

Advanced R Visualizations.

STAT UN2102 *Statistical Computing & Introduction to Data Science*

Gabriel Young
Columbia University

February 13, 2018

Base R Graphics

Some More Plotting with Base R

Recall,

- Visualization variation (of a single variable):
 - `hist()` – Histograms.
 - `barplot()` – Bargraphs.
- Visualizing covariation (of multiple variables):
 - `plot()` – Scatterplots.
 - `boxplot()` – Boxplots (box-and-whisker plots).

The `plot()` function.

- The foundation of many of R's graphics functions.
- Often one builds up the graph in stages with `plot()` as a base.
- Each call to `plot()` begins a new graph window.
- Takes arguments, called *graphical parameters*, to change various aspects of the plot. (`?par`)

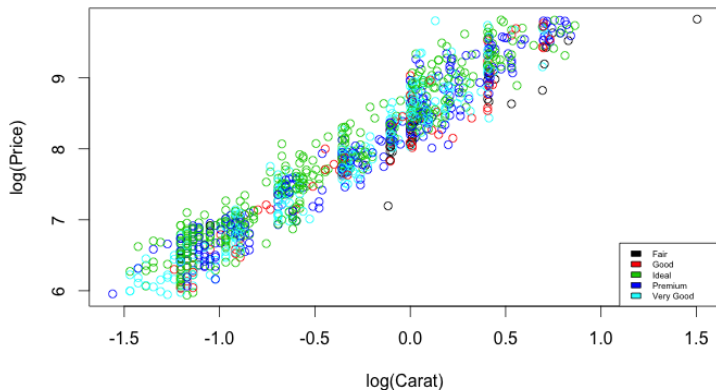
Diamonds Dataset

- Recall the diamonds data set. (diamonds.csv)
- Run `diamonds <- read.csv("diamonds.csv", as.is = TRUE)`.

```
> diamonds <- read.csv("diamonds.csv", as.is = T)
> diamonds$cut <- factor(diamonds$cut)
> diamonds$color <- factor(diamonds$color)
> diamonds$clarity <- factor(diamonds$clarity)
> set.seed(1)
> rows <- dim(diamonds)[1]
> diam <- diamonds[sample(1:rows, 1000), ]
```

Building a Visualization: An Example

```
> plot(log(diam$carat), log(diam$price), col = diam$cut)
> legend("bottomright", legend = levels(diam$cut),
+       fill = 1:length(levels(diam$cut)), cex = .5)
```

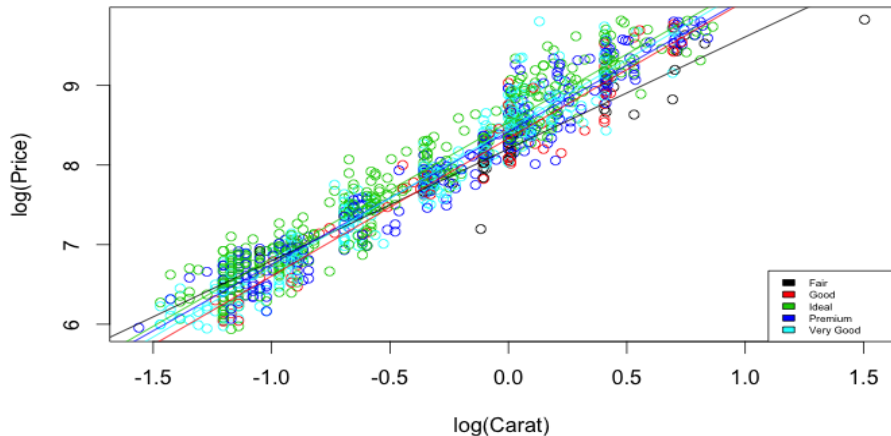


Building a Visualization: An Example

Let's instead plot a regression line for each cut separately.

```
> cuts      <- levels(diam$cut)
> col_counter <- 1
> for (i in cuts) {
+   this_cut    <- diam$cut == i
+   this_data   <- diam[this_cut, ]
+   this_lm     <- lm(log(this_data$price)
+                     ~ log(this_data$carat))
+   abline(this_lm, col = col_counter)
+   col_counter <- col_counter + 1
+ }
```

