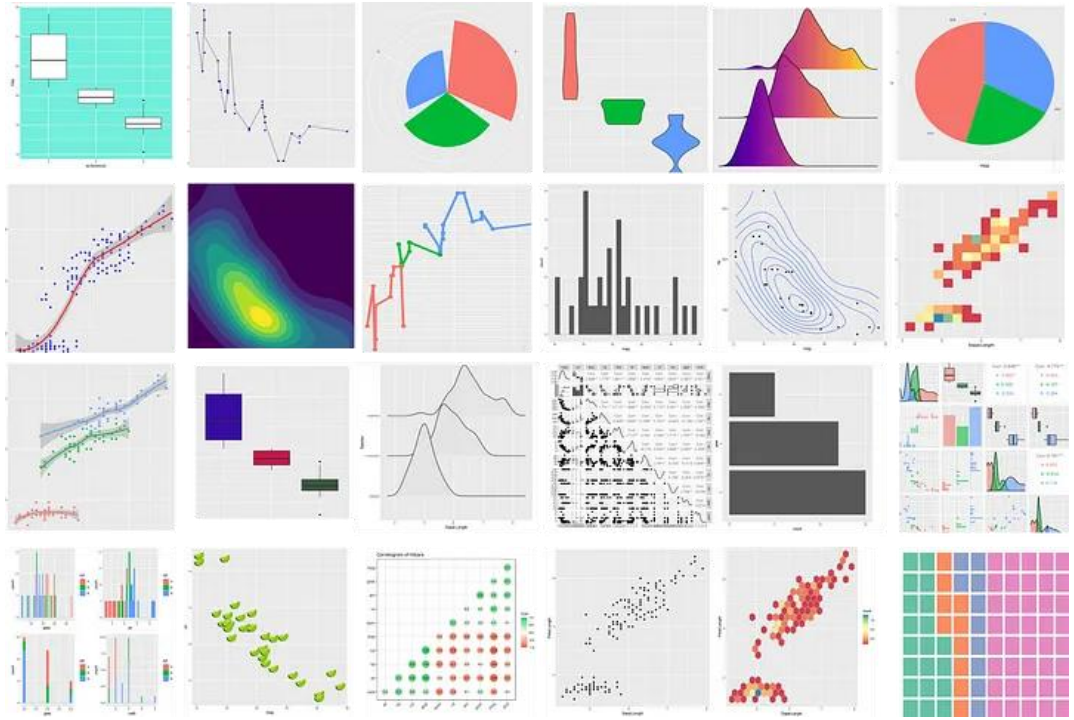
Data visualization with ggplot

Overview

Data visualization using ggplot

- Quick review of the grammar of graphics
- Using ggplot to create data visualizations



If there is time

- ggplot bonus features

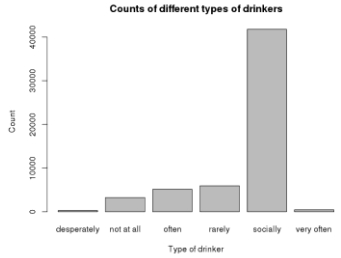
Homework 3

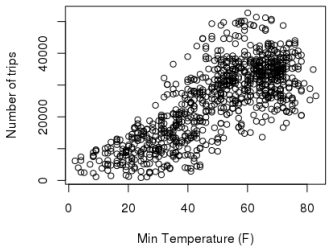
It is due on Gradescope by 11pm on Monday July 21st

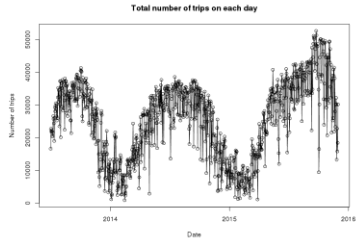
How is the homework going so far?

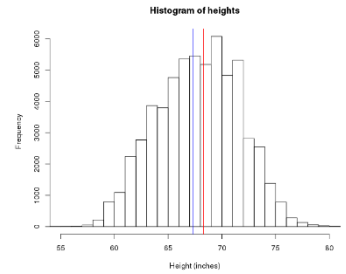
Review of the grammar of graphics and ggplot

The grammar of graphics





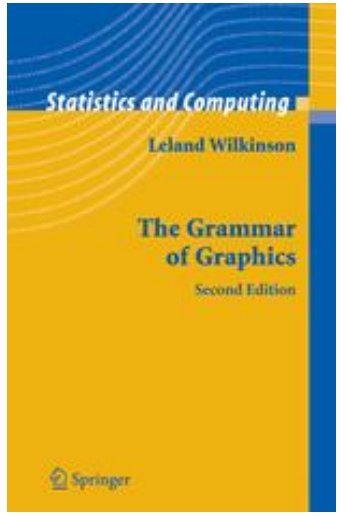


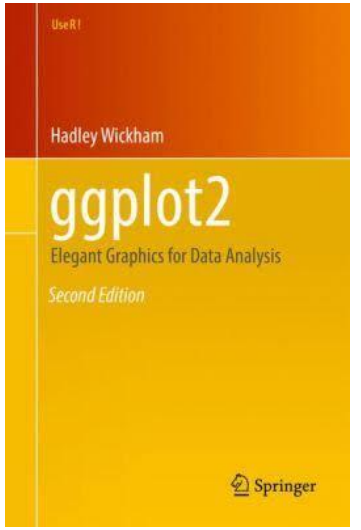


Leland Wilkinson noticed similarities between many graphs and tried to generate a ‘grammar’ that could be used to express a graph

