

# ***DATA VISUALIZATION IN R***

## ***An Introduction with ggplot2***

***Sam Tyner, Ph.D.***

***12 October 2021***

# OUTLINE

- Introduction to data visualization
- Visualizing 1, 2, or 3+ variables
- Introduction to the **g**rammar of **g**raphics
- Building a visualization with `ggplot2`
- Beyond the basics of `ggplot2`
- `ggplot2` extras

# ***INTRODUCTION TO DATA VISUALIZATION***

# ***CHOOSE A VISUALIZATION TYPE***

How many variables?

- One variable
- Two variables
- Three variables

What kind of variables?

- Numeric / Continuous: The range of possible values is (theoretically) infinite. Examples: height, weight, age
- Categorical / Discrete: There are a finite number of categories to which all observations belong. Examples: country, survey responses on Likert scales, age categories
  - ordered: categories have a natural order. Example: income brackets
  - unordered: categories don't have order. Example: eye color

# DATA EXAMPLE

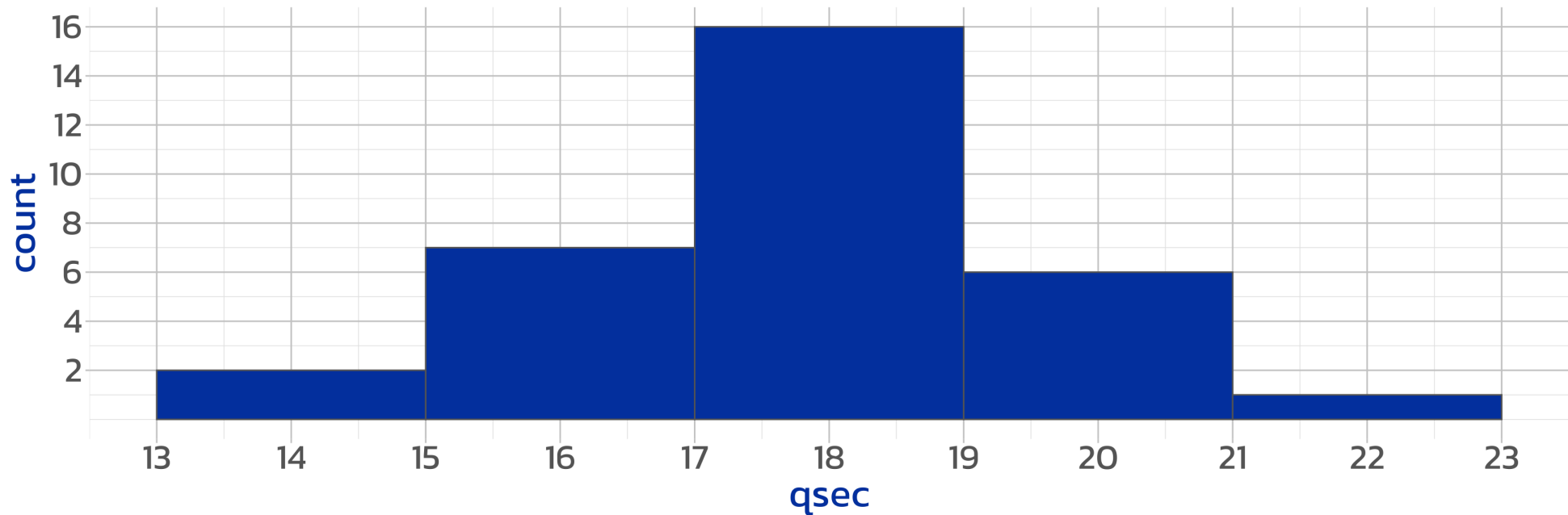
`mtcars`: Data from the 1974 Motor Trend US magazine. Fuel consumption and 10 aspects of 32 automobiles (1973–74 models). First 6 cars below.

car make/model		character: unordered categorical data type		double: numeric/continuous data type		factor: unordered categorical data type		ordered: ordered categorical data type				
car		mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
<chr>		<dbl>	<ord>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fct>	<fct>	<ord>	<ord>
## 1	Mazda RX4	21	6	160	110	3.9	2.62	16.5	v-sha...	manual	4	4
## 2	Mazda RX4...	21	6	160	110	3.9	2.88	17.0	v-sha...	manual	4	4
## 3	Datsun 710	22.8	4	108	93	3.85	2.32	18.6	strai...	manual	4	1
## 4	Hornet 4 ...	21.4	6	258	110	3.08	3.22	19.4	strai...	autom...	3	1
## 5	Hornet Sp...	18.7	8	360	175	3.15	3.44	17.0	v-sha...	autom...	3	2
## 6	Valiant	18.1	6	225	105	2.76	3.46	20.2	strai...	autom...	3	1
miles per US gallon		# cylinders		displacement (cubic in.)	gross horsepower	rear axle ratio	weight (x1000lbs)	1/4 mile time	engine type	transmission type	# forward gears	# carburetors

# ***VISUALIZING ONE VARIABLE***

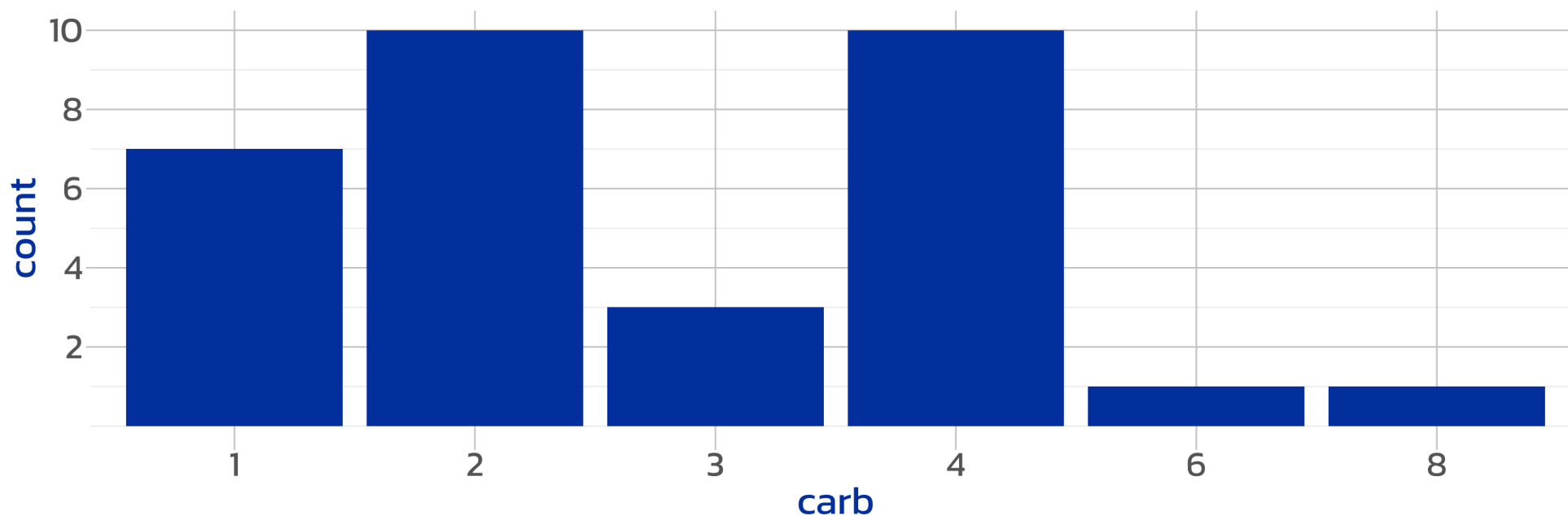
# HISTOGRAM

- For one continuous variable
- Width of bar shows a range of values of the variable
- Height shows the number of observations in that range



# BAR CHART

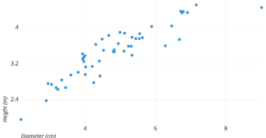
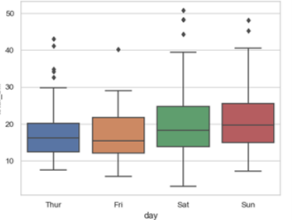
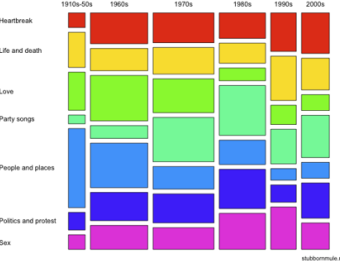
- For one categorical variable
- Each bar represents one category
- Height of bars represents number of observations in that category