Building a Visualization: An Example



Check Yourself

Exercise:

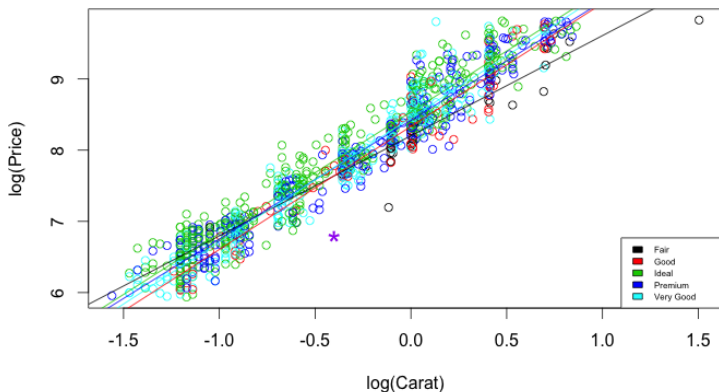
Use the built-in `iris` dataset.

- Create a new column `Setosa` that takes a 1 if the iris is a setosa and a 0 otherwise.
- Plot `iris Sepal.Width` on the x-axis and `Sepal.Length` on the y-axis. Color the points according to whether the iris is a setosa or not.
- Plot two regression lines on the plot, one for the setosa iris and one for non-setosa iris.

Building a Visualization: An Example

We add a new point for a diamond that is \$898 and 0.67 carats.

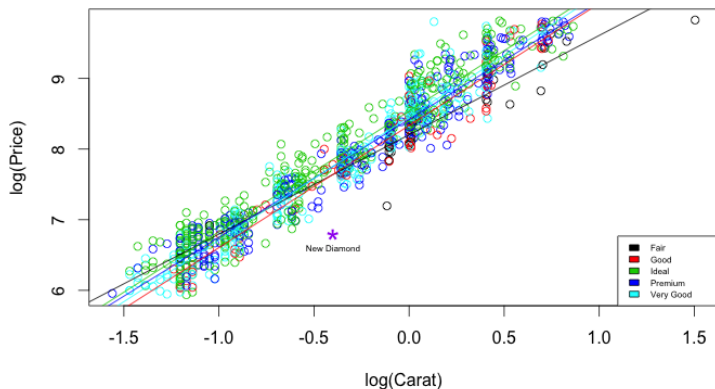
```
> points(-0.4, 6.8, pch = "*", col = "purple")
```



Building a Visualization: An Example

We add text to the new point we just added.

```
> text(-0.4, 6.8 - .2, "New Diamond", cex = .5)
```



Useful Graphical Parameters

The table below lists a selection of R's graphical parameters. More info at <http://www.statmethods.net/advgraphs/parameters.html> or using `?par`.

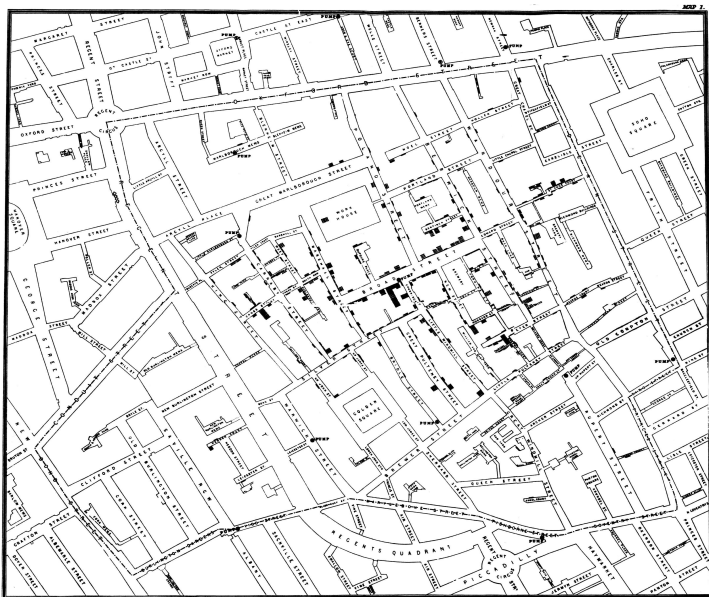
Parameter	Description
<code>pch</code>	<i>Point Character</i> . Character of the points in the plot.
<code>main</code>	Title of the plot.
<code>xlab, ylab</code>	Axes labels.
<code>lty</code>	<i>Line Type</i> . E.g. 'dashed', 'dotted', etc.
<code>lwd</code>	<i>Line Width</i> . Line width relative to default = 1.
<code>cex</code>	<i>Character Expand</i> . Character size relative to default = 1.
<code>xlim, ylim</code>	The limits of the axes.
<code>mfrow</code>	Plot figures in an array (e.g. next to each other).
<code>col</code>	Plotting color.

