Before we start...

Access the RStudio Cloud Project (Link in https://datafest.psu.edu/go/)

- Workshop - Introduction to Data Visualization in R with ggplot2

OR

1. Download R
2. Download RStudio
3. Install packages `tidyverse` or `ggplot2` (check with library(ggplot2))

Concept of data analysis and maintenance Illustration by WOOBRO DESIGN on Iconscout

Introduction to Data Visualization in R with ggplot2

Penn State, University Libraries, Research Informatics and Publishing

Workshop Housekeeping

Questions?

Use the Q&A, chat, or raise hand feature.

Feedback Survey

After the workshop, please fill out the Qualtrics survey (link in chat).

Credits

Images and content sourced/based on:

ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham

Introduction to data visualization with ggplot2, Data Camp - Rick Scavetta

About Me

- Master’s in Applied Statistics
- Bachelor’s in Computational Statistics

R Experience:

- Self-taught for research (2017-Present)
- Statistics Courses (STAT 184/380)
- Other Projects

Research Consultant, University Libraries

David Chen
dzc89@psu.edu
Research Consultant
github.com/TheDavidChen

Introduce Yourself!

Name, Major, Year,

How do you pronounce the word: Data

General Comments

If you see: > print(“Hello World”)

Run print(“Hello World”) in R Script. Do not include the `>`.

Use `#` to write comments - code after # is not run.

```
> # This is not run
```

Motivation

What is the purpose of Data Visualization?

Agenda

- Base R vs ggplot2
- The logic of ggplot2
- Basic plots
- Data viz cheat sheet
- Faceting
- Themes
- Exporting plots
- The little things

Base R vs ggplot2

Base R you plot based on specific functions

- hist()
- boxplot()
- barplot()
- pie()
- plot()

Base R vs ggplot2

ggplot2 is based on the Grammar of Graphics

Thinks about visualizations as layers/components

Greater flexibility, more personalization, more intricate plots, while simple plots are still simple

ggplot2 layers

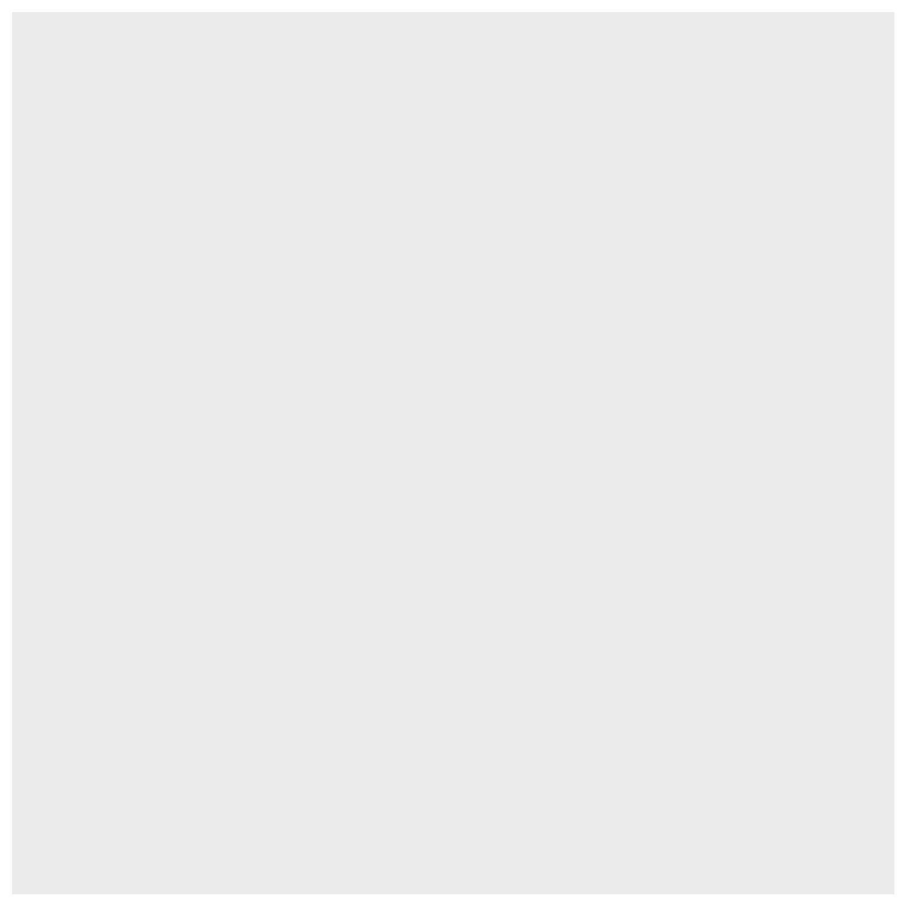
mpg dataset from ggplot2 - “Fuel economy data from 1999 to 2008 for 38 popular models of cars”

Variables

- displ - engine displacement (size)
- year - year of manufacture
- cty - city miles per gallon
- hwy - highway miles per gallon
- class - type of car (suv, pickup, etc.)

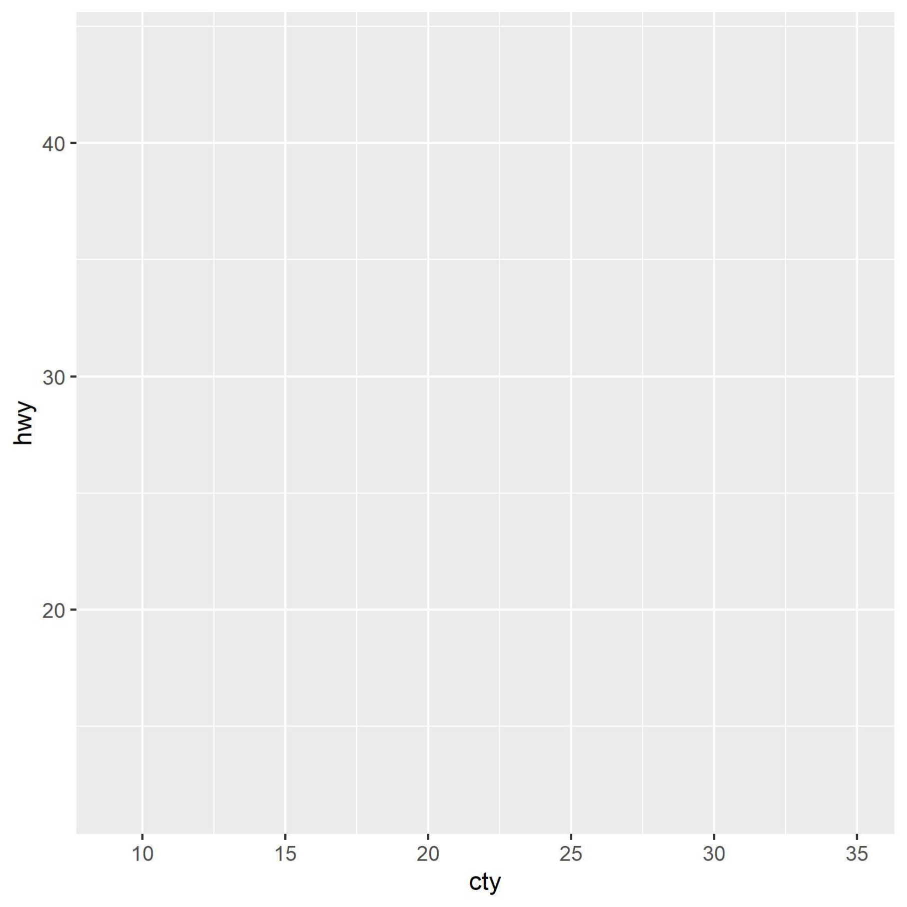
ggplot2 layers

```
ggplot(mpg)
```



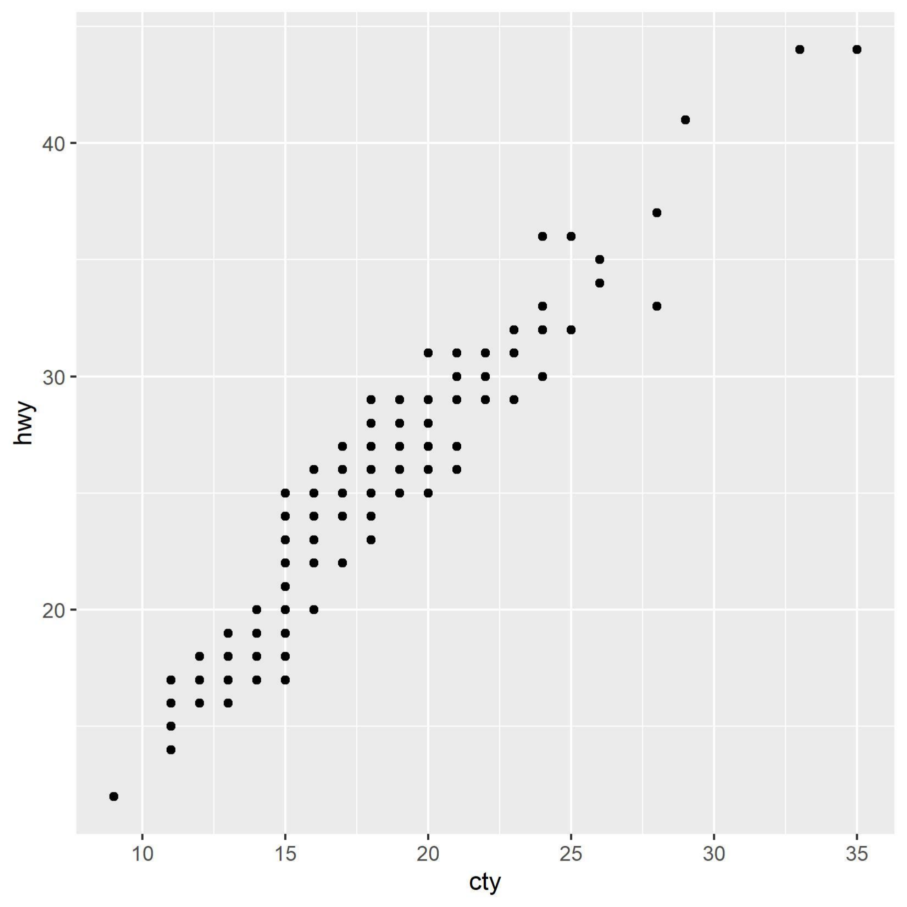
ggplot2 layers

```
ggplot(mpg, aes(x = cty, y = hwy))
```



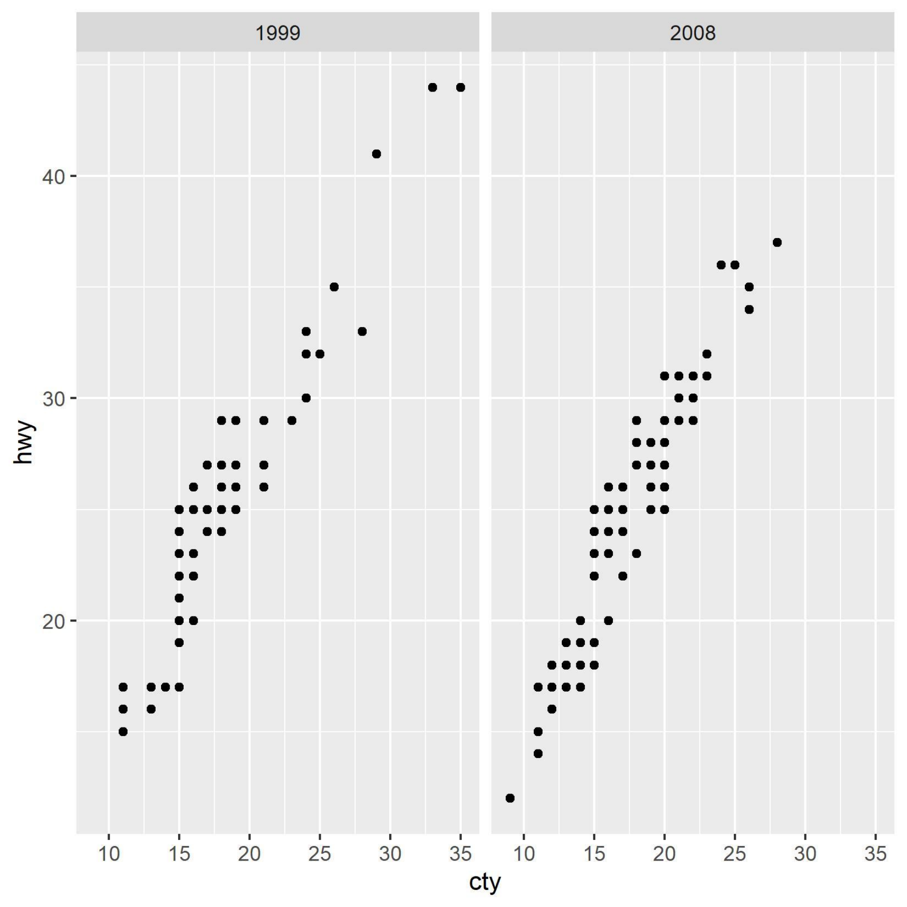
ggplot2 layers

```
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point()
```



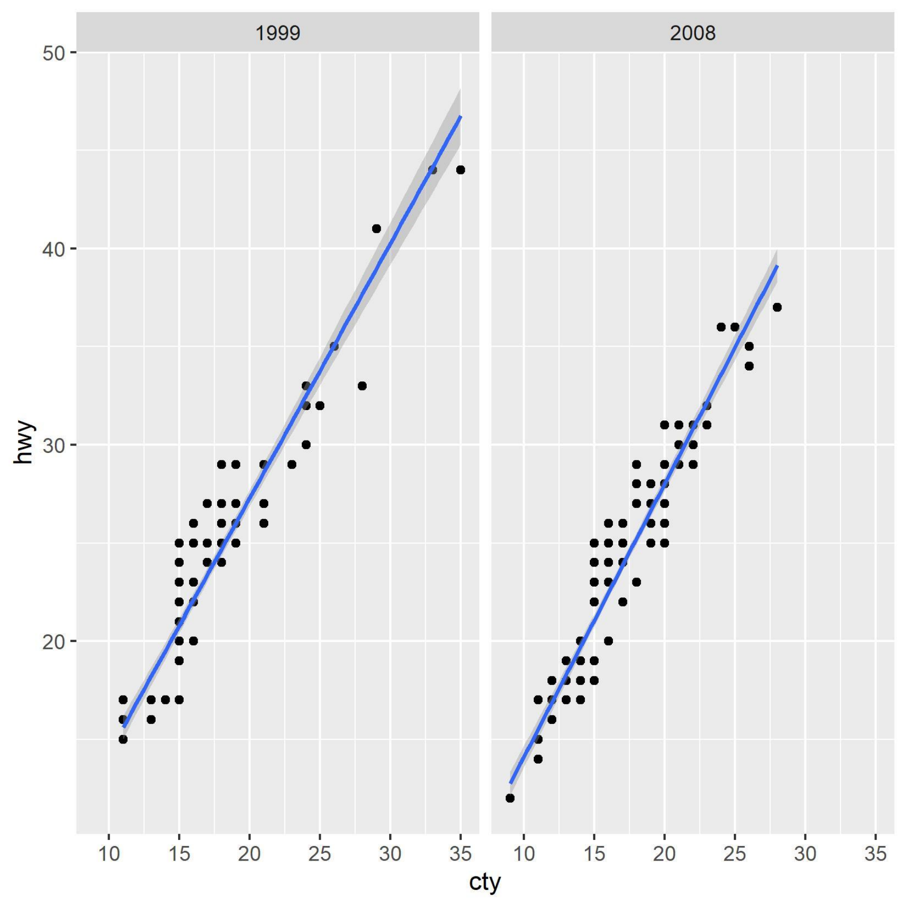
ggplot2 layers

```
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  facet_wrap(~year)
```



