

Introduction to R for Research Data Visualization with ggplot2



Workshop Housekeeping



Questions?

Use the chat or raise hand feature.



Feedback Survey

After each session, please fill out the Qualtrics survey (link in chat and emailed out after session).



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

**How do you
pronounce the
word: Data**

What is the purpose of Data Visualization?

Agenda

- Base R vs ggplot2
- The logic of ggplot2
- Basic plots
- Data viz cheat sheet
- Faceting
- Themes
- Research Example
- Exporting plots
- Steps moving forward



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

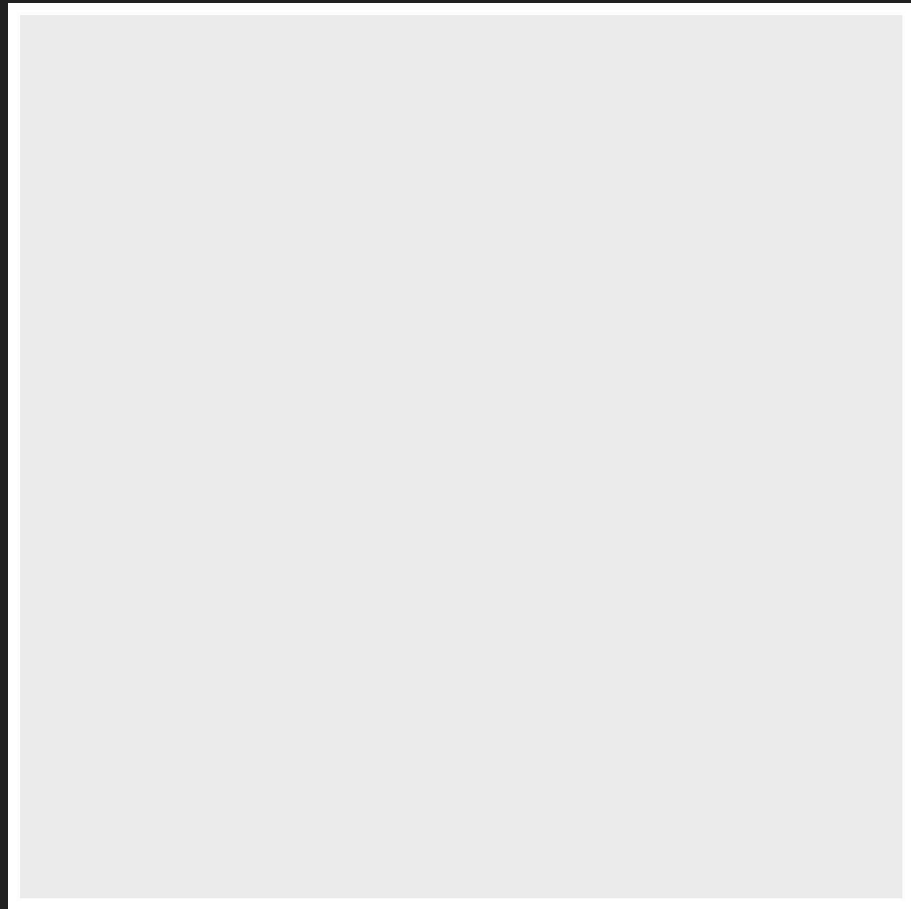
mpg dataset from ggplot2 - “Fuel economy data from 1999 to 2008 for 38 popular models of cars”

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.)

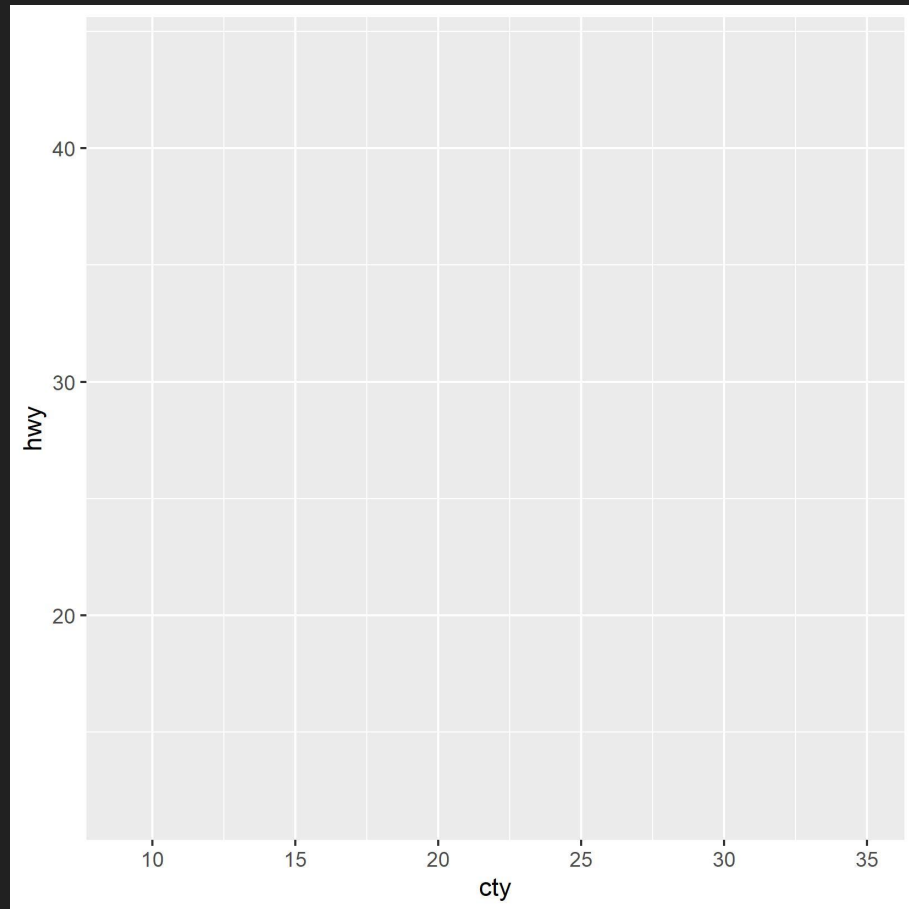
ggplot2 layers

ggplot(mpg)