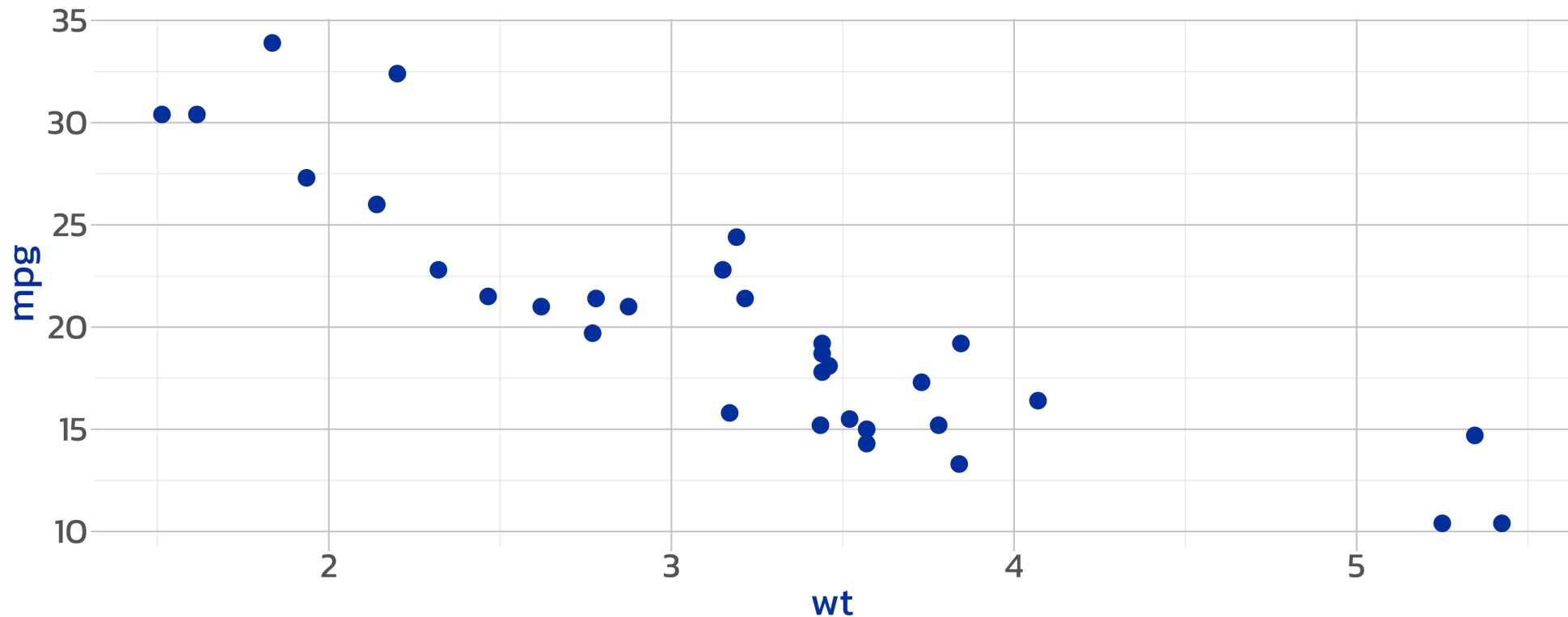
# ***VISUALIZING TWO VARIABLES***

# WHAT KIND OF VARIABLES?

	Numerical	Categorical
Numerical	 <ul style="list-style-type: none"><li>• Scatter plot (point)</li><li>• 2D binning (bin2d, hex)</li><li>• Contour plot (density2d)</li><li>• Quantiles (quantile, qq)</li><li>• Lines (line, smooth)</li><li>• Ribbons (ribbon, area)</li></ul>	 <ul style="list-style-type: none"><li>• Boxplot (boxplot, violin)</li><li>• Counts (count, tile)</li><li>• Error bars (errorbar)</li><li>• Columns (col)</li></ul>
Categorical	<p><b><i>Which ggplot2 data viz is right for your data?</i></b></p> <p>(geoms in parentheses)</p>	 <ul style="list-style-type: none"><li>• Mosaic (ggmosaic::geom_mosaic)</li><li>• Counts (count, tile)</li></ul>

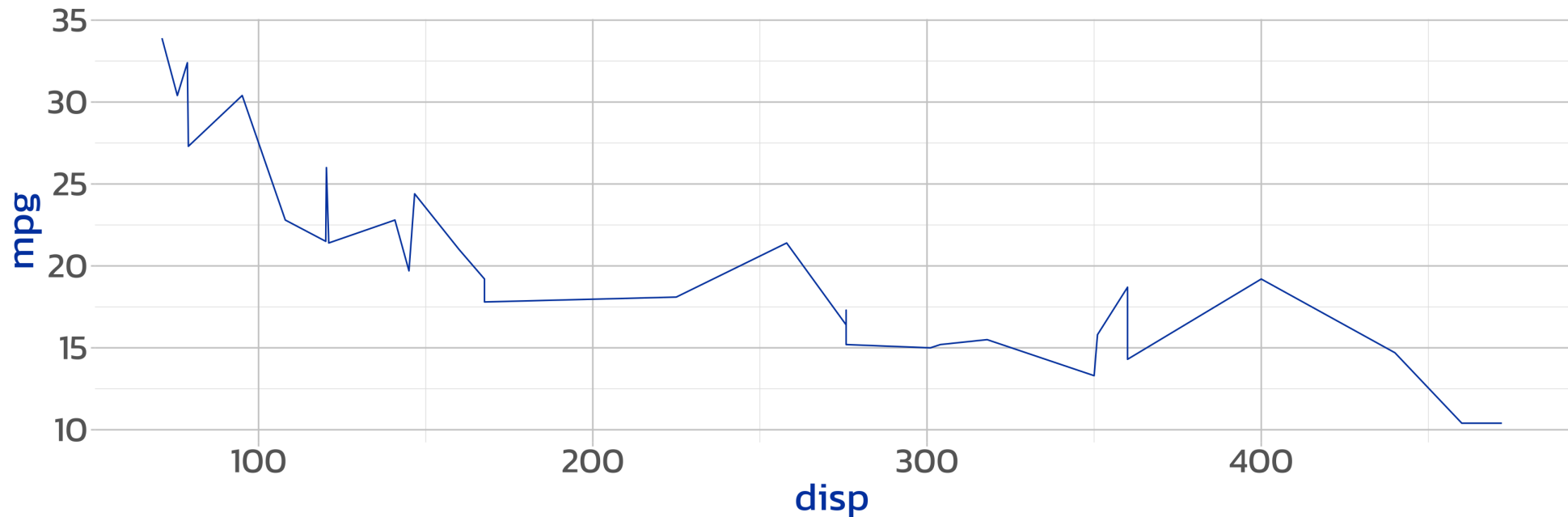
# ***TWO NUMERIC VARIABLES: Scatterplot***

- Draw points in 2 dimensions: one per observation
- Values of one variable on X axis, value of the other on Y axis



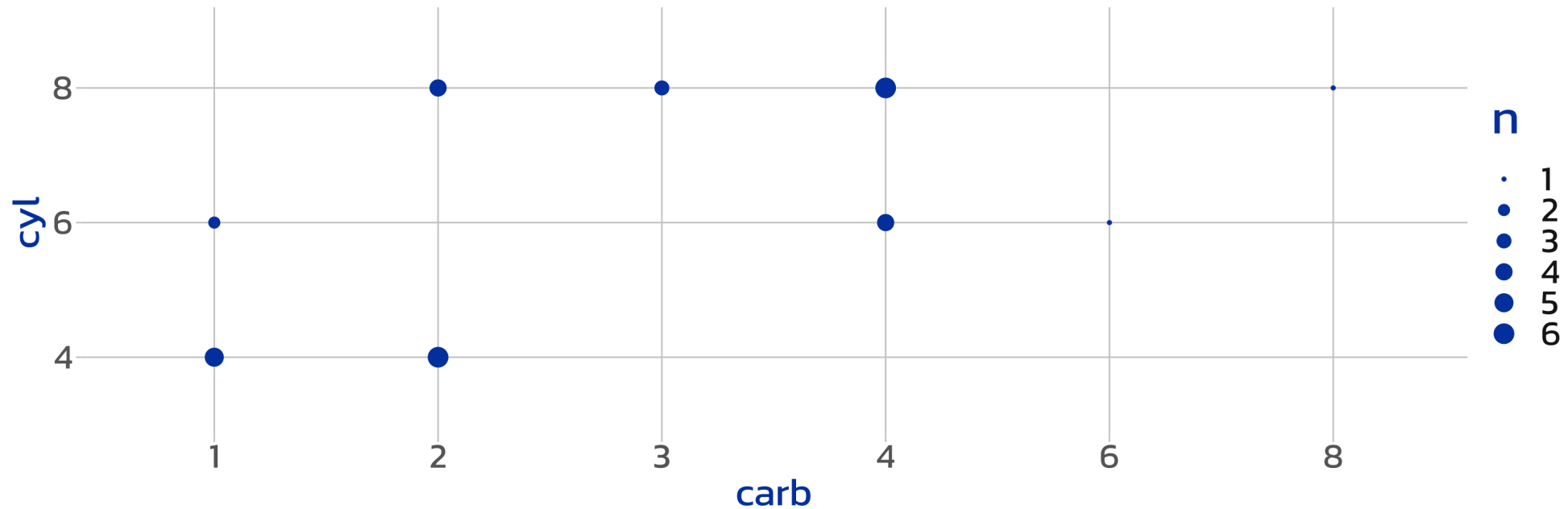
# ***TWO NUMERIC VARIABLES: Line chart***

- Connect points in space that represent 2 variables
- Values of one variable on X axis, value of the other on Y axis
- Special case: time series chart, where the X axis variable is a time metric



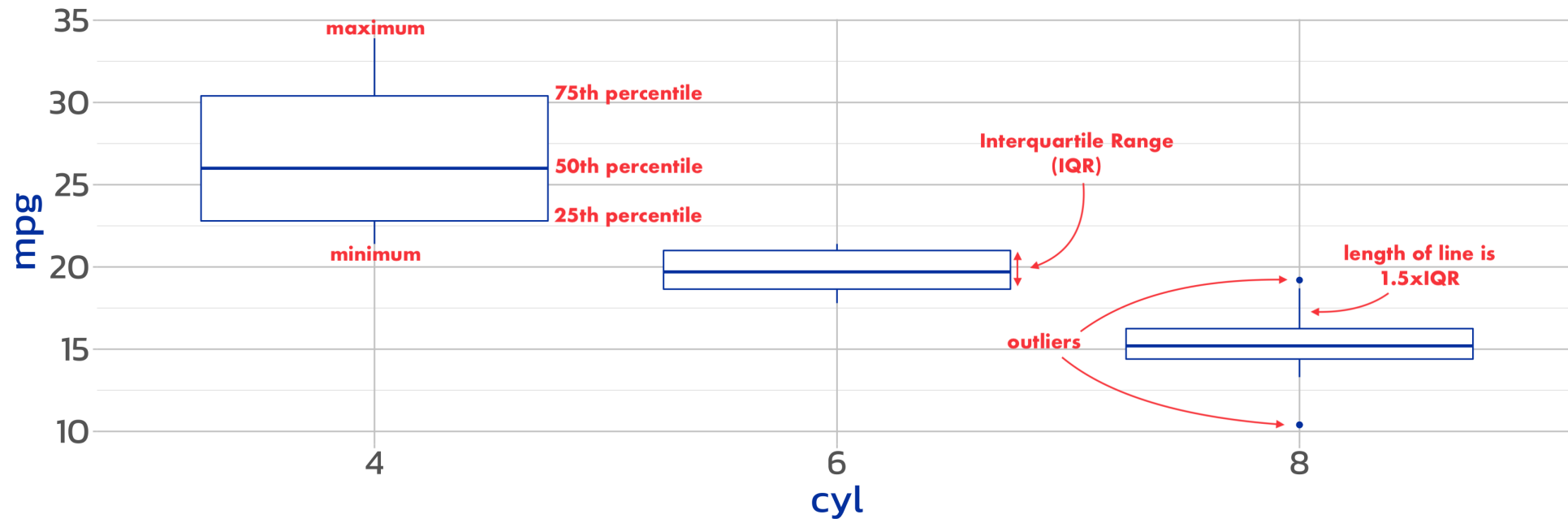
# ***TWO CATEGORICAL VARIABLES: Counts***

- Draws points at the intersections of categories from each variable
- Size of points represents number of observations in each combination of categories