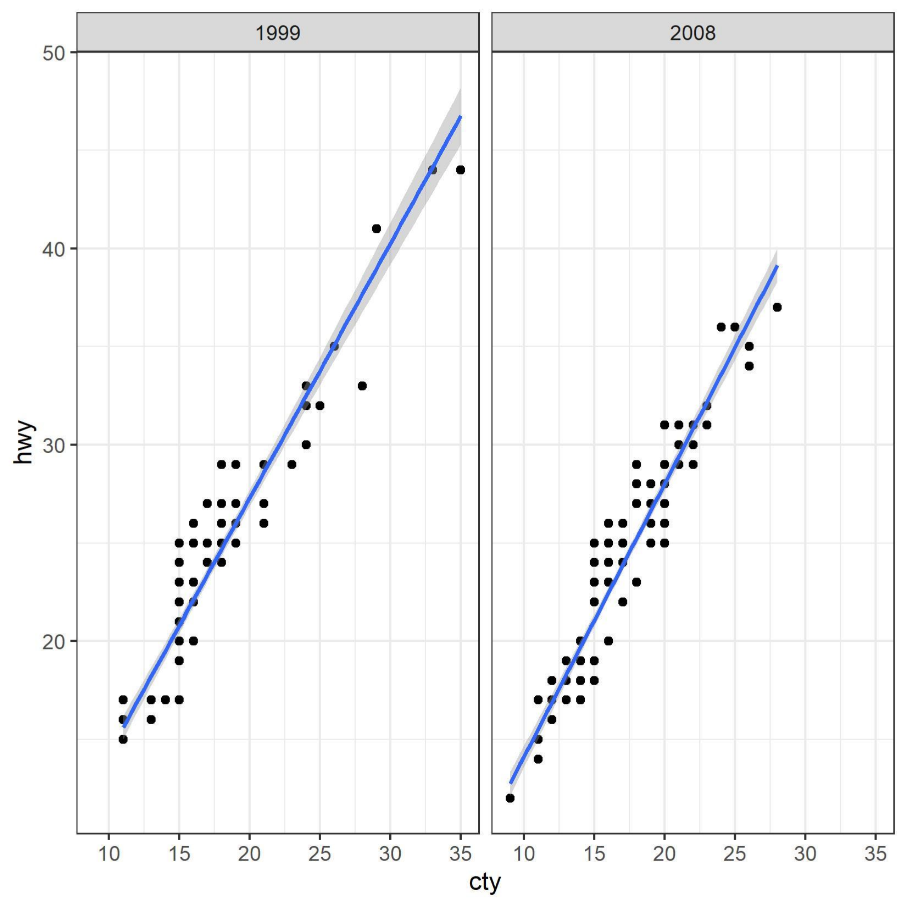
ggplot2 layers

```
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  facet_wrap(~year) +
  geom_smooth(method = "lm")
```



ggplot2 layers

```
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  facet_wrap(~year) +
  geom_smooth(method = "lm") +
  theme_bw()
```



Based on: Introduction to data visualization with ggplot2, Data Camp - Rick Scavetta
https://rpubs.com/odenipinedo/introduction-to-data-visualization-with-ggplot2

ggplot2 layers

Data	**What dataset**
Aesthetics	**Select x and y-axis, colors, shapes, etc.**
Geometries	**Which geometric object (points, bars, lines)**

Based on: Introduction to data visualization with ggplot2, Data Camp - Rick Scavetta
https://rpubs.com/odenipinedo/introduction-to-data-visualization-with-ggplot2

ggplot2 layers

Data	**What dataset**
Aesthetics	**Select x and y-axis, colors, shapes, etc.**
Geometries	**Which geometric object (points, bars, lines)**
Facets	Plot subsets of the data separately

Based on: Introduction to data visualization with ggplot2, Data Camp - Rick Scavetta
https://rpubs.com/odenipinedo/introduction-to-data-visualization-with-ggplot2

ggplot2 layers

Data	**What dataset**
Aesthetics	**Select x and y-axis, colors, shapes, etc.**
Geometries	**Which geometric object (points, bars, lines)**
Facets	Plot subsets of the data separately
Statistics	Statistical transformations

Based on: Introduction to data visualization with ggplot2, Data Camp - Rick Scavetta
https://rpubs.com/odenipinedo/introduction-to-data-visualization-with-ggplot2

ggplot2 layers

Data	**What dataset**
Aesthetics	**Select x and y-axis, colors, shapes, etc.**
Geometries	**Which geometric object (points, bars, lines)**
Facets	Plot subsets of the data separately
Statistics	Statistical transformations
Coordinates	Coordinate system (Cartesian, polar, map)

Based on: Introduction to data visualization with ggplot2, Data Camp - Rick Scavetta
https://rpubs.com/odenipinedo/introduction-to-data-visualization-with-ggplot2

ggplot2 layers

Data	**What dataset**
Aesthetics	**Select x and y-axis, colors, shapes, etc.**
Geometries	**Which geometric object (points, bars, lines)**
Facets	Plot subsets of the data separately
Statistics	Statistical transformations
Coordinates	Coordinate system (Cartesian, polar, map)
Themes	Font size, background color, etc.

Breaking down ggplot2

Before we start plotting

Make sure to load tidyverse (or just ggplot2)

```
> library(tidyverse)
```

We will load datasets like mpg by doing the following:

```
> data(mpg)
```

To learn about the dataset, type:

```
> ?mpg
```

Before we start plotting

Plots will appear under the `Plots` tab in the bottom right section of RStudio

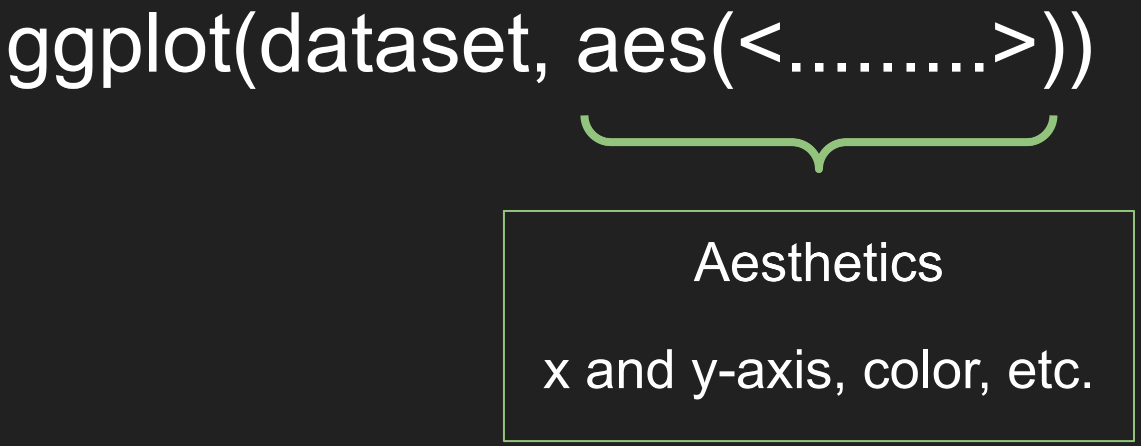
Click the arrow buttons to go between plots

Data

```
ggplot(dataset)
```

Specify dataset desired for the plot

Aesthetics



```
ggplot(dataset, aes(<.........>))
```

Aesthetics

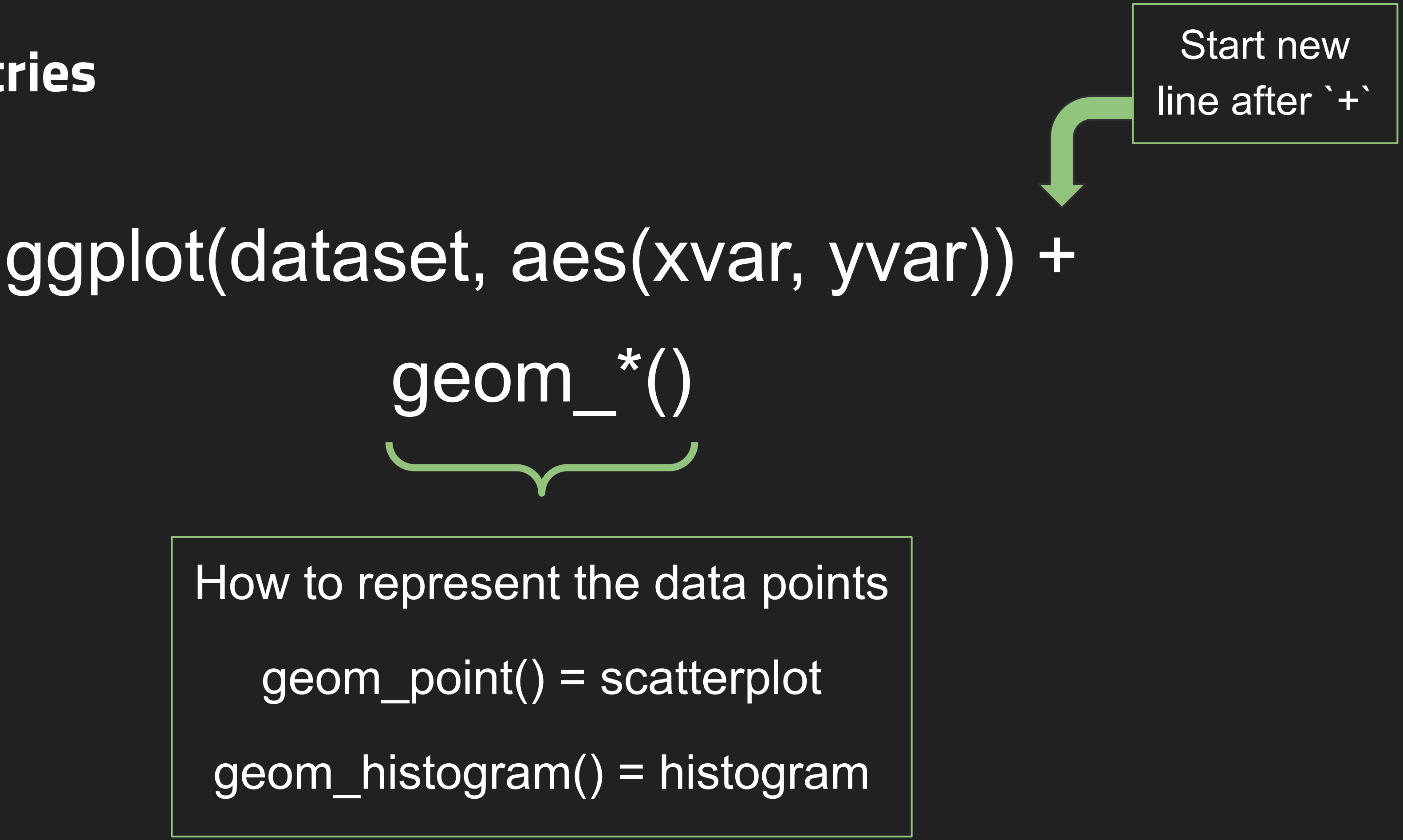
x and y-axis, color, etc.

Based on: Introduction to data visualization with ggplot2, Data Camp - Rick Scavetta
https://rpubs.com/odenipinedo/introduction-to-data-visualization-with-ggplot2

Aesthetics

Aesthetic	Description
x	X axis
y	Y axis
fill	Fill color
color	Color of points or outlines of other geoms
size	Size of points, thickness of lines
alpha	Transparency

Geometries



Start new line after `+`

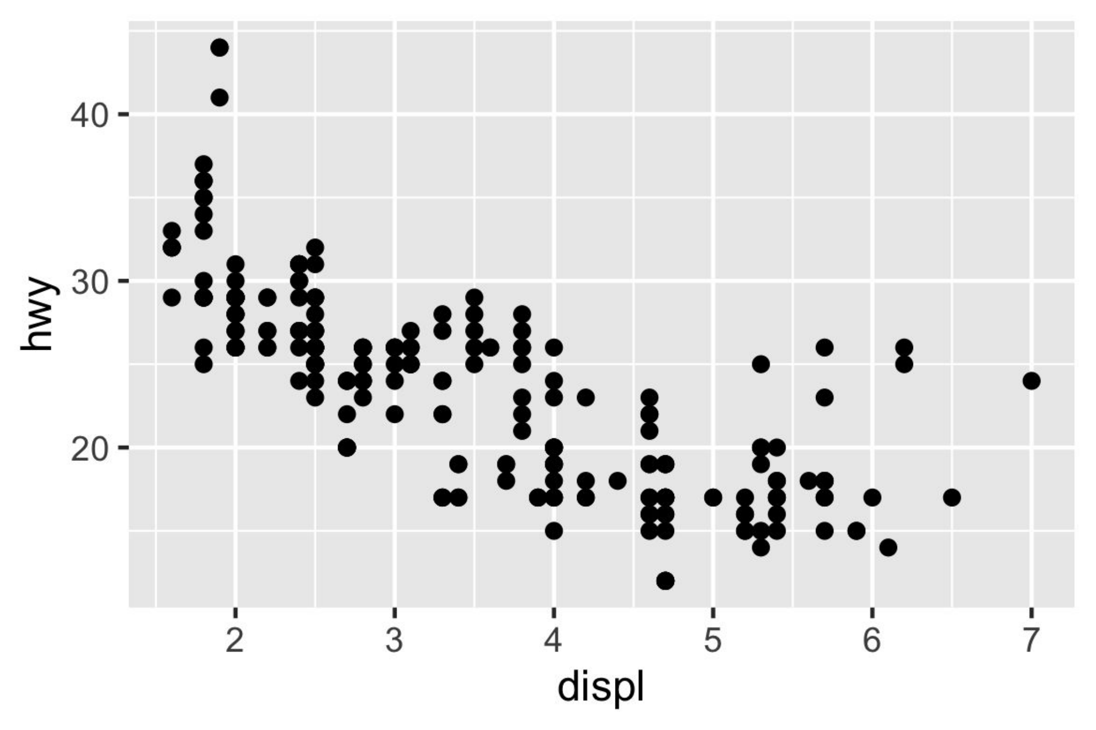
```
ggplot(dataset, aes(xvar, yvar)) +
geom_*()
```