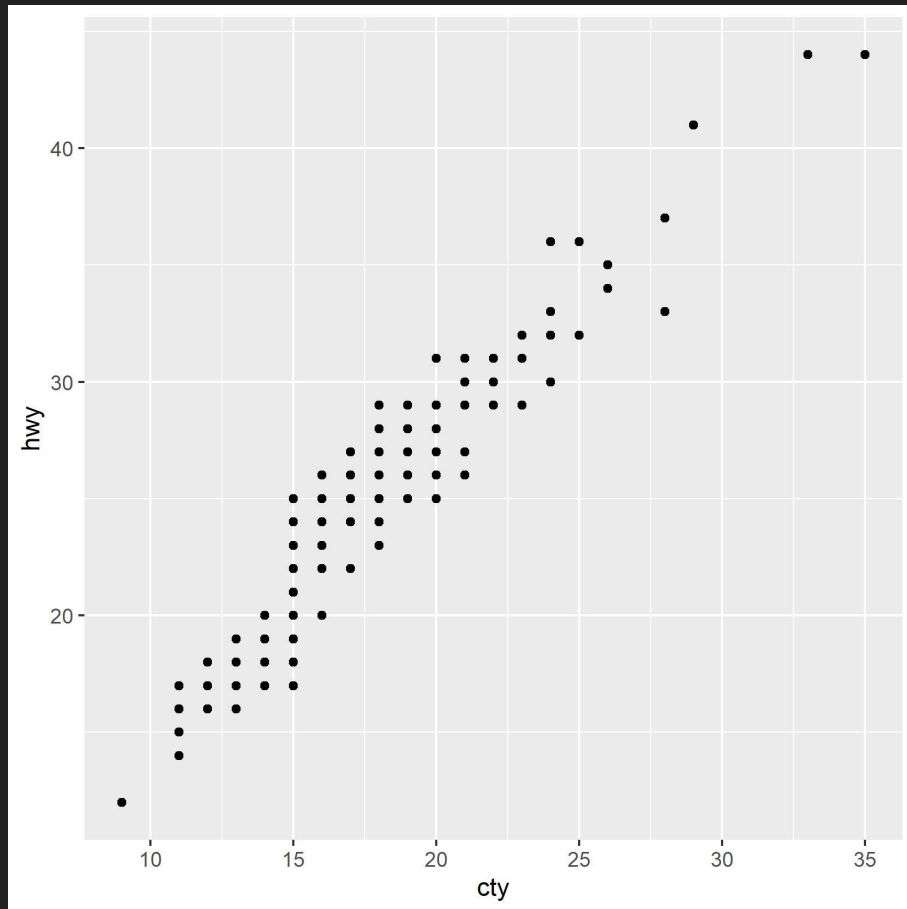
ggplot2 layers

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



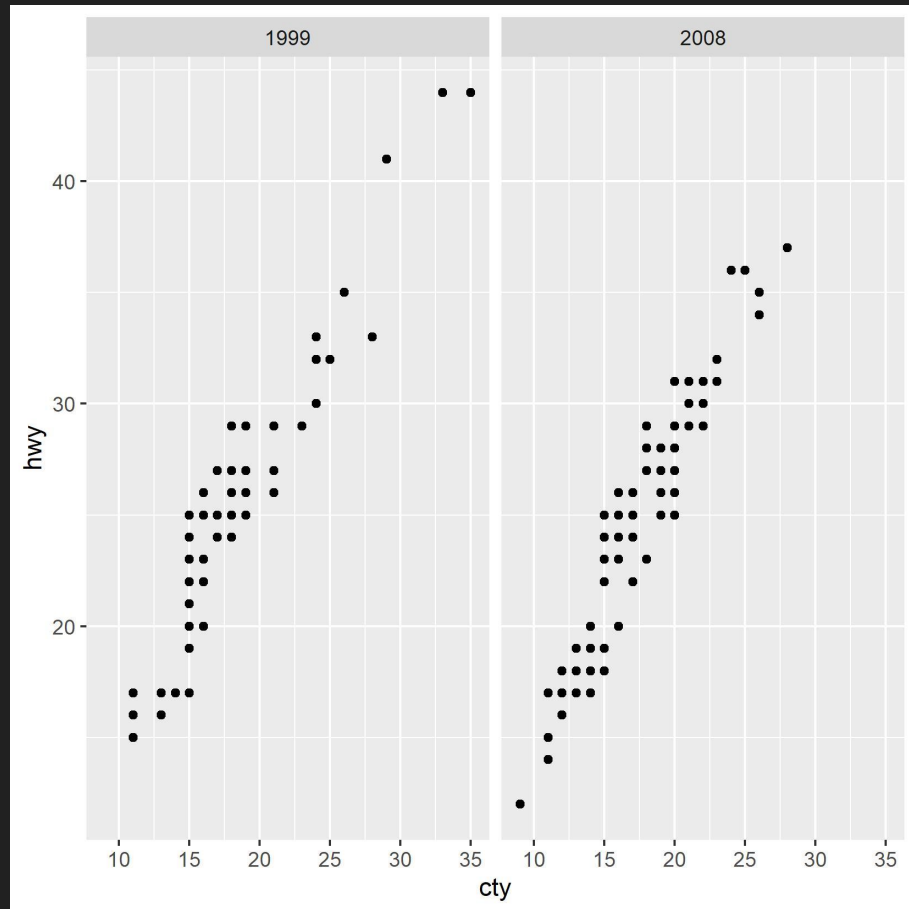
ggplot2 layers

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



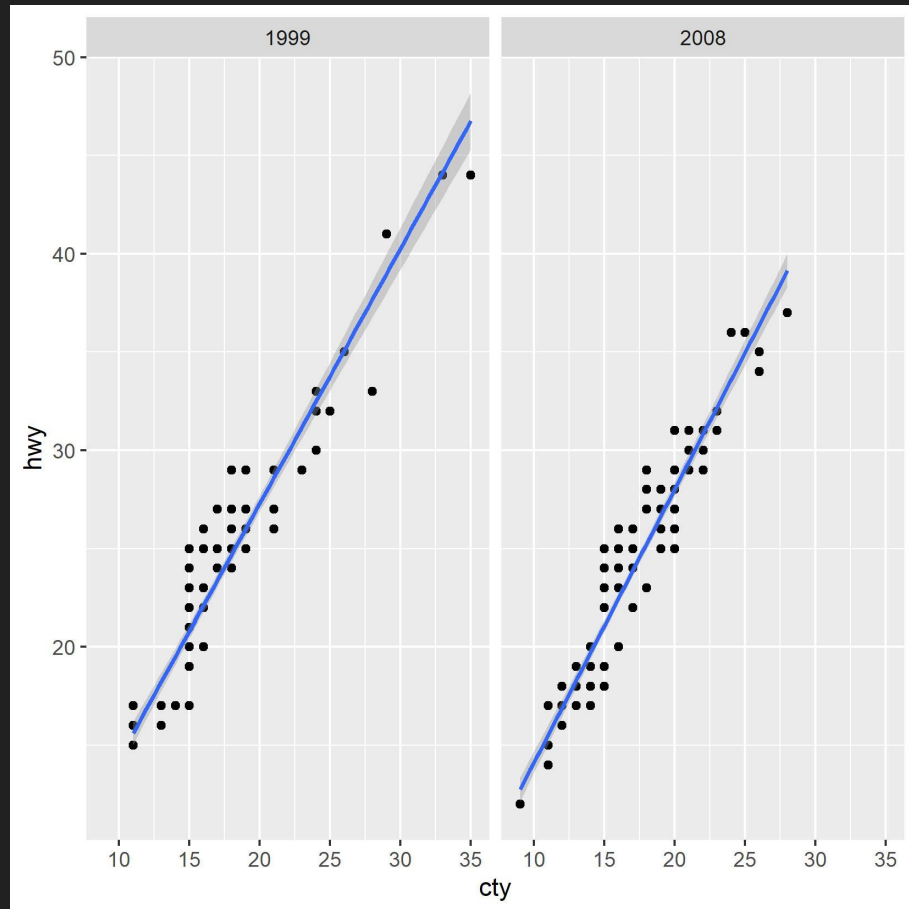
ggplot2 layers

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



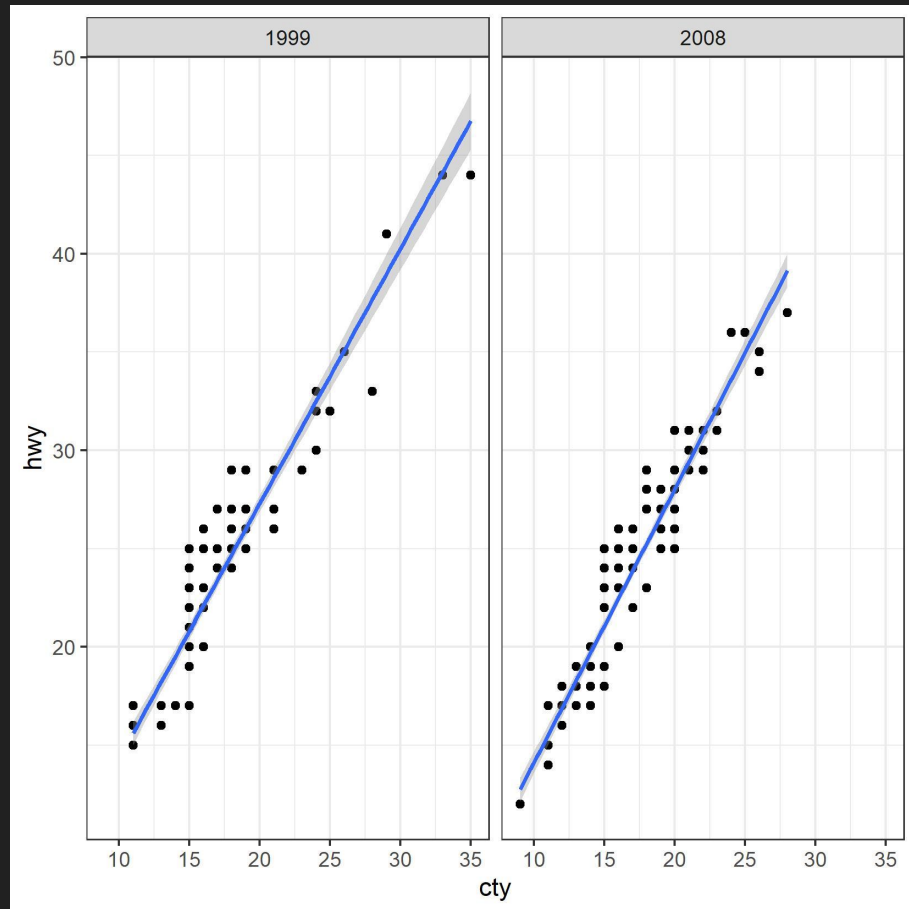
ggplot2 layers

```
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()
```



ggplot2 layers

Data

What dataset

Aesthetics

Select x and y-axis, colors, shapes, etc.

Geometries

Which geometric object (points, bars, lines)

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

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

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)

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 left 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.

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 `+`

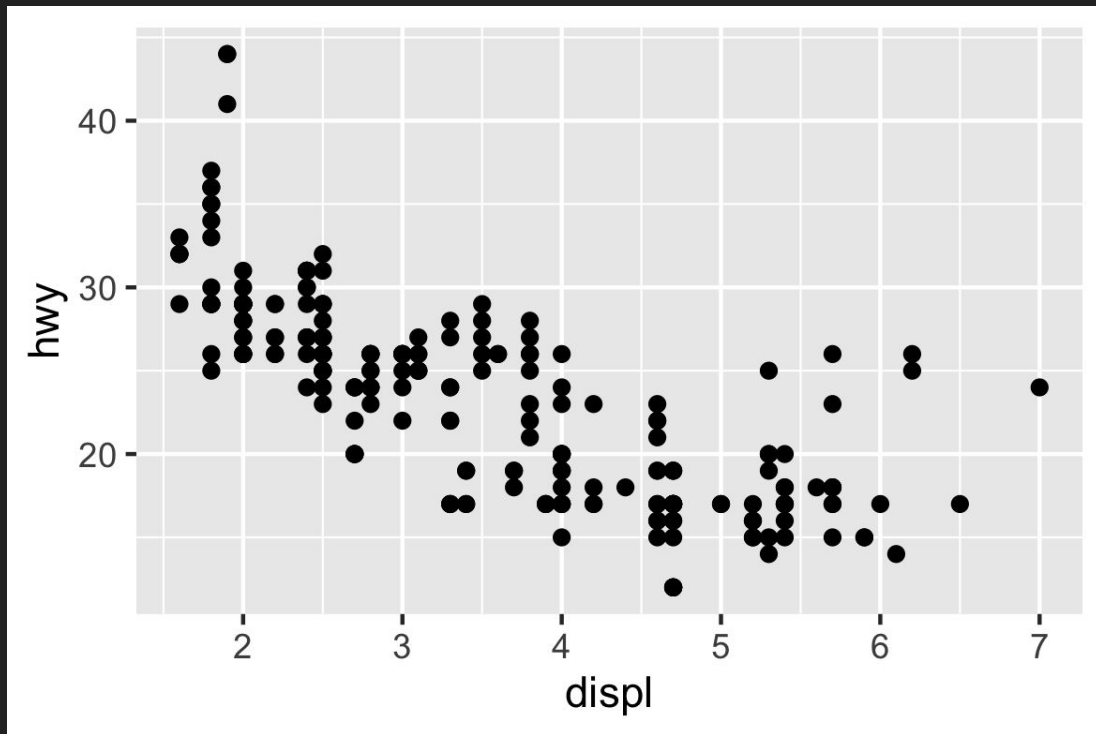
```
ggplot(dataset, aes(xvar, yvar)) +  
  geom_*()
```

How to represent the data points

`geom_point()` = scatterplot

`geom_histogram()` = histogram

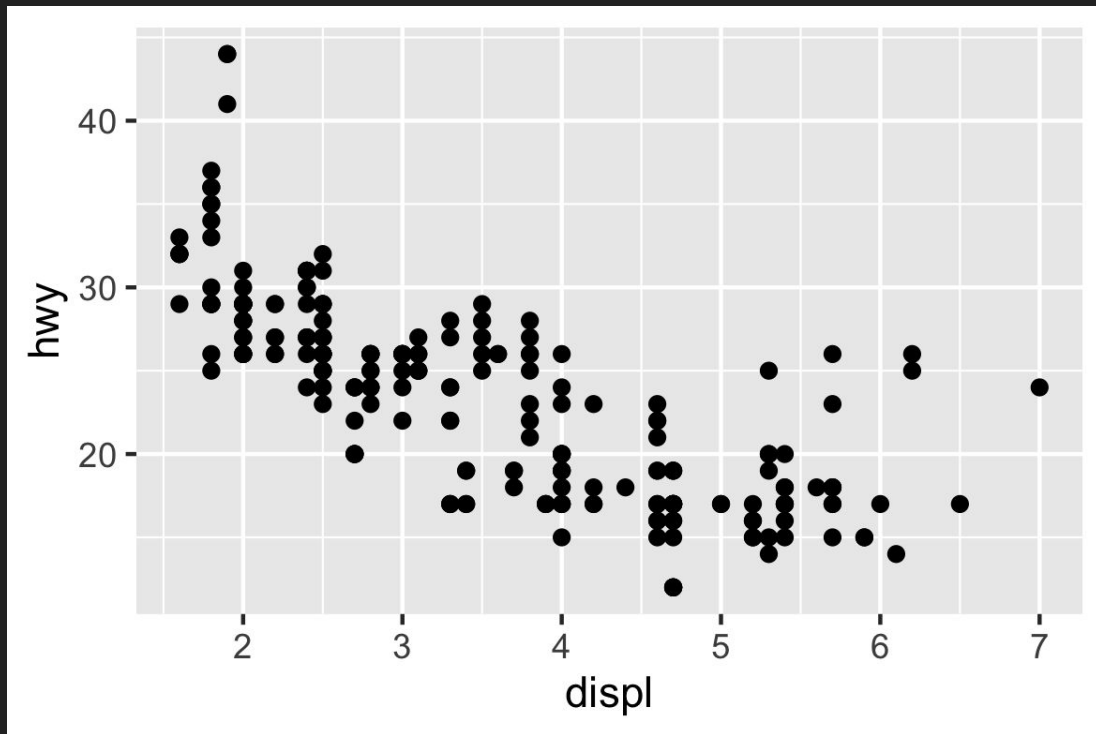
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point()
```



ggplot2: Elegant Graphics for Data Analysis,
by Hadley Wickham

<https://ggplot2-book.org/getting-started.html>

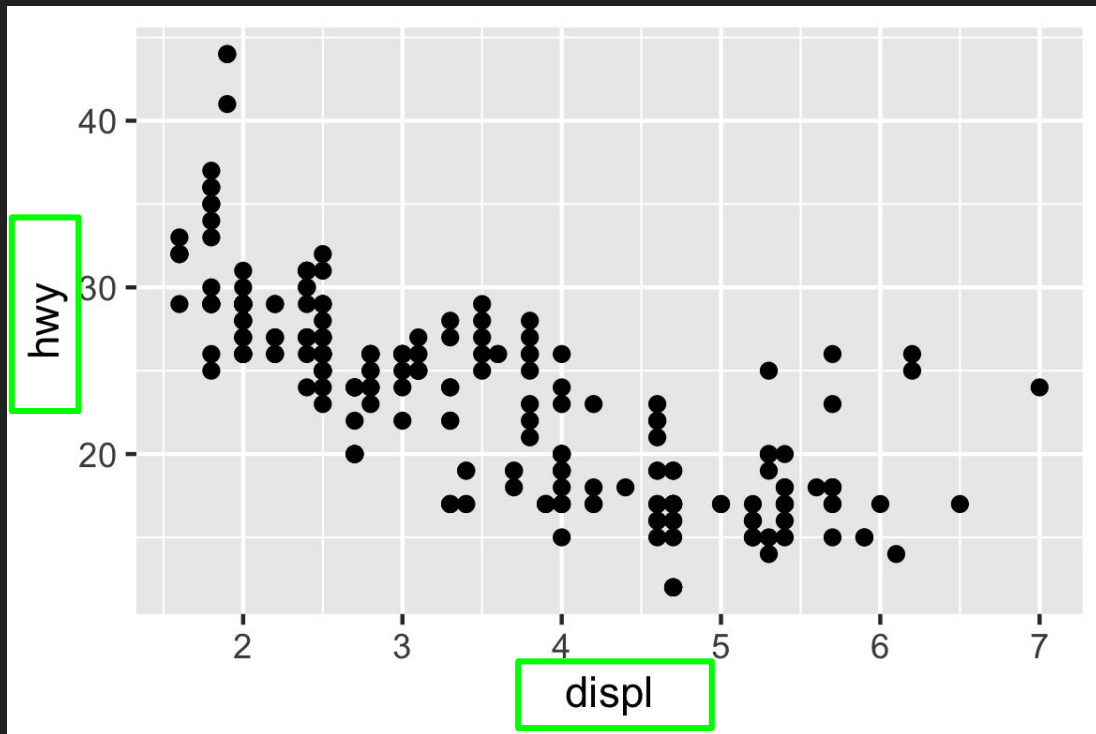
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point()
```