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

Hadley Wickham implemented these ideas in R in the ggplot2 package





Graphs are composed of…

A Frame: Coordinate system on which data is placed

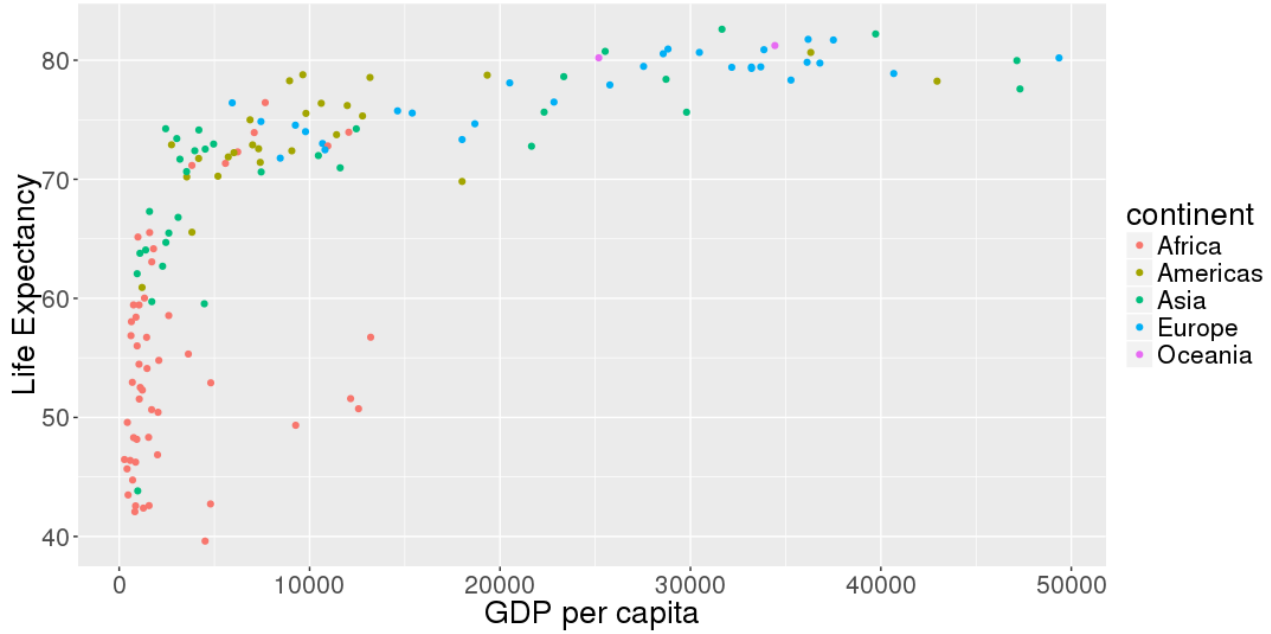
- ggplot() +

Glyphs: basic graphic unit representing cases or statistics

- Data is **mapped** onto these aesthetics such as: shape, color, size, etc. and/or aesthetics can be set to a fixed value
  - geom_point(aes(x = gdpPercap, y = lifeExp, color = continent)) geom_point(aes(x = gdpPercap, y = lifeExp), color = "red")

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

- scale_x_continuous(trans = "log10") scale_color_brewer(type = "qua", palette = 2)



Plots can also contain…

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

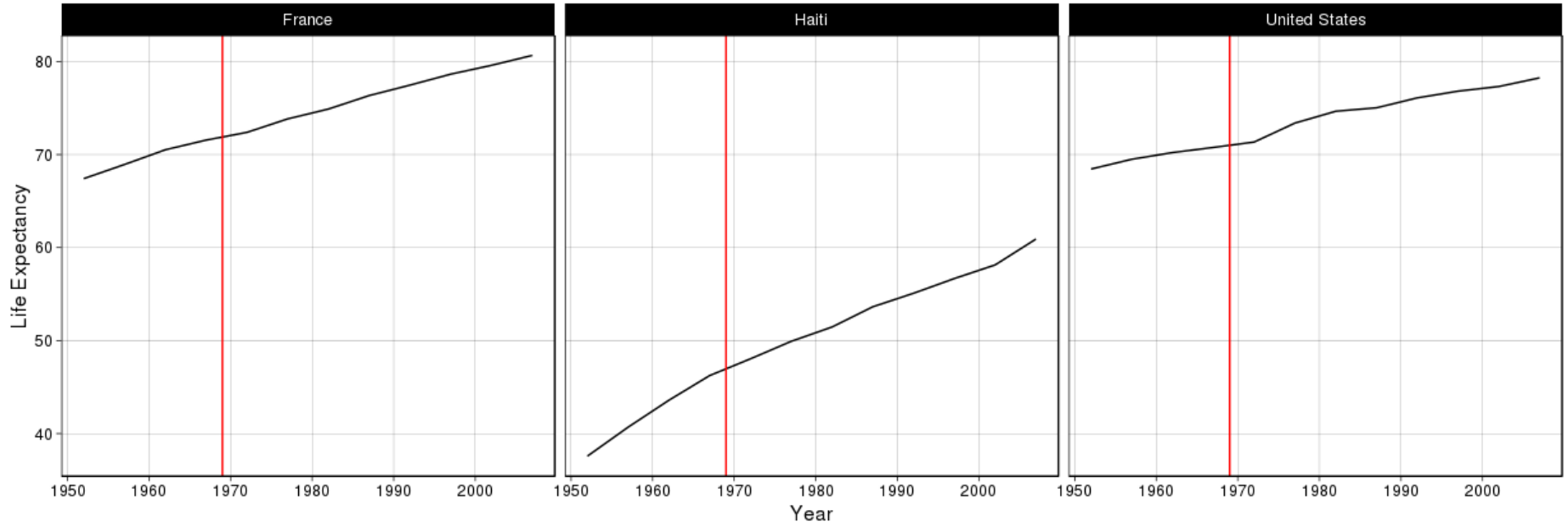
- facet_wrap(~country)

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

- geom_vline(xintercept = 1969, col = "red")

Theme: contains finer points of display (e.g., font size, background color, etc.)

- theme_wsj()



Aesthetic mappings

Data Frame

state	date	organization	leaning	leaning_detail
Michigan	29-Aug	Roth	Leaning Dem	Leaning
Kentucky	31-Aug	NYT	Likely Rep	86% Rep.
Louisiana	4-Aug	538	Leaning Rep	55% Rep.
Arkansas	27-Aug	WaPo	Leaning Rep	65% Rep.
Georgia	22-Aug	Cook	Tossup	Tossup
Alaska	31-Aug	NYT	Leaning Dem	52% Dem.

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.

Q: What are the mappings between each variable and visual attribute?

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.

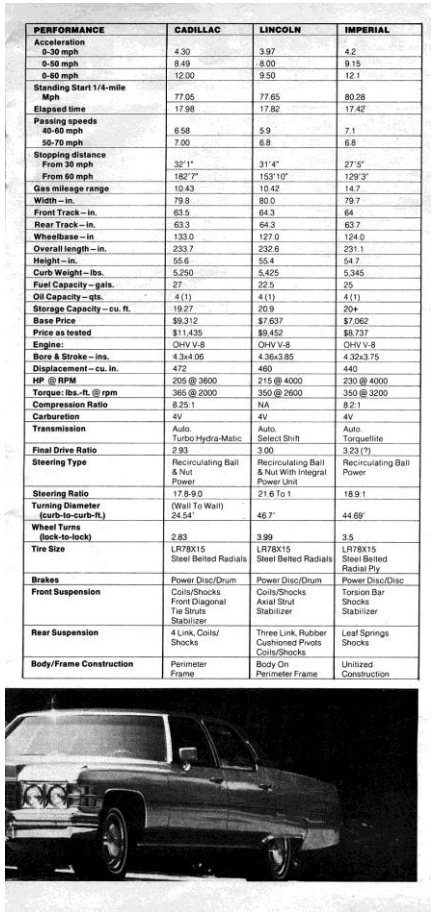
The very qualities that appear to condemn these sail-less luxury liners will very likely ensure their perpetuation. In days past, the block-long Cadillac and shiny Lincoln with paint jobs three feet deep flaunted a socio-economic position we could never hope to achieve. They did, however, constantly 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

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

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

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.