# ONE OF EACH: Box Plot

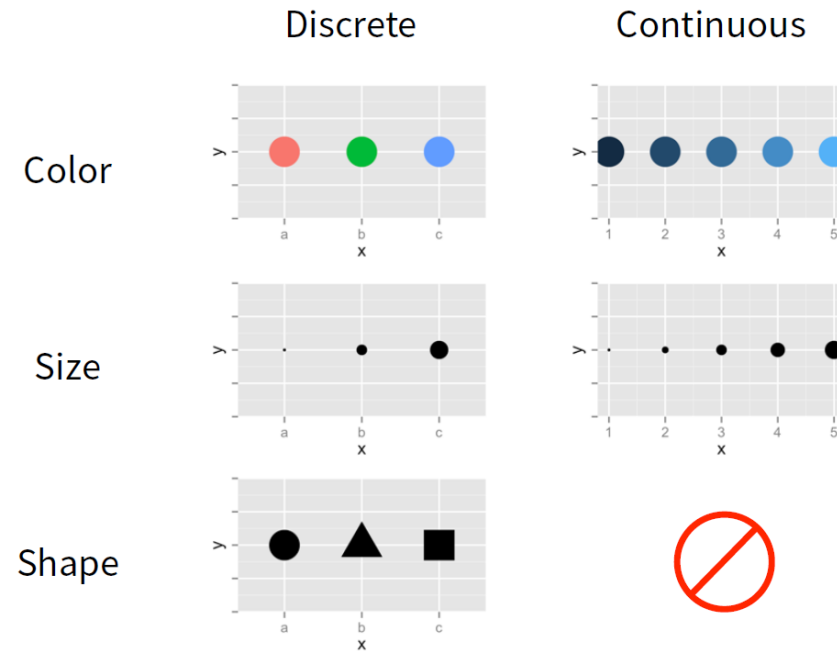
- Draw a summary box plot of a numeric variable for each value of the categorical variable
- Categorical variable can go on either X or Y axis



# ***VISUALIZING THREE VARIABLES***

# 3 VARIABLE STRATEGY

Start with a 2-variable visualization, then add color, shape, or size:

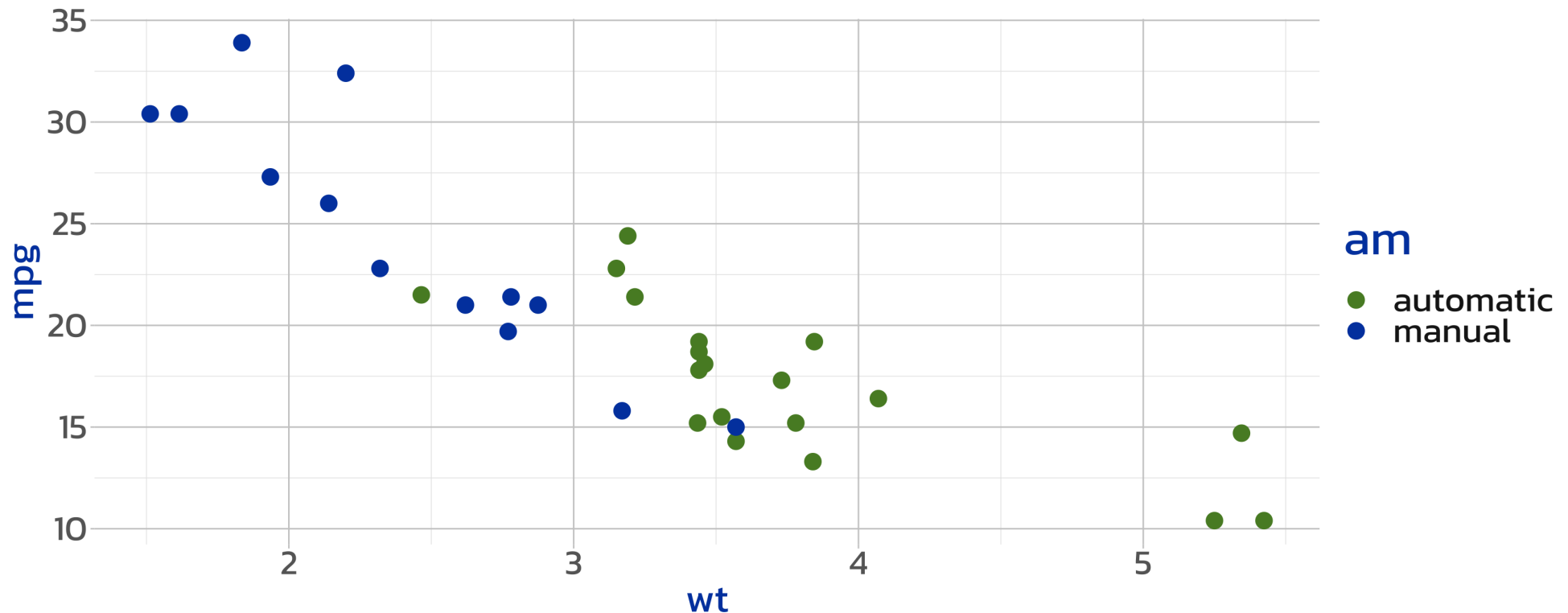


Example: adding 3rd variable to a scatterplot. Source: Garret Grolmund



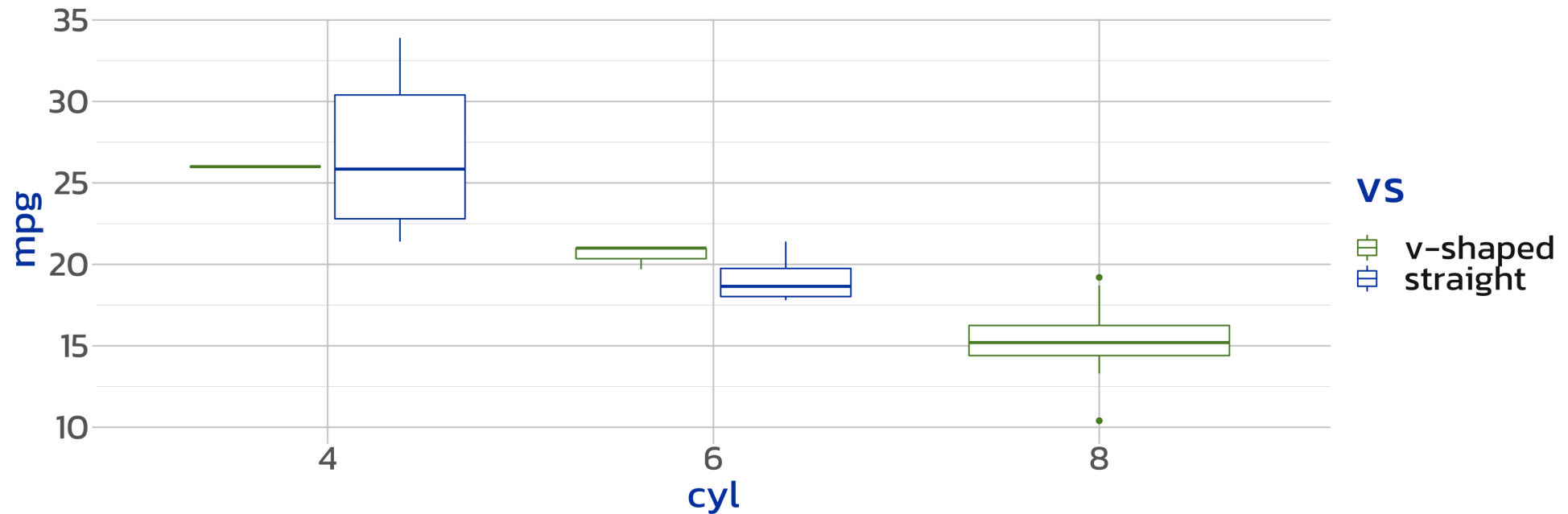
# 3 VARIABLE EXAMPLE: Scatterplot

- Color the points according to the transmission type
- We see that: automatic cars are heavier and have lower MPG than manual cars



# 3 VARIABLE EXAMPLE: Box Plot

- Color by engine type
- 4 cylinder straight engines have the highest MPGs, and 8 cylinder engines are only v-shaped.



# ***INTRODUCTION TO THE GRAMMAR OF GRAPHICS***

# WHAT IS A GRAMMAR?

*grammar* (noun): (1) the study of the classes of words, their inflections, and their functions and relations in the sentence; (2) the principles or rules of an art, science, or technique

The grammar of a language is used to construct sentences.

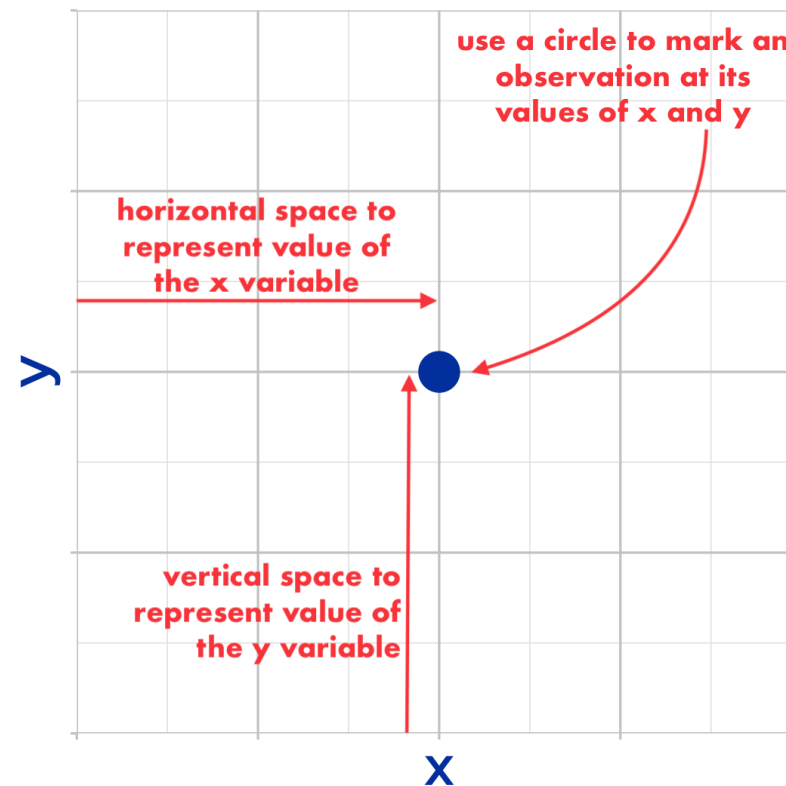
The grammar of graphics<sup>1</sup> is used to construct data visualizations.