ggplot2: Elegant Graphics for Data Analysis,
by Hadley Wickham

<https://ggplot2-book.org/getting-started.html>

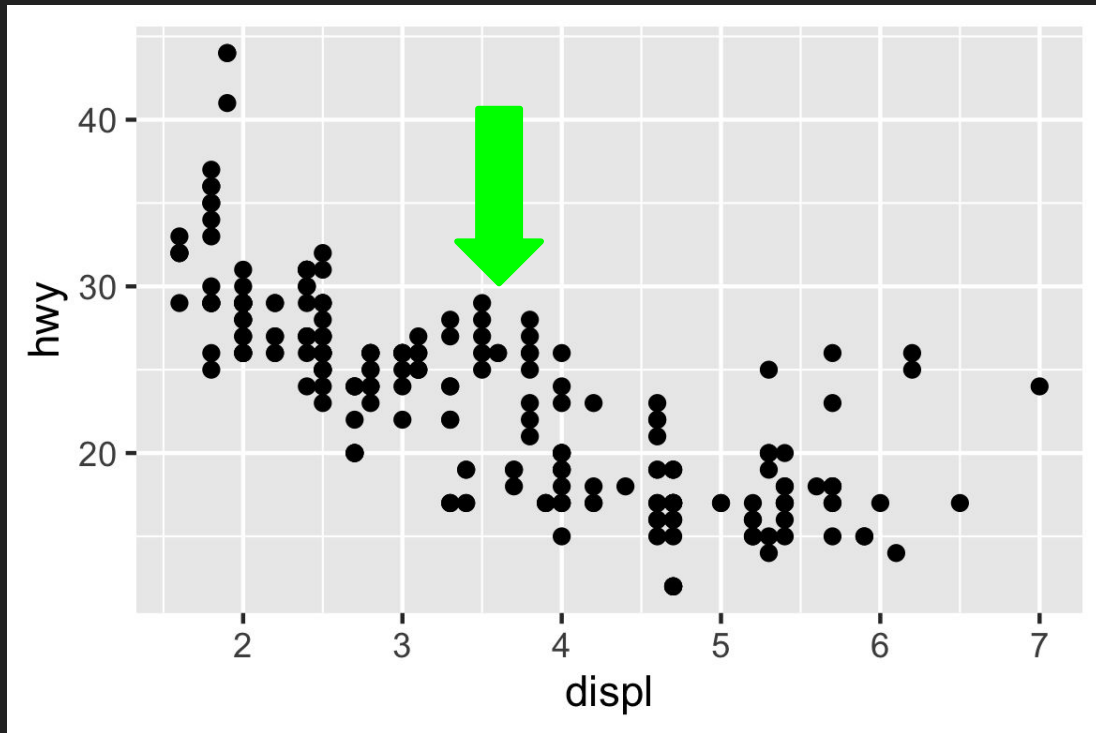
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point()
```



ggplot2: Elegant Graphics for Data Analysis,
by Hadley Wickham

<https://ggplot2-book.org/getting-started.html>

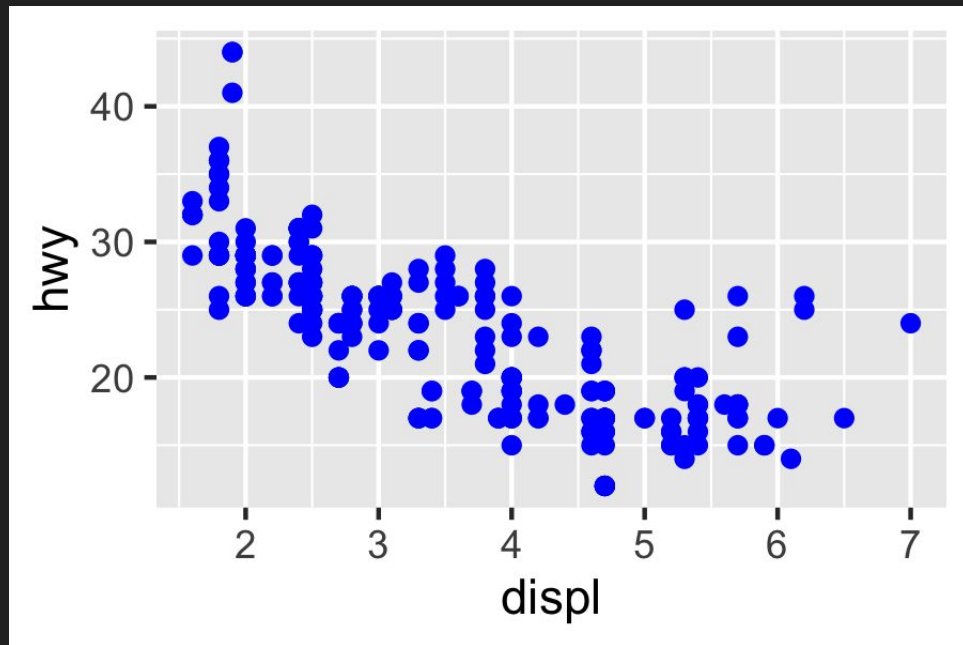
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point()
```



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




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



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)) +
```

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

```
>   geom_point(alpha = 0.2)
```

```
>   geom_point(size = 5, color = "red")
```

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

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

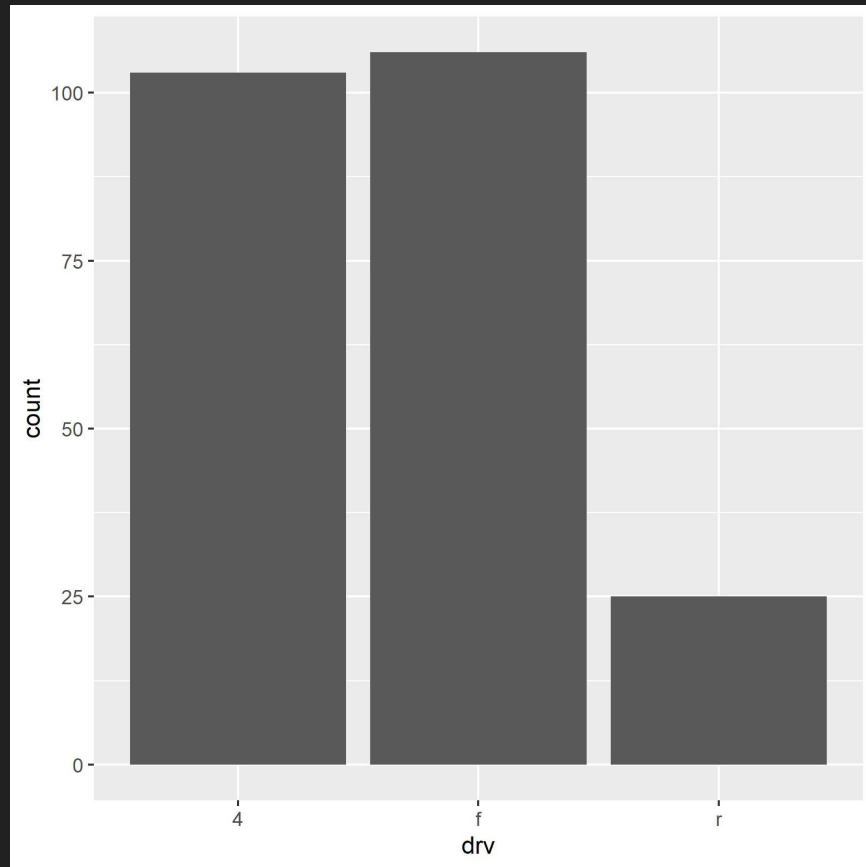
```
>   geom_point(shape = 15)
```

```
>   geom_point()
```

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

Some geometries only need
one variable.

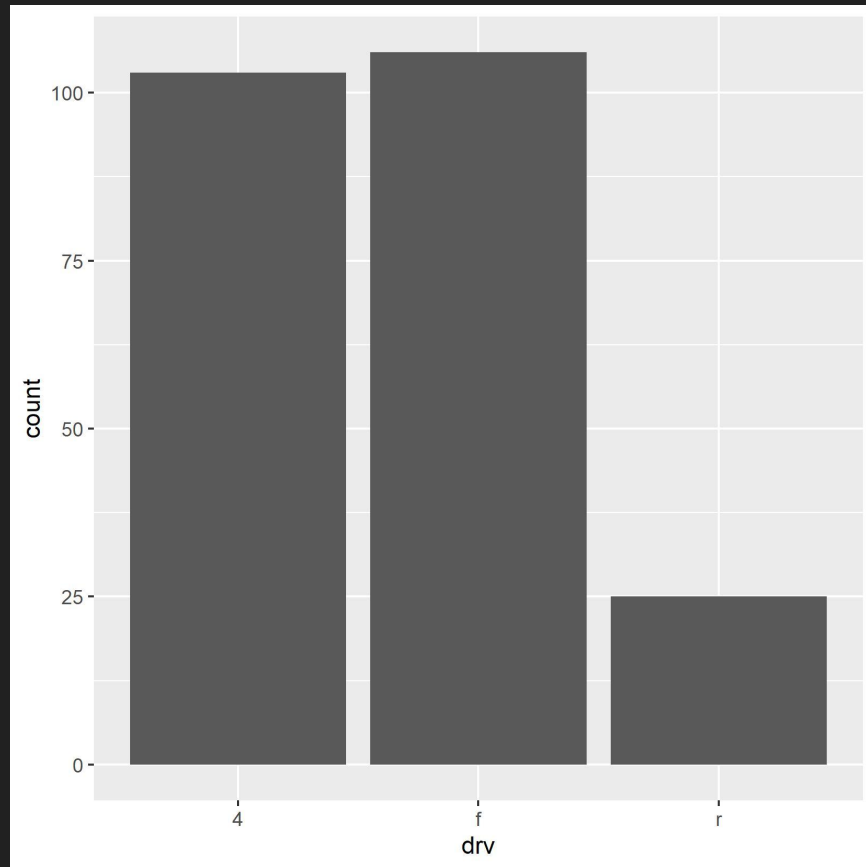
Bar Plots - One (Discrete)
Variable



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

Arguments may require discrete
(categorical) vs continuous variables.

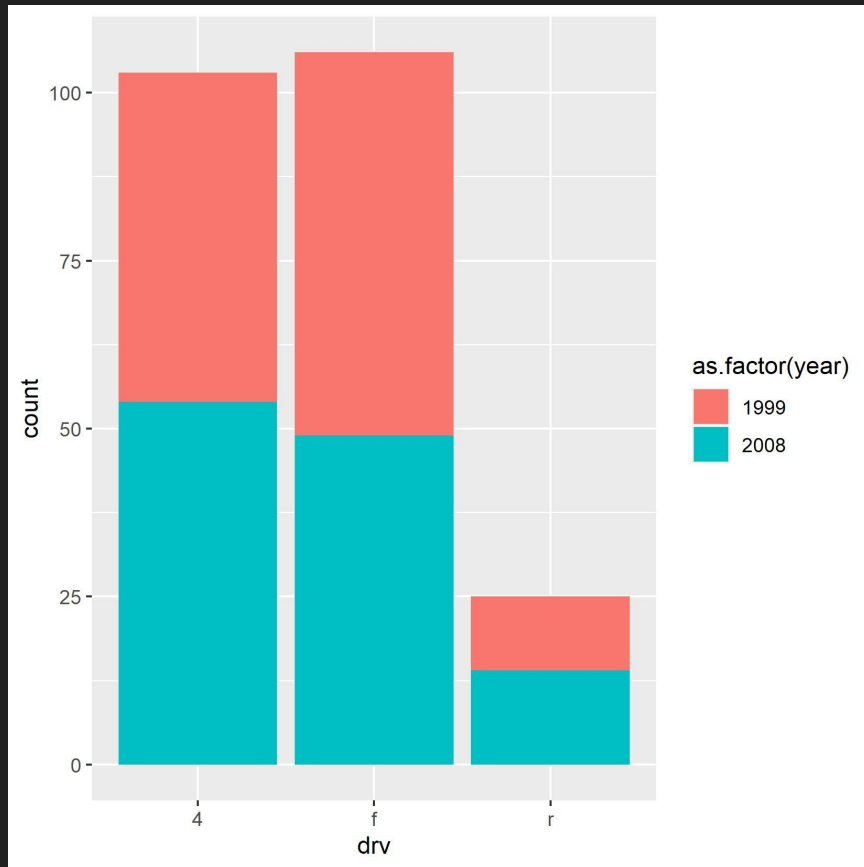
Fill must be categorical, but the data
has it as continuous.



```
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!

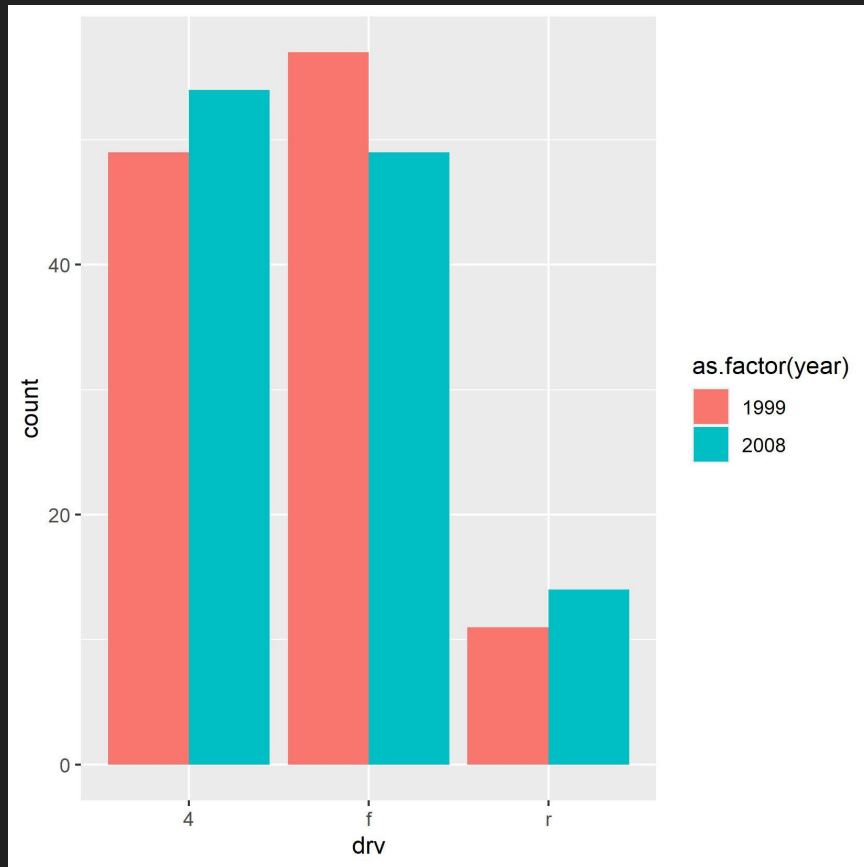
Note: For other plots, the color can be a continuous variable.