JUNE 1974 39



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

Key variables:

- **mpg** Miles/(US) gallon
- **cyl** Number of cylinders
- **hp** Gross horsepower
- **wt** Weight (1000 lbs)
- **am** Transmission
  - 0 = automatic, 1 = manual

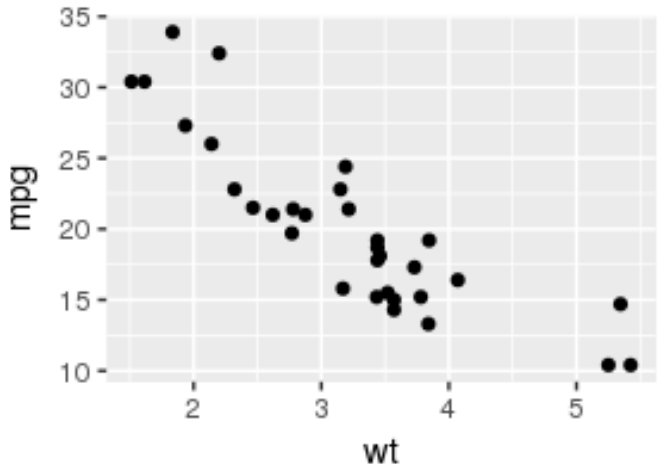
Creating a scatter plot in ggplot

Data frame to be used

Aesthetic mapping

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

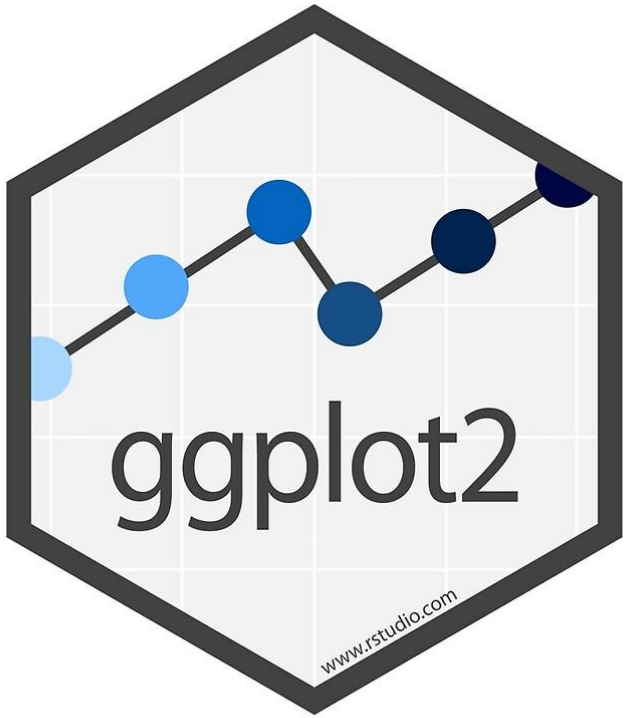
Adds a layer with glyphs



	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.

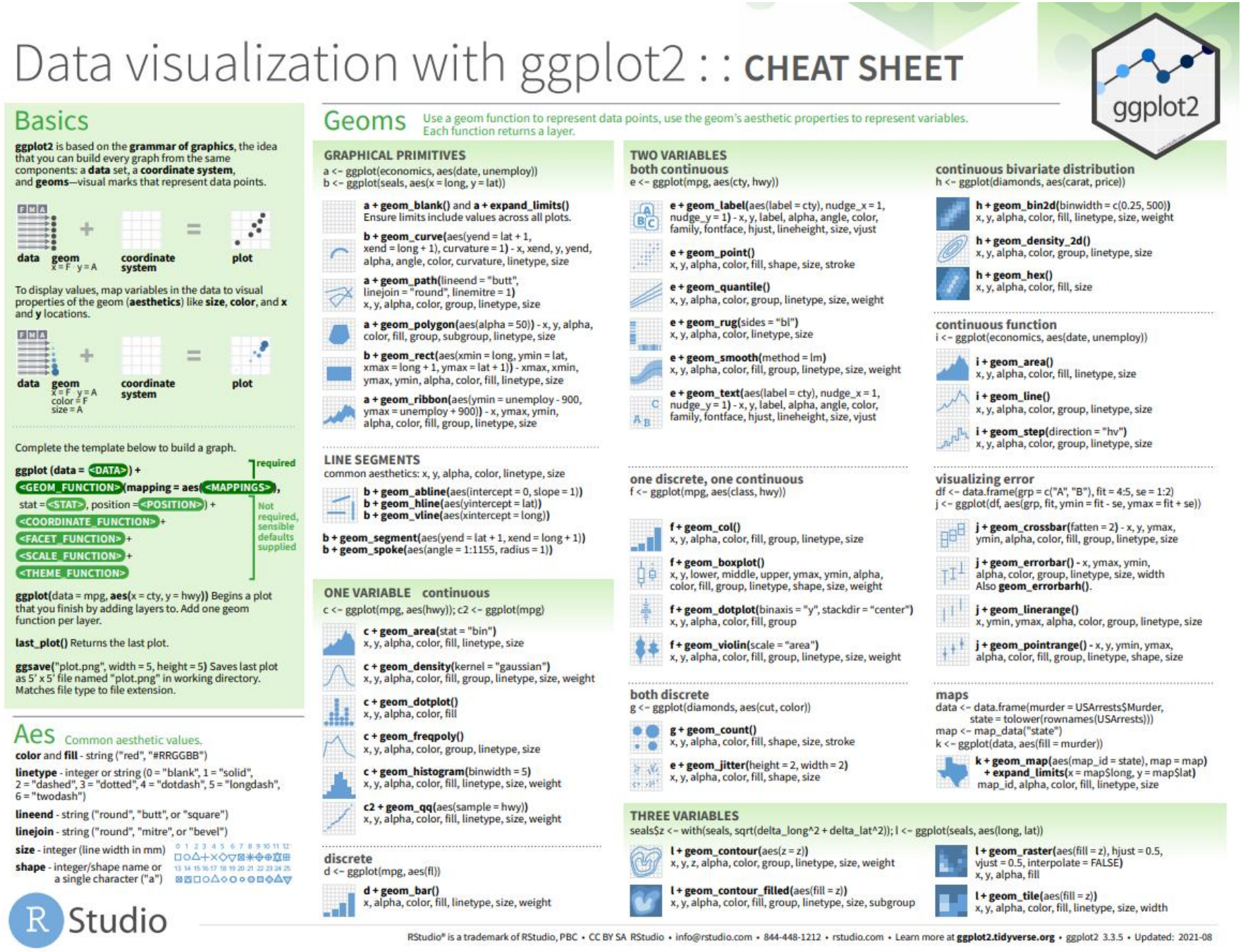


The R Graph Gallery

Let’s try the rest in R!

Questions?