Data Visualization

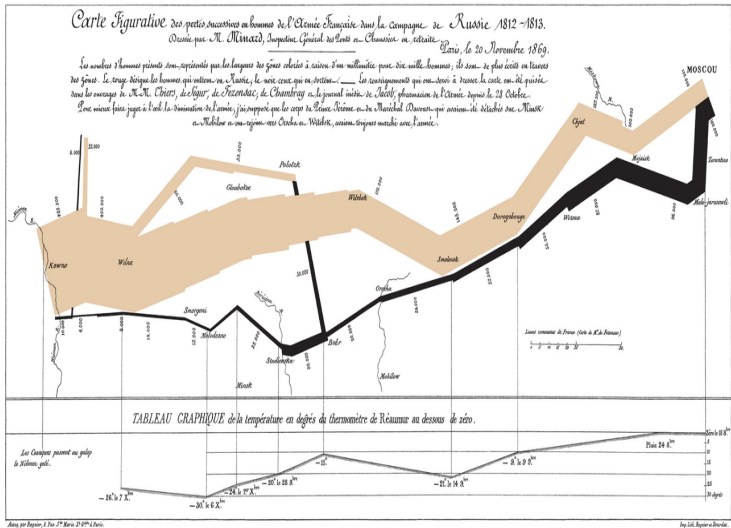
Good Visualizations

In data science, good visualizations should give you more information than you can see in just the data table itself.

Good Visualizations - John Snow 1854 (Wikipedia)



Good Visualizations - Charles Joseph Minard 1896 (Wikipedia)



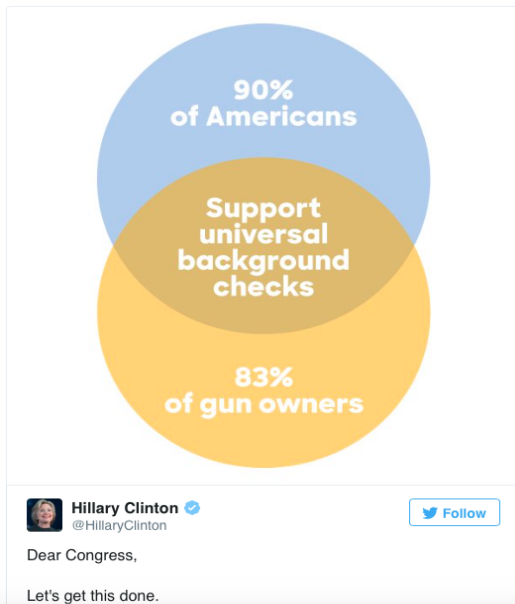
Good Visualizations - Charles Joseph Minard 1896 (Wikipedia)

Minard Graph

Minard shows six variables:

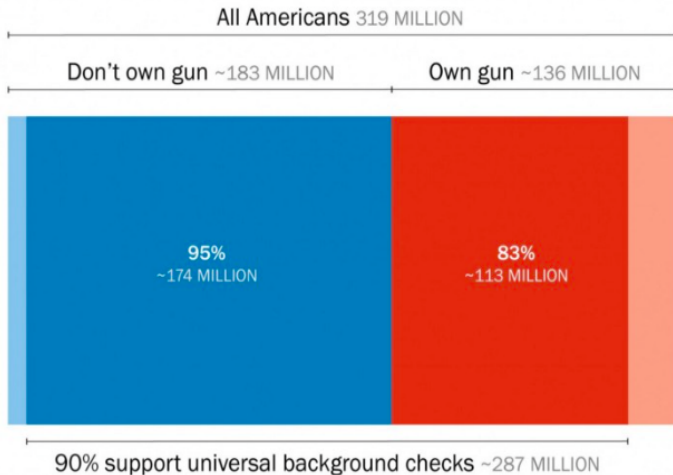
- Number of soldiers,
- Direction of the march,
- Location coordinates,
- Temperature on the return journey,
- Location on dates in November and December.