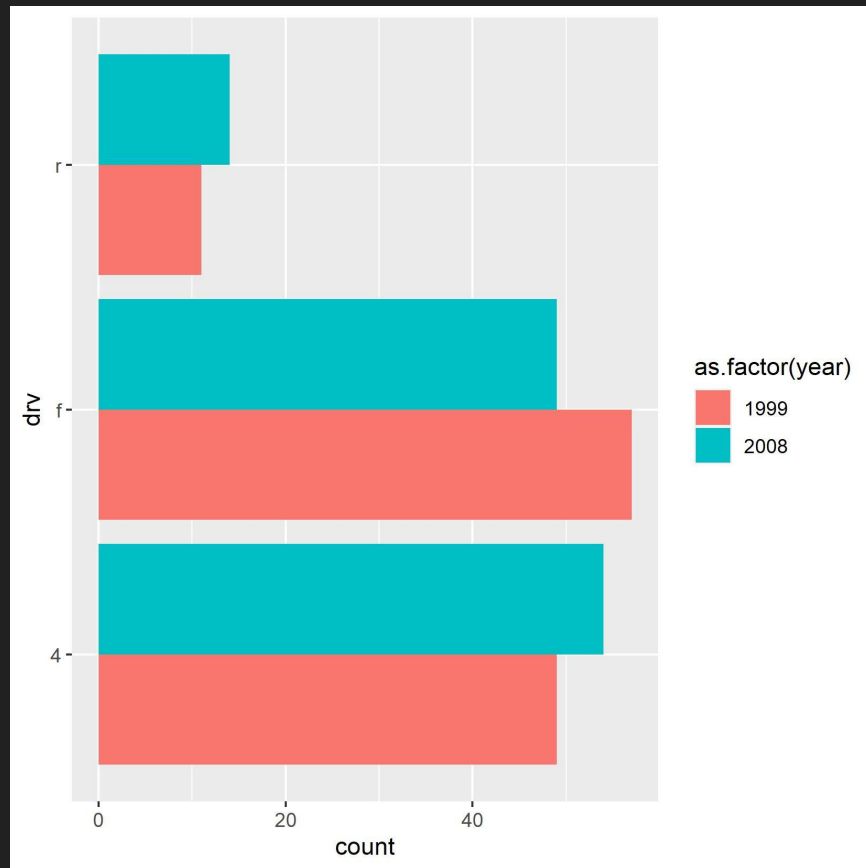
```
ggplot(mpg, aes(x = drv, fill = as.factor(year))) +  
  geom_bar(position = "dodge")
```



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

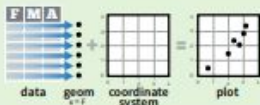
Data Viz Cheat Sheet

Data Visualization with ggplot2 Cheat Sheet



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.



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 = 2)) - 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

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`.

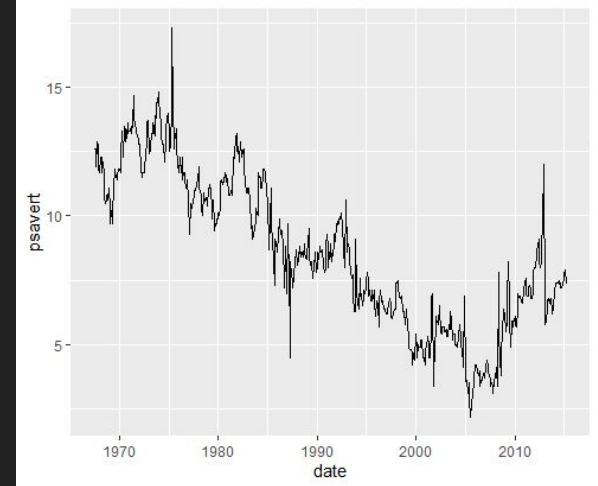
4. Create as many different types of plots for (2) as possible.

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()
```

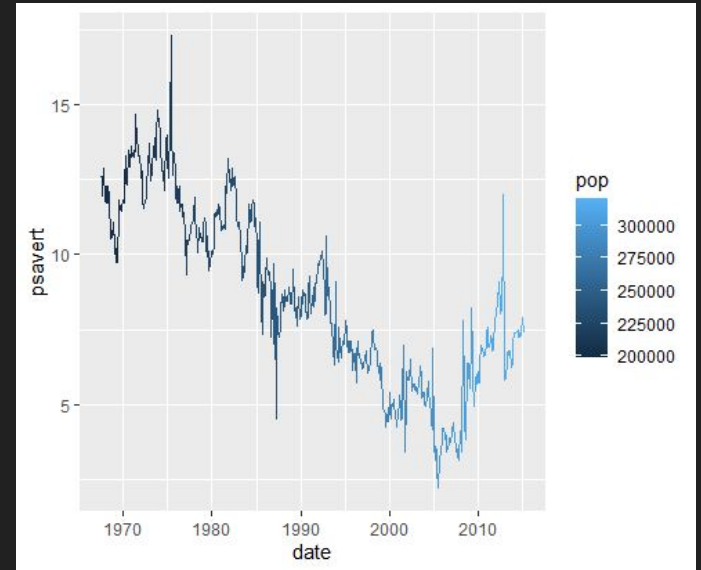


Economics Example - Solutions

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

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

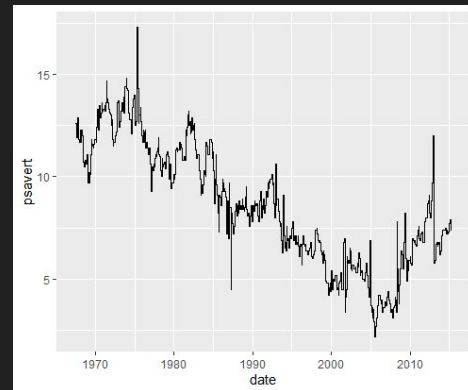
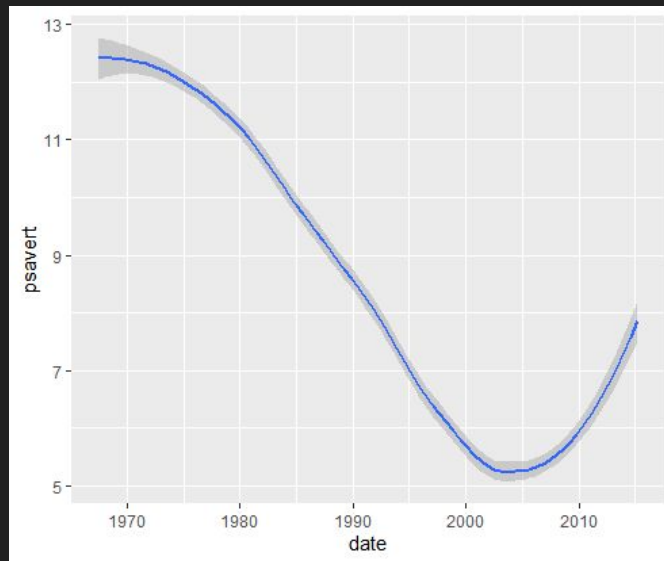
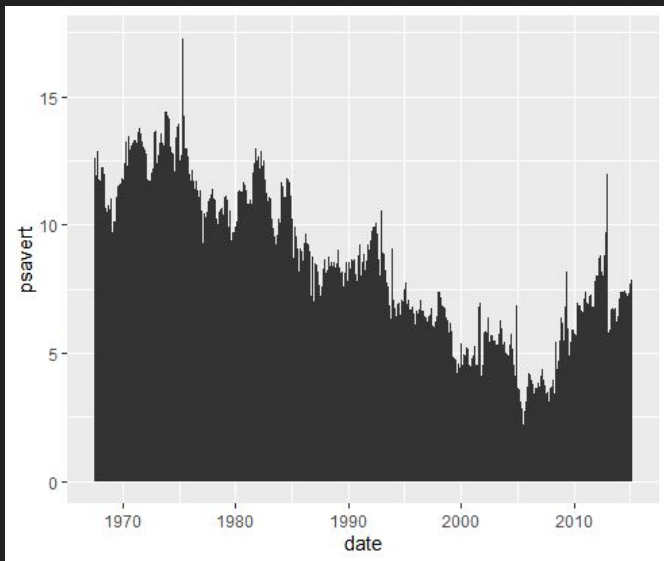
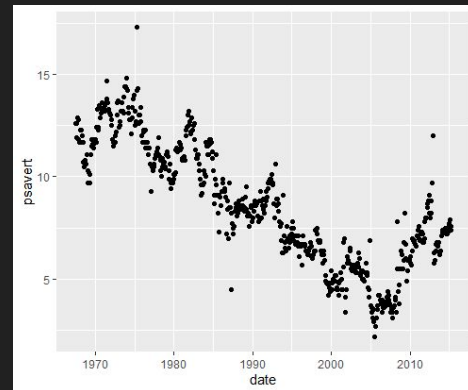
```
> geom_line()
```



Economics Example - Solutions

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

Essentially all the geometries for 2 continuous variables work!



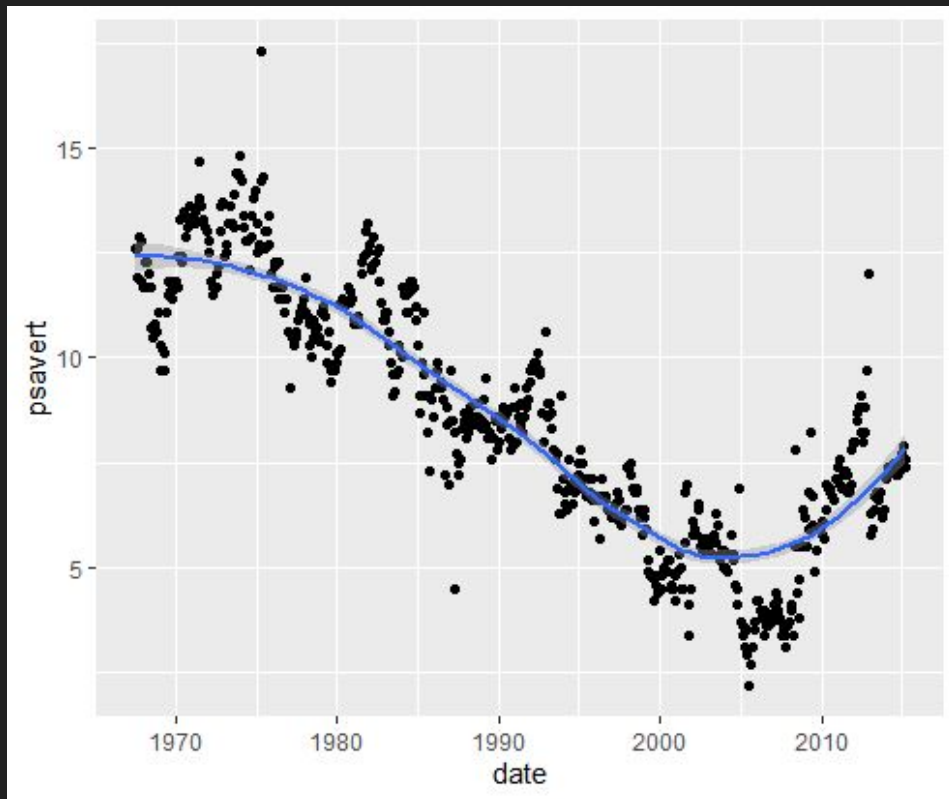
Economics Example - - Solutions

Note: You can overlap geoms!

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

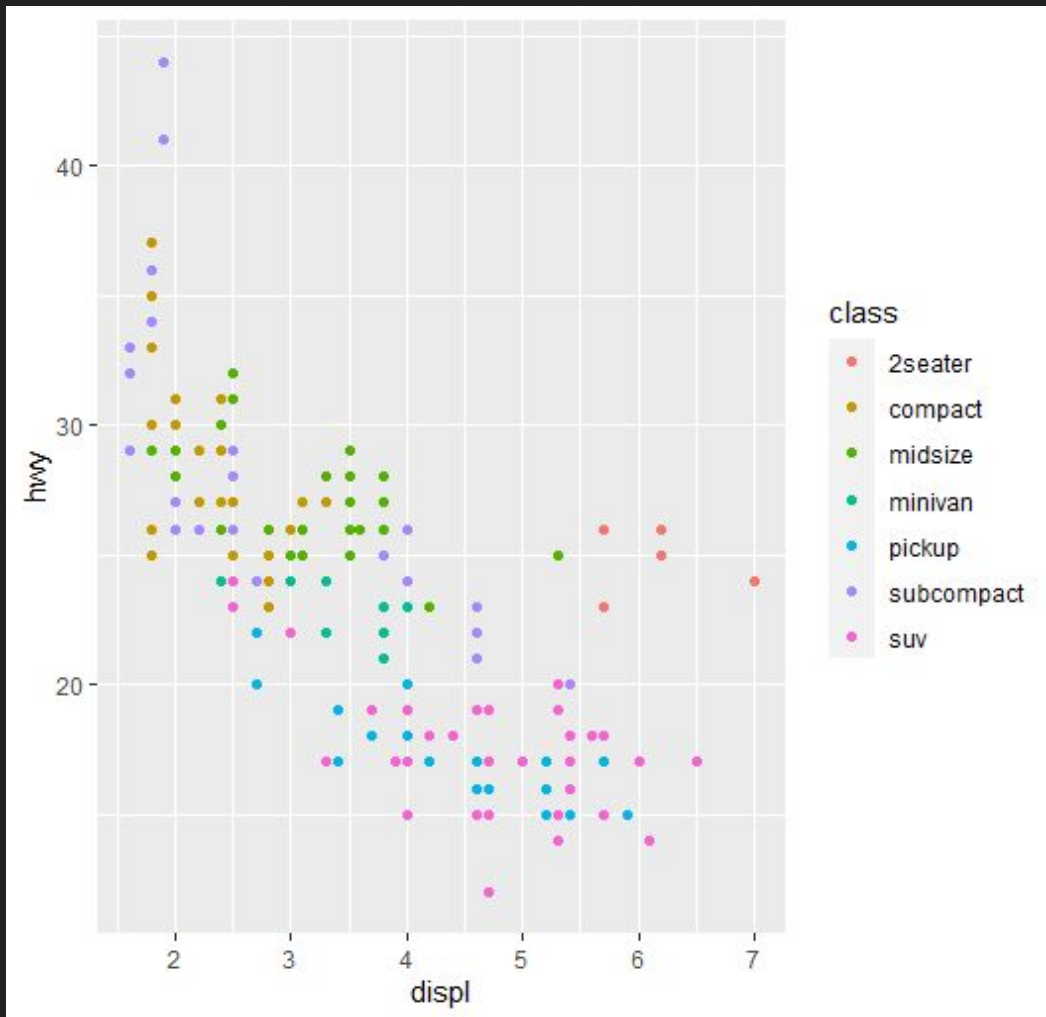
```
  geom_point() +
```