[ggplot2 cheat sheet]



Adding labels to plots

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

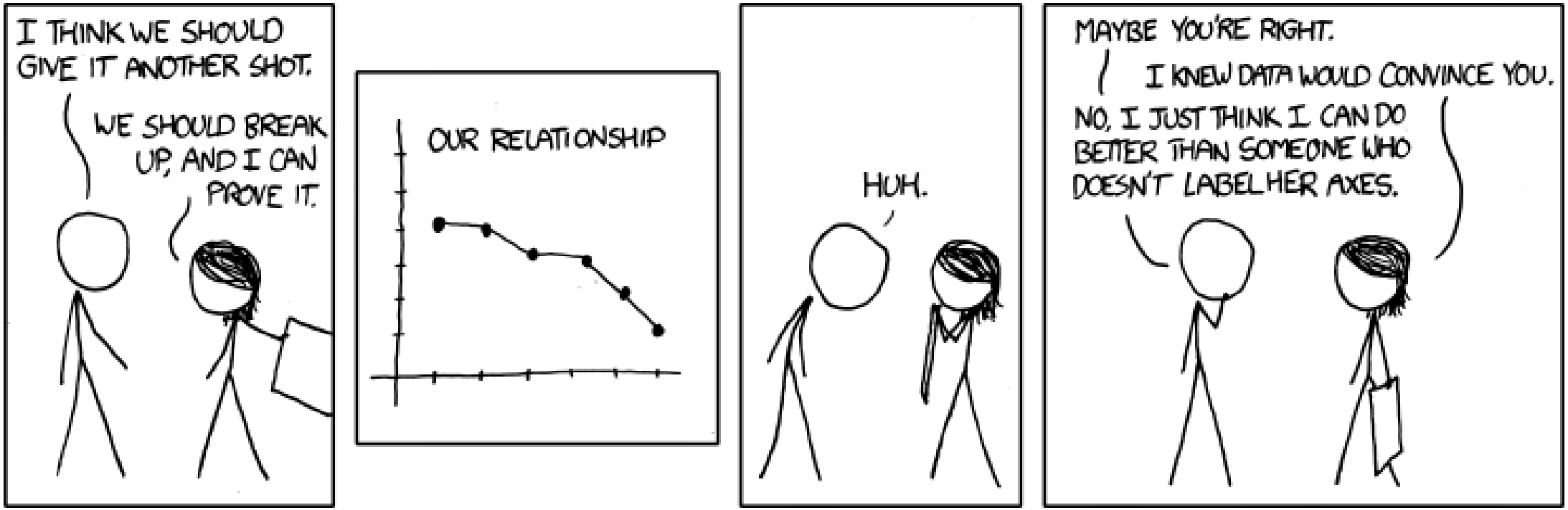
```
> ggplot(mtcars, aes(x = wt, y = mpg)) +
        geom_point()  +
        labs(x = "Weight",
             y = "Miles per Gallon",
             title = "MPG as a function of weight",
             subtitle = "ggplot is cool")
```

Adding labels to plots alternative

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")
```



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… (?)

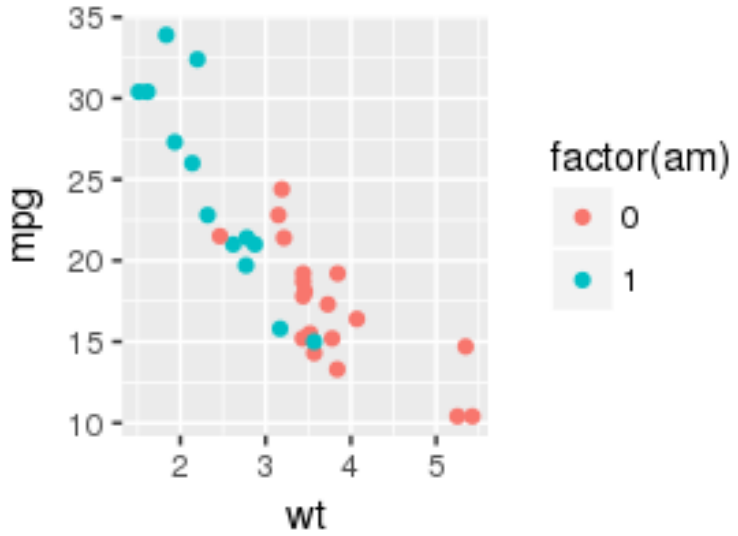
```
> ggplot(mtcars, aes(x = wt, y = mpg, col = am)) +
        geom_point()

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

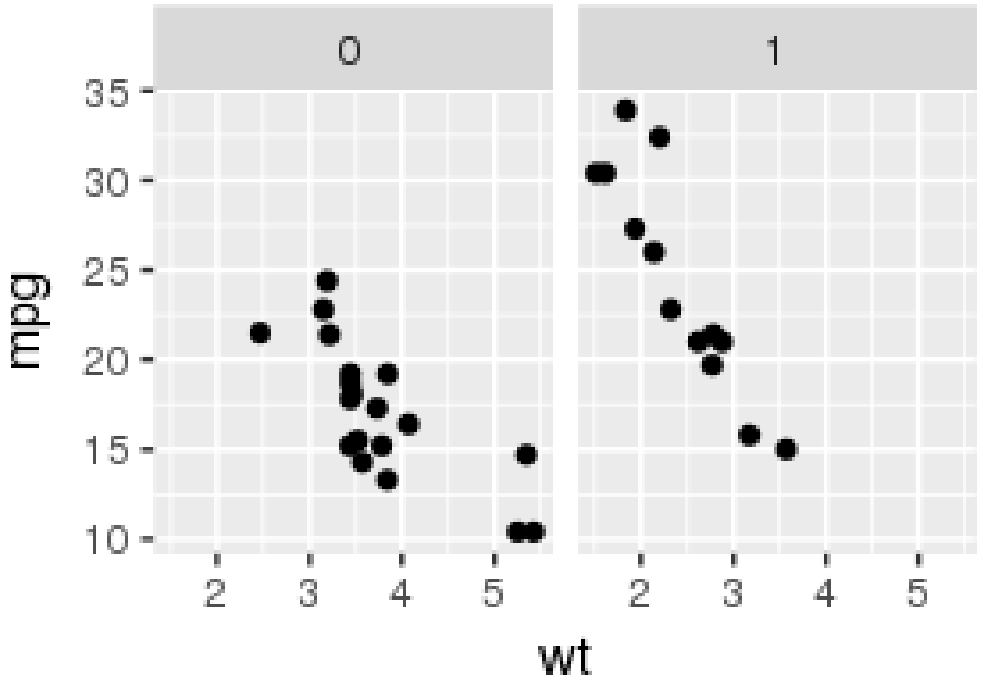
Outside the aesthetic mapping!