How to represent the data points

geom_point() = scatterplot

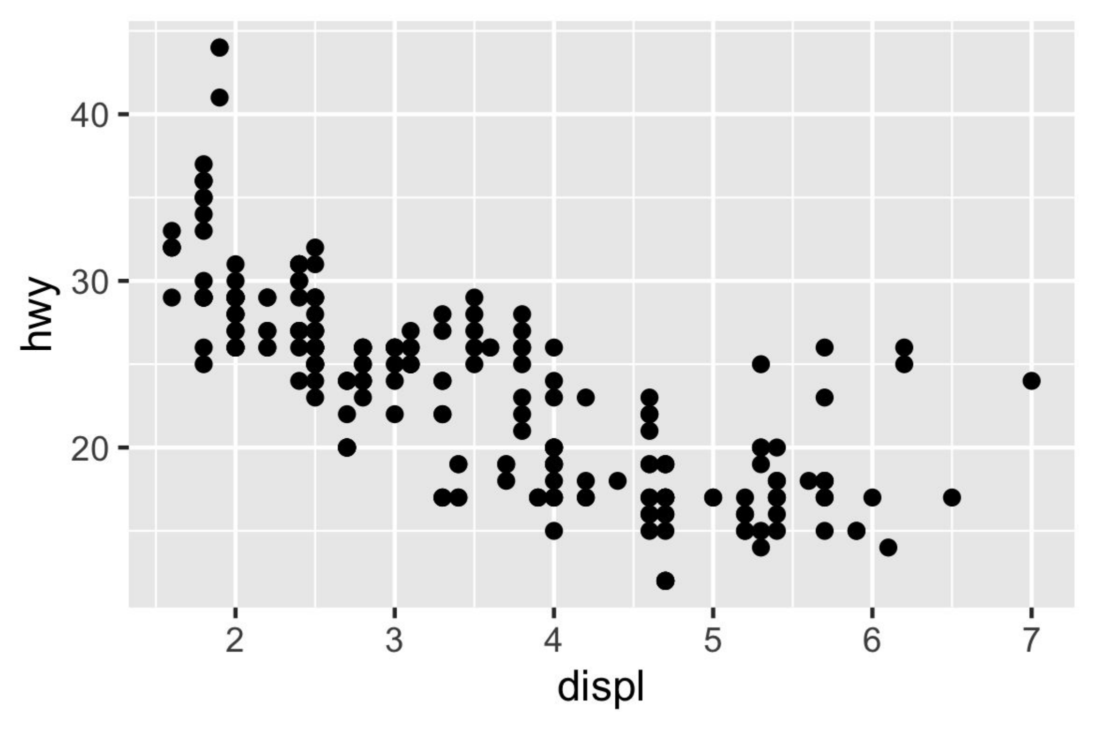
geom_histogram() = histogram

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



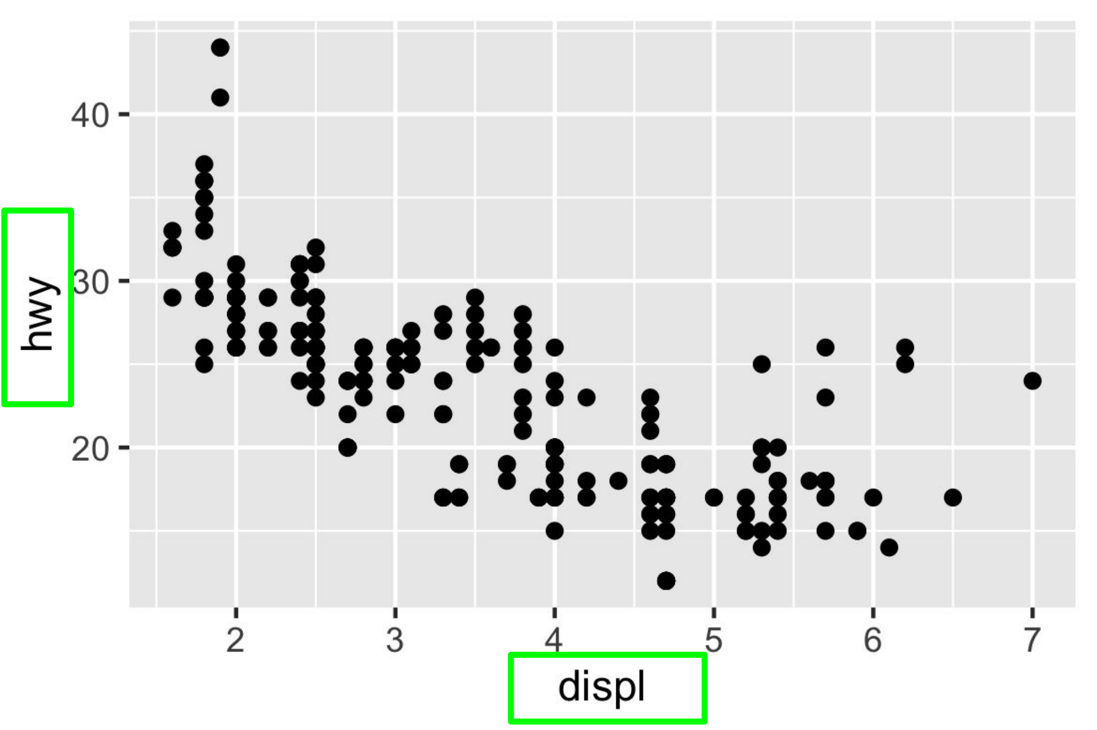
ggplot2: Elegant Graphics for Data Analysis, by Hadley Wickham
https://ggplot2-book.org/getting-started.html

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



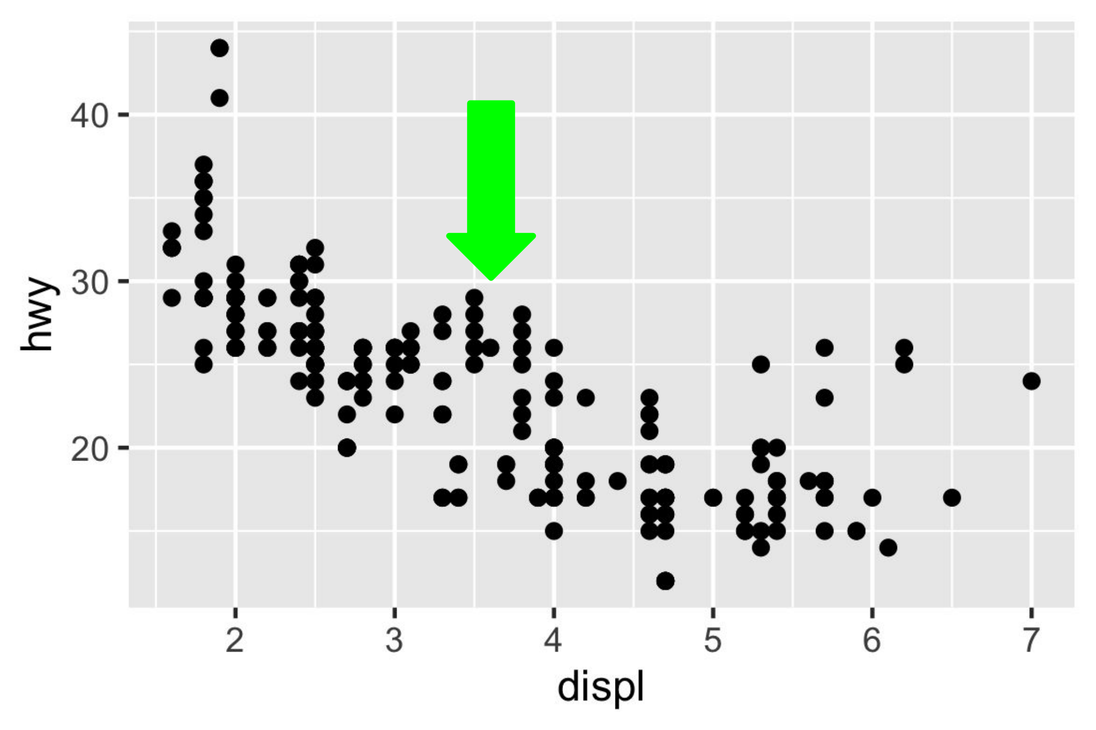
ggplot2: Elegant Graphics for Data Analysis, by Hadley Wickham
https://ggplot2-book.org/getting-started.html

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



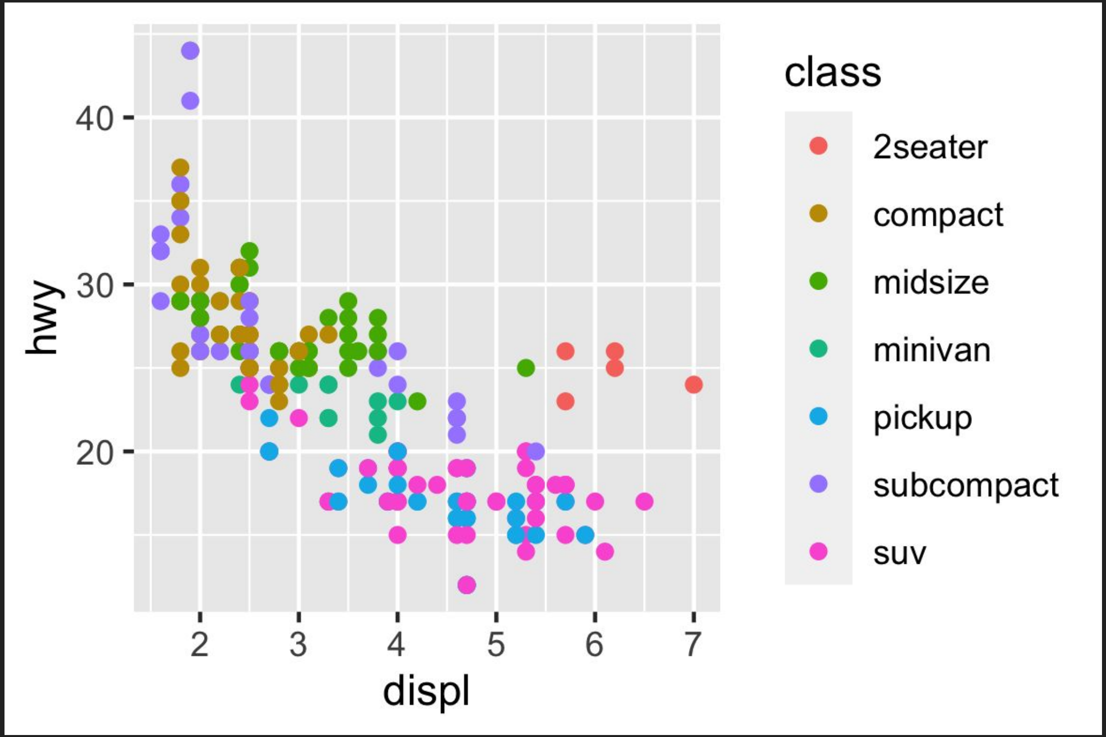
ggplot2: Elegant Graphics for Data Analysis, by Hadley Wickham
https://ggplot2-book.org/getting-started.html

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



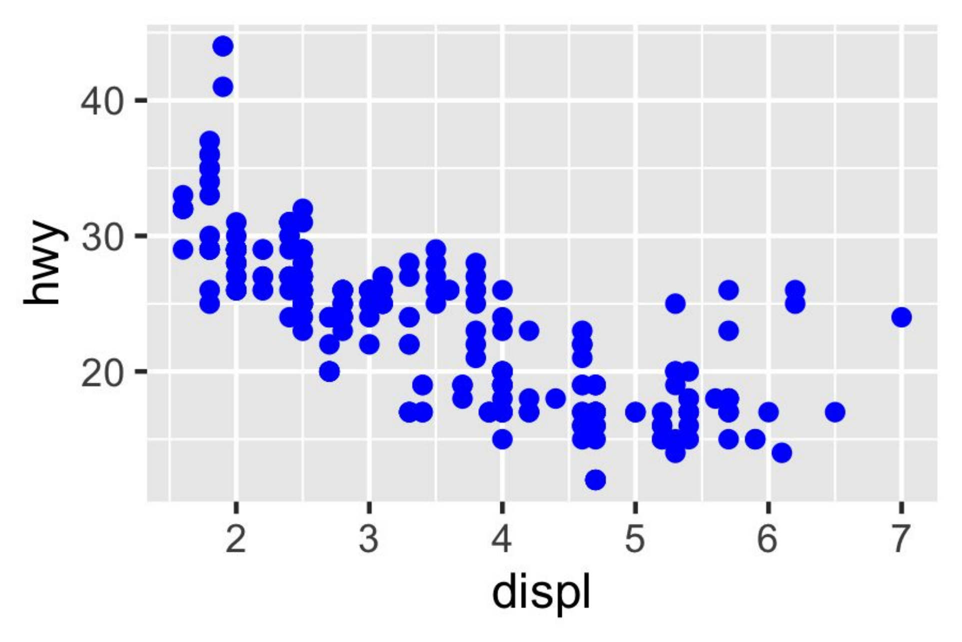
ggplot2: Elegant Graphics for Data Analysis, by Hadley Wickham
https://ggplot2-book.org/getting-started.html

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



ggplot2: Elegant Graphics for Data Analysis, by Hadley Wickham
https://ggplot2-book.org/getting-started.html

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(color = “blue”)
```



ggplot2: Elegant Graphics for Data Analysis,
by Hadley Wickham
https://ggplot2-book.org/getting-started.html

mpg Example

1. Load the mpg dataset (from the ggplot2 package)

```
> data(mpg)
```

```
> ?mpg # To learn about the variables
```

2. Experiment with different aesthetics (alpha, shape, size, color)

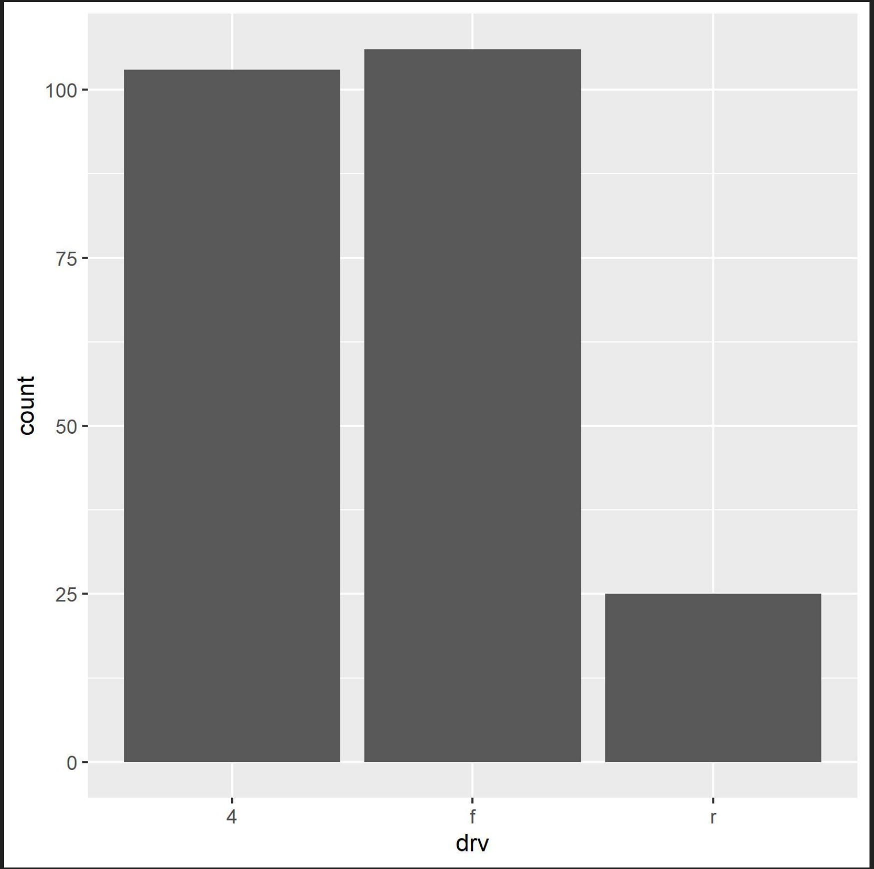
```
> ggplot(mpg, aes(x = displ, y = hwy)) +
>   geom_point(alpha = 0.2)
```

```
> ggplot(mpg, aes(x = displ, y = hwy)) +
>   geom_point(shape = 15)
```

```
> ggplot(mpg, aes(x = displ, y = hwy)) +
>   geom_point(size = 5, color = “red”)
```

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

```
ggplot(mpg, aes(x = drv)) +
  geom_bar()
```



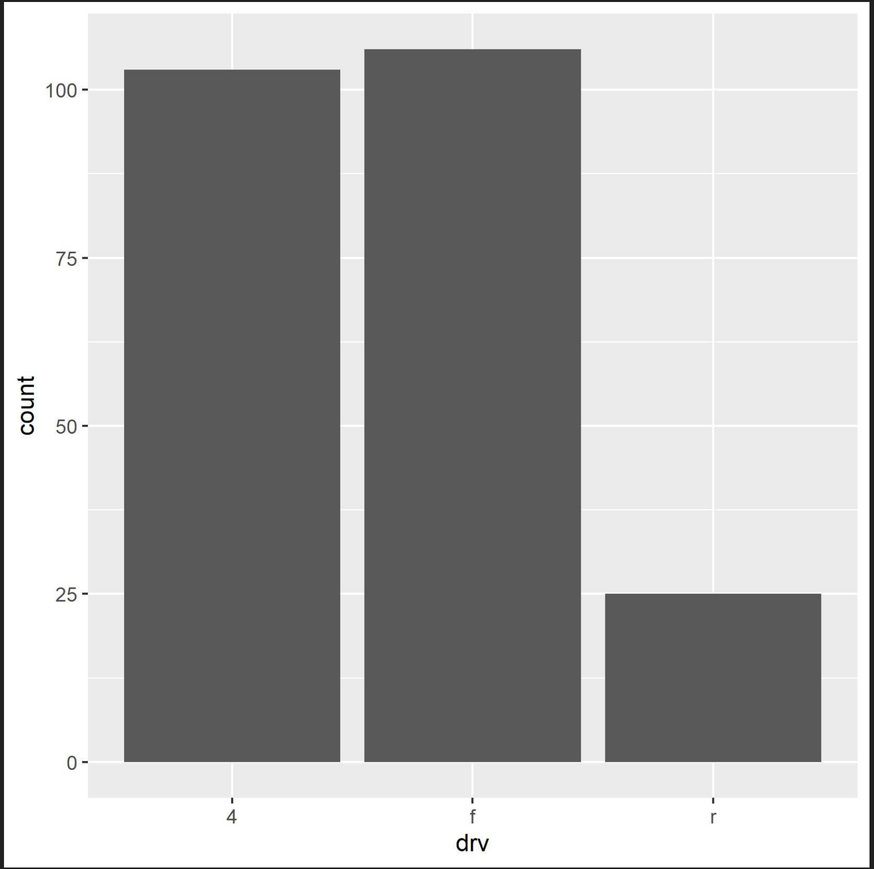
Some geometries only need one variable.

Bar Plots - One (Discrete) Variable