The `gg` in `ggplot2` stands for "grammar of graphics"

[1] *The Grammar of Graphics* is the title of a 1999 book by Leland Wilkinson

# DATA VIZ "PARTS OF SPEECH"

- Data
  - **What** are you visualizing?
- Coordinate system (almost always Cartesian)
  - What is your **canvas** for visualizing?
- Mapping
  - How are you mapping variables to **visual features** (aesthetics)?
- Geometry (point, line, box plot, etc.)
  - What **shape(s)** are you drawing?



Mapping data to x and y axes

# ***HOW TO BUILD A DATA VISUALIZATION WITH GGLOT2***

# BUILDING A VIZ

1. Need the data loaded in your workspace.

```
head(mtcars)
```

```
## # A tibble: 6 × 12
##   car          mpg cyl  disp  hp  drat    wt  qsec vs      am  gear carb
##   <chr>        <dbl> <ord> <dbl> <dbl> <dbl> <dbl> <dbl> <fct> <fct> <ord> <ord>
## 1 Mazda RX4    21    6    160   110   3.9   2.62  16.5 v-sha... manual 4      4
## 2 Mazda RX4... 21    6    160   110   3.9   2.88  17.0 v-sha... manual 4      4
## 3 Datsun 710    22.8  4    108    93   3.85   2.32  18.6 strai... manual 4      1
## 4 Hornet 4 ... 21.4  6    258   110   3.08   3.22  19.4 strai... autom... 3      1
## 5 Hornet Sp... 18.7  8    360   175   3.15   3.44  17.0 v-sha... autom... 3      2
## 6 Valiant      18.1  6    225   105   2.76   3.46  20.2 strai... autom... 3      1
```

# ***BUILDING A VIZ***

2. Need the `ggplot2` package loaded.

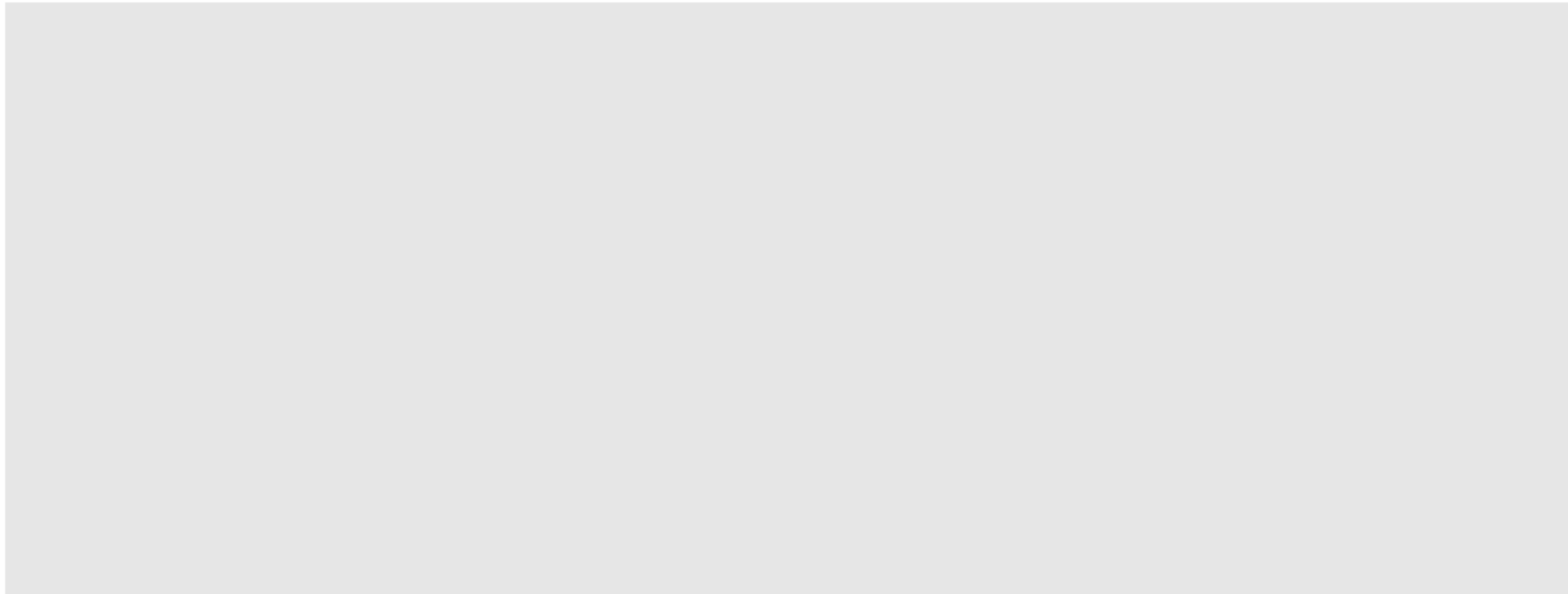
```
library(ggplot2)
```



# ***BUILDING A VIZ***

3. Set up your canvas by passing the data to the `ggplot()` function.

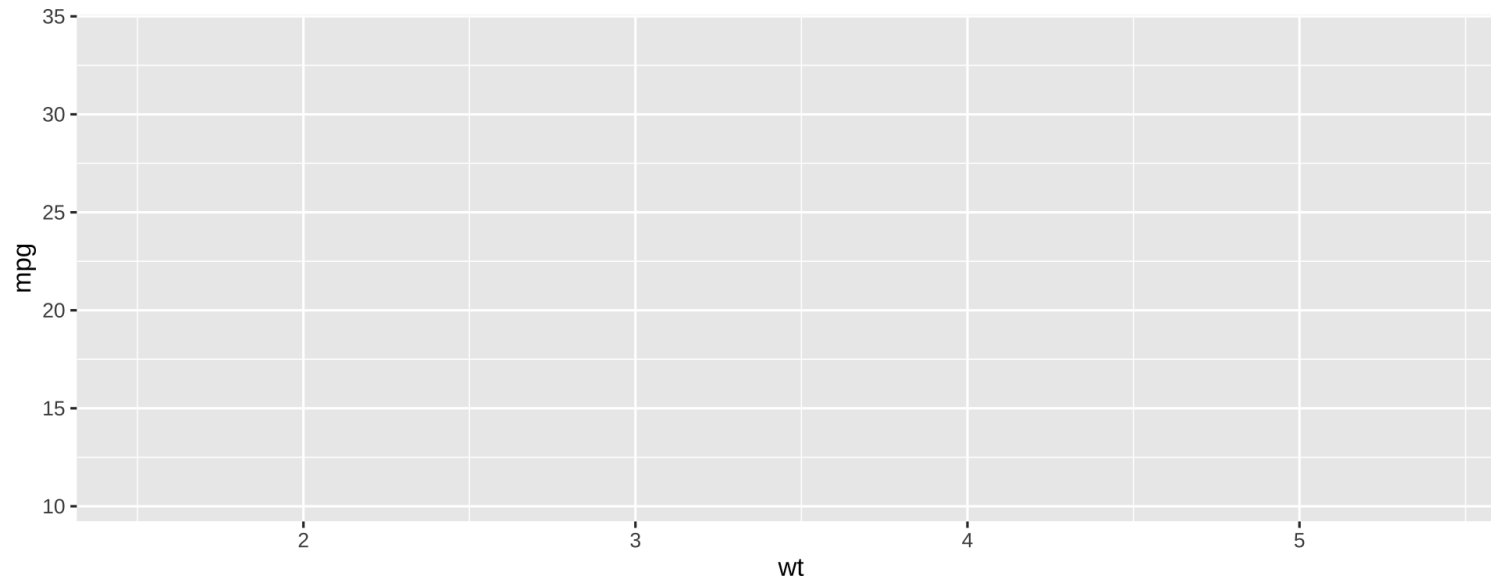
```
ggplot(data = mtcars)
```



# ***BUILDING A VIZ***

4. Set up your mapping inside of `aes()`.

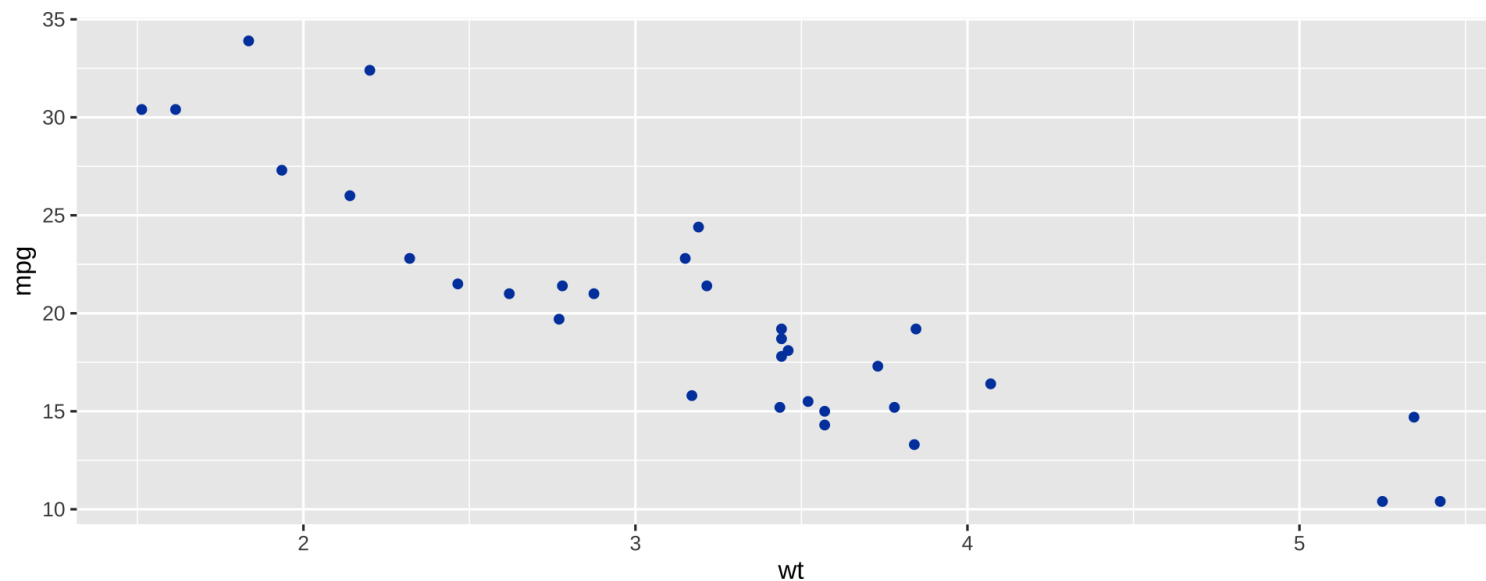
```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg))
```