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)
```

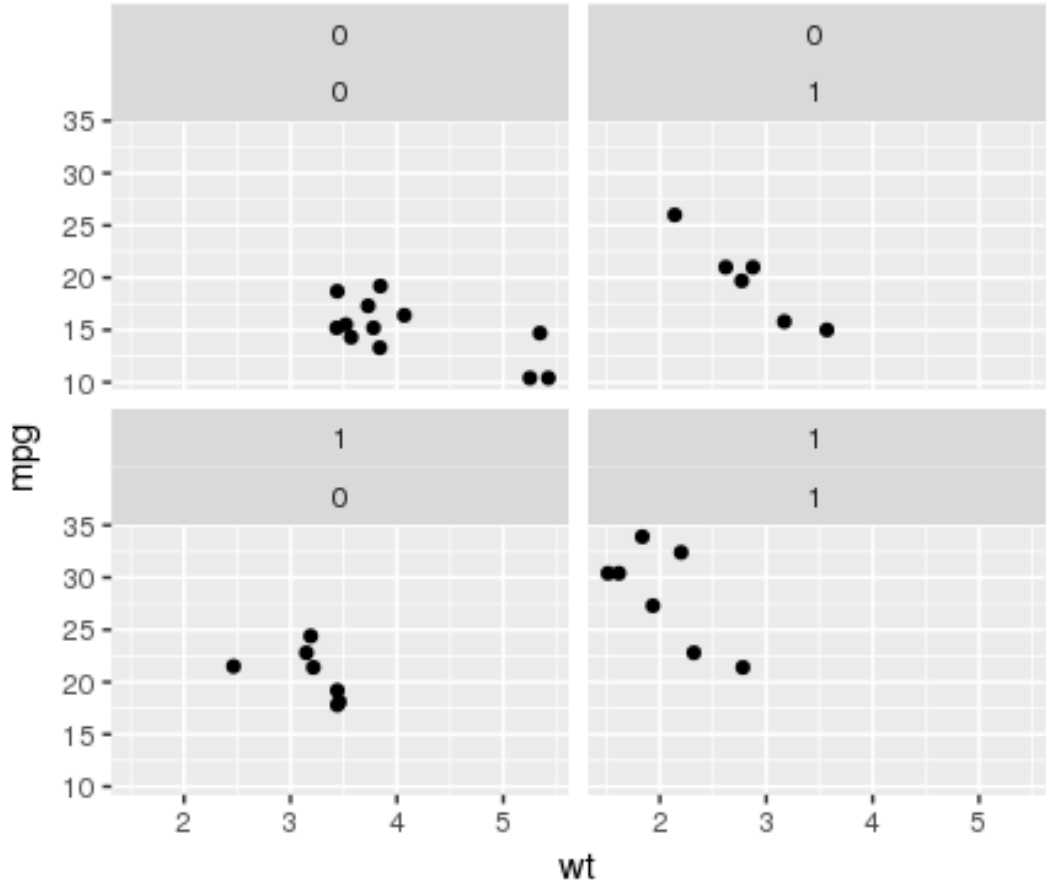
What do facets make it easy to see on this graph?



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)
```



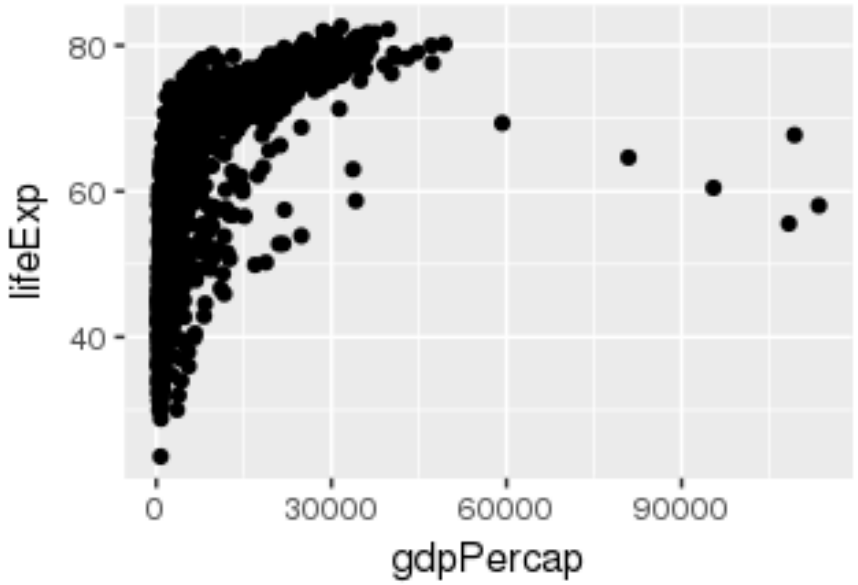
Overplotting

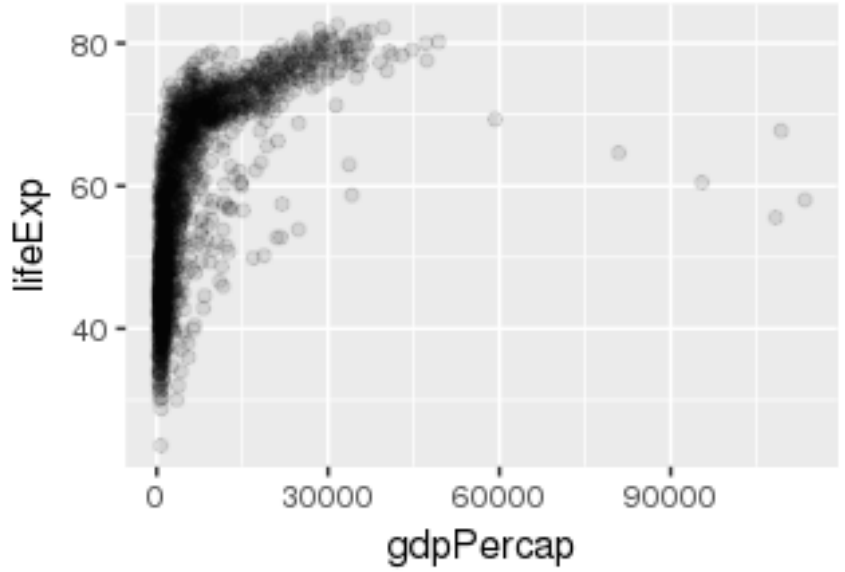
Sometimes points overlap making it hard to estimate the number of points at a particular range of values

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

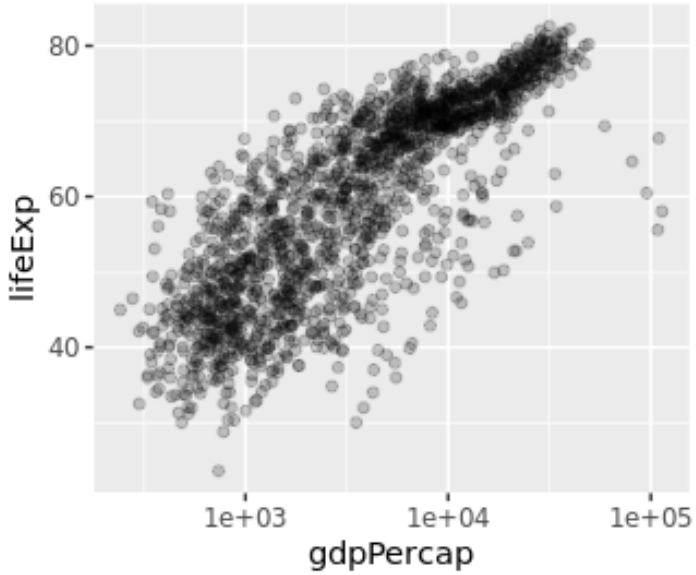
We can change the scale underlying each aesthetic visual feature

- 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

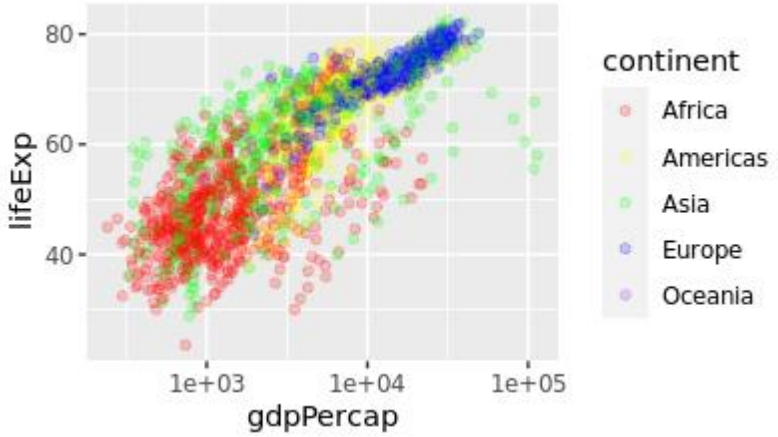
We can change the scale underlying each aesthetic visual feature