```
  geom_smooth()
```



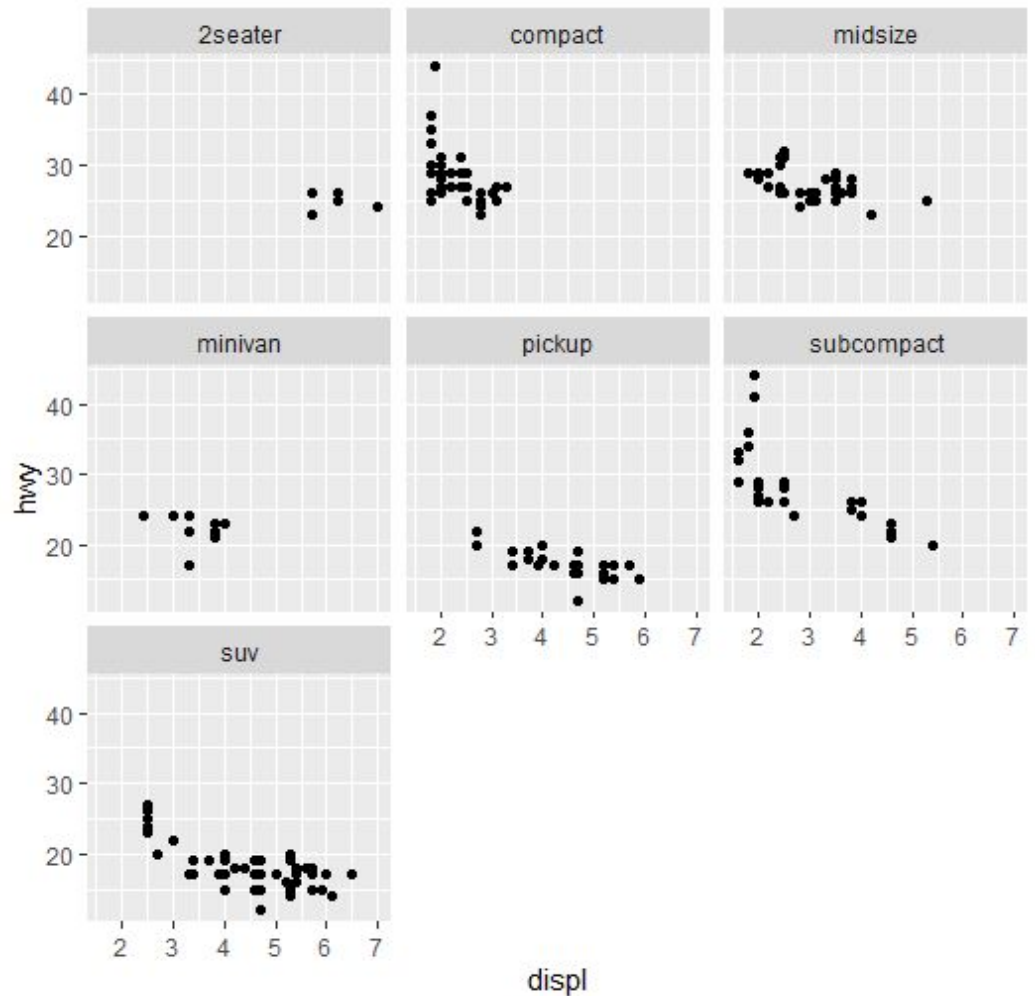
Facets

Grouping by color
may be hard to
differentiate



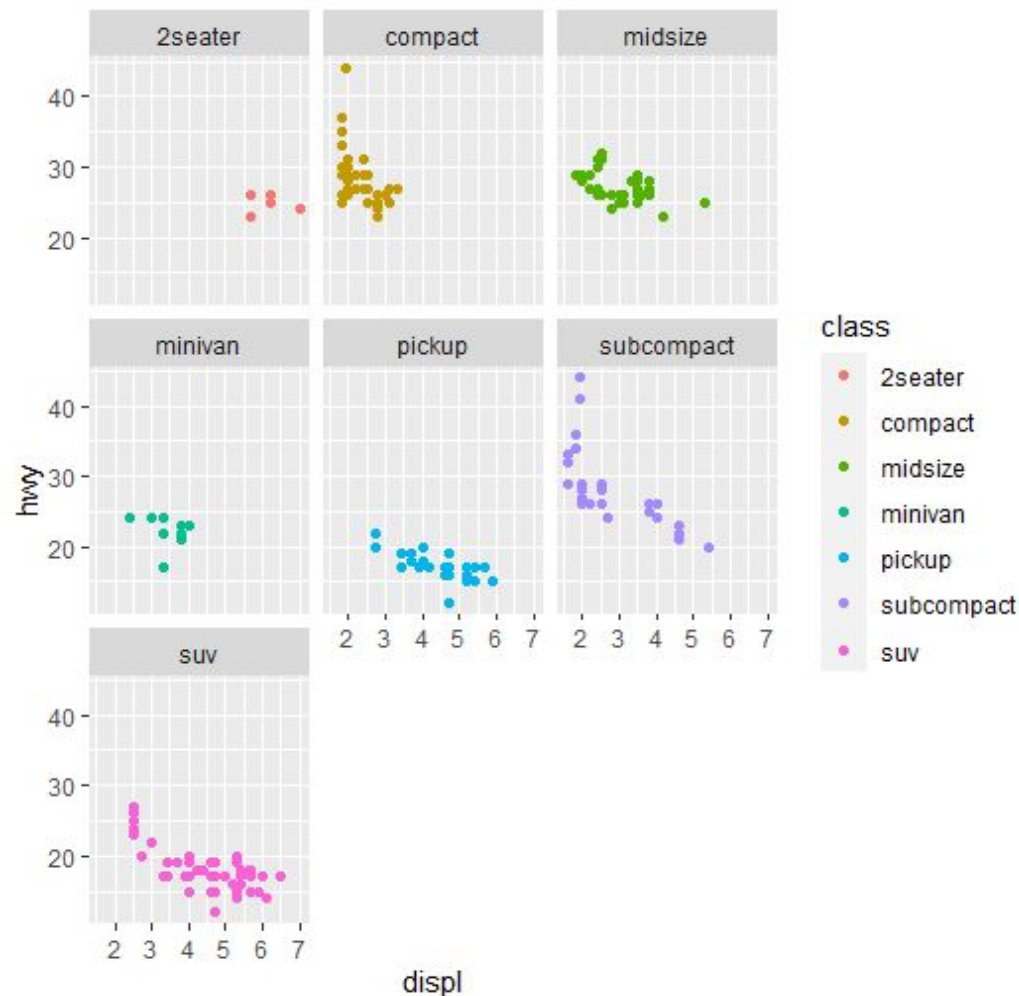
Facets

Facets clearly show groupings

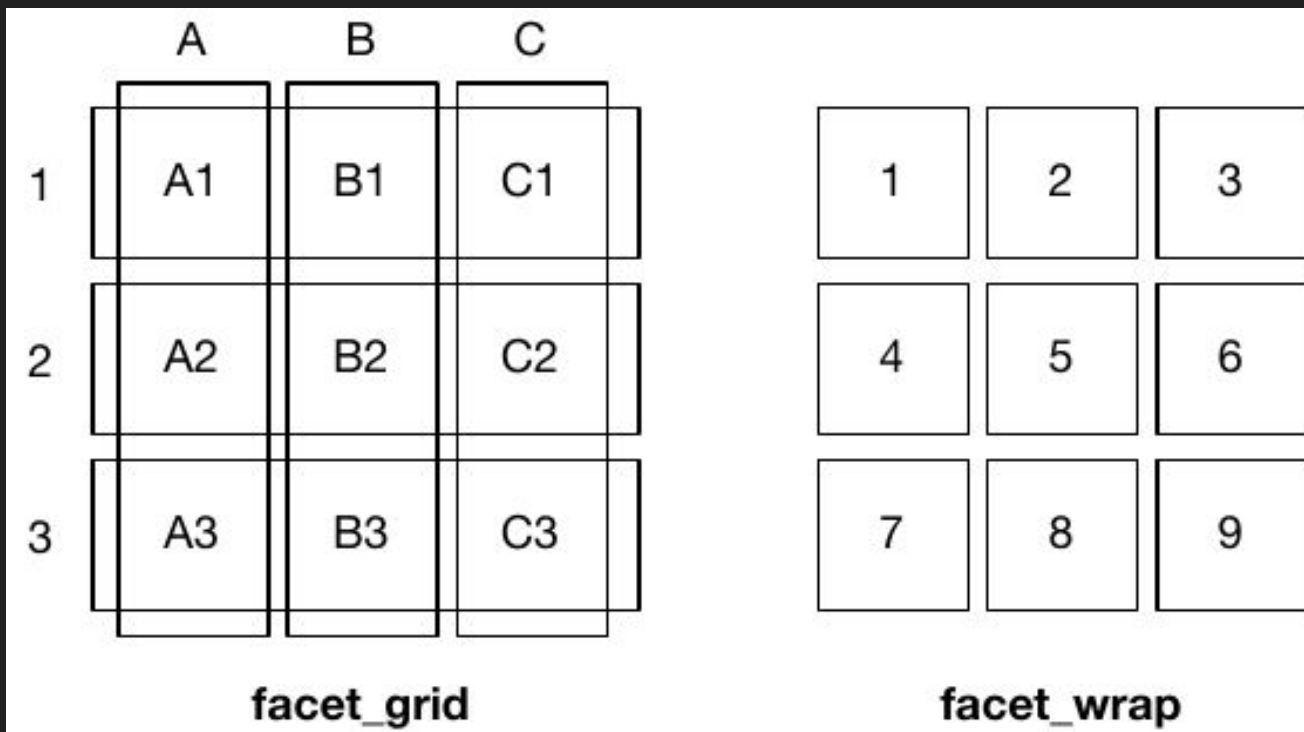


Facets

Facets clearly show groupings



Facets



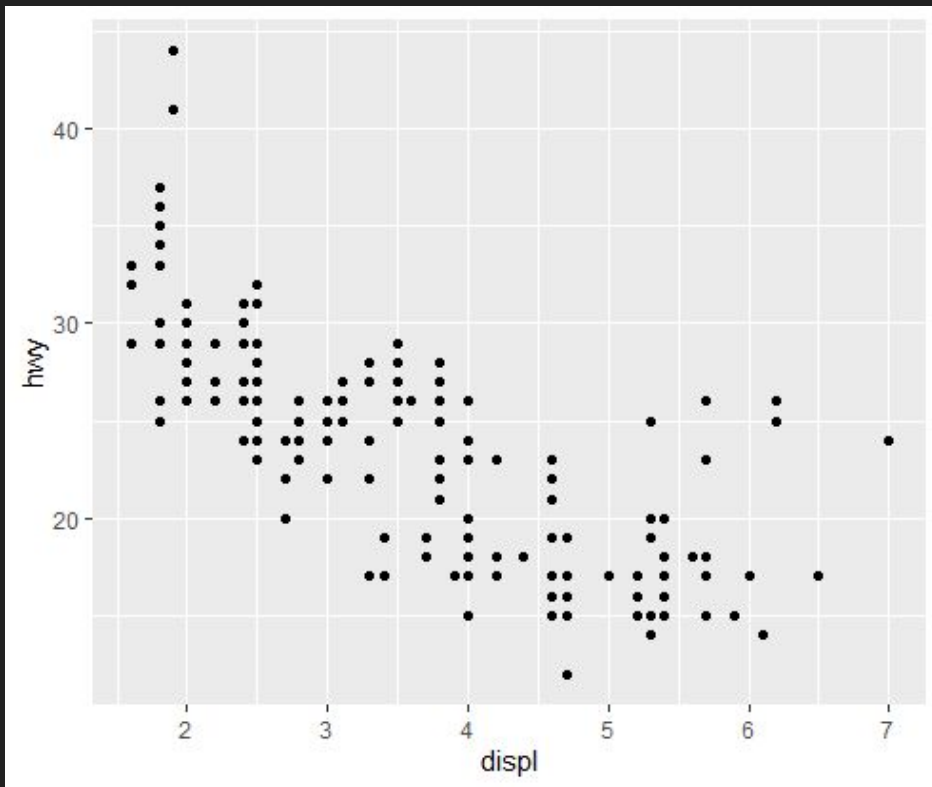
facet_wrap

... + facet_wrap(~x)

Split the plots by
variable x

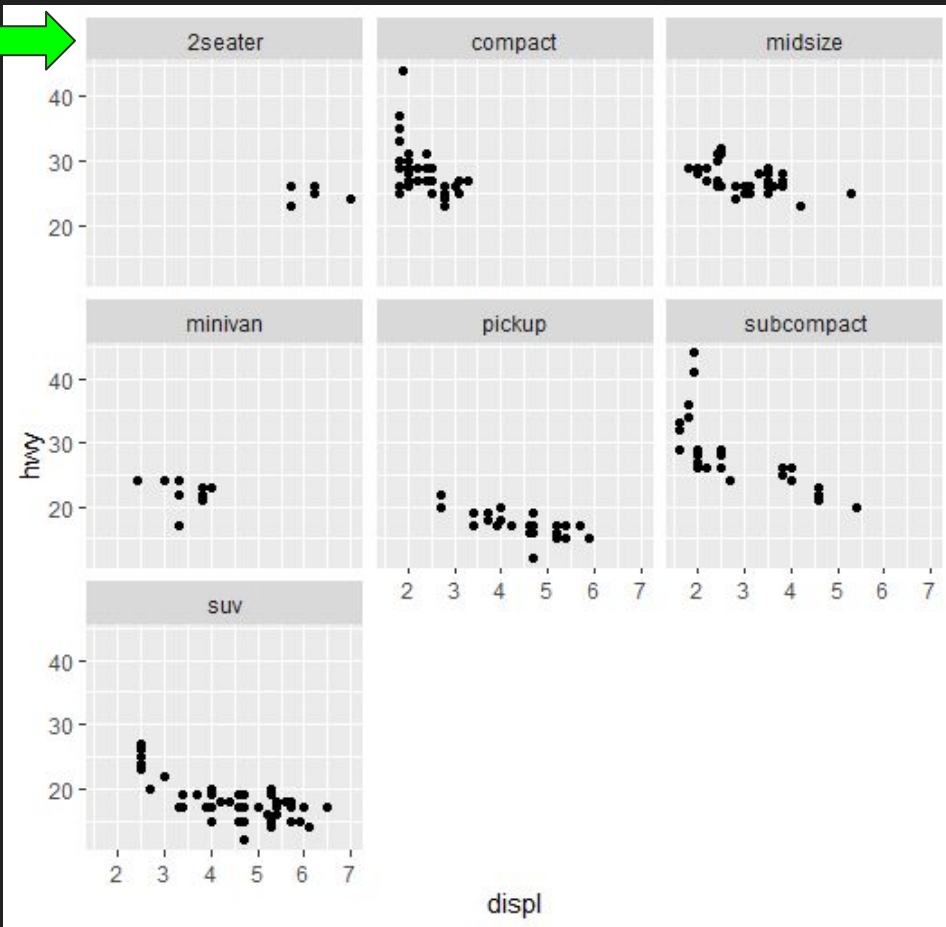
facet_wrap

