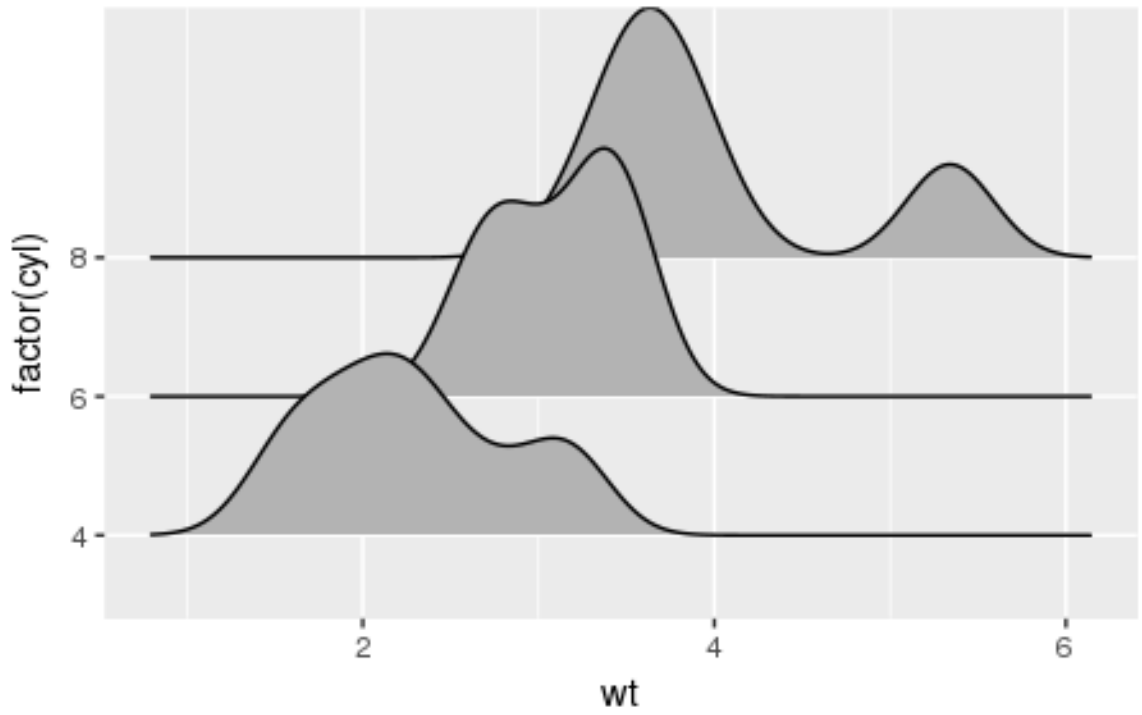
```
> ggplot(mtcars, aes(x = factor(cyl), y = wt)) +
        geom_violin()
```

```
> library("ggridges")
> ggplot(mtcars, aes(y = factor(cyl), x = wt)) +
        geom_density_ridges()
```

Note x and y are reversed

Violin and Joy plots

Any ideas why they are called joy plots?



Multiple layers

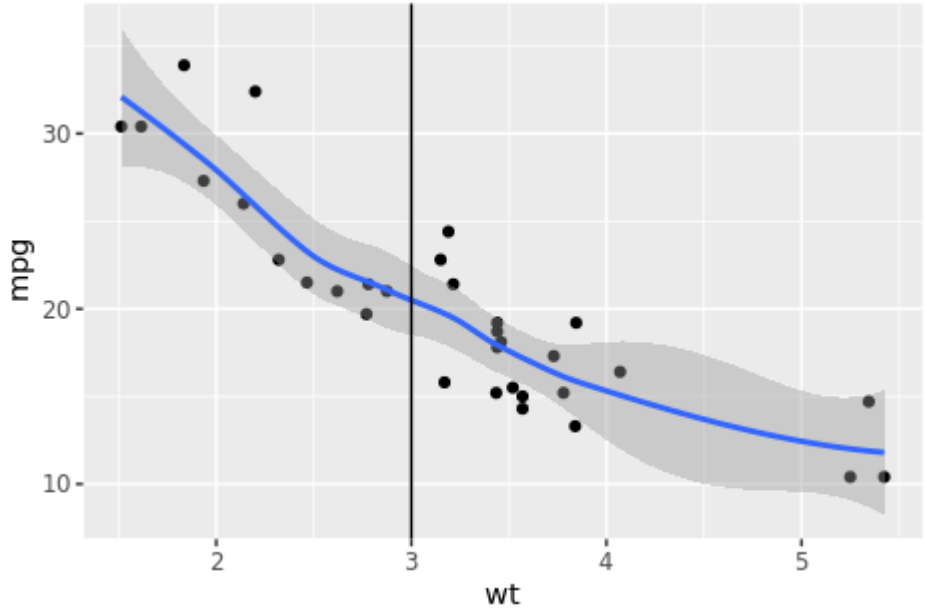
We can also have multiple geom layers on a single graph by using the + symbol