```
ggplot(mpg, aes(x = drv, fill = year)) +
  geom_bar()
```

Arguments may require discrete (categorical) vs continuous variables.

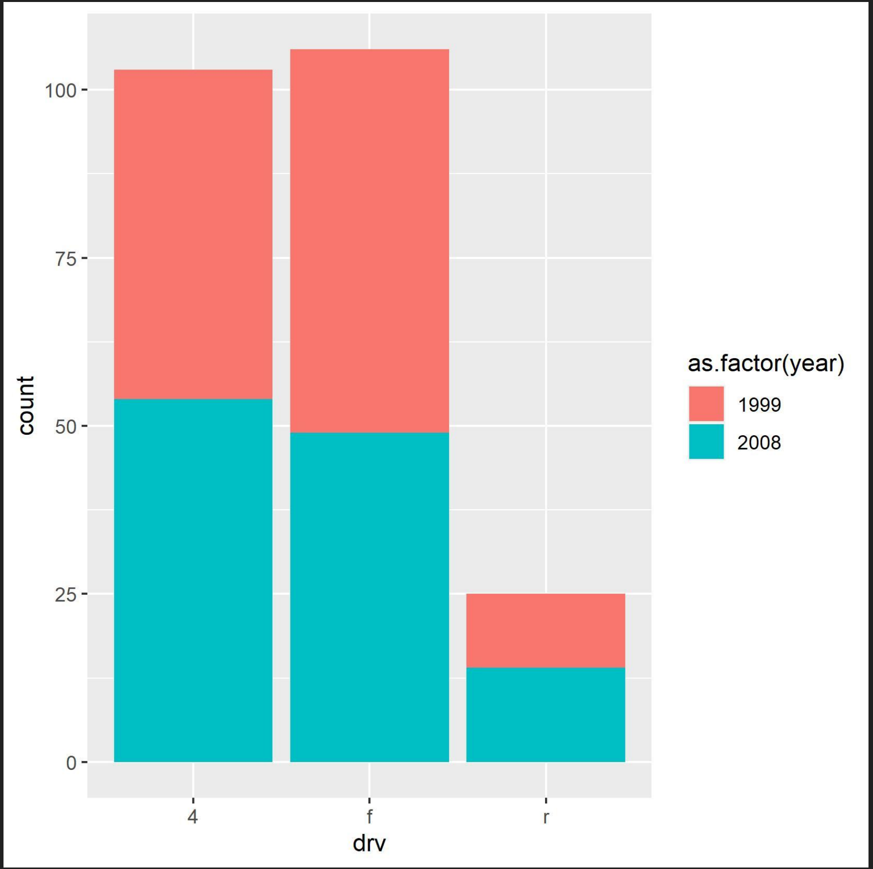
Fill must be categorical, but the data has it as continuous.



```
ggplot(mpg, aes(x = drv, fill = as.factor(year))) +
  geom_bar()
```

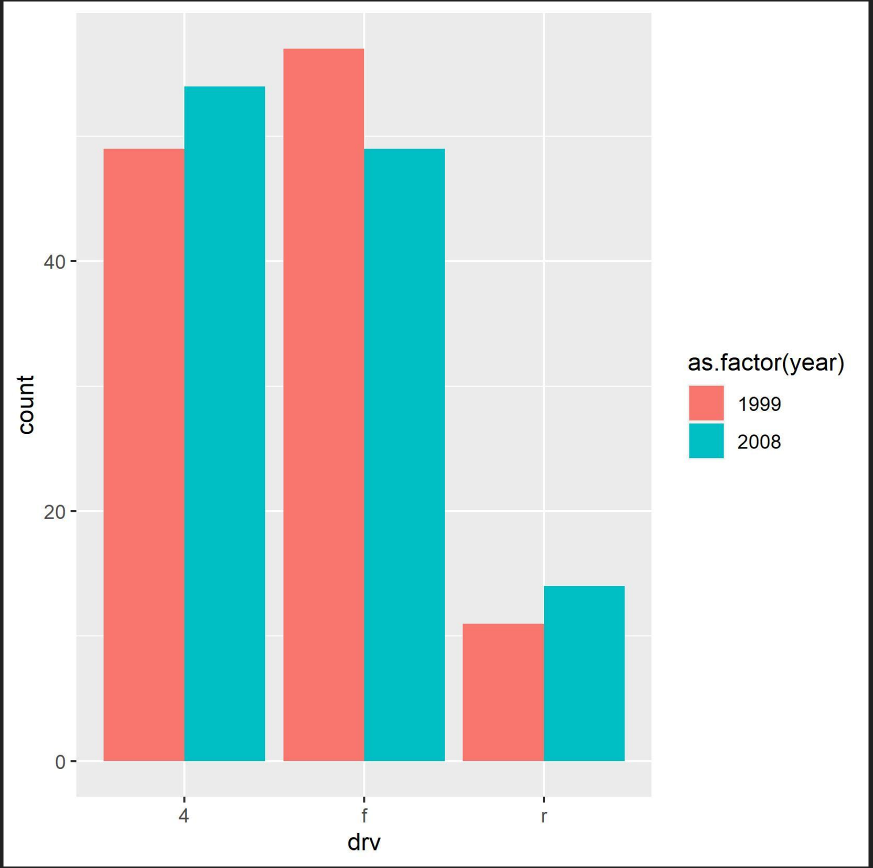
Setting the year (continuous) as a factor makes it categorical. We can now use it to fill the colors!

Note: For other plots, the color can be a continuous variable.

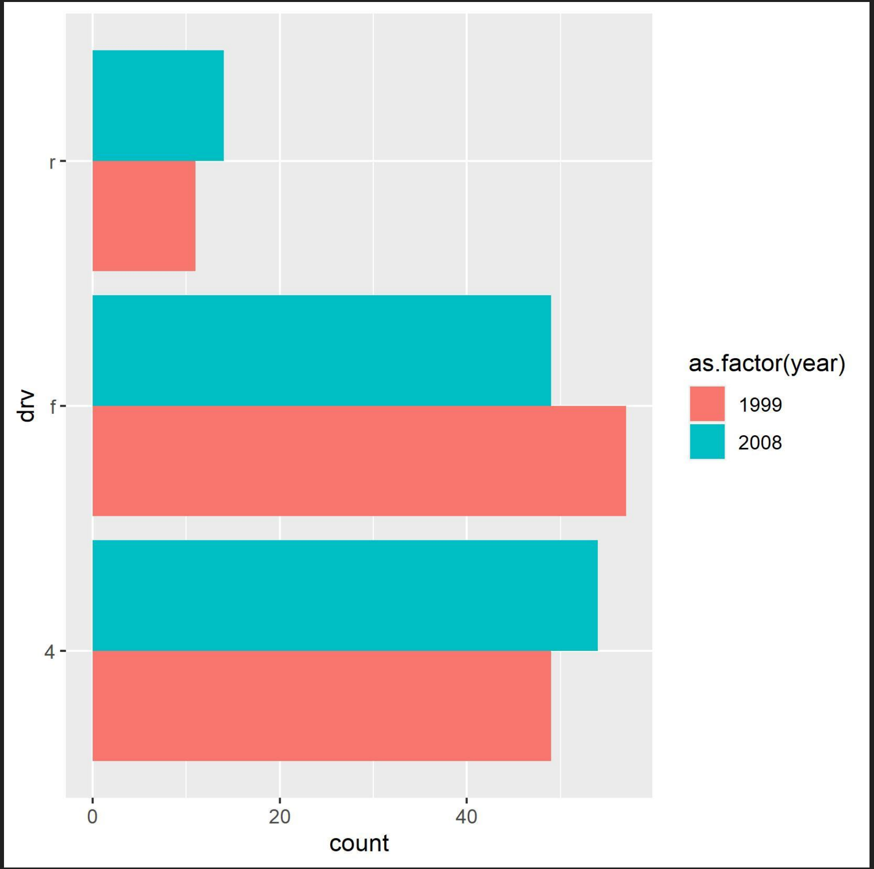


```
ggplot(mpg, aes(x = drv, fill = as.factor(year))) +
  geom_bar(position = "dodge")
```

Different `geom_*` have different arguments



```
ggplot(mpg, aes(y = drv, fill = as.factor(year))) +
  geom_bar(position = "dodge")
```



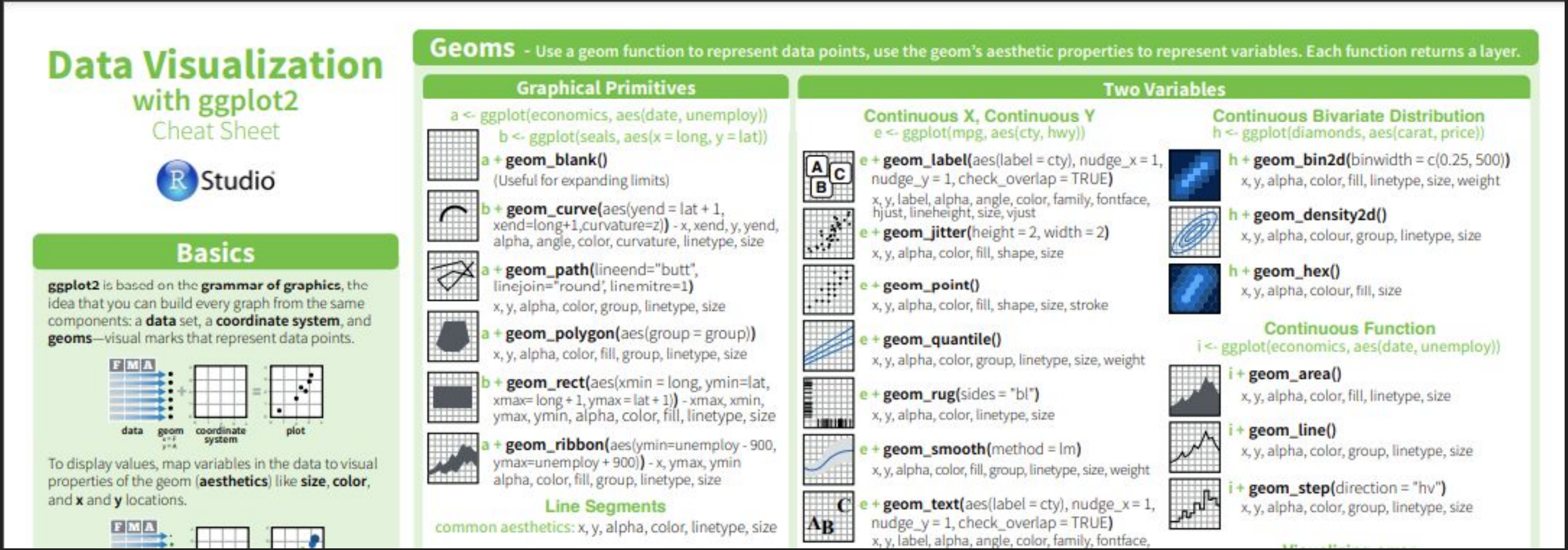
Data Viz Cheat Sheet

ggplot does not suggest what plot to make

How do you decide what type of plot to make?

- Number of variables
- Continuous vs Discrete
- Goal of the visualization

Data Viz Cheat Sheet



Data Visualization with ggplot2 Cheat Sheet

RStudio

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.

F M A
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.

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()**
(Useful for expanding limits)

b + **geom_curve(**aes(yend = lat + 1, xend=long+1,curvature=z)**)** - 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(group = group)**)**
x, y, alpha, color, fill, group, 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

Two Variables

Continuous X, Continuous Y

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

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

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

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, check_overlap = TRUE**)**
x, y, label, alpha, angle, color, family, fontface,

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_density2d()**
x, y, alpha, colour, group, linetype, size

h + **geom_hex()**
x, y, alpha, colour, 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

https://www.rstudio.com/wp-content/uploads/2016/11/ggplot2-cheatsheet-2.1.pdf

Economics Example

1. Load the `economics` dataset:

```
> data(economics)        > ?economics # To learn about the variables
```

2. Create a line plot to visualize the trend in personal savings rate (`psavert`) over time (`date`)