## ***BUILDING A VIZ***

5. Pick your geometry with a function that starts with `geom_*()`

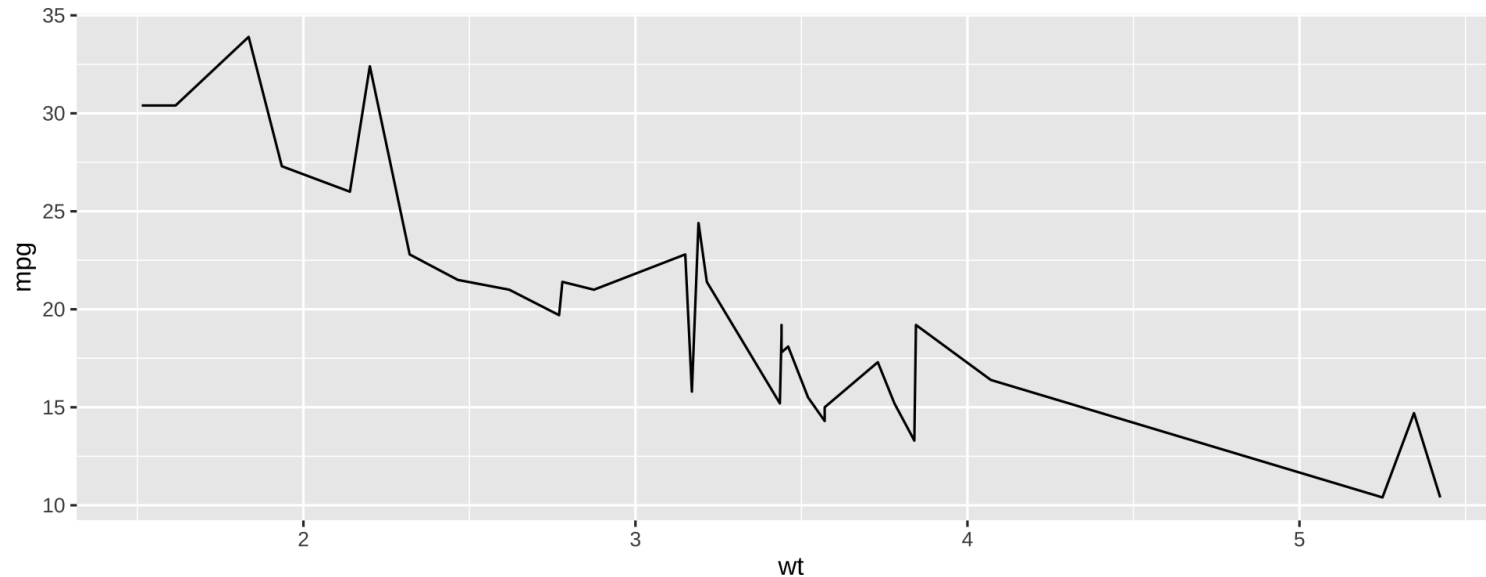
```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point()
```



# ***BUILDING A VIZ***

5. Pick your geometry with a function that starts with `geom_*()`

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_line()
```



# ***BEYOND THE BASICS***

***AESTHETICS: aes()***

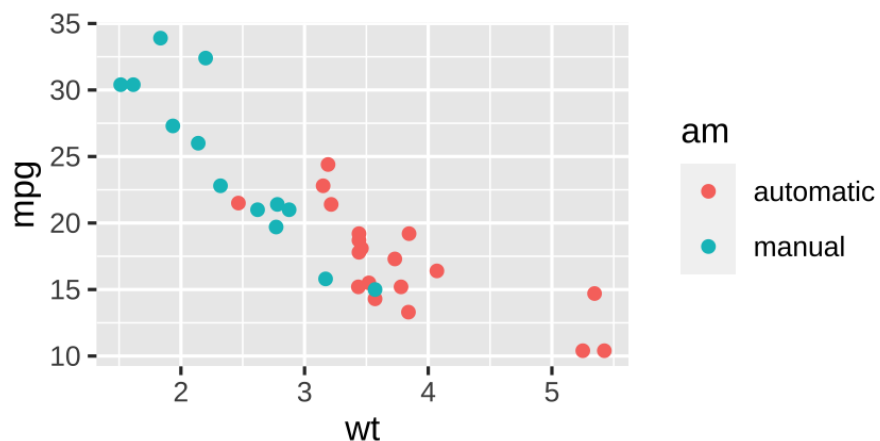
# ***USING aes() FOR MAPPING***

- `x`, `y` and other values passed to `aes()` are called **aesthetics**
- Aesthetics can be mapped to data, as in `aes(x = wt , y = mpg)`
- Aesthetics can be mapped to a single value for a geometry, outside of `aes()`
- The mapping function (`aes()`) can go inside of `ggplot()` or inside of a `geom_*()` or **both**

# EXAMPLE 1

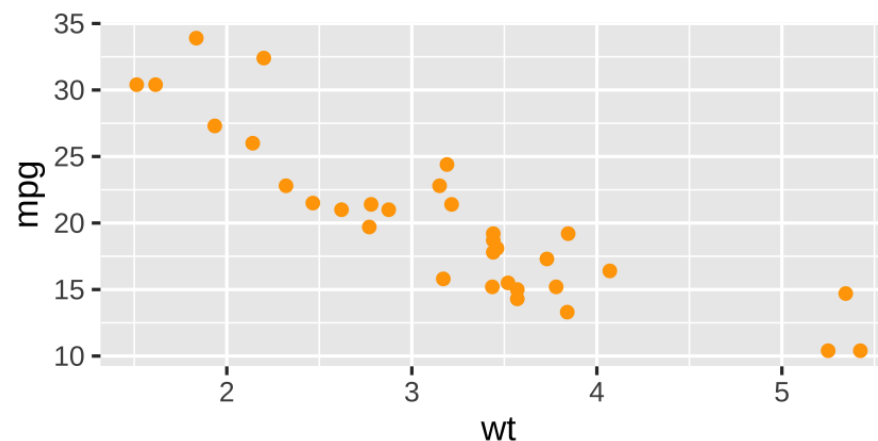
## Inside aes()

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point(aes(color = am))
```



## Outside aes()

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point(color = "orange")
```

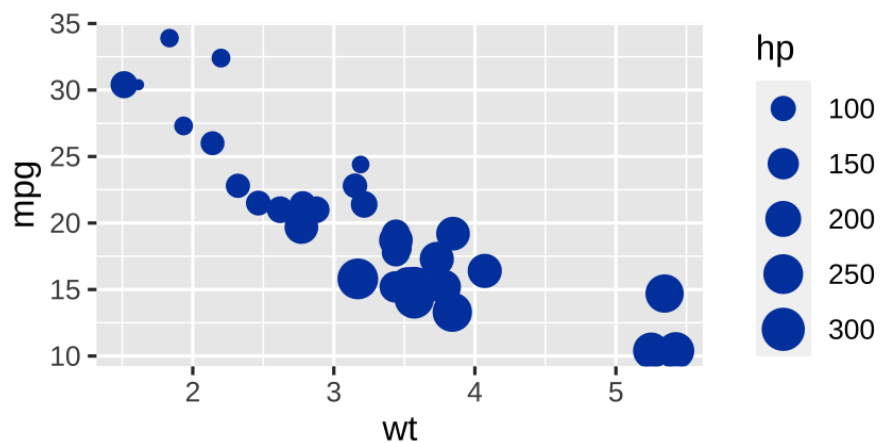




# EXAMPLE 2

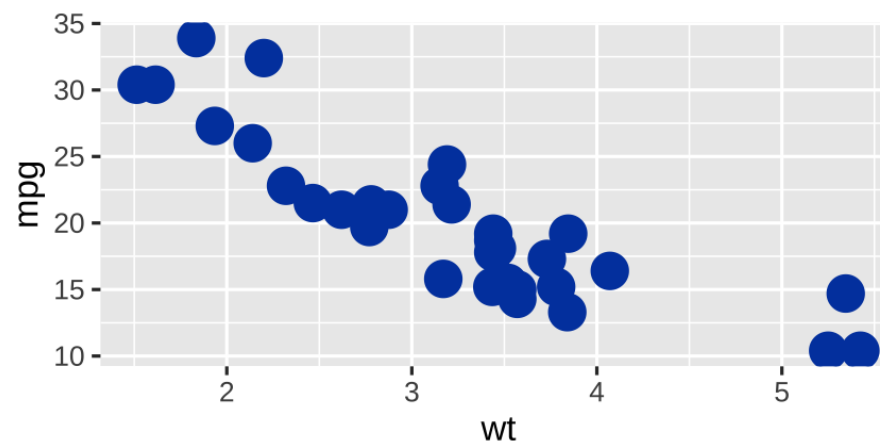
## Inside aes()

```
ggplot(data = mtcars, aes(x = wt,  
                           y = mpg, size = hp)) +  
  geom_point()
```



## Outside aes()

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point(size = 5)
```



***LABELS: labs()***

# ***USING `labs()` FOR PROPER LABELLING***

The `labs()` function labels plot elements, including:

- `title`: plot title
- `subtitle`: plot subtitle
- `caption`: plot caption (bottom-right)
- `tag`: tag label (top-left)
- `alt`: alt-text for the plot
- Any aesthetic variable mappings:
  - `x`: x-axis label
  - `y`: y-axis label
  - `color`: if there's a color mapped, the color legend's title
  - etc.

