ggplot2: An Introduction to the Basics

Jennifer Thompson, MPH

Department of Biostatistics
jennifer.l.thompson@vanderbilt.edu
http://biostat.mc.vanderbilt.edu/JenniferThompson

What is ggplot2?

- Graphics package available for R written by Hadley Wickham
- Something I am NOT an expert in
- Uses code based on R's `grid` graphics system
- Uses philosophy of Wilkinson's *Grammar of Graphics*

  "...the grammar tells us that a statistical graphic is a mapping from data to aesthetic attributes (colour, shape, size) of geometric objects (points, lines, bars). The plot may also contain statistical transformations of the data and is drawn on a specific coordinate system."

Emphasis: Data visualization.

Example data set

```
library(ggplot2)
head(diamonds)
```

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
0.24	Very Good	J	VVS2	62.8	57	336	3.94	3.96	2.48

Getting Started: qplot()

`qplot()` is a basic plotting function from ggplot2 which allows straightforward plots to be generated quickly.

Syntax is similar to `plot()` from base graphics.

```
plot(x = diamonds$carat, y = diamonds$price)
qplot(x = carat, y = price, data = diamonds)
```

The drawback of `qplot()` is that all its layers must use the same data and aesthetic values, so its capabilities are more limited than the full `ggplot()` syntax.

ggplot(): The Real Deal

The `ggplot()` syntax is the core of ggplot2.

- Built in layers
- Each layer can use default data and aesthetics, or specify its own
- Scales allow control over colors, axes
- Options and themes allow fine-tuning of overall plot

ggplot() Syntax

`ggplot() +`

Initializes figure, sets default data and aesthetics

`geom_smooth() +`

Plots one layer using a geom; figure could be finished here

`geom_point() +`

Plots second layer, adding a geom to present raw data as scatter plot

`scale_colour_gradient() +`

Uses scales to manipulate color, X and Y axes

`opts()`

Fine tunes plot settings like background color, etc

What the Heck is a Geom?

A geom is a shape which represents raw data, or a transformation of that raw data. Examples:

- Point geoms to represent raw data points (scatterplot)
- Line geoms to represent regression line or density
- Ribbon geoms to represent confidence bounds
- Others: smoother, histogram, area, polygon, tiles...

In essence, you tell `ggplot()` what shape you want the data to be using a geom.

OK, Now What the Heck Is an Aesthetic?

An aesthetic is a characteristic of a layer which is determined by the value of a variable. Examples:

- Size of a point determined by patient weight
- Color of a line determined by patient gender
- Fill of a bar graph determined by study site
- X and Y positions

OK, Now What the Heck Is an Aesthetic?

Aesthetics can be set in the first `ggplot()` call, to be the default for all plot layers, or they can be set specifically for each layer.

Defaults for entire plot:

```
ggplot(diamonds, aes(x = carat, y = price, colour = cut)) +
    geom_point()
```

Change for a specific layer:

```
ggplot(diamonds, aes(x = carat, y = price)) +
    geom_smooth() +
    geom_point(aes(colour = color))
```

(more examples in code file)

OK, Now What the Heck Is an Aesthetic?

- Color aesthetics by default are based on the color wheel. Example: A categorical variable with five levels (e.g., `cut` in the diamonds data) will be mapped to five colors evenly spaced around the wheel.
- Another potentially useful aesthetic is the `group` aesthetic, good for spaghetti plots.
- Different geoms have different aesthetics depending on their characteristics. `geom_histogram()` has a "fill" aesthetic, `geom_line()` has a "linetype" aesthetic...

Didn't You Say Something About Stats?

A "statistic" (or "stat"), in ggplot2 language, is a statistical transformation of data. A few examples:

- Binning (i.e., histogram)
- Smoothing
- Boxplots

Geoms often call stats internally in order to determine how to present the data. For example, `geom_histogram()` calls `stat_bin()`.

Didn't You Say Something About Stats?

These internal calls often create new variables that can be used to summarize the data. For example, `stat_bin()` creates the new variable `..density..`, which can be used to create a histogram showing data density rather than counts.

Tell `geom_histogram()` to use density instead, and tell `stat_bin()` to decrease the bin width:

```
ggplot(diamonds, aes(x = carat)) +
    geom_histogram(aes(y = ..density..), binwidth = 0.1)
```

Using the stats directly can allow more control over the visuals. For example, `stat_bin()` can change the typical bar histogram to an area geom instead:

```
ggplot(diamonds, aes(x = carat)) +
    stat_bin(geom = 'area')
```

Faceting

Faceting is a way to easily subset your data and see it all at once.

- `facet_wrap()` subsets on one variable, creating panels for each of its levels, and coerces them as much as possible into a square shape.
- `facet_grid()` subsets on two variables, creating a grid where the data for one group can easily be compared with another.

By default, the X and Y scales for all the facets are the same. You can allow these to vary with the `scales` argument:

- `scales = 'fixed'`: X and Y limits are the same for all facets (default)
- `scales = 'free'`: Both X and Y limits can vary
- `scales = 'free_x'` or `scales = 'free_y'`: Only X or Y axis is allowed to vary

Faceting

Histogram of diamond price, faceting by cut:

```
ggplot(diamonds, aes(x = price)) +
    geom_histogram() +
    facet_wrap(~ cut)
```

Histogram of diamond price, faceting by cut and color and allowing both axis limits to vary:

```
ggplot(diamonds, aes(x = price)) +
    geom_histogram() +
    facet_grid(color ~ cut, scales = 'free')
```

Let's Step On The Scale

Each aesthetic is linked to a scale. Scales are how you can tweak colors, labels, axis breaks, etc.

- Each scale has a `name` argument, which lets you change legend or axis titles.
- Color scales define colors of the given variable, based on the color wheel.
- Position scales define the limits, breaks, and labels of the X and Y axes.

Let's tweak a basic plot:

```
ggplot(diamonds, aes(x = carat)) +
  geom_histogram() +
  geom_point(aes(y = price, colour = x), alpha = I(1/10))
```

Let's Step On The Scale

```
ggplot(diamonds, aes(x = carat)) +
```
Initializes figure, sets default data and aesthetics
```
  geom_histogram() +
```
Plots histogram of carat
```
  geom_point(aes(y = price, colour = x), alpha = I(1/5)) +
```
Plots raw data as points, with Y = price; sets x measurement as color aesthetic; sets opacity level
```
  scale_colour_gradient(name = 'X Measurement', low =
'lightgreen', high = 'black') +
```
Changes color value of x measurement to start at light green and end at black, changes legend title
```
  scale_x_continuous(name = 'Number of carats', limits =
c(0, 4), breaks = seq(0, 4, 0.5), labels = seq(0, 4, 0.5))
```
Changes X axis label, subsets on carat size <4, and sets axis breaks and labels

Valuable References

- Wickham, Hadley (2009). *ggplot2: Elegant Graphics for Data Analysis.* New York: Springer.
- Hadley's web site: `had.co.nz/ggplot2/`
- ggplot2 Google Group: `groups.google.com/group/ggplot2`