Bad Visualizations - Hillary Clinton



Improving the Clinton campaign's terrible graph

Population estimates from the Census Bureau. Gun ownership estimate based on calculations from Gallup compared with Census household data. Percentages based on Clinton campaign figures.



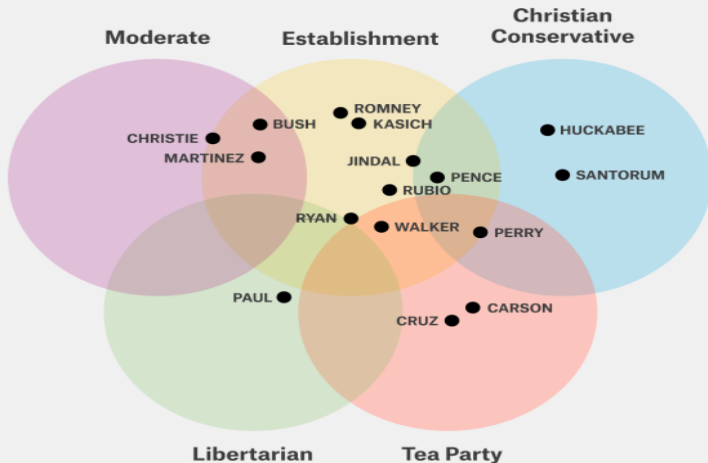
Bad Visualizations

Even statisticians are sometimes bad at making visualizations!

Bad Visualizations - Nate Silver

The Republicans' Five-Ring Circus

A graphic conception of the GOP field



Bad Visualizations - Nate Silver

Candidate	Moderate	Establishment	Christian Conservative	Libertarian	Tea Party
Bush	X	X			
Carson					X
Christie	X	X			
Cruz					X
Huckabee			X		
Jindal		X	X		
Kasich		X			
Martinez	X	X			
Paul				X	
Pence		X	X		
Perry		X	X		X
Romney		X			
Rubio		X			
Ryan		X			
Santorum			X		
Walker		X			X

Check Yourself – Good or Bad?

National Household Survey (NHS)

Search NHS

By topic

By geography

Products

Analytical products

Data products

Reference products

Other links

NHS corrections/updates

Accessing my NHS information

Census

Geography

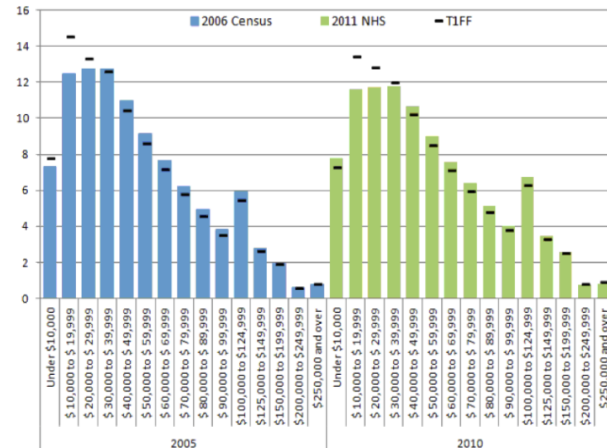
Open data

Figure 2

Distribution of after-tax income of census family units for Canada, 2005 and 2010

[Description for figure 2](#)