- E.g ggplot(…) + geom_type1() + geom_type2()

Create a scatter plot of miles per gallon as a function of weight and then add:

- a smoothed line using geom_smooth()
- a vertical line using geom_vline()

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point() +
        geom_smooth() +
        geom_vline(xintercept = 3)
```



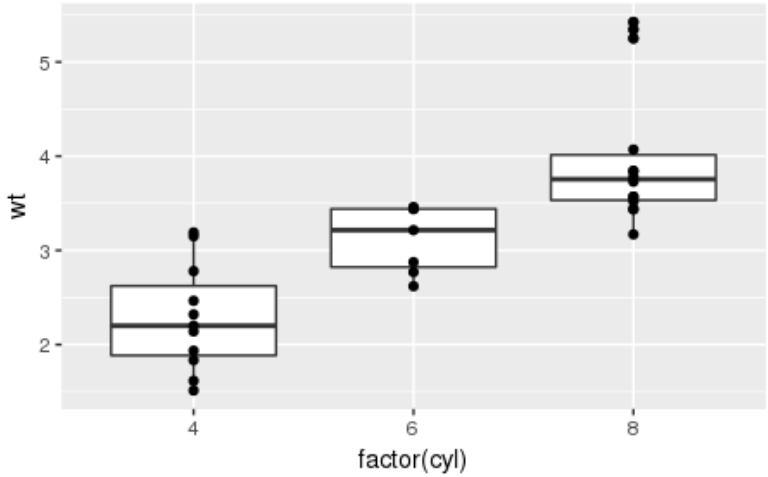
Multiple layers

We can also have multiple geom layers on a single graph by using the + symbol

- E.g ggplot(…) + geom_type1() + geom_type2()

Recreate a boxplot of weight (wt) grouped by the factor of cylinders (cyl), and then add points using geom_point()

```
> ggplot(mtcars, aes(x = factor(cyl), y = wt)) +
        geom_boxplot() +
        geom_point()
```



Themes

We can also use different types to change the appearance of our plot

Add theme_classic() to your plot

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point()  +
        xlab("Weigth") +
        ylab("Miles per Gallon") +
        theme_classic()
```

Also see the theme_fivethirtyeight() from the ggthemes package

Themes

We can also create a customized theme using theme()

```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point()  +
        theme_classic() +
        theme(
                axis.text.y = element_blank(),
                plot.background = element_rect(fill = "red")
              )
```

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** locations.

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

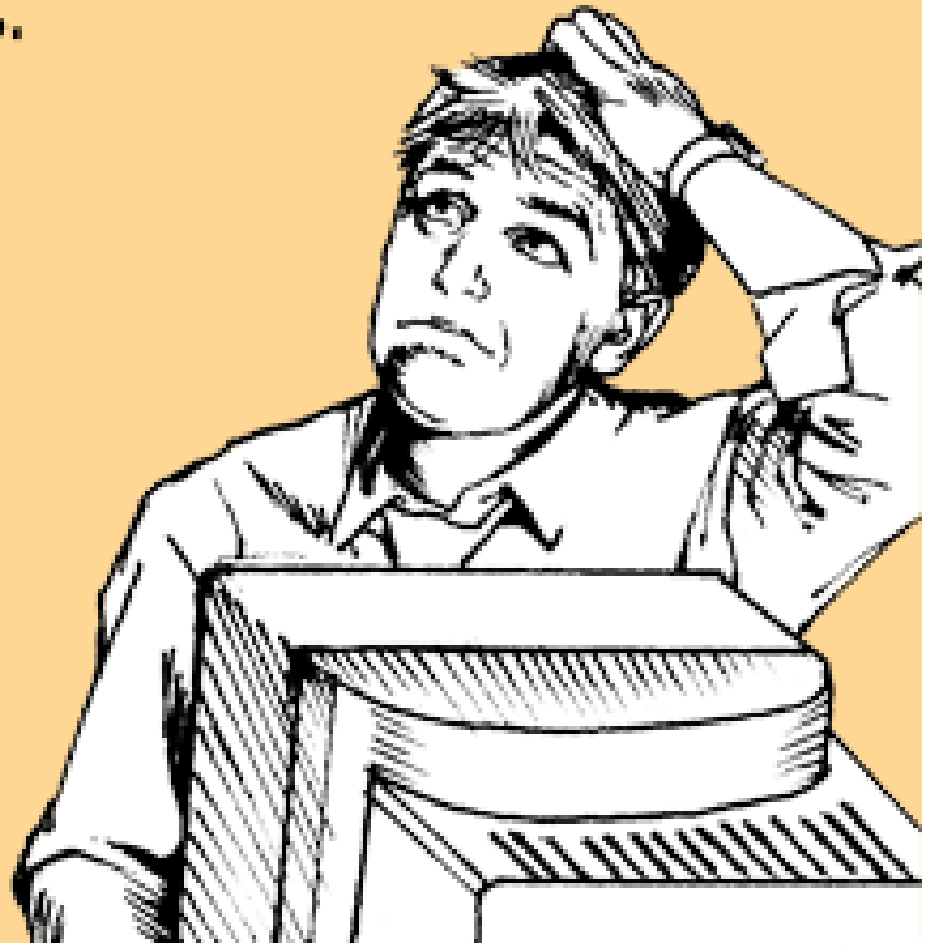
l + geom_contour_filled(aes(fill = z)**)**
x, y, alpha, color, fill, group, linetype, size, subgroup