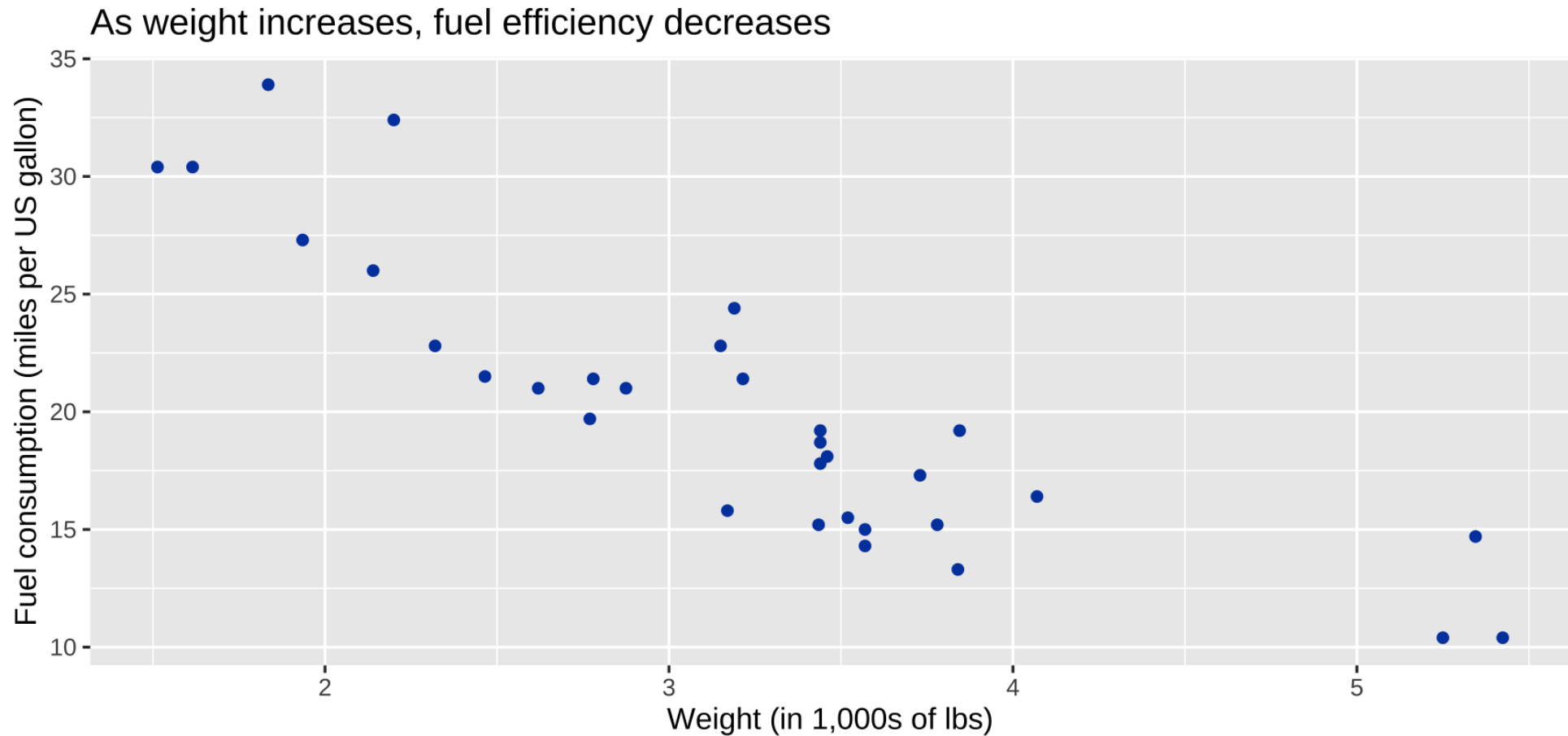
# ***HOW TO LABEL A DATA VIZ***

- Axes should be clearly labeled with units where appropriate:
  - "Fuel Consumption in Miles per Gallon" not "mpg"
  - "Age (in years)" not "age"
- Legends and their values should also be clearly labeled:
  - "Transmission type", "automatic", "manual" not "am", "0", "1"
  - "Country", "Scotland", "England", "Northern Ireland", "Wales" not "ctry", "SCT", "ENG", "NIR", "WLS"
- The title should clearly convey the message of the plot
- When necessary, provide your data source in the caption.

# EXAMPLE 1

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point() +  
  labs(x = "Weight (in 1,000s of lbs)",  
       y = "Fuel consumption (miles per US gallon)",  
       title = "As weight increases, fuel efficiency decreases",  
       caption = "Data from the datasets R package")
```

# EXAMPLE 1 RESULT

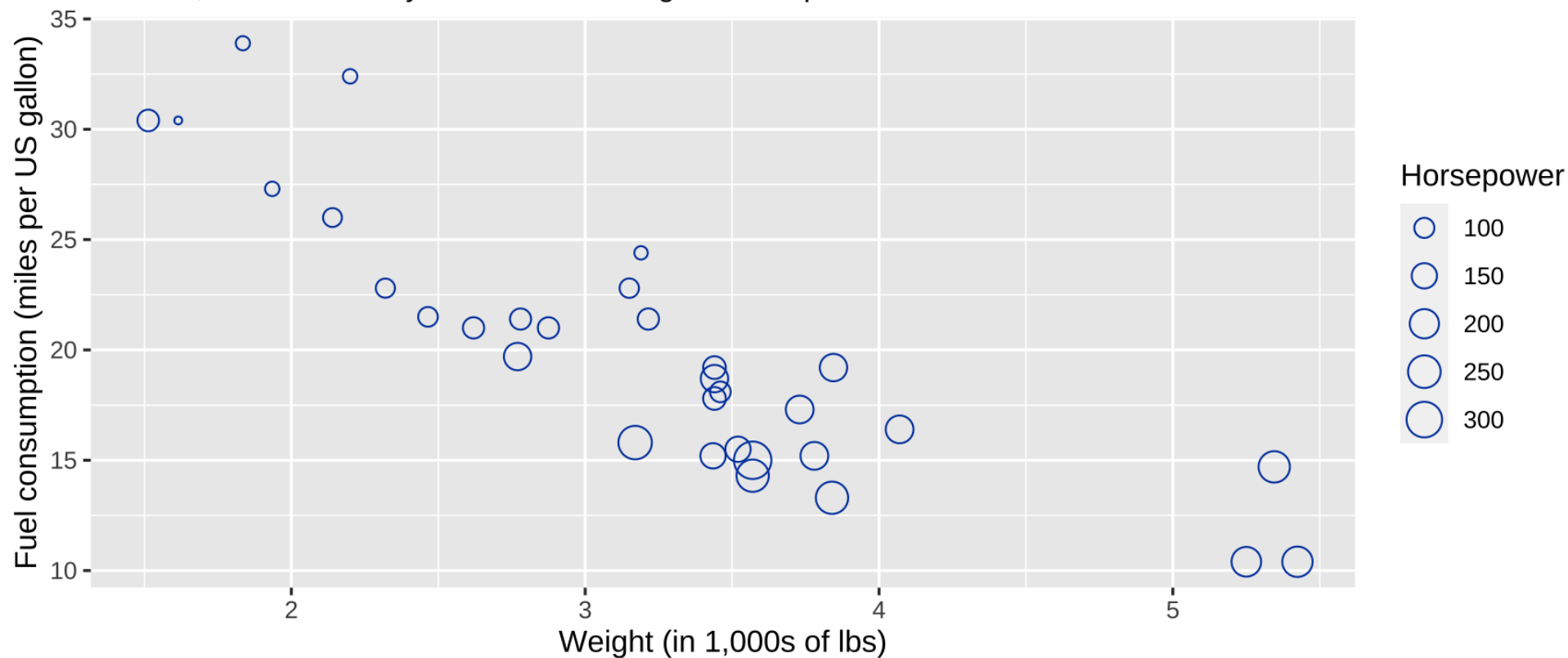


## EXAMPLE 2

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point(aes(size = hp), shape = 21) +  
  labs(x = "Weight (in 1,000s of lbs)",  
       y = "Fuel consumption (miles per US gallon)",  
       size = "Horsepower",  
       title = "As weight increases, fuel efficiency decreases",  
       subtitle = "Heavier, lower efficiency cars also have higher horsepower",  
       caption = "Data from the datasets R package")
```

# EXAMPLE 2 RESULT

As weight increases, fuel efficiency decreases  
Heavier, lower efficiency cars also have higher horsepower



Data from the datasets R package



# ***APPEARANCE: `scale_()` FUNCTIONS***

# ***USING `scale_()` FUNCTIONS FOR STYLING***

The `scale_()` family of functions is used to modify how the aesthetics are drawn:

- Change how axes are labeled
- Change color scheme
- Change axis limits
- Transform axis (e.g. log-transform)
- Change the number of breaks / labels

# ***USING `scale_()` FUNCTIONS FOR STYLING***

The `scale_()` family of functions is used to modify how the aesthetics are drawn:

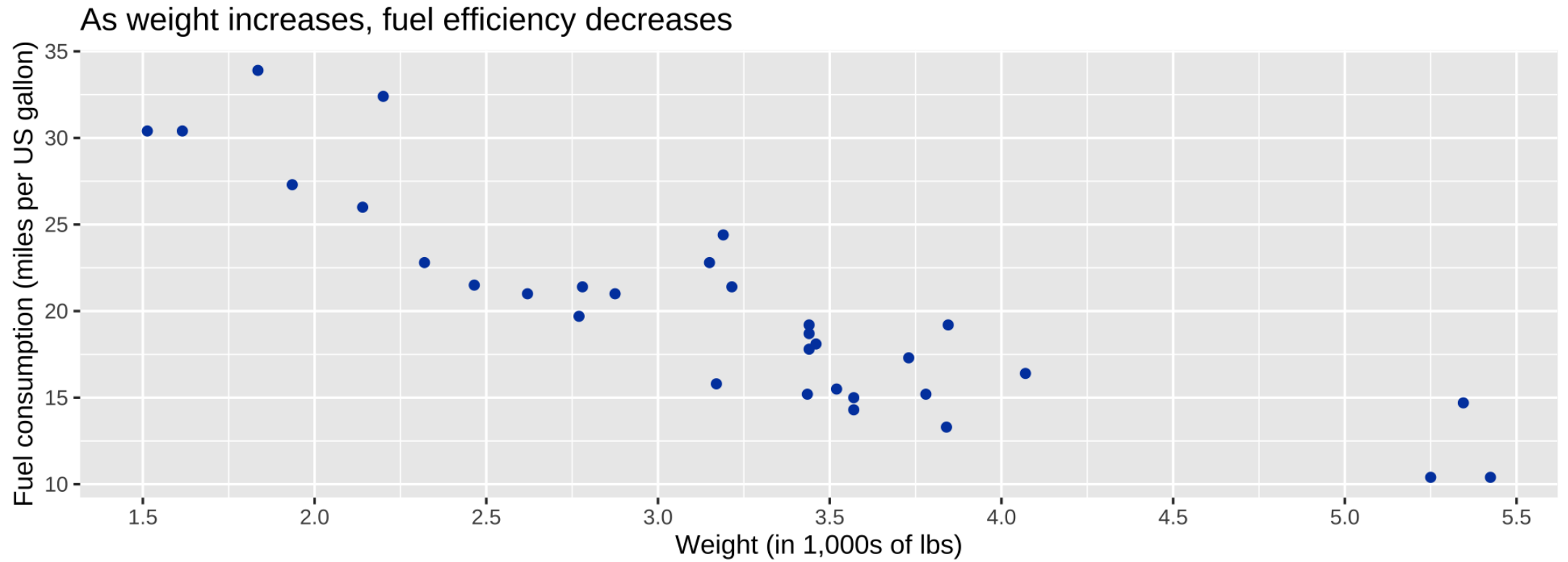
- `scale_x_*`() functions modify the x-axis
- `scale_y_*`() functions modify the y-axis
- `scale_color_*`() functions modify the color `aes()` mapping
- `scale_fill_*`() functions modify the fill `aes()` mapping
- `scale_shape_*`() functions modify the shape `aes()` mapping
- etc.