percentage



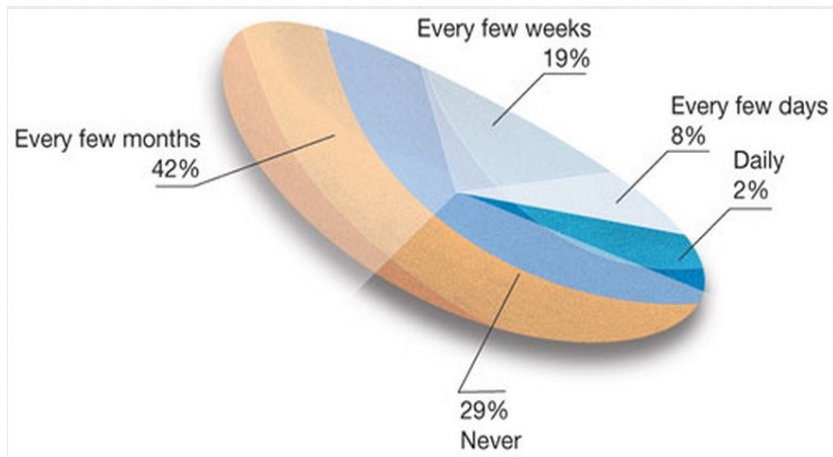
Bad Visualizations

Pie charts are a poor way to illustrate information.

Bad Visualizations

Pie charts are a poor way to illustrate information.

Especially this one:



Identify what the following function does.

Code

```
> pie.chart <- function(data) {  
+   print("I suck")  
+ }
```

Task

Identify what the following function does.

Code

```
> pie.chart <- function(data) {  
+   print("I suck")  
+ }
```

The `pie.chart` function prints "I suck"

```
> pie.chart(c("Red", "Red", "Blue"))  
[1] "I suck"
```

Good Visualizations

- Keep things simple in terms of color and presentation!
- Try not adding non-needed dimensions to a plot, i.e., 3D bar chart describing one categorical variable.
- Showing more dimensions on lower a dimensional plot is encouraged, i.e, diamond price versus carat split by cut.
- Barcharts are a better way to summarize categorical data compared to piecharts.

Advanced Visualization Techniques

- R has several systems for making graphs (we've looked at the base R functions).
- ggplot2 is one of the most elegant and flexible.
- ggplot2 uses a coherent system (or 'grammar') for describing and building graphs.

- R has several systems for making graphs (we've looked at the base R functions).
- ggplot2 is one of the most elegant and flexible.
- ggplot2 uses a coherent system (or 'grammar') for describing and building graphs.

Need to run `install.packages("ggplot2")` now and `library("ggplot2")` every time you want to use it!

We study ggplot2 using the mpg dataset. Let's try to answer the question:
do cars with bigger engines use more fuel than cars with small engines?

ggplot2

We study ggplot2 using the mpg dataset. Let's try to answer the question:
do cars with bigger engines use more fuel than cars with small engines?

Read about the data using `?mpg`.

```
> dim(mpg)
```

```
[1] 234  11
```

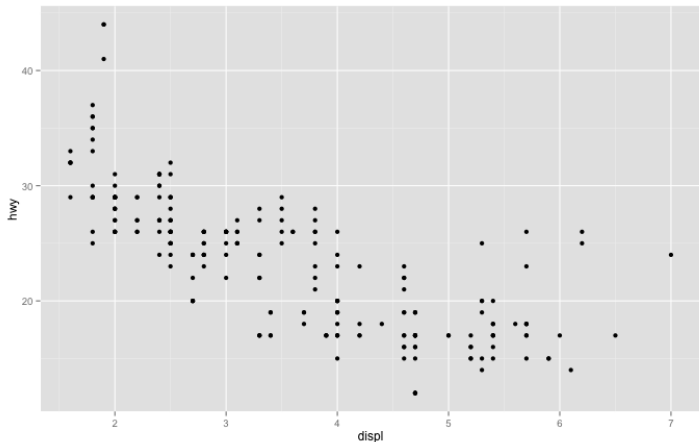
```
> head(mpg, 3)
```

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31

	fl	class
1	p	compact
2	p	compact
3	p	compact

A First Plot

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



A First Plot

Let's break apart the code:

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

A First Plot

Let's break apart the code:

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

- Begin a plot with `ggplot()`.
 - It creates the coordinate axis that you add to.
 - The first argument is the dataset

A First Plot

Let's break apart the code:

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

- Begin a plot with `ggplot()`.
 - It creates the coordinate axis that you add to.
 - The first argument is the dataset
- Next you want to add layers to the plot.
 - In our example: `geom_point()` adds a layer of points.
 - Lots of different `geom` functions doing different things.