- 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"))
```



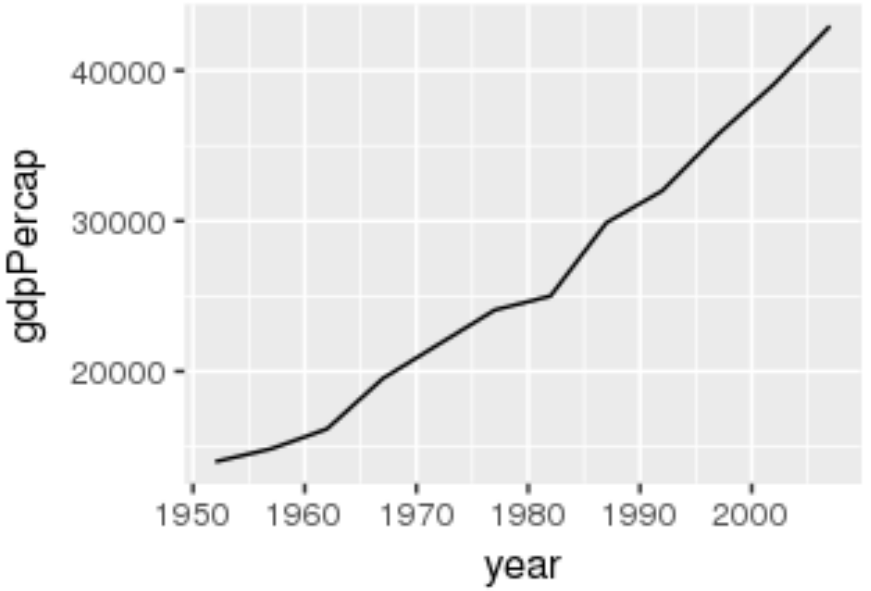
Geometries: line plot

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…

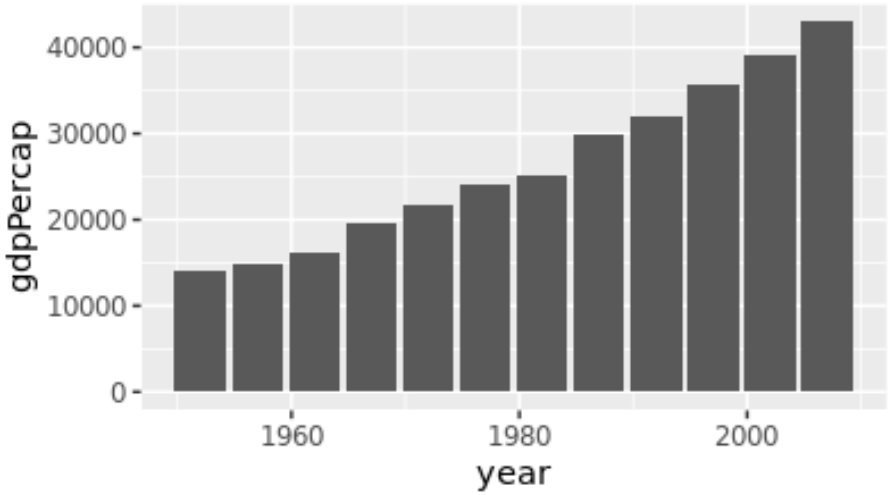
```
> gapminder |>
      filter(country == 'United States') |>
      ggplot(aes(x = year, y = gdpPercap)) +
      geom_line()
```



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()
```

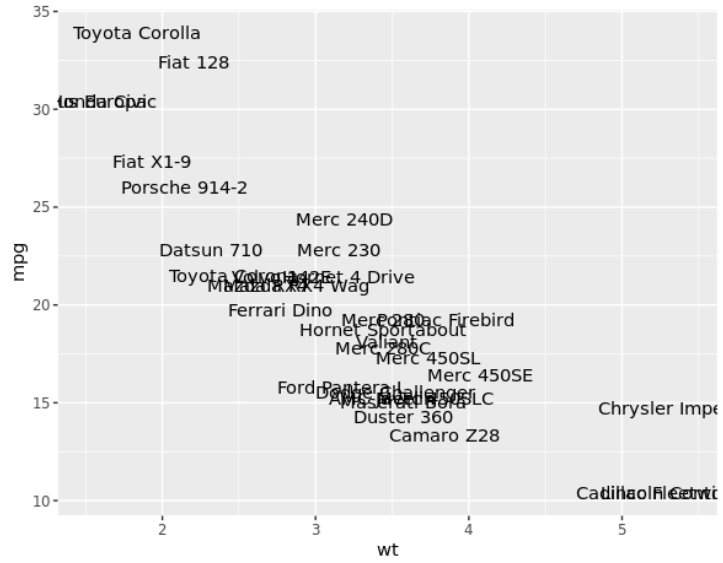


Geometries: text

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()