```
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)



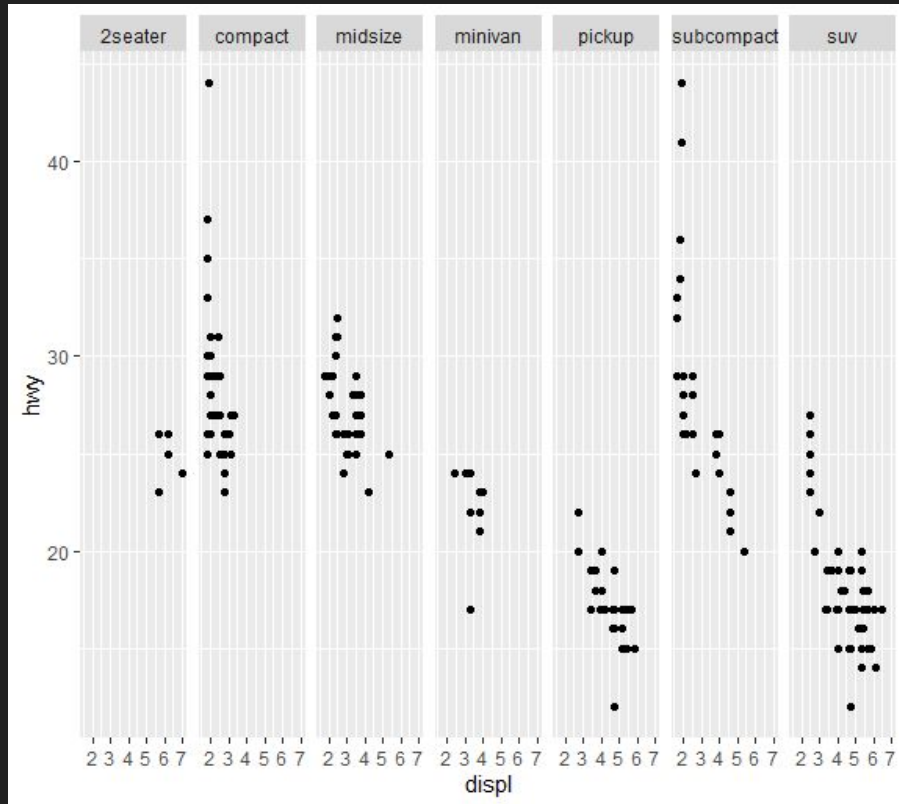
y variable on columns
x variable on rows

facet_grid

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



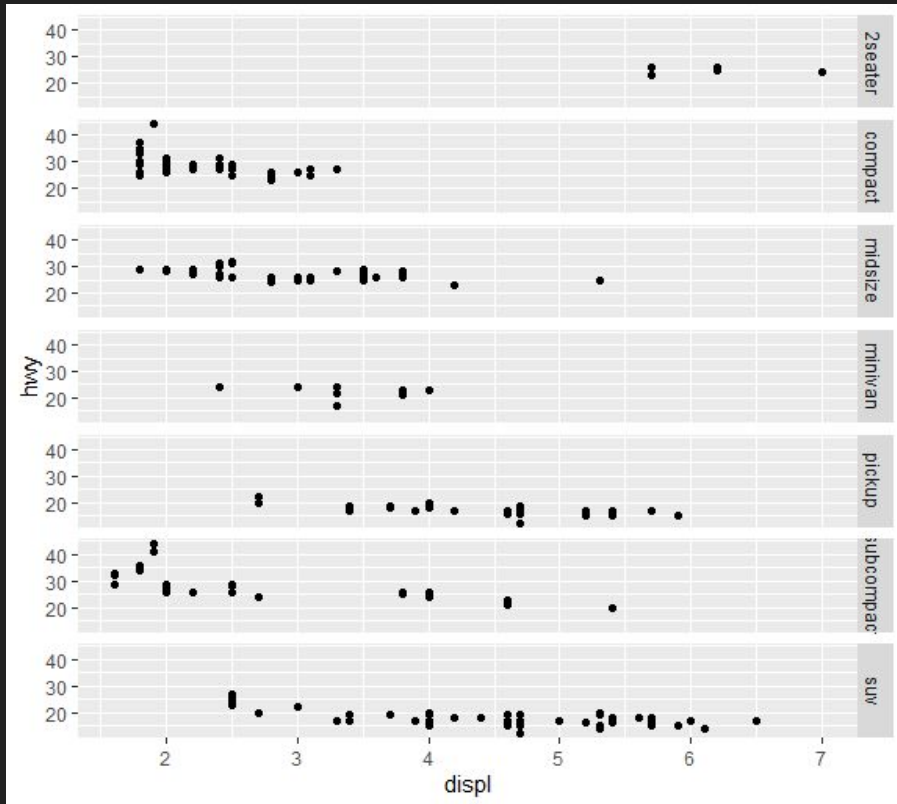
`class` as columns



1800



```
`class` as rows
```

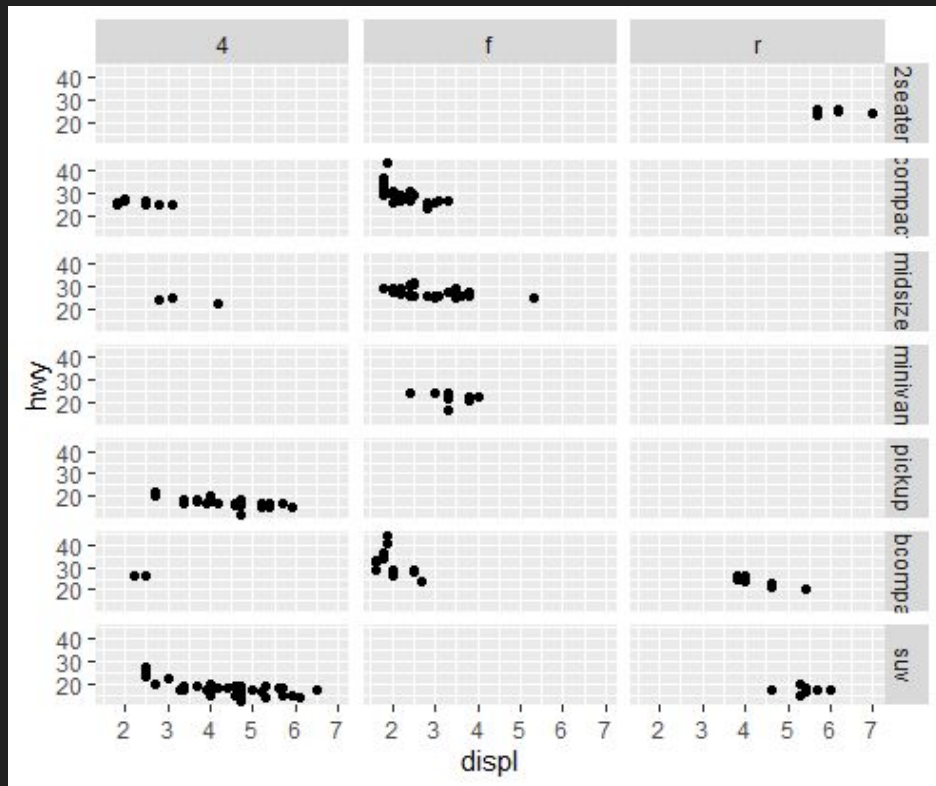


facet_grid

```
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 = “_____”)

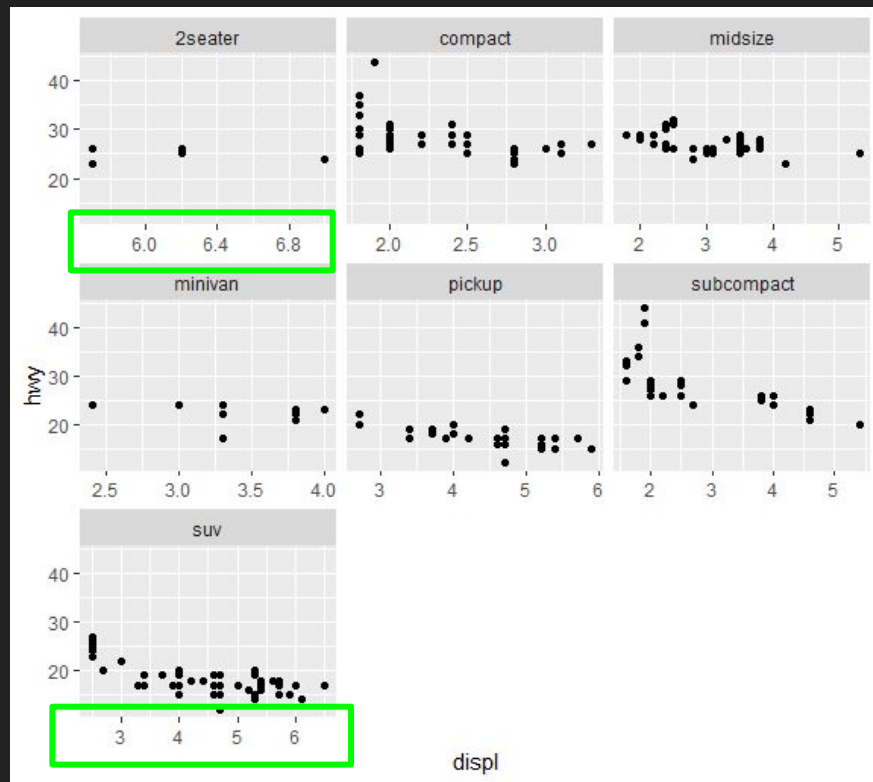
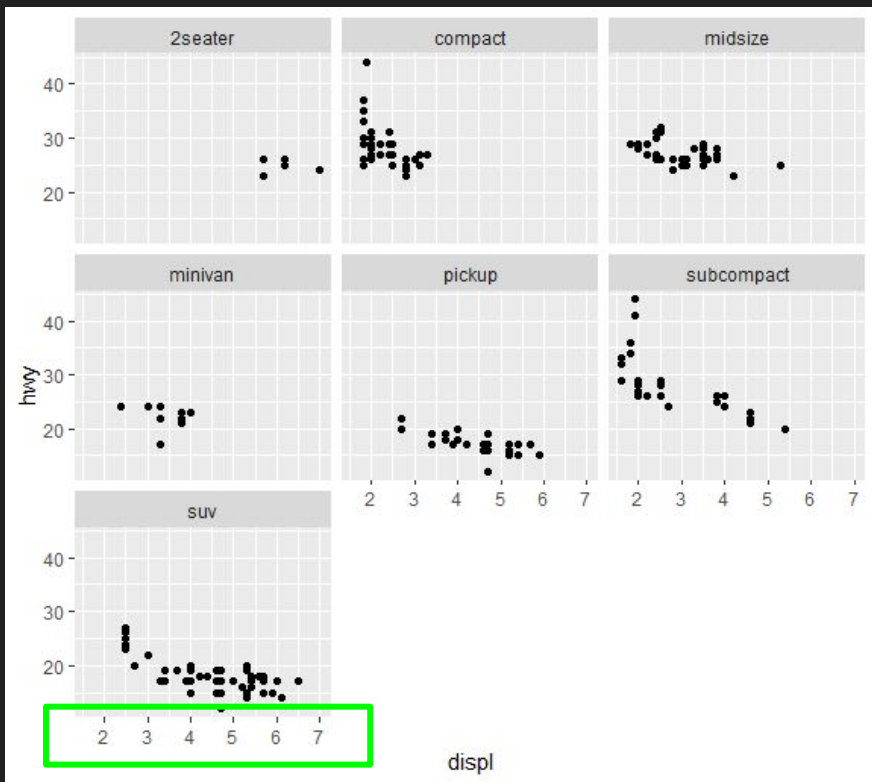
“fixed” = x and y fixed

“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()`.

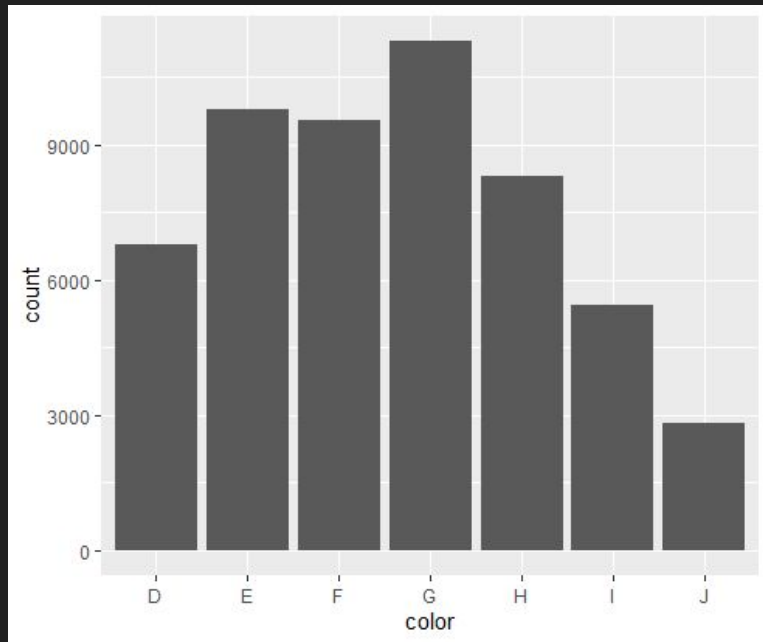
4. Set the facet scales to be “free_y”, and color by `cut`

Diamonds Example - Solutions

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

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