```
> ggplot(economics, aes(x = ____, y = _____) +
```

```
>   geom______()
```

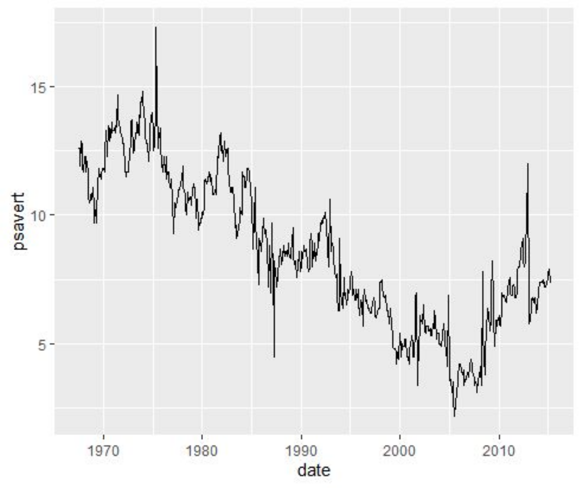
3. Color the above plot based on the population size `pop`.

4. Create as many different types of plots for (2) as possible.

Economics Example - Solutions

2. Create a line plot to visualize the trend in personal savings rate (`psavert`) over time (`date`)

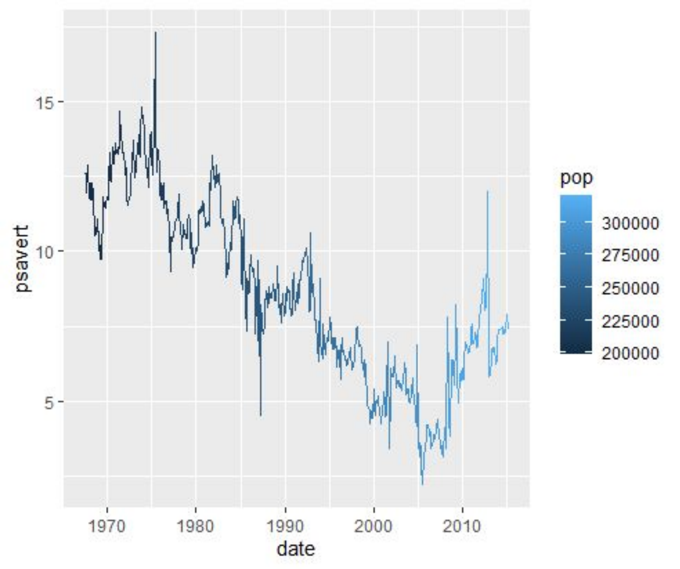
```
> ggplot(economics, aes(x = date, y = psavert) +
>   geom_line()
```



Economics Example - Solutions

3. Color the above plot based on the population size `pop`.

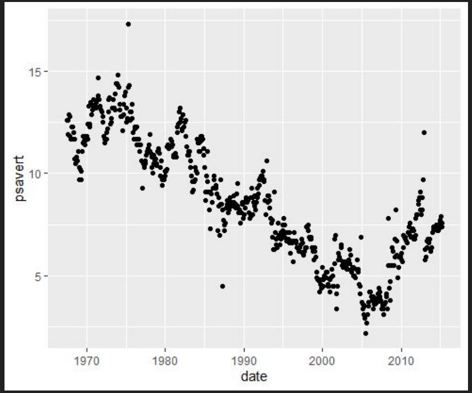
```
> ggplot(economics, aes(x = date, y = psavert, color = pop)) +
>   geom_line()
```

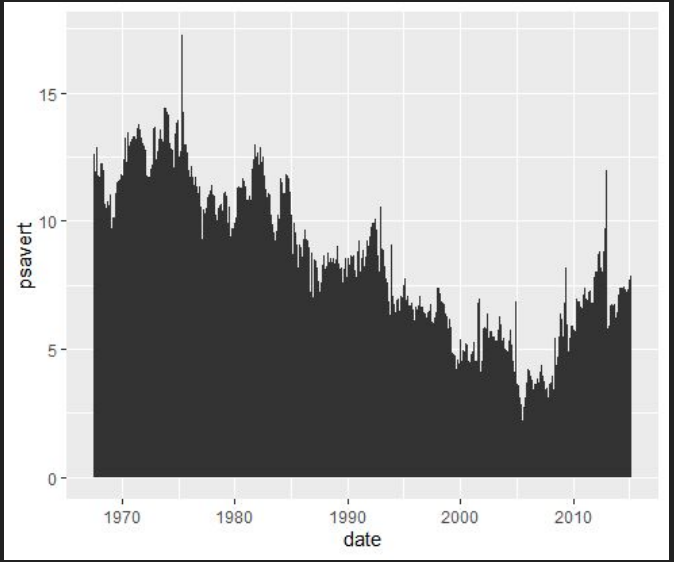


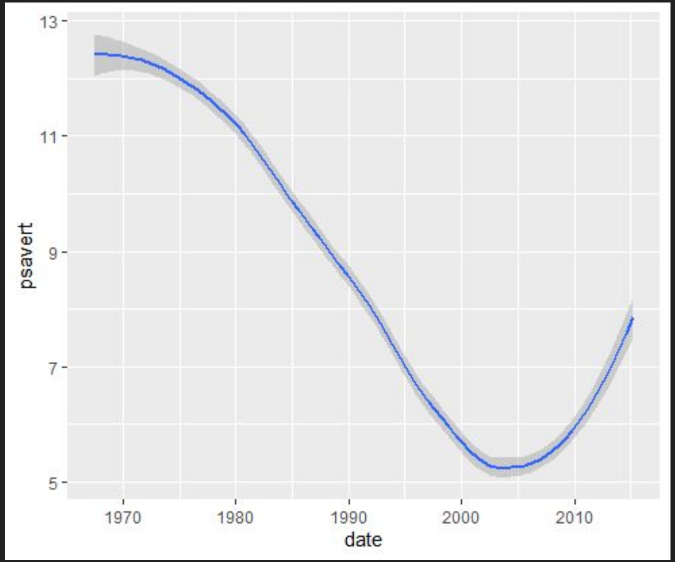
Economics Example - Solutions

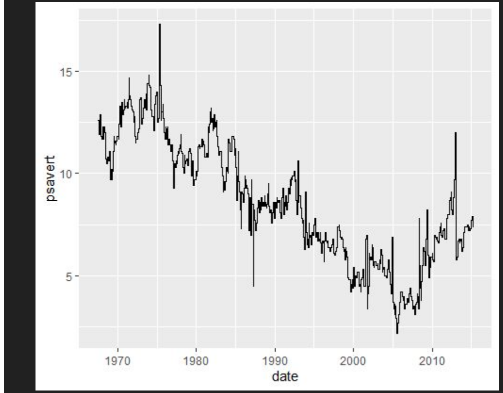
4. Create as many different types of plots for (2) as possible.

Essentially all the geometries for 2 continuous variables work!





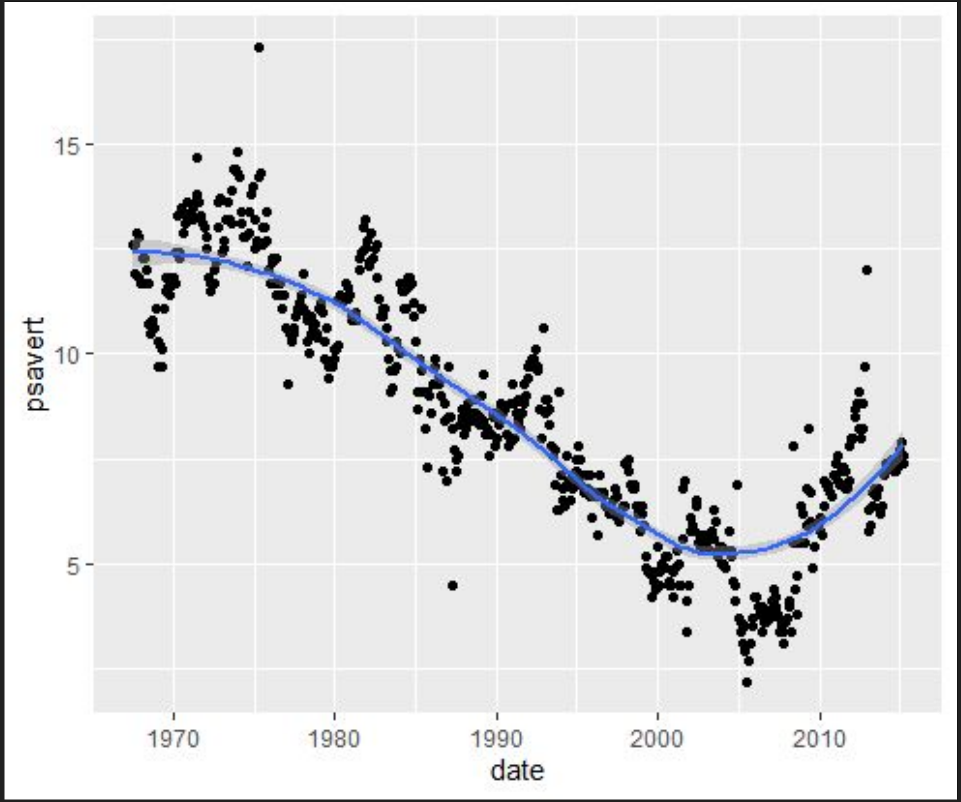




Economics Example - Solutions

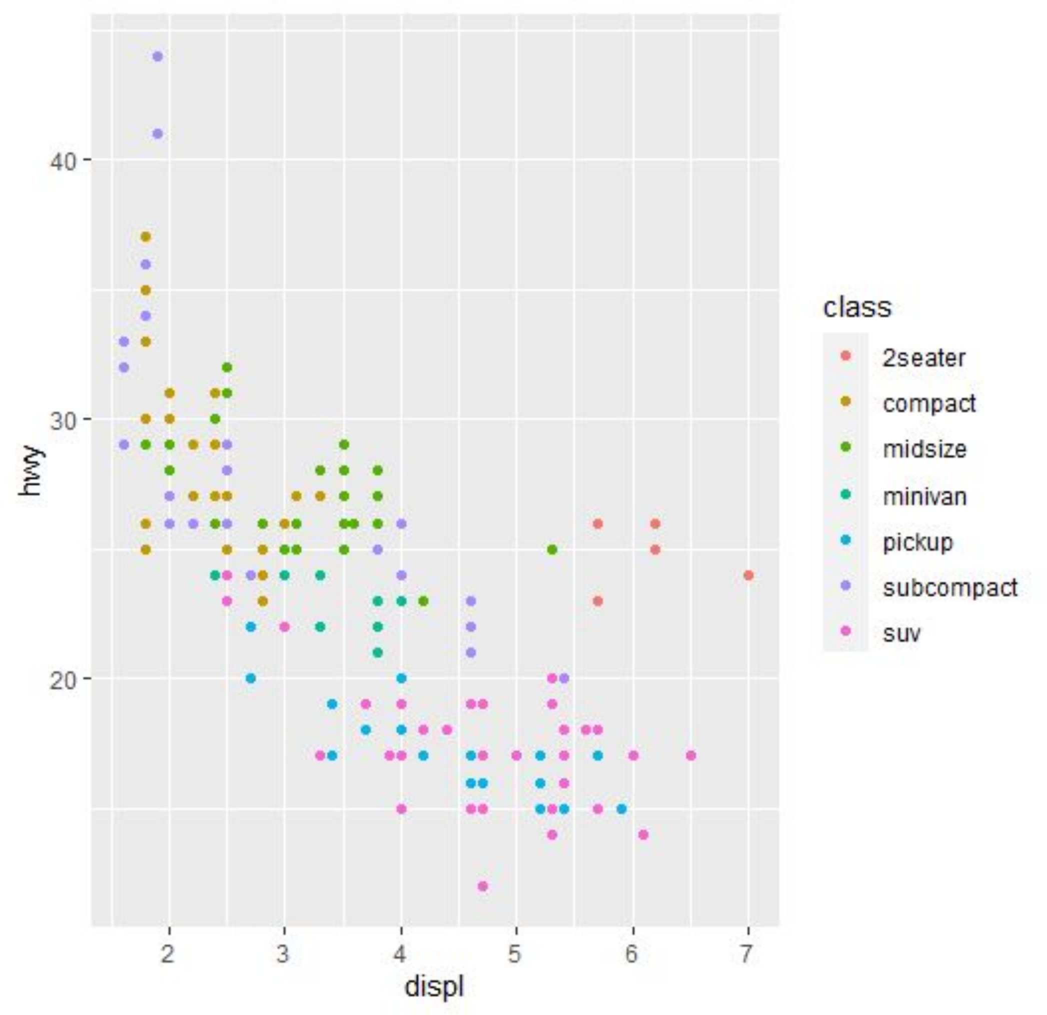
Note: You can overlap geoms!

```
ggplot(economics, aes(x = date, y =
psavert)) +
  geom_point() +
  geom_smooth()
```



Facets

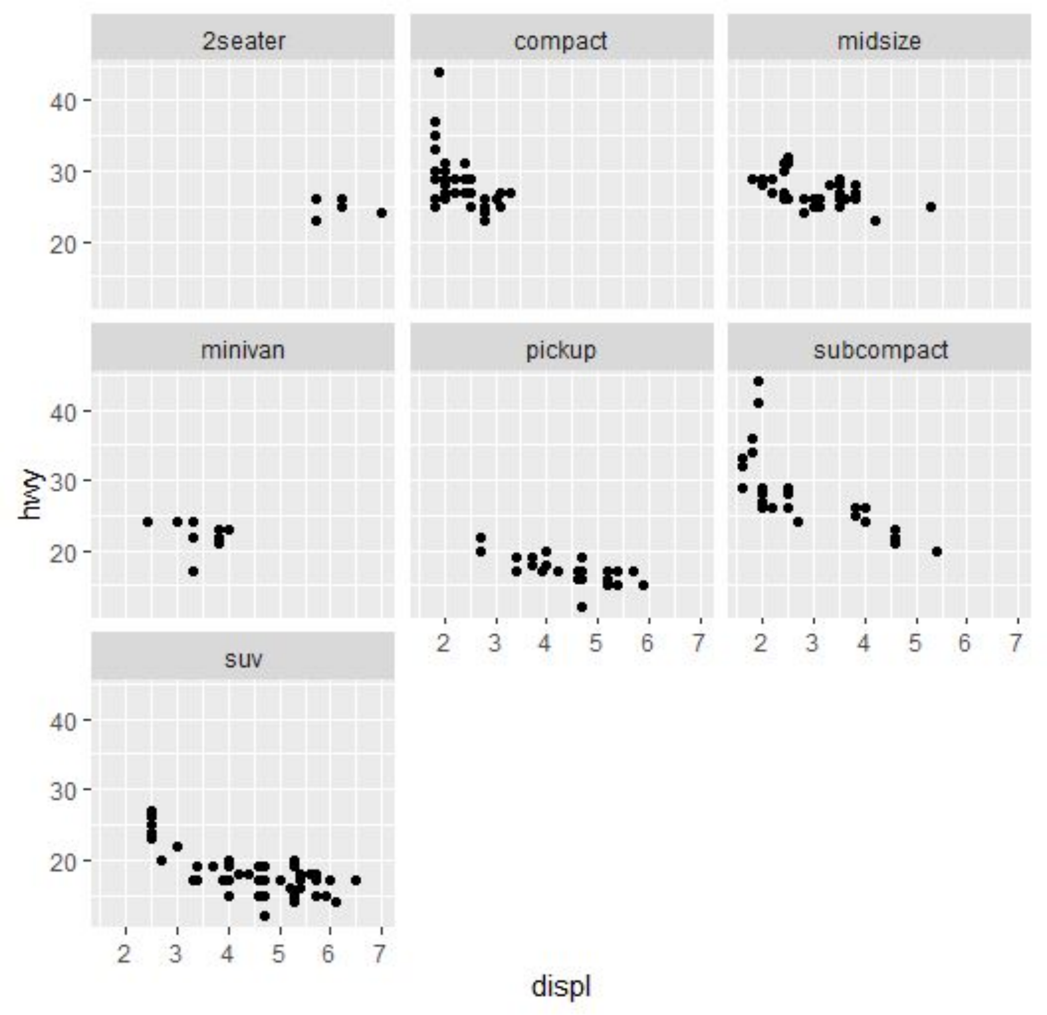
Grouping by color may be hard to differentiate



Based on: https://ggplot2-book.org/getting-started.html

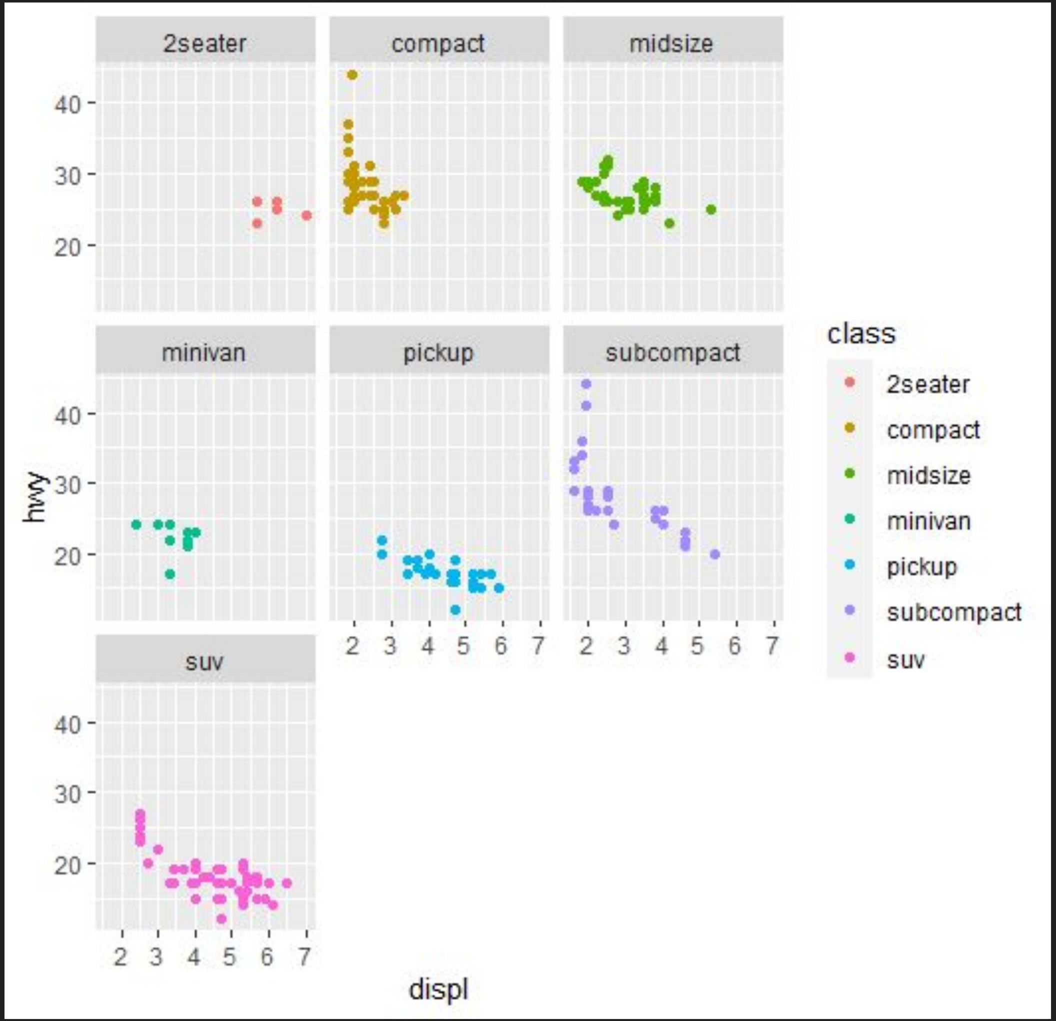
Facets

Facets clearly show groupings

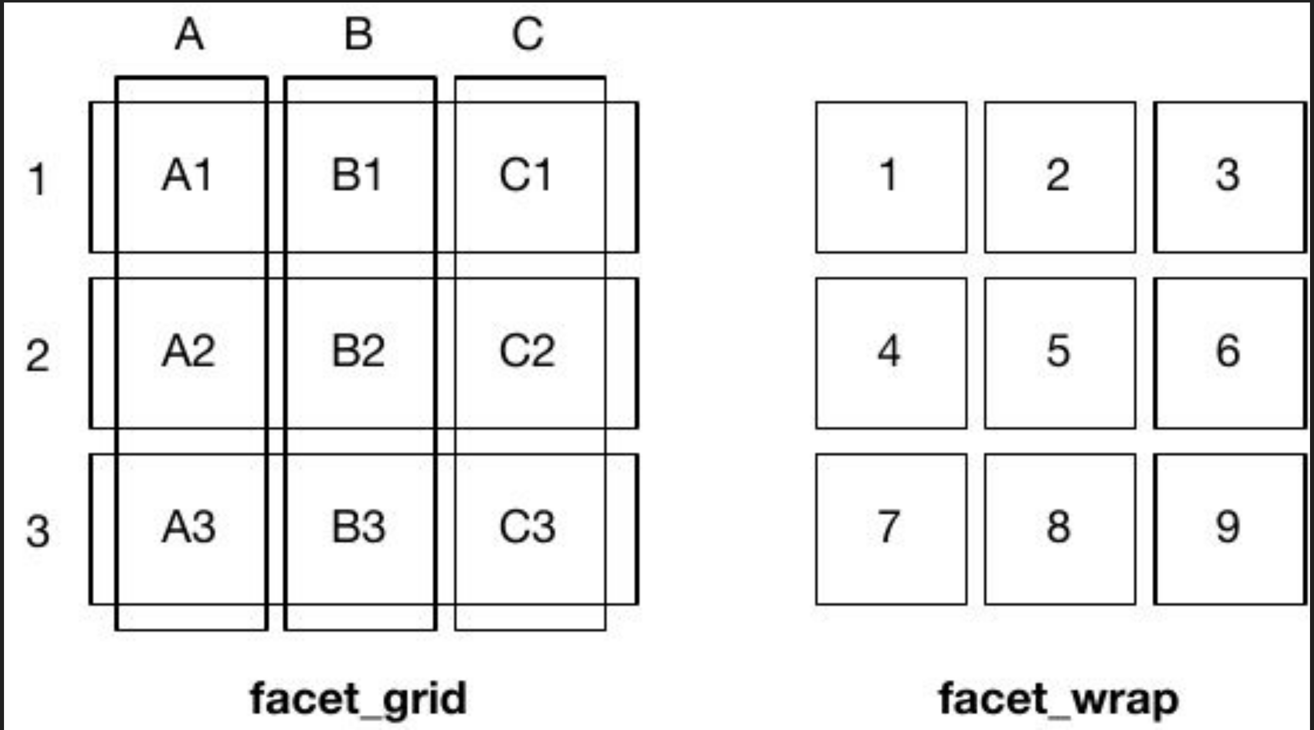


Facets

Facets clearly show groupings



Facets



Source: https://ggplot2-book.org/facet.html

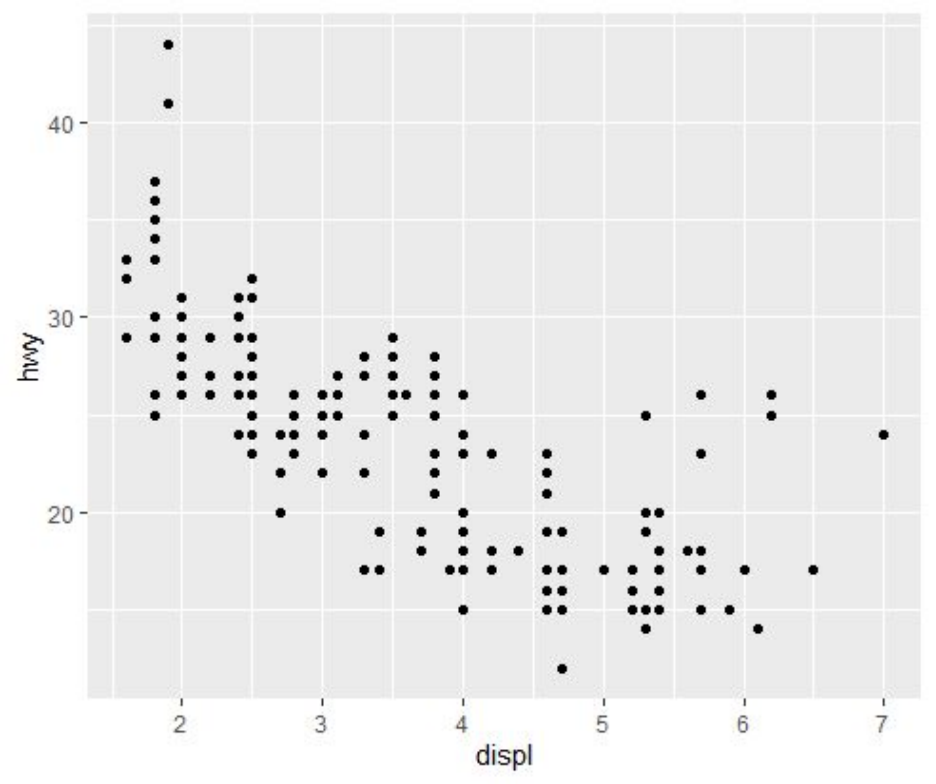
facet_wrap

... + facet_wrap(~x)

Split the plots by variable x

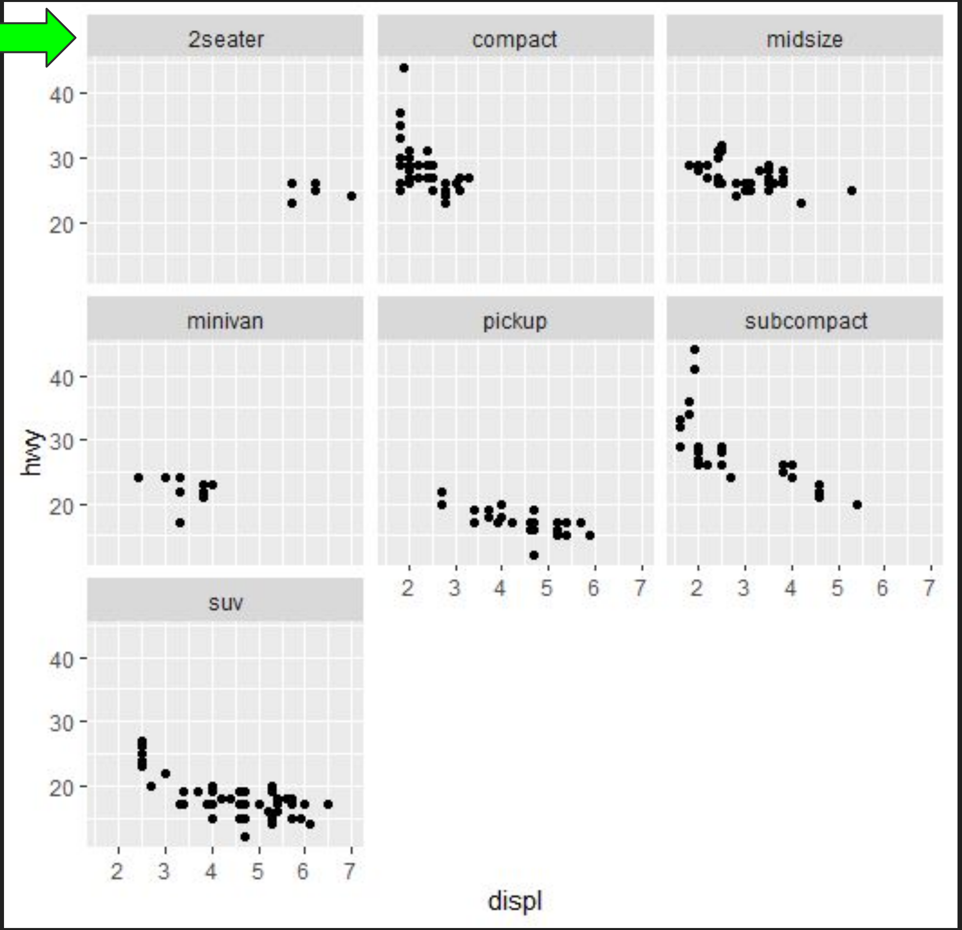
facet_wrap

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



facet_wrap

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



facet_grid