A First Plot

Let's break apart the code:

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

- Begin a plot with `ggplot()`.
 - It creates the coordinate axis that you add to.
 - The first argument is the dataset
- Next you want to add layers to the plot.
 - In our example: `geom_point()` adds a layer of points.
 - Lots of different `geom` functions doing different things.
- `geom` functions take `mapping` arguments.
 - Defines how variables in your dataset are mapped to visual properties.
 - Always paired with `aes()`.
 - The `x` and `y` arguments specify which variables to map to the axes.

A First Plot

General structure:

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

To create a plot, replace the bracketed sections in the code above with a dataset, a geom function, and a set of mappings.

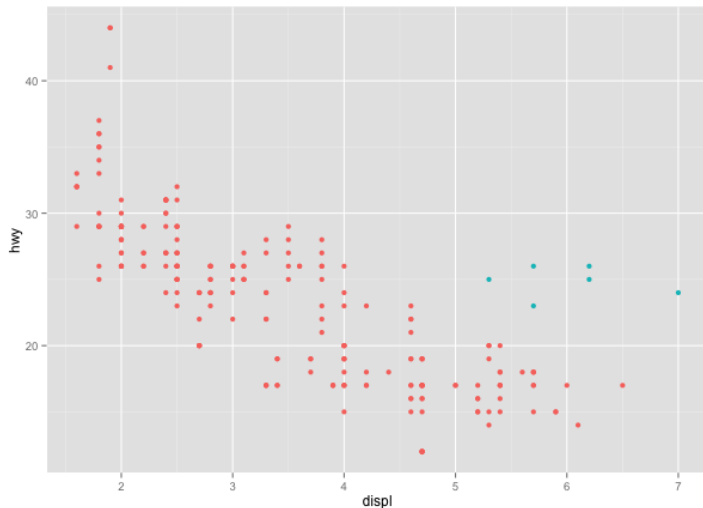
From this template, we can make many different kinds of graphs using ggplot.

Check Yourself

Tasks

- Plot just `ggplot(data = mpg)`. What do you get?
- Make a scatterplot of `hwy` vs. `cyl`.
- Make a scatterplot of `class` vs. `drv`. Why is this plot not useful?

Aesthetic Mappings



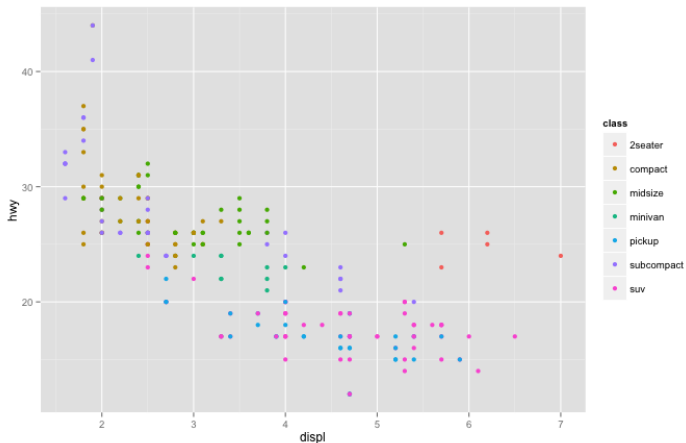
The blue points seem to have a different trend than the rest – possibly hybrids? We study car class to find out.

Aesthetic Mappings

- We can add a third variable to a scatterplot by mapping it to an **aesthetic**.
- An **aesthetic** is a visual property of the objects in the plot.
- Things like size, color, shape of points.

Mapping Aesthetics

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



Check Yourself

Tasks

- Instead of mapping class to the color aesthetic, map it to the alpha aesthetic or the size aesthetic.
- Instead of mapping class to the color aesthetic, map it to the shape aesthetic. Note that `ggplot()` will only use 6 shapes at a time. What does this mean for our plot?

- What does the following code do?