# EXAMPLE 1

Using a `scale_*()` function to change the axes:

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point() +  
  labs(y = "Fuel consumption (miles per US gallon)",  
       title = "As weight increases, fuel efficiency decreases",  
       caption = "Data from the datasets R package") +  
  scale_x_continuous(name = "Weight (in 1,000s of lbs)",  
                    breaks = seq(1.5, 5.5, .5))
```

# EXAMPLE 1 RESULT



Data from the datasets R package

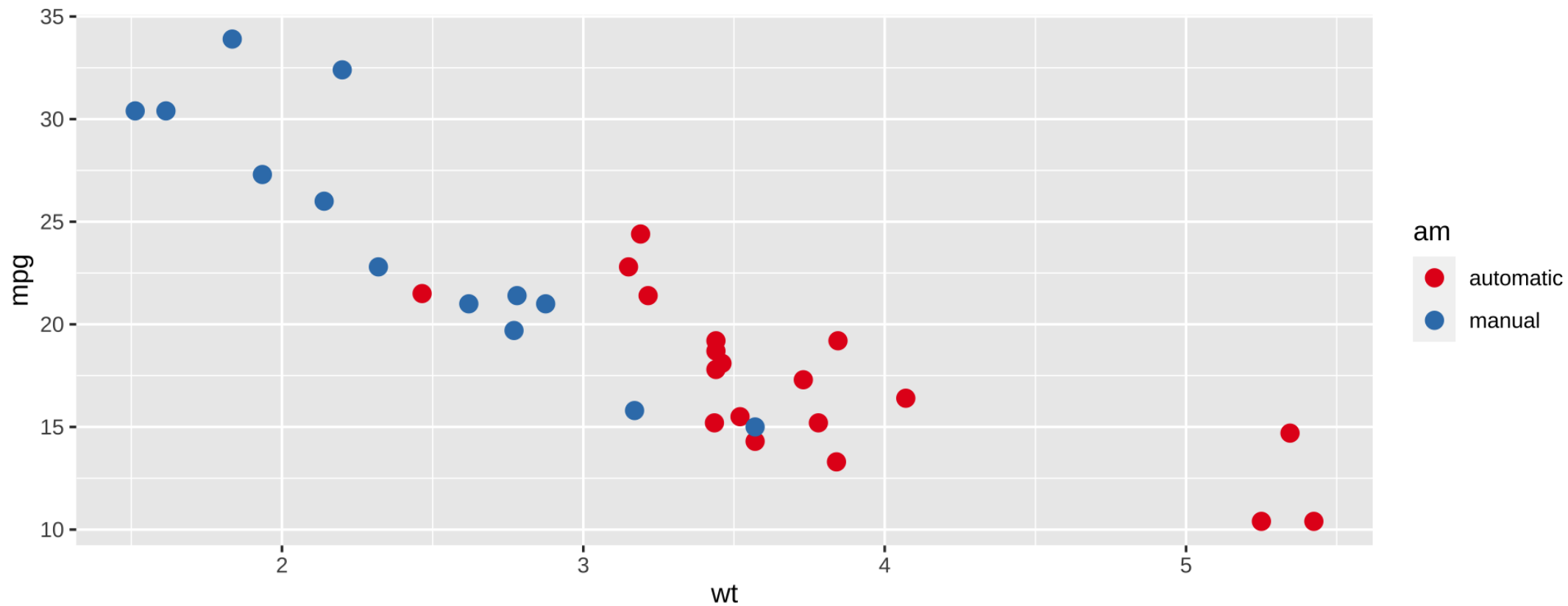
## EXAMPLE 2

Using a `scale_()` function to change color scheme

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point(aes(color = am)) +  
  scale_color_brewer(palette = "Set1")
```

The website [colorbrewer2.org](https://colorbrewer2.org) has a lot of colors & color palettes to choose from

## EXAMPLE 2 RESULT

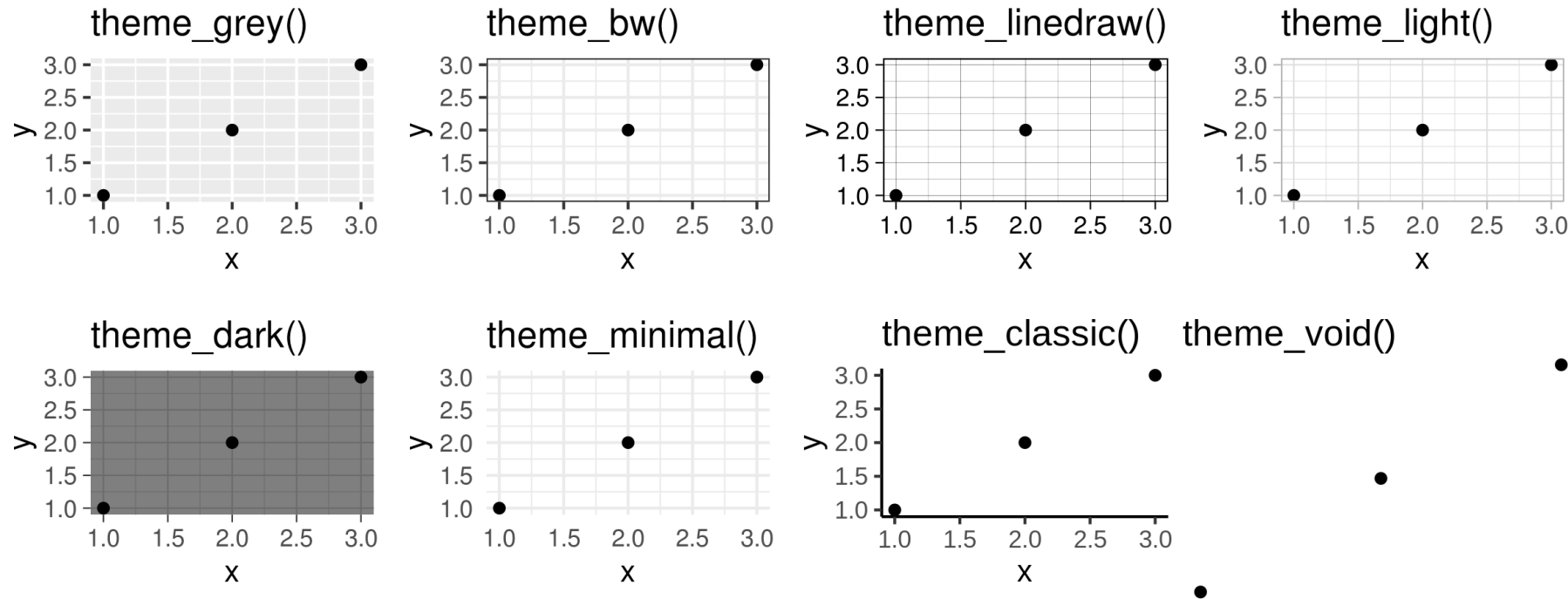


***APPEARANCE: theme()***



# USING theme FUNCTIONS FOR STYLING

ggplot2 has several built-in theme functions. The default is `theme_grey()`:



# ***USING theme FUNCTIONS FOR STYLING***

The `theme()` function gives you control over every single non-data visual element of the plot:

- fonts used for labels
- background color
- axis text positioning, size, etc.
- legend appearance, position, etc.

At time of writing, `theme()` takes 92 arguments! You'll never use them all. I typically use only 2-3, and rarely use more than 10.

# theme ( ) *ELEMENTS*

There are 4 functions used to modify the theme:

- `element_rect()`: Change appearance of a rectangle element
- `element_line()`: Change appearance of a line element
- `element_text()`: Change appearance of a text element
- `element_blank()`: Do not draw an element

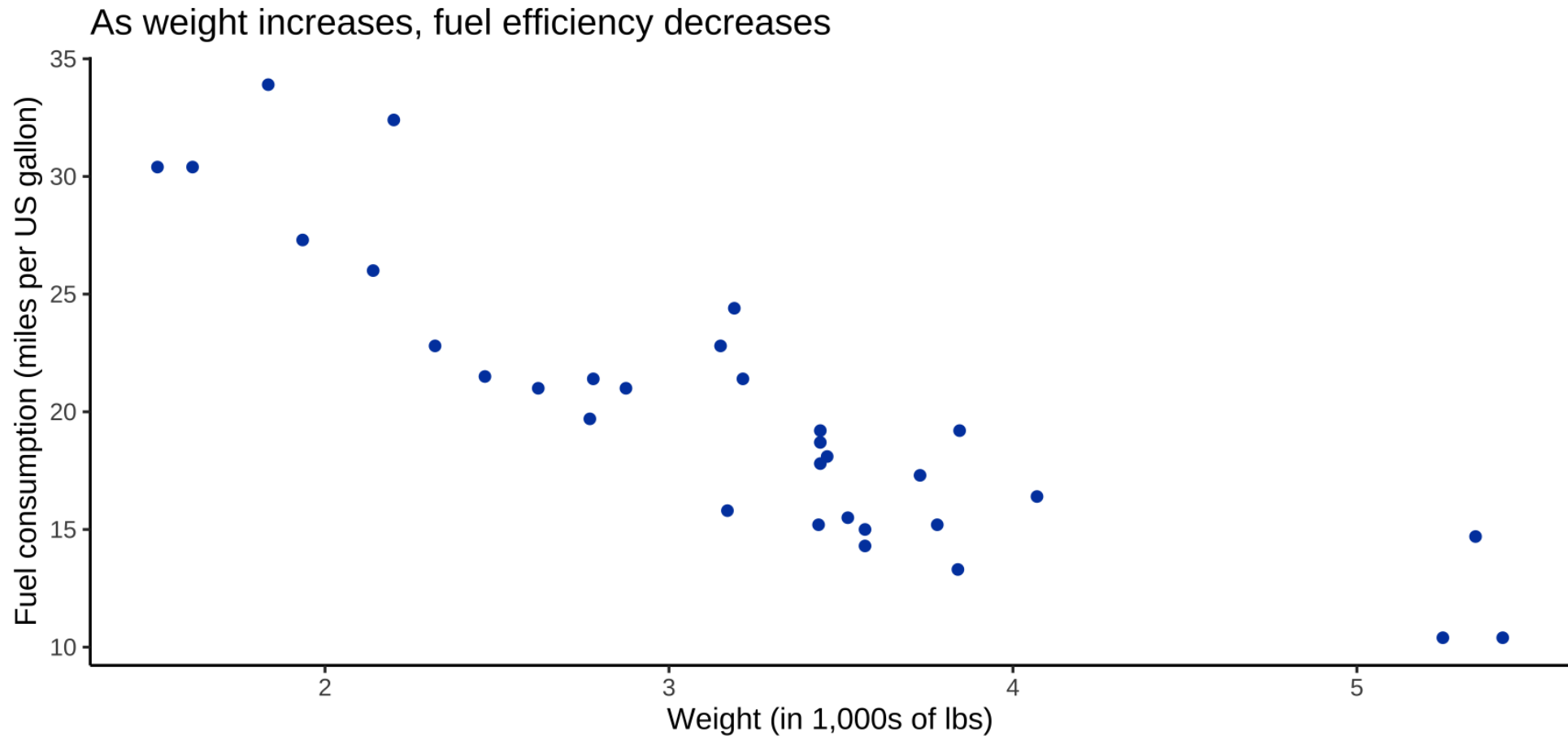
The documentation of the `theme()` function tells you when to use each function