```
... + facet_grid(y~x)
```

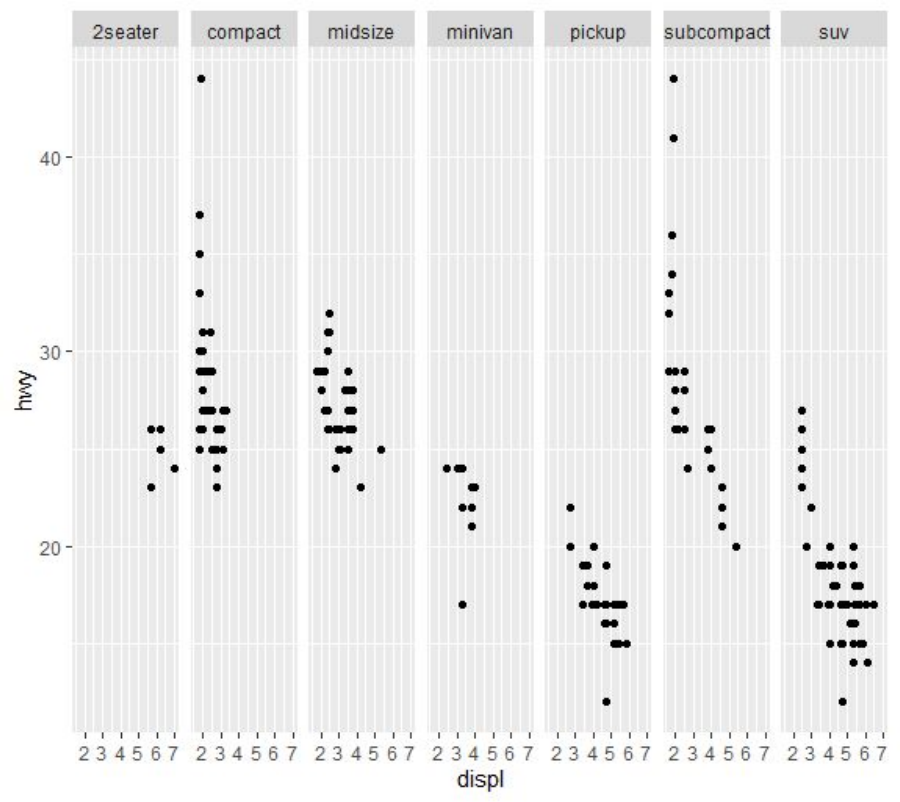
y variable on columns

x variable on rows

facet_grid

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_grid(class~.)
```

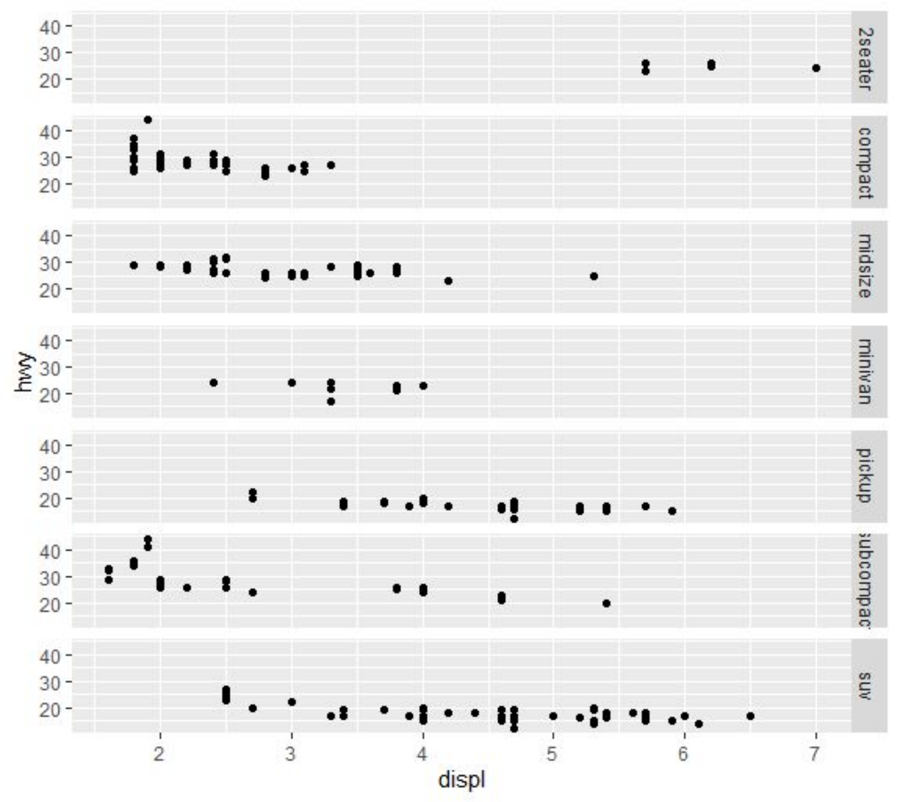
`class` as columns



facet_grid

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_grid(.~class)
```

`class` as rows

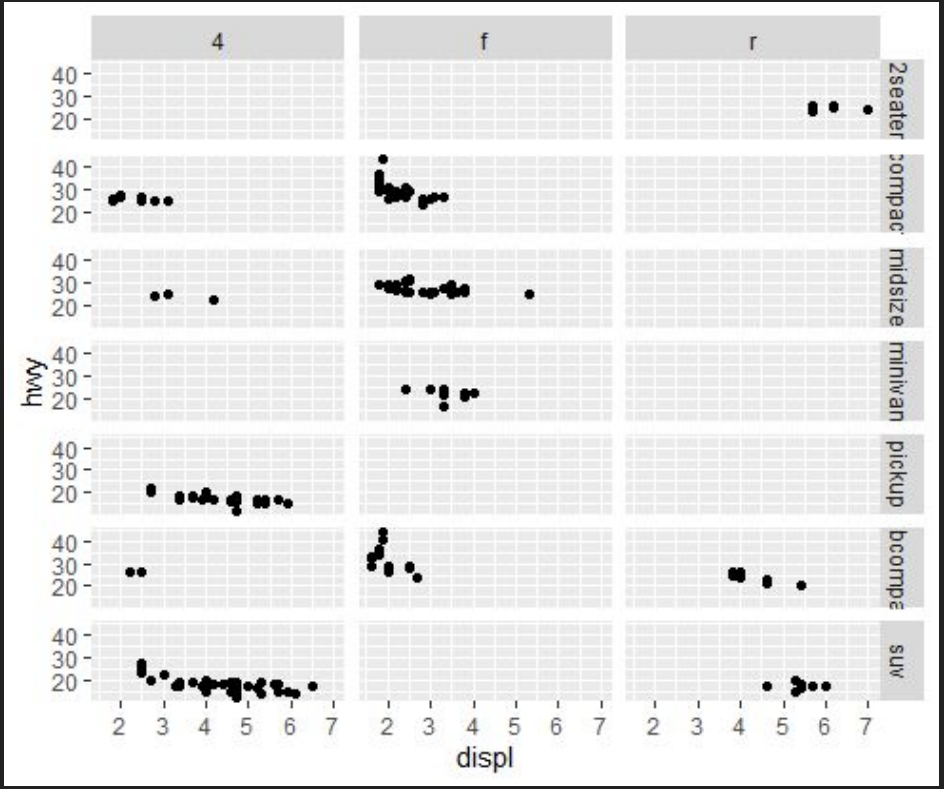


facet_grid

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_grid(drv ~ class)
```