```
>   geom_bar()
```



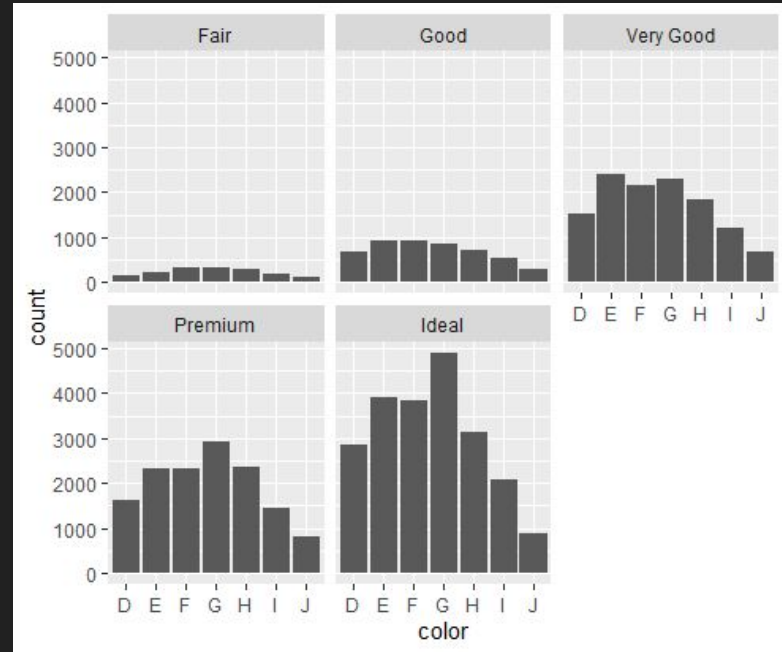
Diamonds Example - Solutions

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() +
```

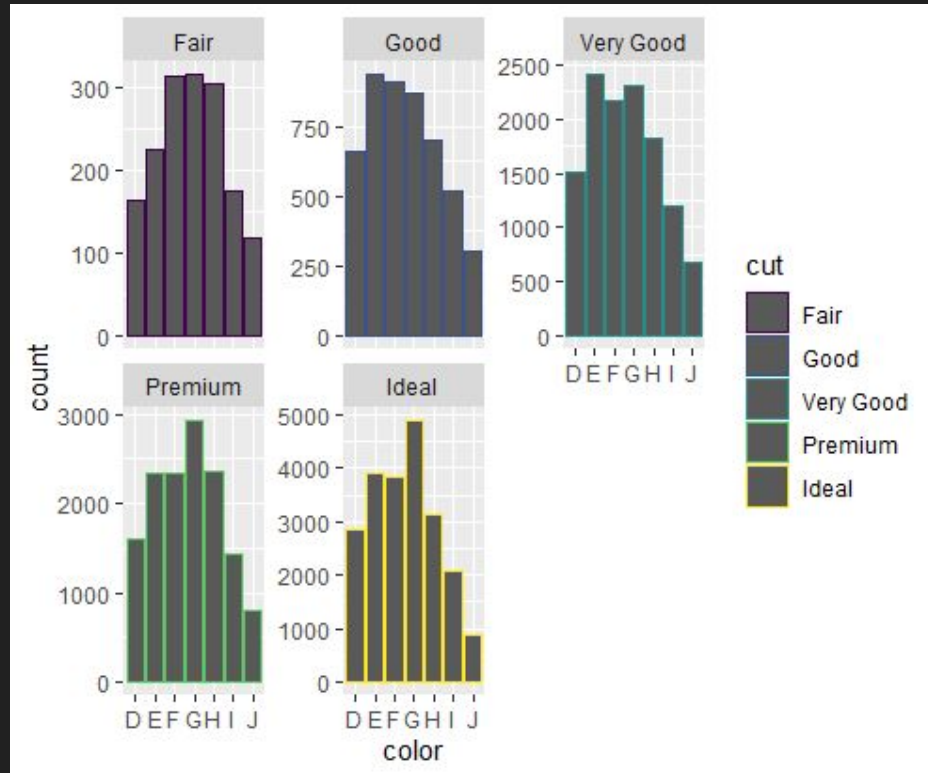
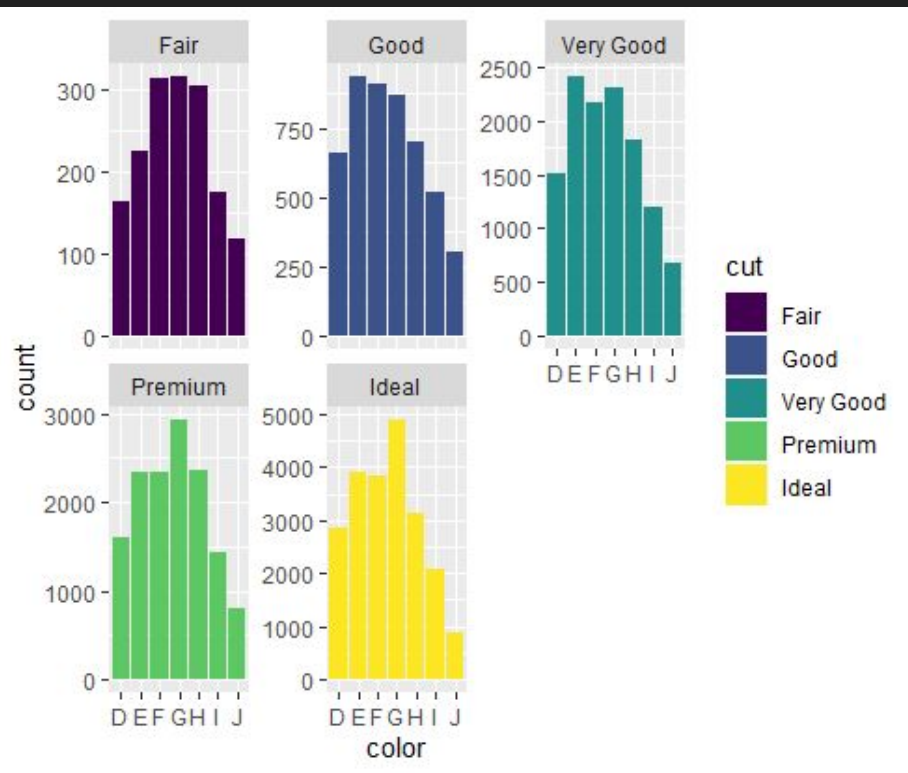
```
>   facet_wrap(~cut, scales = “free_y”)
```

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

```
>   geom_bar() +
```

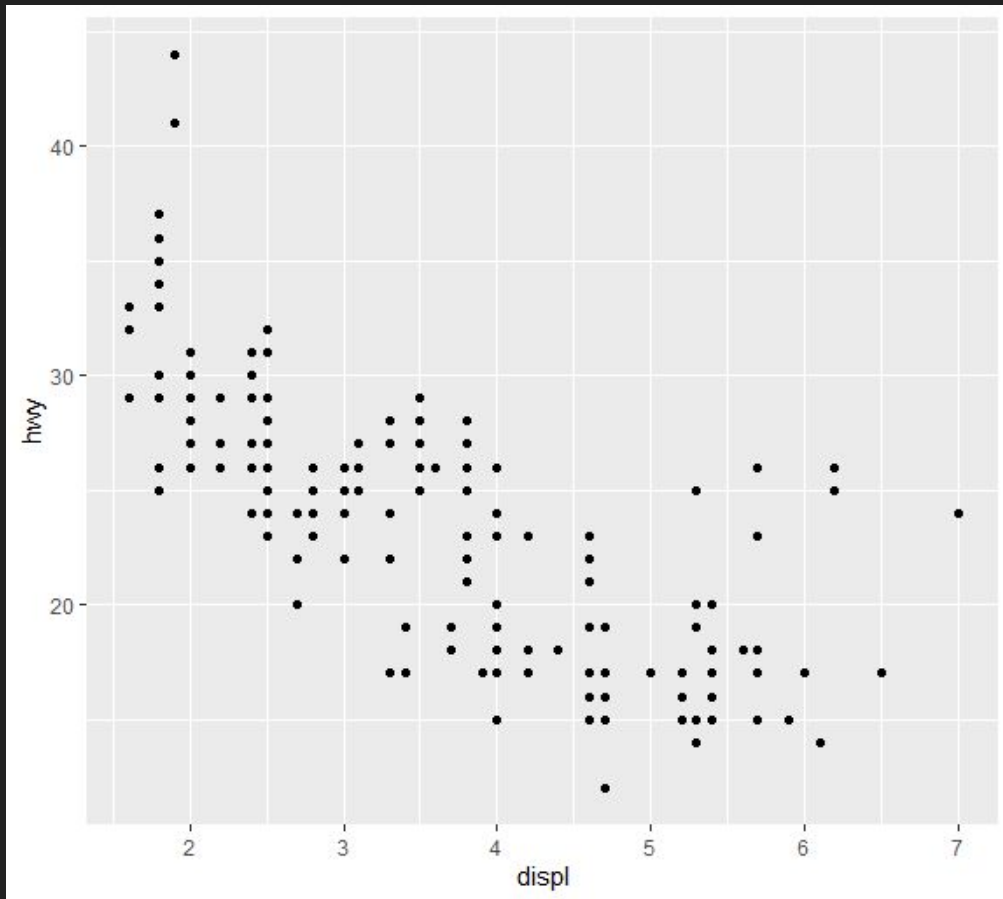
```
>   facet_wrap(~cut, scales = “free_y”)
```

Diamonds Example - Solutions



labs - Adding labels

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



labs - Adding labels

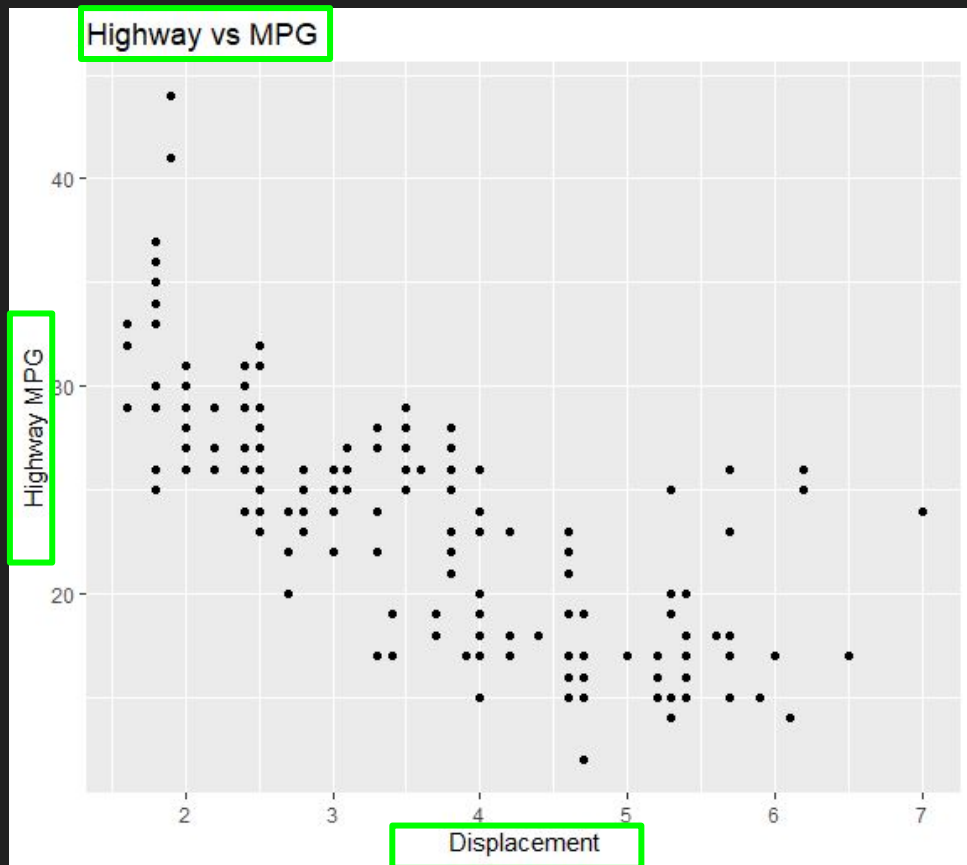
```
ggplot(mpg, aes(x = displ, y = hwy)) +
```

```
  geom_point() +
```

```
  labs(x = "Displacement",
```

```
       y = "Highway MPG",
```

```
       title = "Highway vs MPG")
```