# EXAMPLE 1

Using a `theme_*()` function:

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point() +  
  labs(x = "Weight (in 1,000s of lbs)",  
       y = "Fuel consumption (miles per US gallon)",  
       title = "As weight increases, fuel efficiency decreases",  
       caption = "Data from the datasets R package") +  
  theme_classic()
```

# EXAMPLE 1 RESULT



## EXAMPLE 2

Using the `theme()` function:

```
ggplot(data = mtcars,  
       aes(x = wt, y = mpg)) +  
  geom_point(aes(size = hp), shape = 21) +  
  labs(x = "Weight (in 1,000s of lbs)",  
       y = "Fuel consumption (miles per US gallon)",  
       size = "Horsepower",  
       title = "As weight increases, fuel efficiency decreases",  
       subtitle = "Heavier, lower efficiency cars also have higher horsepower",  
       caption = "Data from the datasets R package") +  
  theme(axis.title.y = element_text(angle = 0),  
        plot.title.position = "plot",  
        panel.grid = element_line(color = "red"),  
        panel.background = element_rect(fill = "white"),  
        panel.grid.minor = element_blank())
```

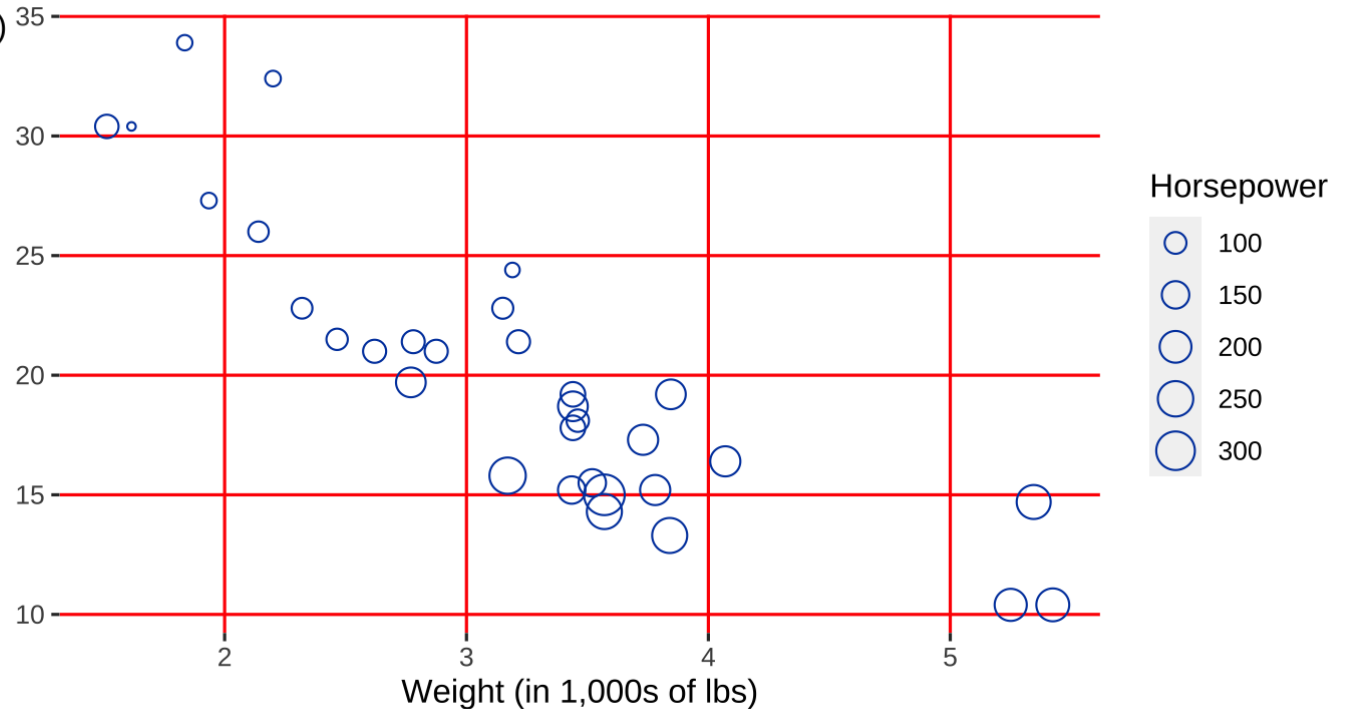
# EXAMPLE 2 RESULT

Using the `theme()` function:

As weight increases, fuel efficiency decreases

Heavier, lower efficiency cars also have higher horsepower

Fuel consumption (miles per US gallon)



Data from the datasets R package

# ***ggplot2 EXTRAS***



# ggplot2 EXTENSIONS

Maintainers of packages can put their `ggplot2` extension on [exts.ggplot2.tidyverse.org/gallery](https://exts.ggplot2.tidyverse.org/gallery)

ggplot2 extensions - gallery

Add Your Extension! [exts.ggplot2.tidyverse.org](https://exts.ggplot2.tidyverse.org)

66 registered extensions available to explore

Sort  
Github stars

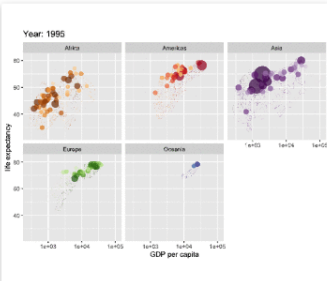
Text Filter  
search name, author, describe

Author Filter

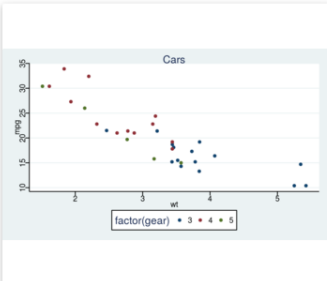
Tag Filter

CRAN Only


Showing 53 of 66



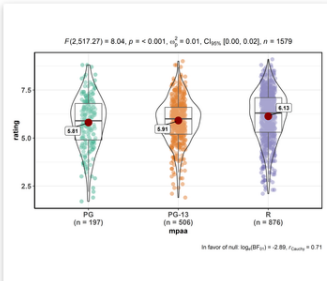
**gganimate** Star 1488  
A Grammar of Animated Graphics.  
■ author: thomasp85  
■ tags: visualization, general  
■ js libraries:



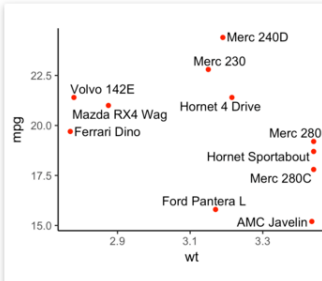
**ggthemes** Star 1039  
Some extra geoms, scales, and themes for ggplot.  
■ author: jrnold  
■ tags: visualization, general, themes  
■ js libraries:



**esquisse** Star 882  
Explore and Visualize Your Data Interactively with ggplot2  
■ author: dreamrs  
■ tags: visualization, interface  
■ js libraries:



**ggstatsplot** Star 781  
'ggstatsplot' provides a collection of functions to enhance 'ggplot2' plots with results from statistical tests.  
■ author: IndrajeetPatil  
■ tags: visualization, statistics  
■ js libraries:



**ggrepel** Star 714  
Repel overlapping text labels away from each other.  
■ author: slowkow  
■ tags: visualization, general  
■ js libraries:

# EXAMPLES OF EXTENSIONS

## Domain specific:

- Networks:
  - `geomnet`
  - `ggraph`
- Time Series:
  - `ggaluvial`
  - `sugrrants`
- Sciences:
  - `gggenes`
  - `ggtree`
  - `ggseqlogo`
  - `ggspectra`

## Appearance customization:

- Arrange ggplots:
  - `cowplot`
  - `patchwork`
  - `gganimate`
- Custom themes and/or scales:
  - `ggthemes`
  - `ggsci`
  - `ggtech`
  - `ggthemr`
  - `xkcd`
  - `ggpubr`

# ADDITIONAL RESOURCES

- **ggplot2 book**: 3rd edition freely available online! This book provides much more detail and covers advanced topics I didn't cover here, such as **stats**.
- **R for Data Science book**: Freely available online. Check out Chapter 3 on Data Visualization
- **Tidy Tuesday**: Tidy Tuesday is an online learning community. There are many participants worldwide! Use #TidyTuesday on Twitter to find graphics inspiration.
- **My advice for getting help in R**: my top 10 tips for getting help in R when you get stuck.
- Thomas Lin Pedersen's ggplot2 webinar: **part 1** and **part 2**:
- **RStudio Cheat Sheets**
- **Will Chase's Design Talk** at rstudio::conf
- A **4-hour advanced version** of this tutorial is available on RStudio Cloud.
- A **2-hour version** of this tutorial is available on my website.

# THANKS!

## Contact me:

 [sctyner.me](https://sctyner.me)

 [@sctyner](https://twitter.com/sctyner)

 [sctyner](https://github.com/sctyner)

 [/in/sctyner](https://www.linkedin.com/in/sctyner)



Slides created via the R packages: [xaringan](#) and [xaringanthemer](#)