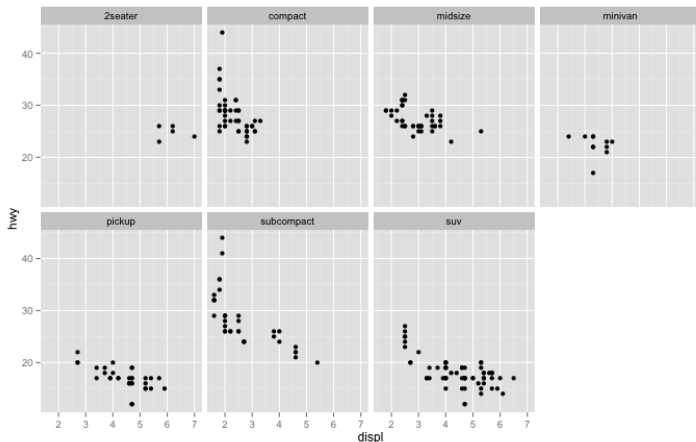
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x=displ, y=hwy), color="blue")
```

- Map a continuous variable in the mpg dataset, like cty, to the alpha, shape, and size aesthetics. What does this do?

- We saw we could add categorical variables to plots using aesthetics.
- Can also do this by splitting the plot into **facets**, which are subplots that each display one subset of the data.
- Use the `fact_wrap()` command to facet a plot by a single variable.
- The argument is a formula created with `~` followed by a variable name.

Facets

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_wrap(~ class, nrow = 2)
```



Tasks

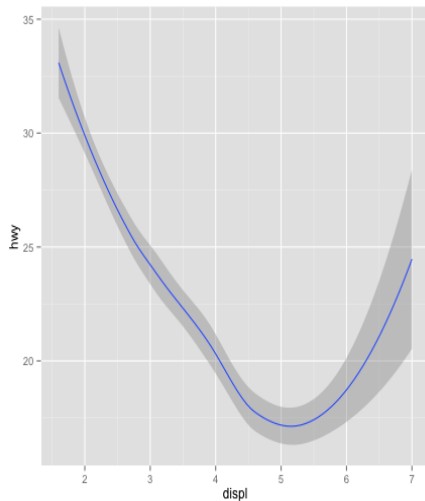
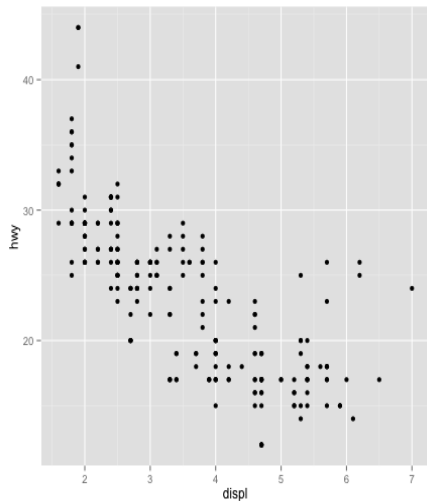
- Facet on two variables use the `facet_grid()` command. An example is the following:

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

What do the empty cells mean?

- Look at `?facet_wrap`. What do `nrow` and `ncol` do? Why doesn't `facet_grid()` have `nrow` and `ncol` arguments?
- What happens if you facet on a continuous variable?

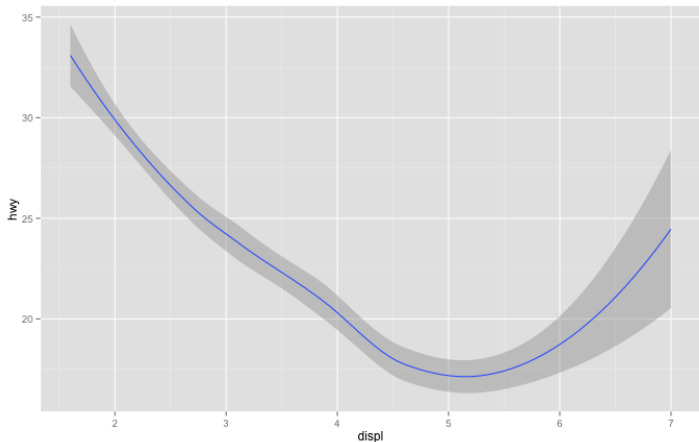
Geometric Objects



- In the previous slide, each plot used a different **visual object** to represent the data.
- Produce this by using different geoms.
- A geom is a geometrical object used to represent data in a plot.
- Often describe plots by the type of geom they use. For example, bar graphs use bar geoms.

Geometric Objects

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



- Every geom takes a mapping argument but not every aesthetic works with every geom.
 - E.g., you can set the shape of a point, but not a line. You can set the linetype of a line.
- ggplot2 has around 30 different geoms.
- Can get help with `?geom_smooth`, for example.