`drv` as columns

`class` as rows



facet_grid and facet_wrap

```
... + facet_grid(y~x, scales = “________”)
```

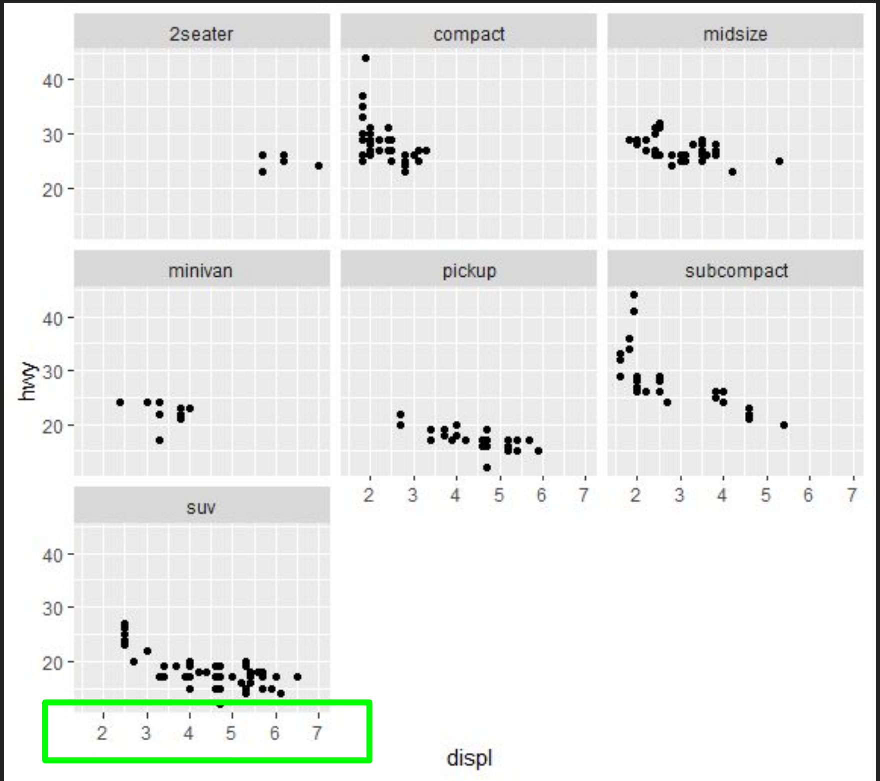
“fixed” = x and y fixed

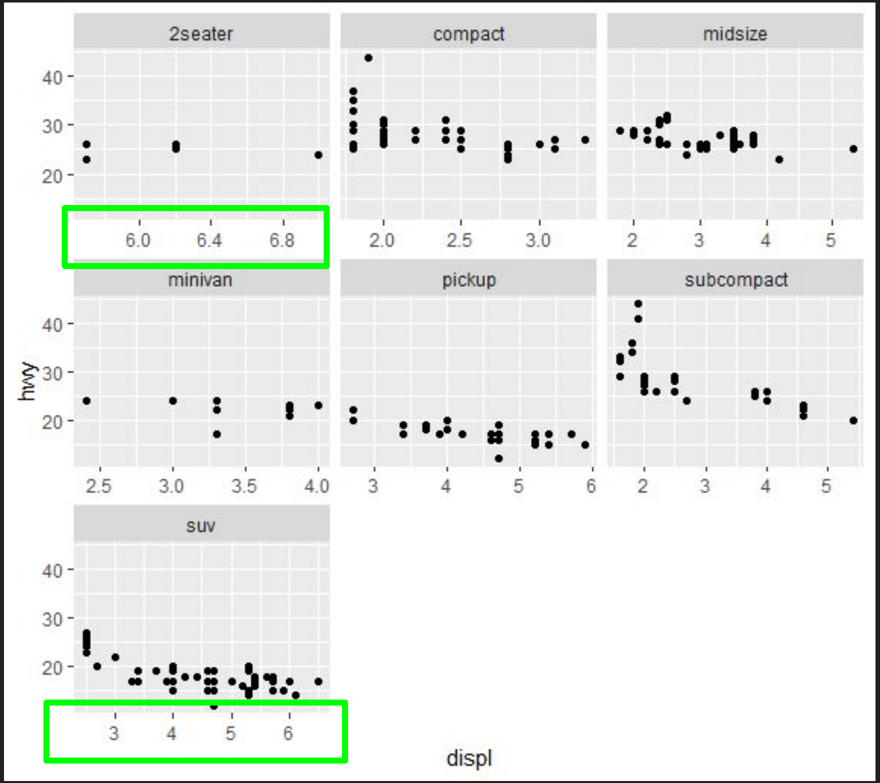
“free_x” = x scale free

“free_y” = y scale free

“free: = x and y free

... + facet_wrap(~class, scales = “free_x”)





Diamonds Example

1. Load the `diamonds` dataset:

```
> data(diamonds)          > ?diamonds # To learn about the variables
```

2. What is the most common diamond color (`color`)?

```
> ggplot(________, aes(x = ____) +

>   geom______()
```

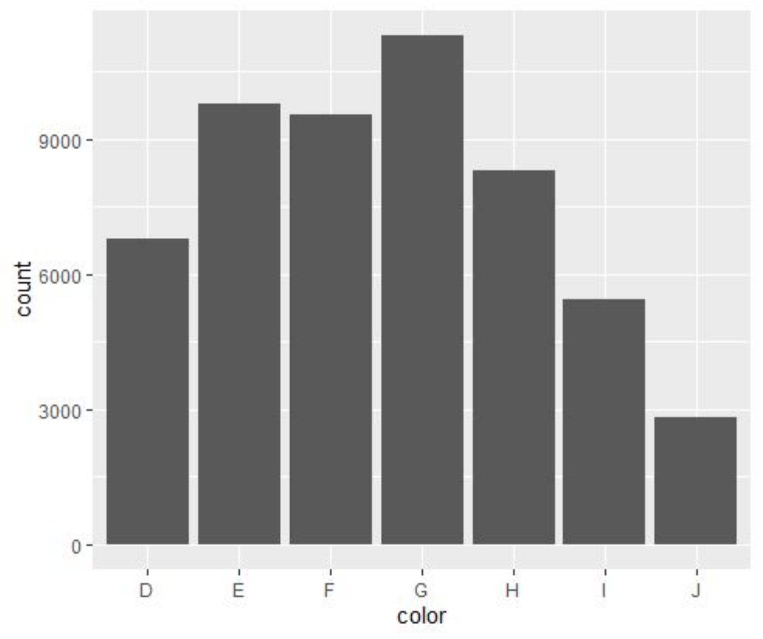
3. What is the most common diamond color (`color`) for each type of cut `cut`? Use facet_wrap() or facet_grid().

4. Set the facet scales to be “free_y”, and color by `cut`

Diamonds Example - Solutions

2. What is the most common diamond color (`color`)? Answer with a plot.

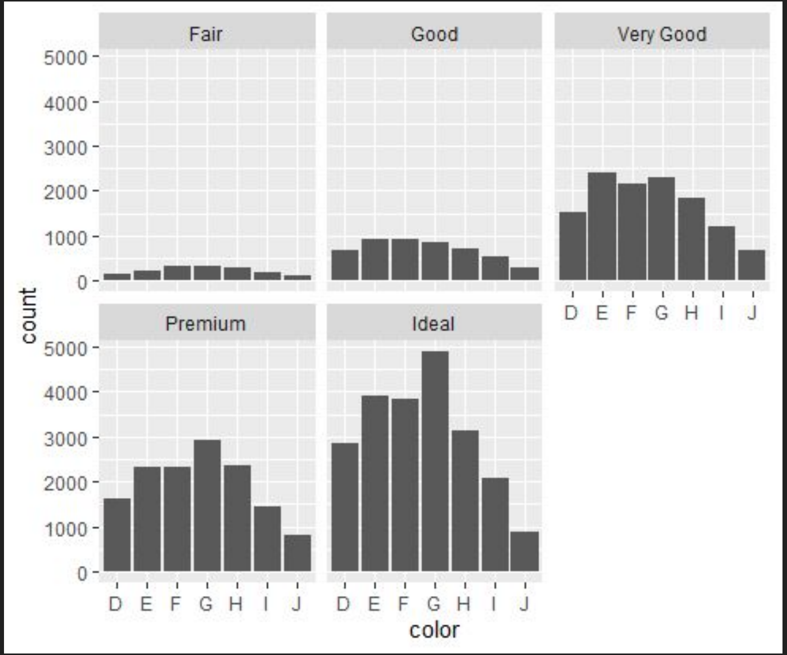
```
> ggplot(diamonds, aes(x = color) +
>   geom_bar()
```



Diamonds Example - Solutions

3. What is the most common diamond color (`color`) for each type of cut `cut`? Use facet_wrap() or facet_grid().

```
> ggplot(diamonds, aes(x = color)) +
>   geom_bar() +
>   facet_wrap(~cut)
```

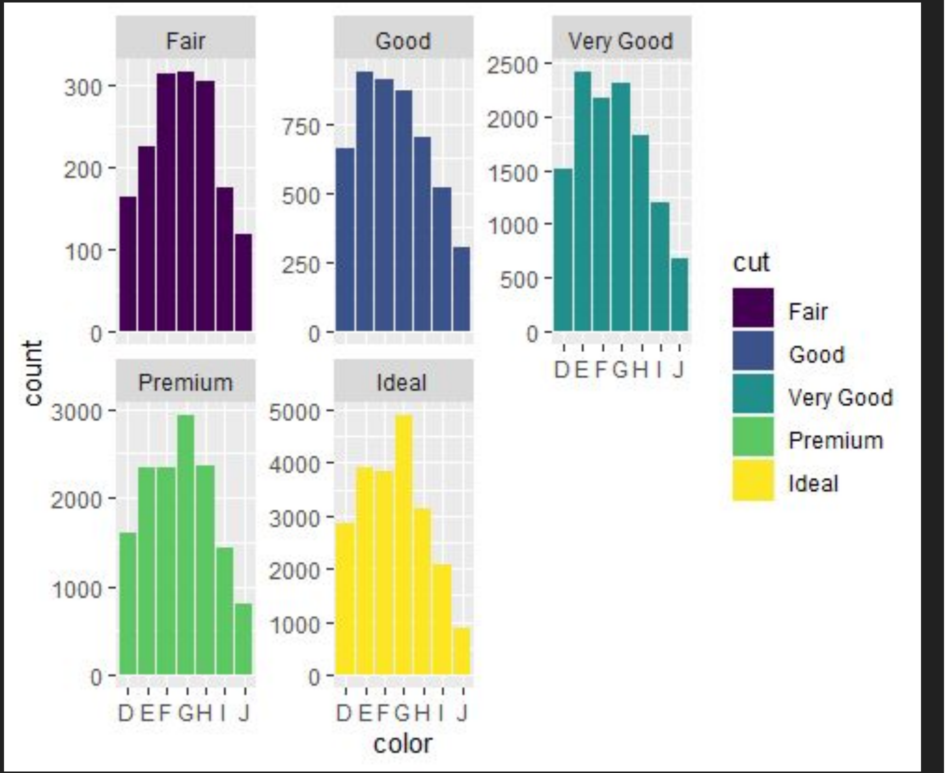


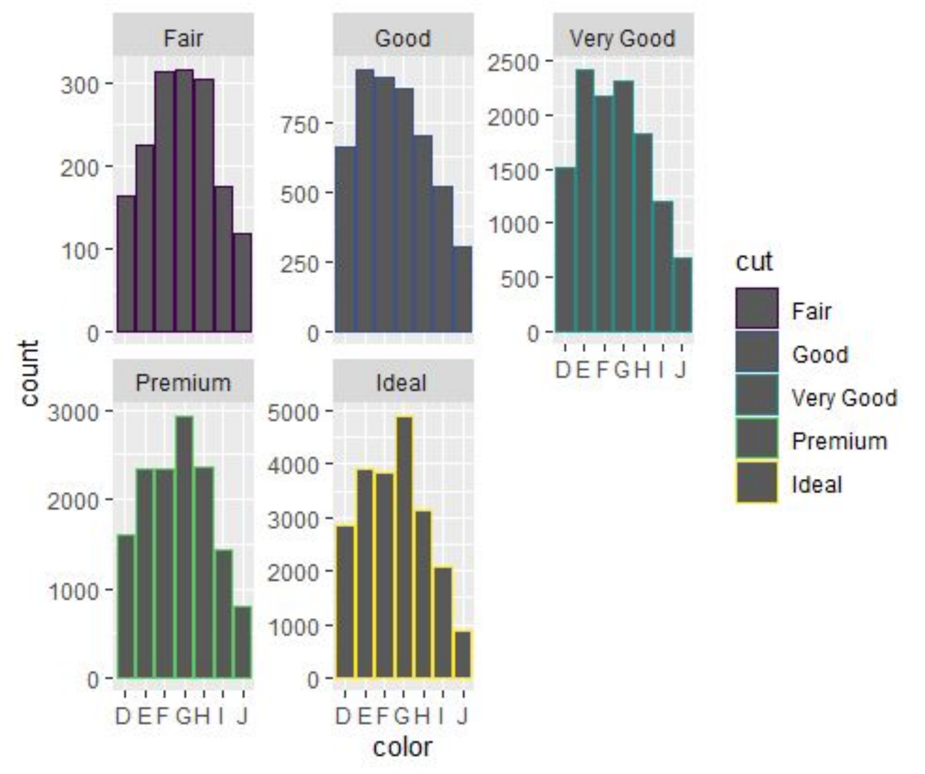
Diamonds Example - Solutions

4. Set the facet scales to be “free_y”, and color by `cut`

```
> ggplot(diamonds, aes(x = color, fill = cut)) +
>   geom_bar() +
>   facet_wrap(~cut, scales = “free_y”)
> ggplot(diamonds, aes(x = color, color = cut)) +
>   geom_bar() +
>   facet_wrap(~cut, scales = “free_y”)
```