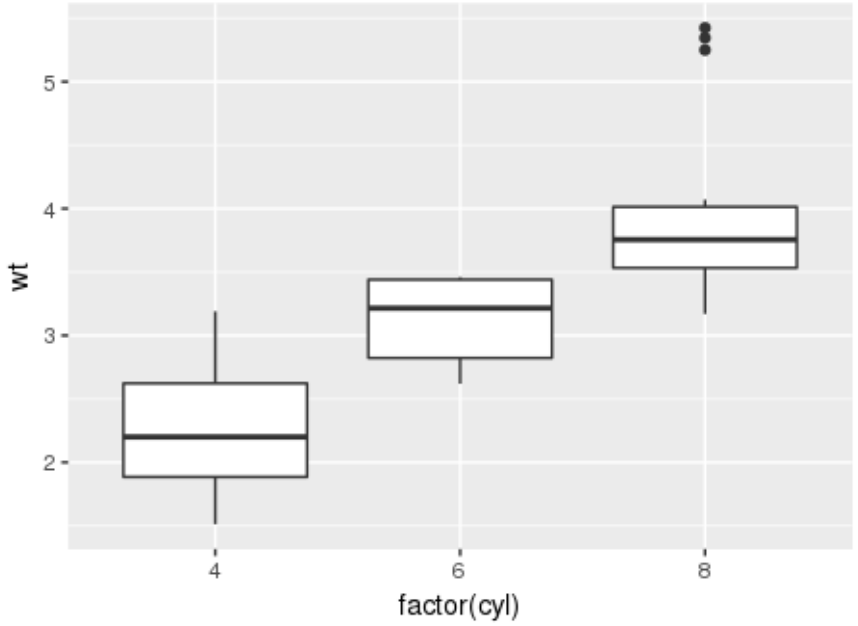
Plot a boxplot of the weights of cars

```
> 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

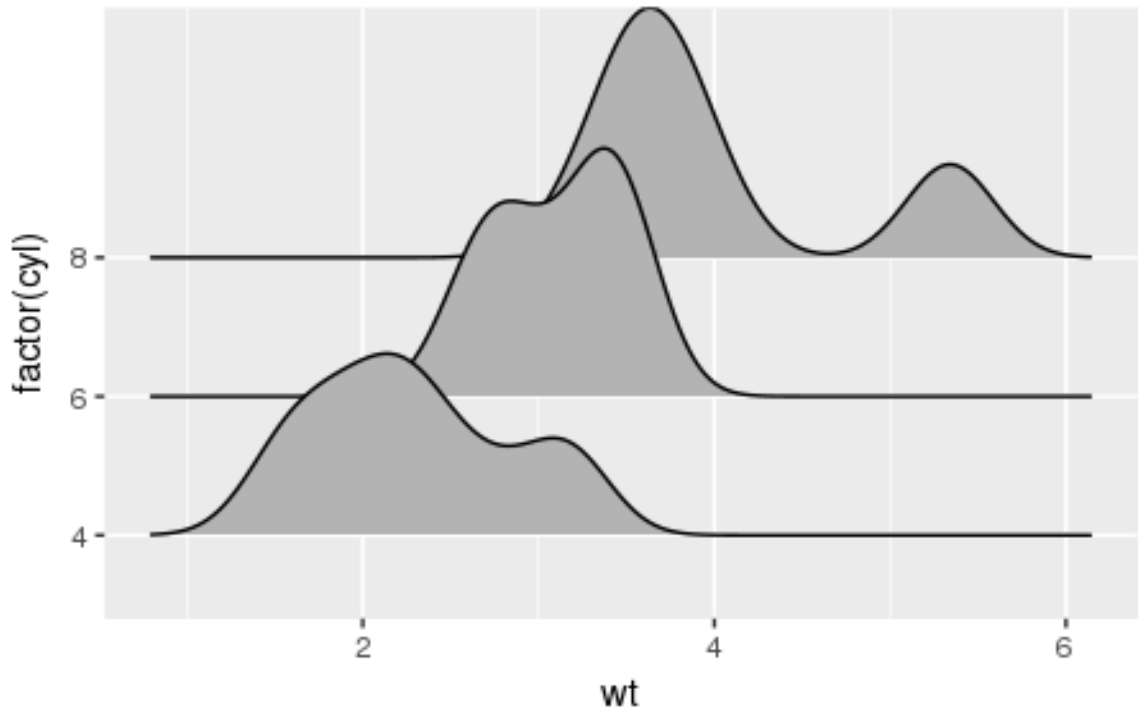
```
> ggplot(mtcars, aes(x = factor(cyl), y = wt)) +
        geom_violin()
```

```
> 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

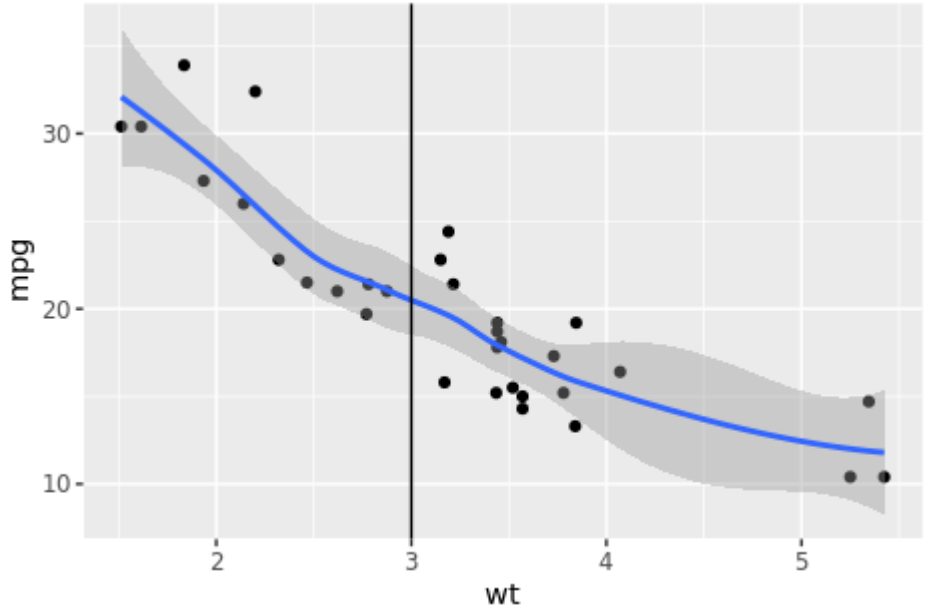
We can also have multiple geom layers on a single graph by using the + symbol