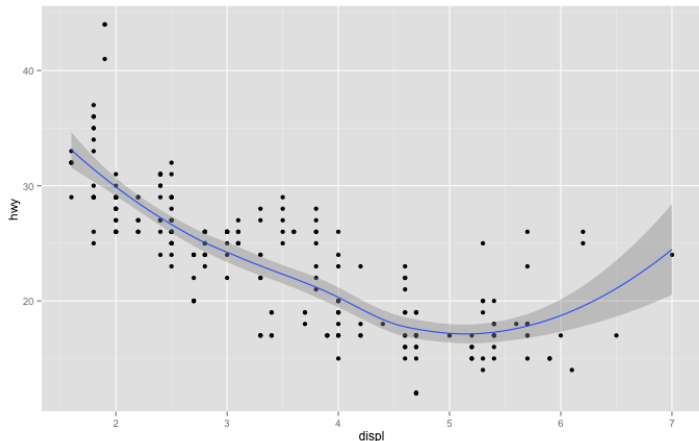
Geometric Objects

Some Commonly-used geoms

geom Name	Used to...	Aesthetics
geom_histogram	Visualize a Continuous Variable	x .
geom_bar	Visualize a Discrete Variable	x.
geom_point	Visualize a Two Continuous Variables	x, y.
geom_text	Add Labels to a Plot	x, y, label.
geom_boxplot	Visualize Continuous and Discrete Variables	x, y.
geom_jitter	Visualize a Two Variables	x, y.
many more ...		

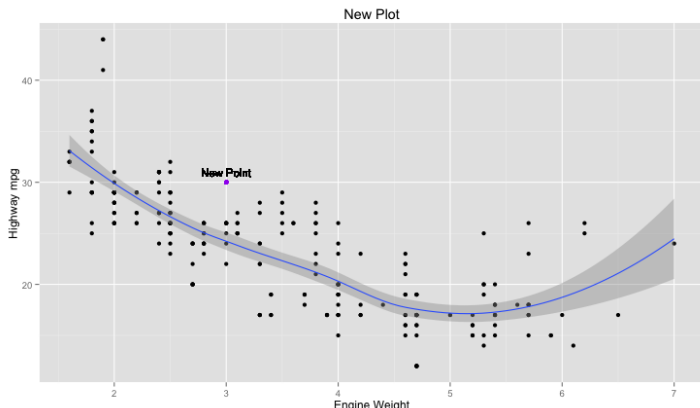
Layering geoms

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



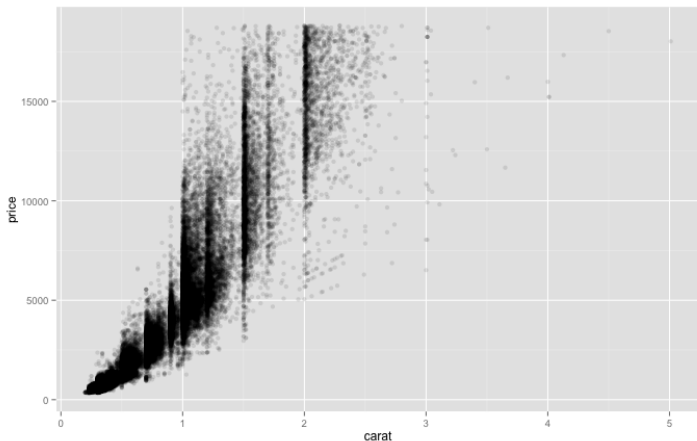
Adding Axis Labels

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(x=3, y=30, color = "purple")) +  
  geom_text(mapping = aes(x=3, y=31, label = "New Point"), size=4) +  
  labs(title = "New Plot", x = "Engine Weight", y = "Highway mpg")
```



Layering geoms

```
> ggplot(data = diamonds) +  
+   geom_point(mapping = aes(x = carat, y = price),  
+                             alpha = 1/10)
```



Check Yourself

Exercise:

Use the built-in `iris` dataset.

- Plot `iris Sepal.Width` on the x-axis and `Sepal.Length` on the y-axis. Color the points according to whether the iris is a setosa or not.
- Plot two regression lines on the plot, one for the setosa iris and one for non-setosa iris. Hint: Use `geom_abline(intercept, slope)`.