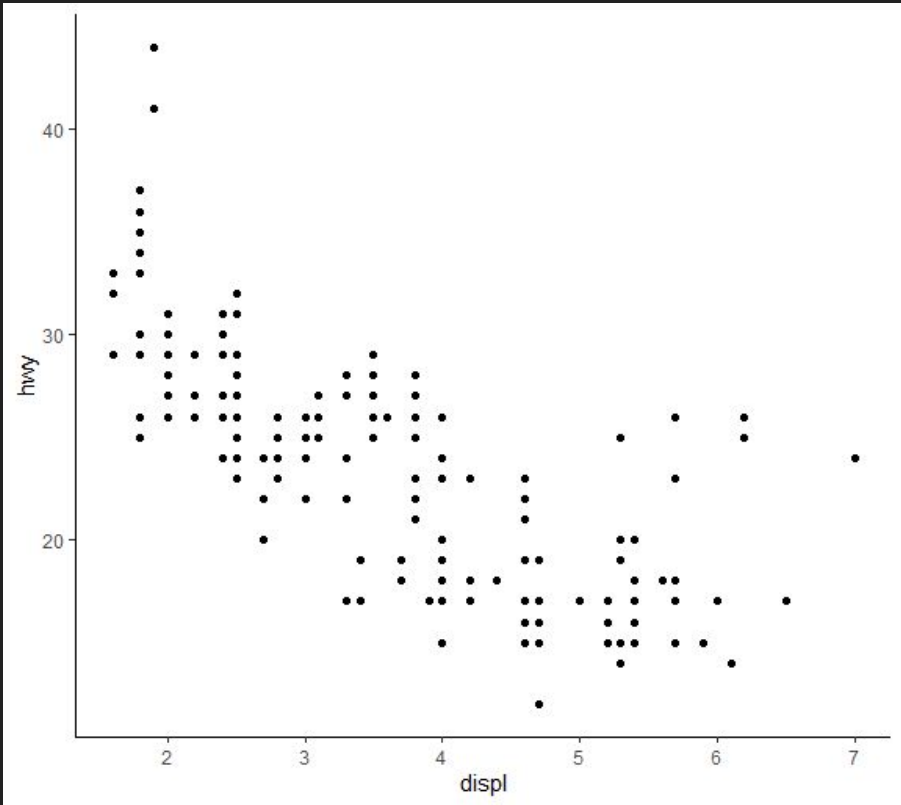
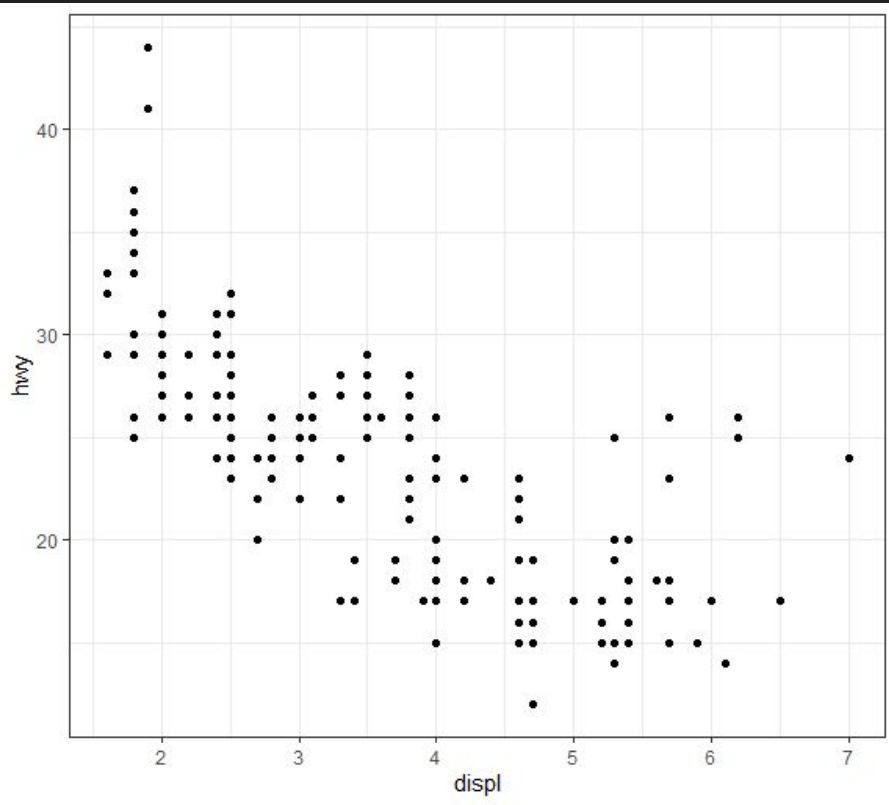


Themes

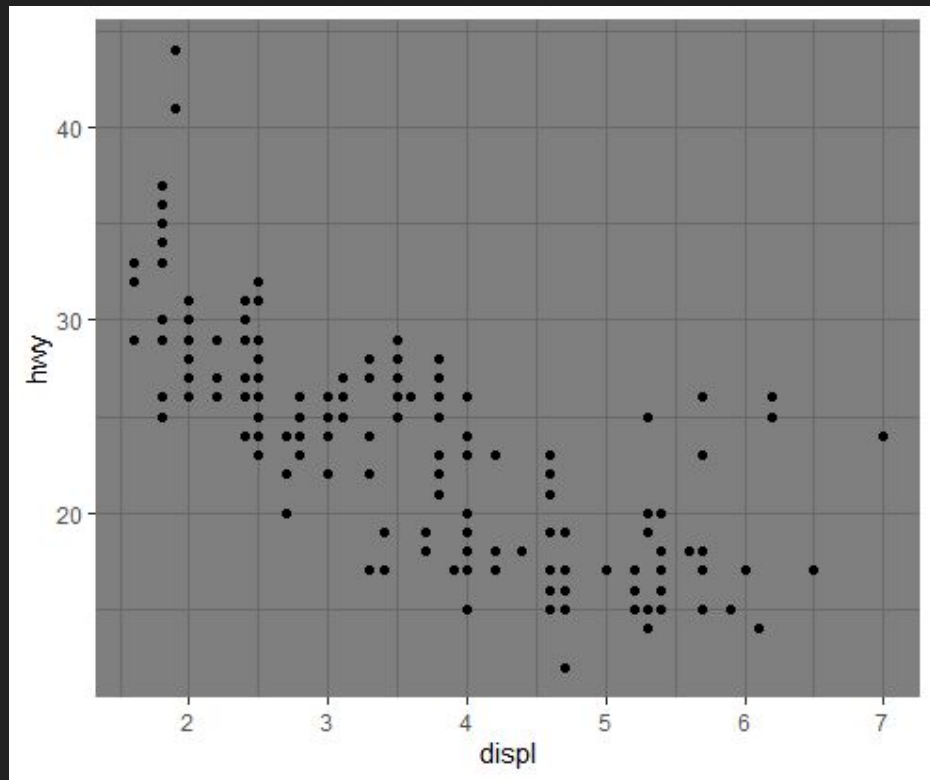
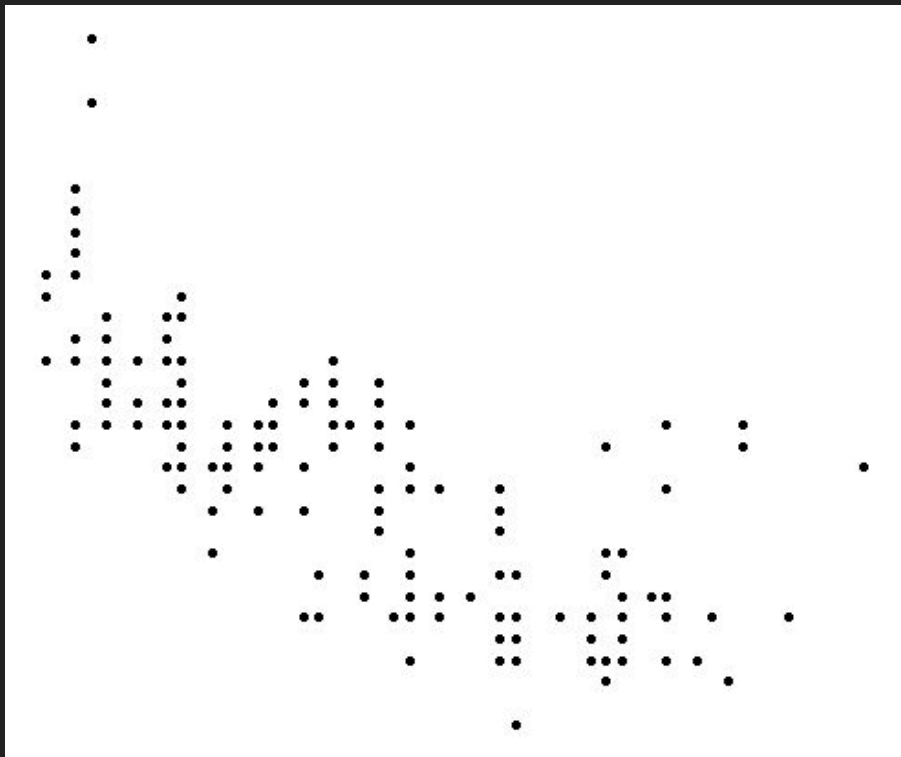
- `theme_bw()`
- `theme_linedraw()`
- `theme_light()`
- `theme_dark()`
- `theme_minimal()`
- `theme_classic()`
- `theme_void()`

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

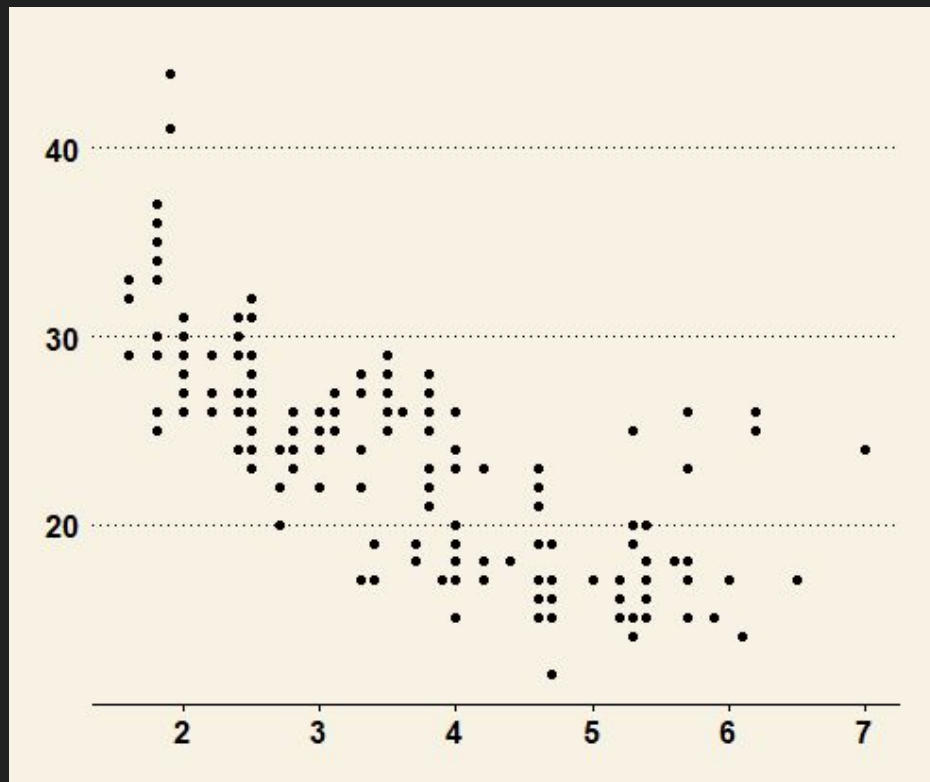
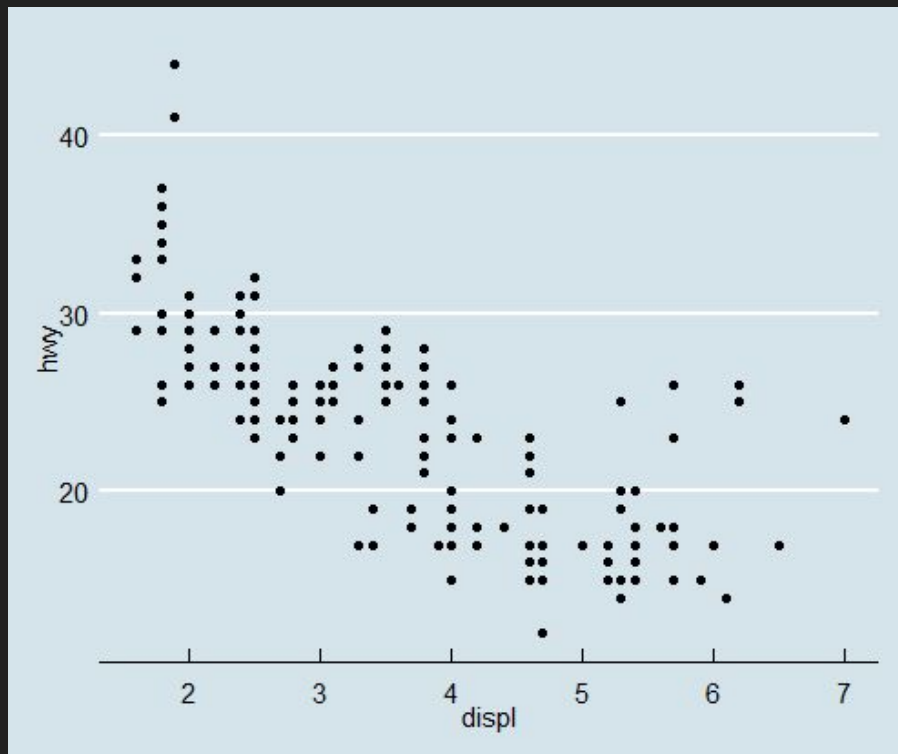
Themes



Themes



Themes - Extra themes from package `ggthemes`



CO2 Example

1. Load the `CO2` dataset (from `datasets` package):

```
> data(CO2)
```

```
> ?CO2 # To learn about the variables
```

You are a researcher trying to identify the cold tolerance of the grass species *Echinochloa crus-galli*.

You are interested in the differences in `conc` and `uptake` between `Plant`, `Type`, and `Treatment`.

2. Make plots to help you draw conclusions about the effect of `Plant`, `Type`, and `Treatment` on the two continuous variables.

Hints: `conc` and `uptake` are continuous, while the rest are discrete.

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



Steps moving forward

Explore opportunities to use R

- Classwork/research
- Take on projects of interest to you
- DataFest

Good Resources

- R for Data Science by Hadley Wickham & Garrett Grolemund
- Google's R Style Guide