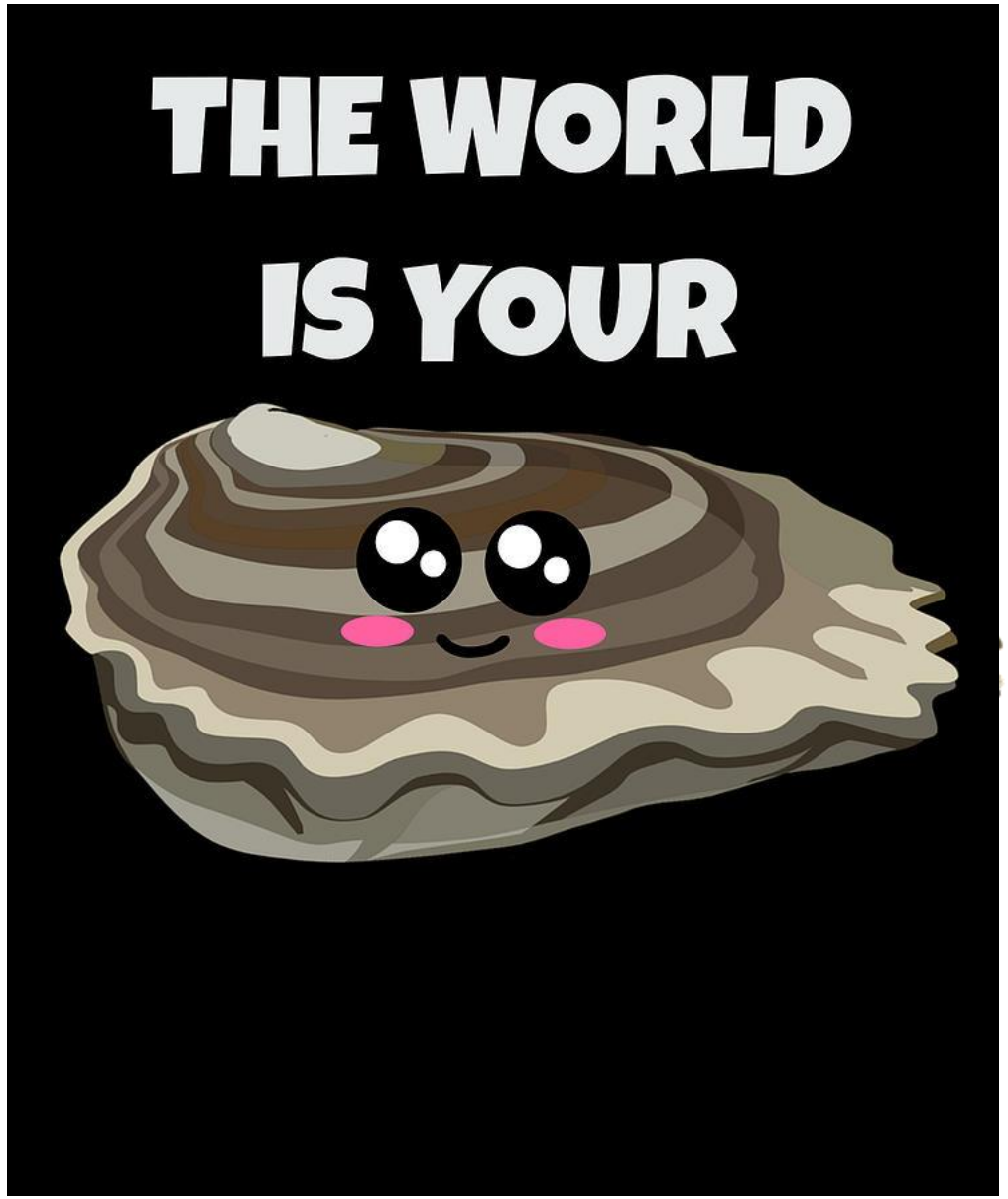
l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE**)**
x, y, alpha, fill

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

 · info@posit.co · posit.co · Learn more at ggplot2.tidyverse.org · ggplot2 3.3.5 · Updated: 2021-08

If life gives you questions, Google gives you answers.




THE WORLD
IS YOUR