- 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)
```



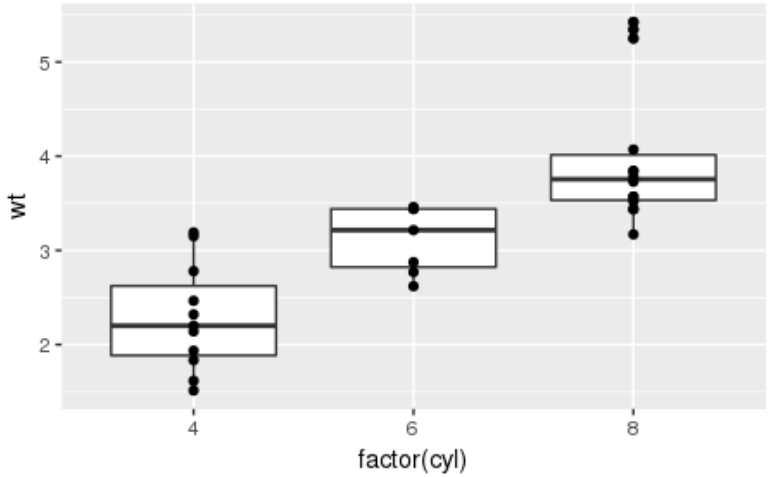
Multiple layers

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

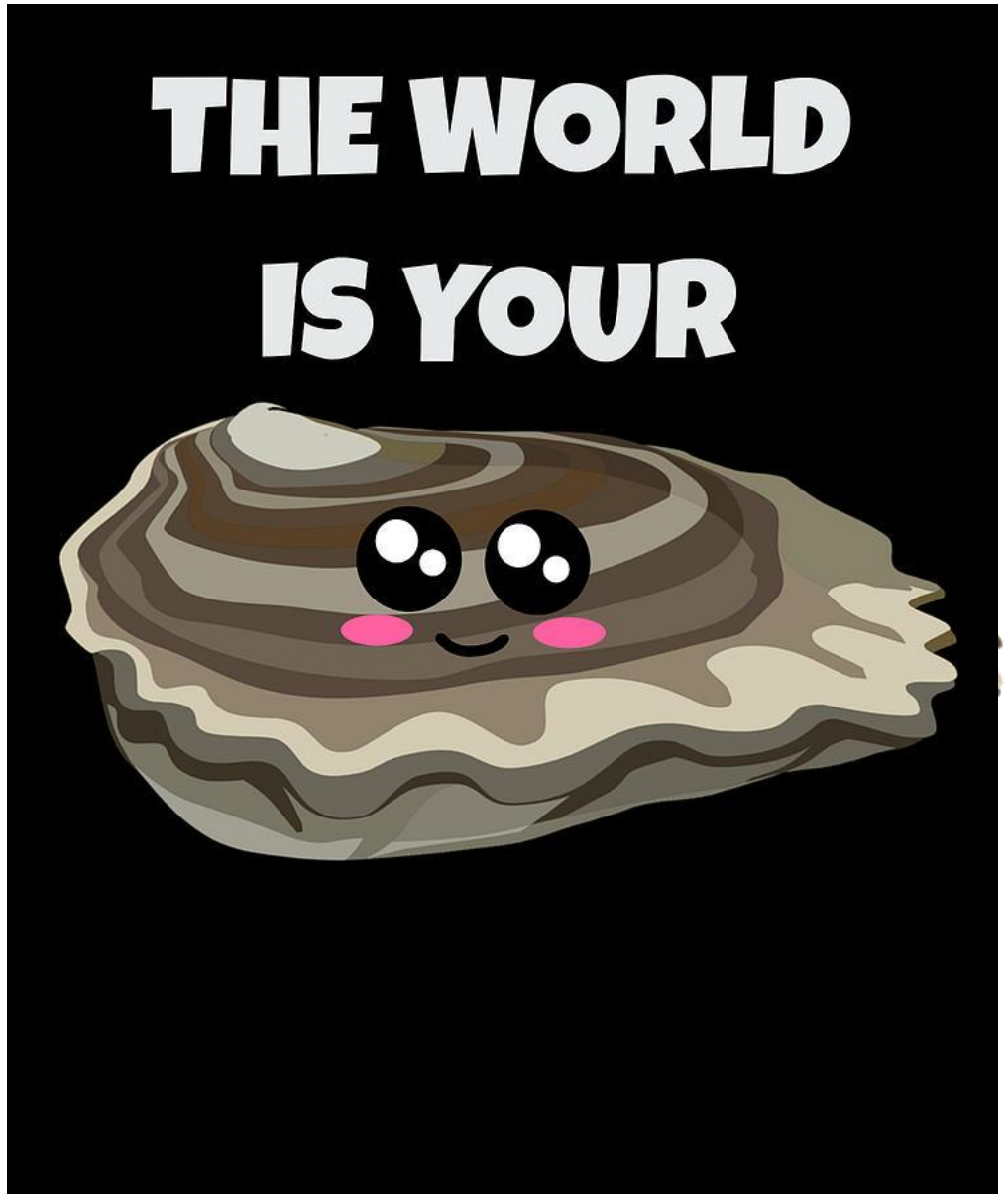
We can also create a customized theme using theme()

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

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!")
```


THE WORLD
IS YOUR

ggplot bonus features

ggplot bonus features: plotly

We can use the plotly package to make interactive graphs

The easiest way to do this is to save our ggplot to an object and then use the ggplotly() function

```
g <- gapminder |>
        filter(year == 2007) |>
        ggplot(aes(x = gdpPercap, y = lifeExp, col = continent, name = country)) +
        geom_point()

ggplotly(g)
```

ggplot bonus features: emojis

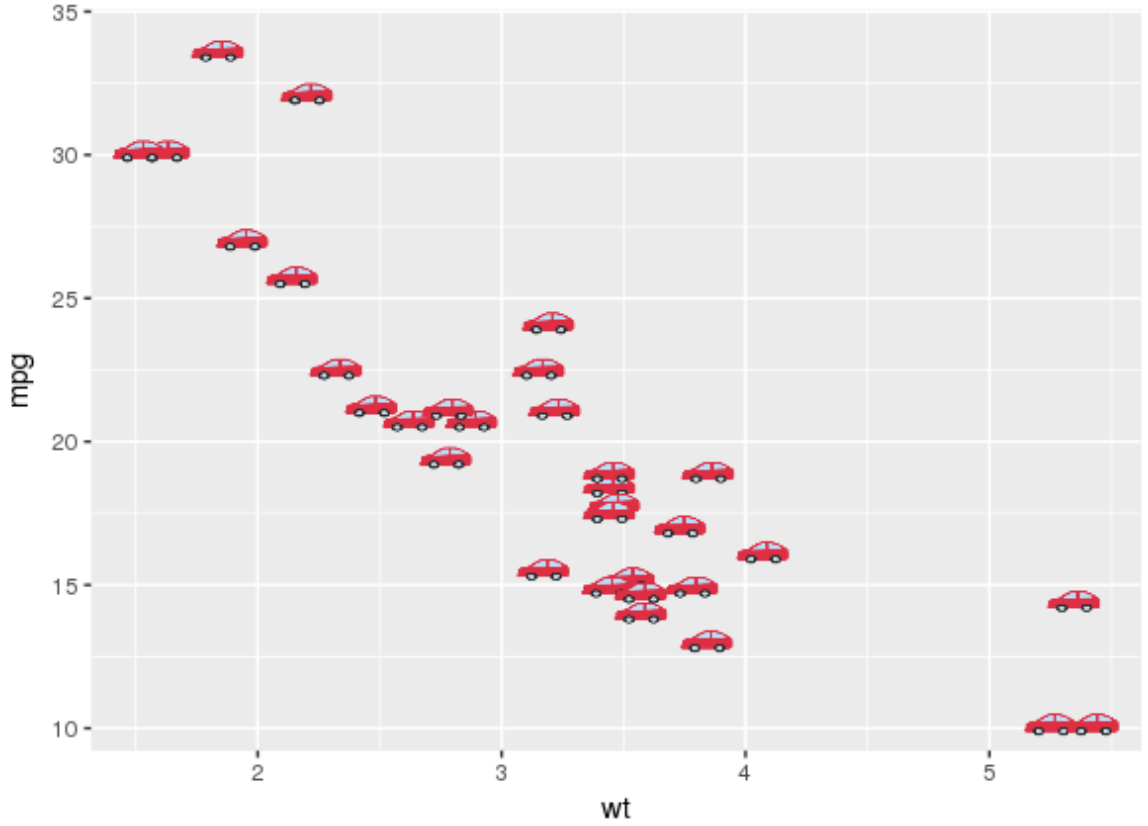
There are also additional packages that add more geoms

```
library(emojifont)
load.emojifont('OpenSansEmoji.ttf')

ggplot(mtcars, aes(x = wt, y = mpg, label = emoji("car"), col = factor(cyl))) +
        geom_text(family="OpenSansEmoji", size=6)
```

ggplot bonus features alternative: emojis

There are also additional packages that add more geoms

```
>  library(emoGG)

> ggplot(mtcars, aes(wt, mpg)) +
        geom_emoji(emoji="1f697")
```



ggplot bonus features: animation

We can create animated images (gifs) using the gganimate package

```
library(gganimate)

ggplot(gapminder, aes(gdpPercap, lifeExp,
          size = pop, col = continent)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_x_log10() +
 # Here comes the gganimate specific parts
   labs(title = 'Year: {frame_time}',
            x = 'GDP per capita', y = 'life expectancy') +
   transition_time(year) +
   ease_aes('linear')
```