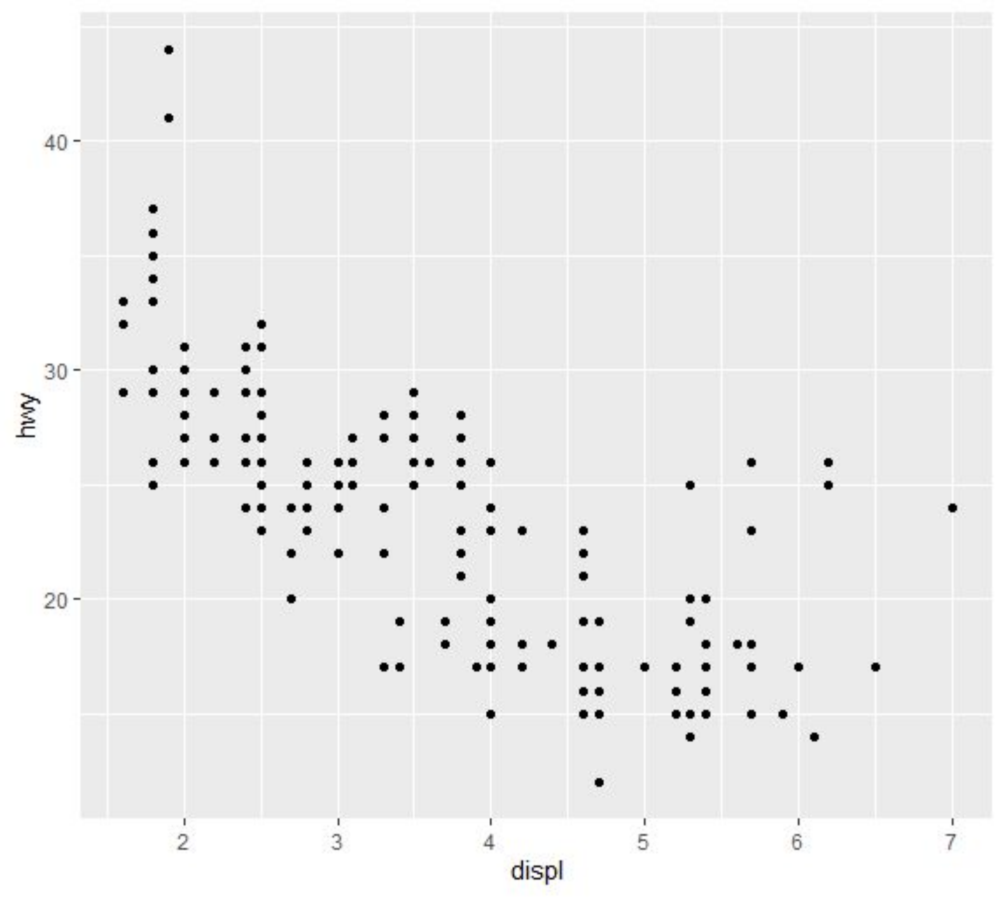
Diamonds Example - Solutions





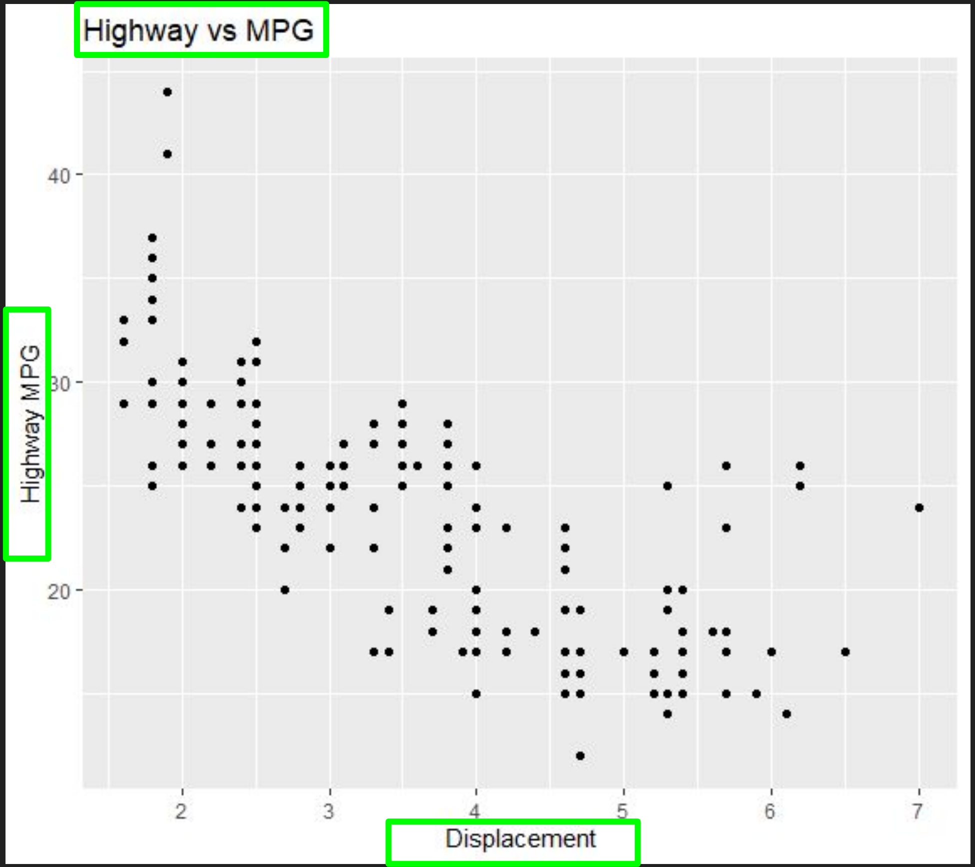
labs - Adding labels

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



labs - Adding labels

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  labs(x = “Displacement”,
         y = “Highway MPG”,
         title = “Highway vs MPG”)
```

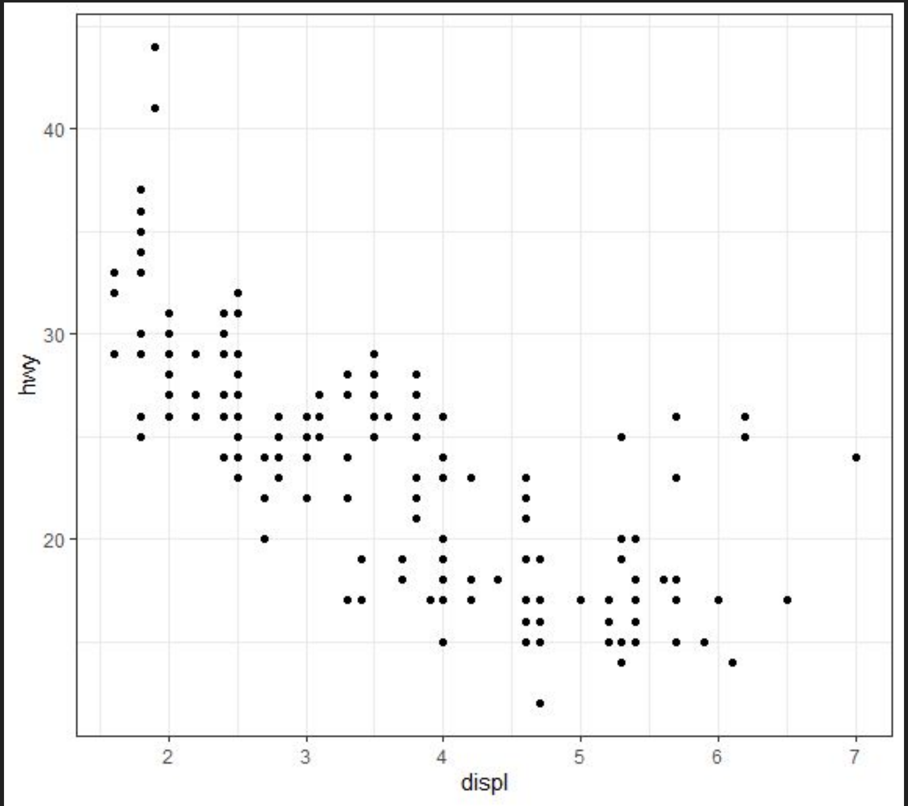


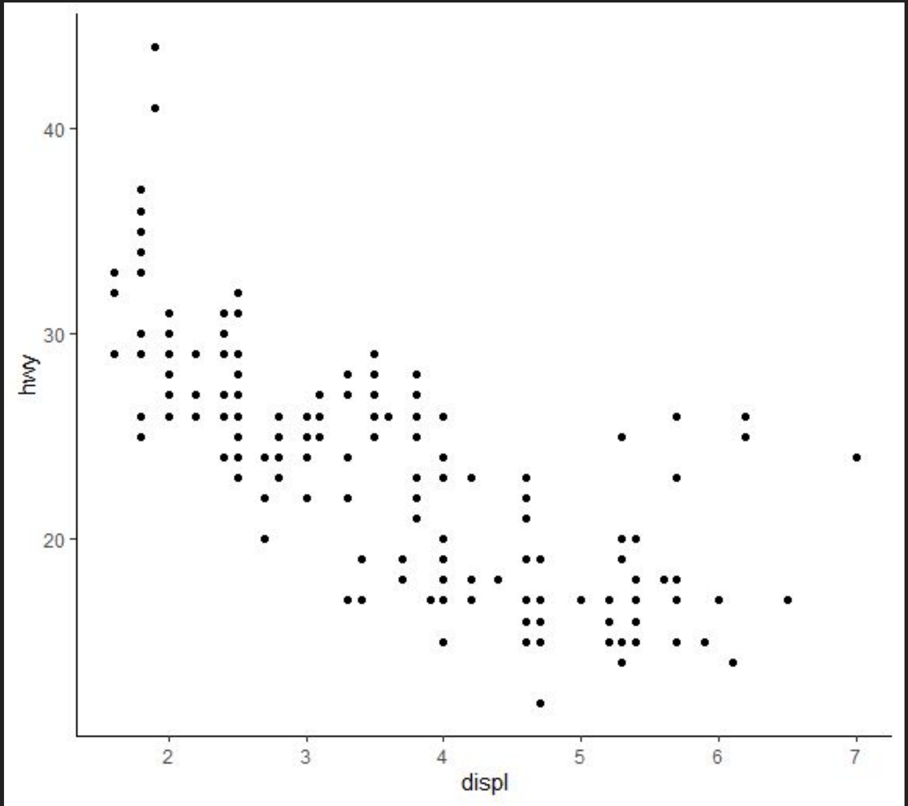
Themes

- theme_bw()
- theme_linedraw()
- theme_light()
- theme_dark()
- theme_minimal()
- theme_classic()
- theme_void()

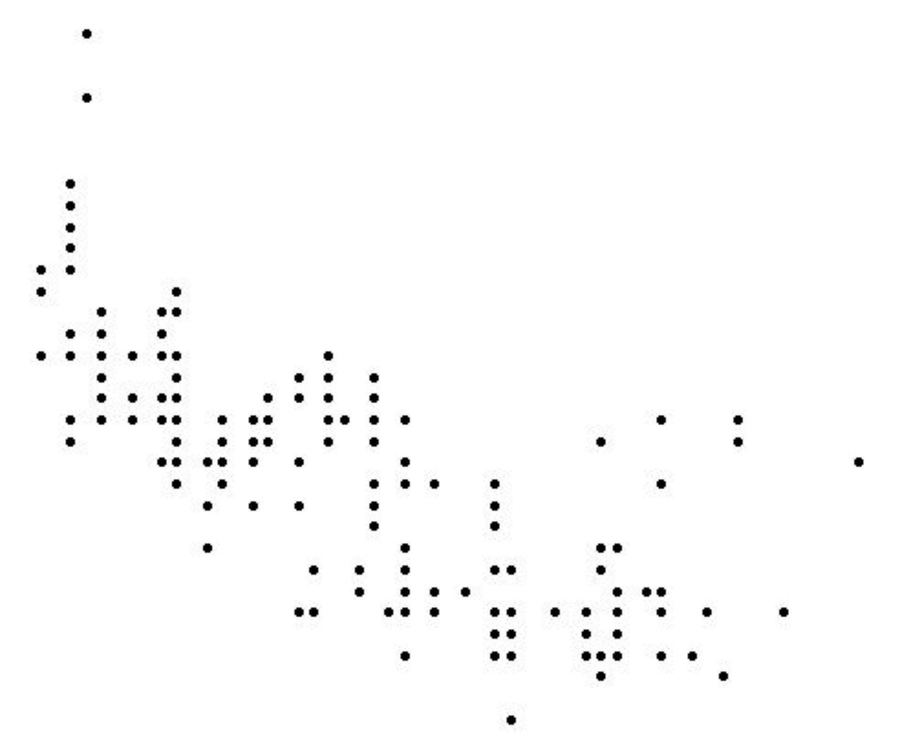
Note: These are complete themes, you can manually adjust colors/scales/fonts too

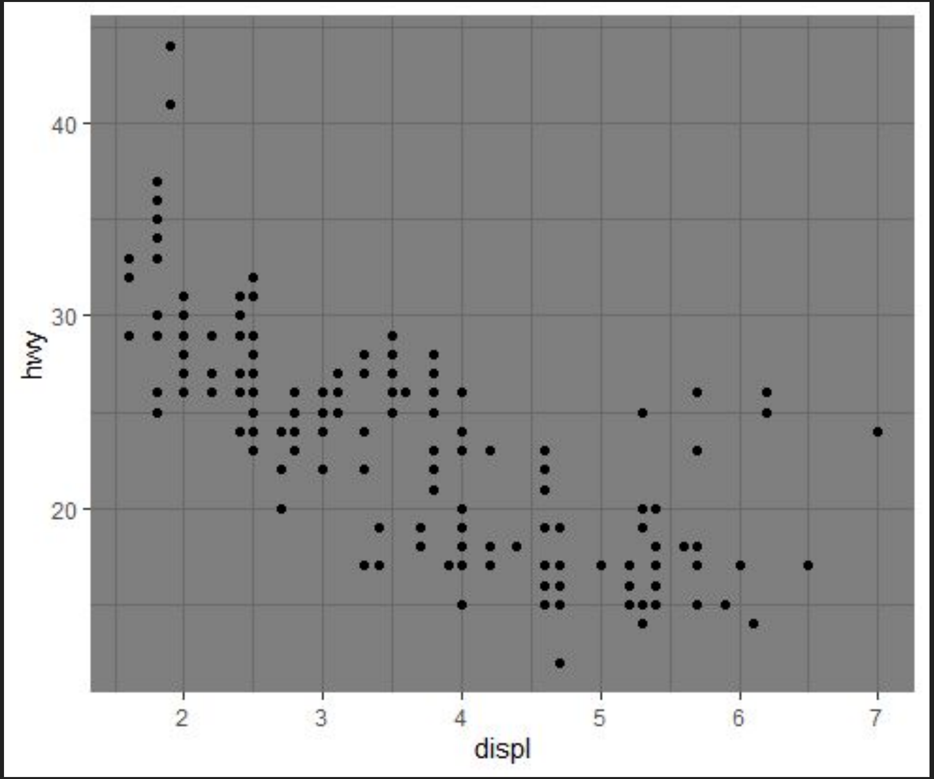
Themes



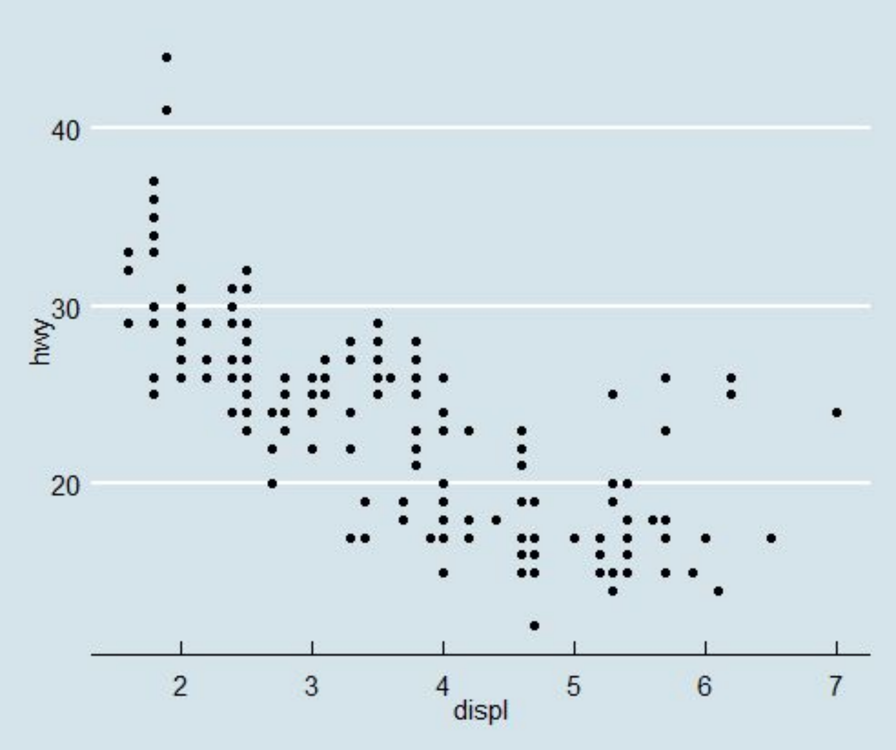


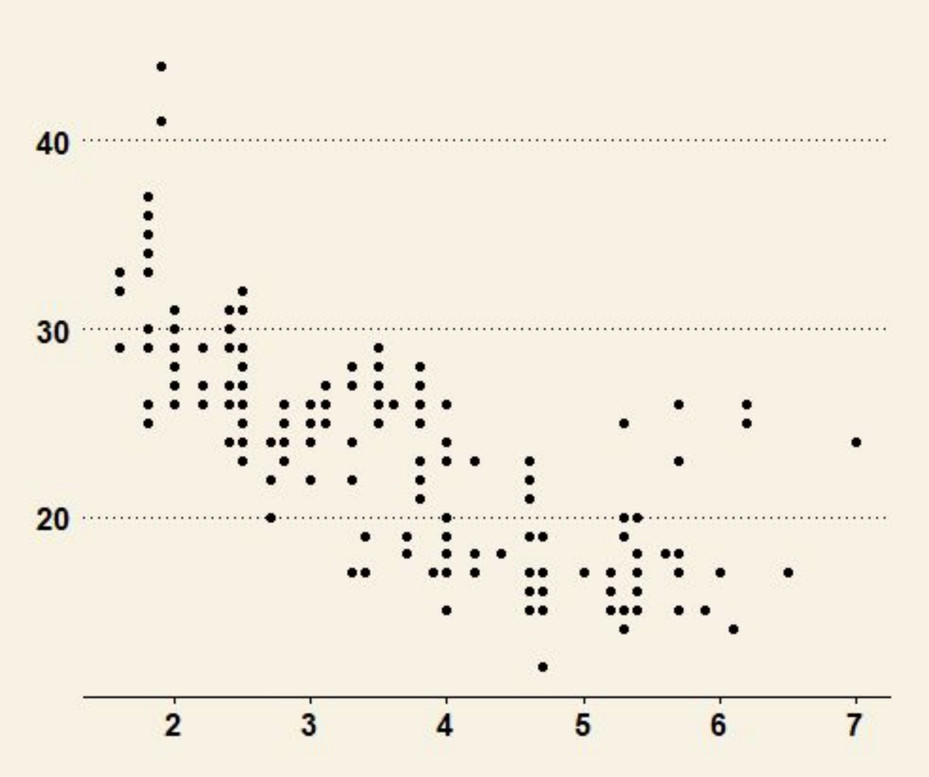
Themes





Themes - Extra themes from package `ggthemes`





Exporting ggplots

```
ggsave(“plotname.png”)
```

Saves most recently displayed ggplot

Can save as a png, jpg, pdf, svg, etc.

Can specify height and width of output

The Little Things

1. One plot = one primary message.
2. Do not use the default ggplot2 theme
3. Adjust axes as necessary
    a. Extreme outliers breaking scales - `scale_x_log10()`
    b. Manually select tick marks - `scale_<x, y>_manual()`
4. Rotate x labels if they are overlapping
    a. `+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))`
5. Alternative: Flip x and y axes if x has a lot of words
    a. `coord_flip()`
6. Want animated plots? Use the `**gganimate**` package

Useful Resources

Fundamentals of Data Visualization by Claus O. Wilke

- https://clauswilke.com/dataviz/

ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham

- https://ggplot2-book.org/index.html

Good luck!

Fill out the feedback survey!