Making graphs with ggplot2

Brian Diggs

Department of Surgery
Oregon Health & Science University

January 20, 2011 / Portland R User Group

Outline

1. Background and Philosophy
    - Statistical Graphics
    - Philosophical Structure
2. Elements of the Grammer
    - Initializer
    - Geoms
    - Layers
    - Scales
    - Facets
    - Groups
    - Stats

Designed for statistical or data graphics: where quantities of the data are represented by position, shape, color, etc.

Not designed for

- Organization charts
- Flow charts
- Infographics
- Posters

Philosophical Structure

- Data in a data.frame in “long” format
- Columns of data are mapped to aesthetics
- Non-data related aspects controlled by theme

Data in long format

data.frame with an entry for each measurement and the variables associated with describing the circumstances of that measurement

```
> str(warpbreaks)
```

```
'data.frame':       54 obs. of  3 variables:
 $ breaks : num  26 30 54 25 70 52 51 26 67 18 ...
 $ wool   : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1
 $ tension: Factor w/ 3 levels "L","M","H": 1 1 1 1 1 1 1 1
```

Aesthetics

- Examples
  - Position along an axis
  - Shape
  - Color
  - Width
  - Line type
  - Replicate (horizontal and/or vertical)
- Legend is mapping between visual element and the data values
- Any aesthetic can be fixed to a specific (non-data) value

Themes

Themes control non-data related aspects of the graphic

- Size of title
- Font of tick lables
- Location of legend
- Background colors

qplot vs. ggplot

Two ways to start making a plot

- qplot
  - Designed to be like plot
  - Eases transition
  - Obscures details
- ggplot
  - Core of the actual grammer
  - Less familiar

ggplot

Arguments

data Default data.frame of the data to be plotted

mapping Default aesthetic mappings

```
> ggplot(warpbreaks)
```

```
Error: No layers in plot
```

Incomplete by itself

Geoms

Geoms are the different ways that “ink” is used to show the values

- Points
- Lines connecting points/locations
- Steps connecting points/locations
- Bars
- Points with lines extending from them (error bars)
- Tiles/rectangles
- Ribbons
- Text

Required and optional aesthetics

Geom	Required	Optional
point	x, y (positions)	shape, colour, size, fill, alpha
rect	xmin, xmax, ymin, ymax	colour, fill, size, linetype, alpha
text	x, y, label	colour, size, angle, hjust, vjust, alpha

Aesthetics are specified by calls to the aes function with arguments that are aesthetic name and the variable being mapped to it.

Examples of geoms

cw is a subset of the ChickWeight data.frame in datasets

```
> ggplot(cw) + geom_point(aes(x = Time,
+       y = weight))
```

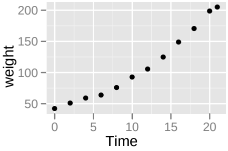
```
> ggplot(cw) + geom_line(aes(x = Time,
+       y = weight))
```

```
> ggplot(cw) + geom_path(aes(x = Time,
+       y = weight))
```

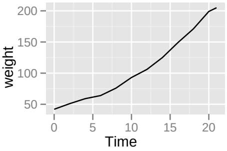
```
> ggplot(cw) + geom_step(aes(x = Time,
+       y = weight))
```

Examples of geoms (results)

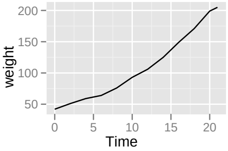
geom_point



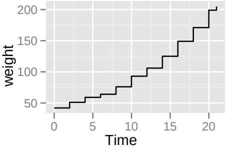
geom_line



geom_path



geom_step



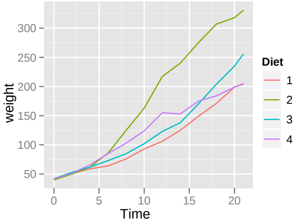
Optional aesthetics

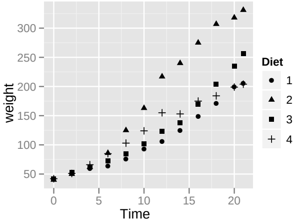
cw2 is a broader subset of ChicWeight

```
> ggplot(cw2) + geom_line(aes(x = Time,
+      y = weight, colour = Diet))

> ggplot(cw2) + geom_point(aes(x = Time,
+      y = weight, shape = Diet))
```

Optional aesthetics (results)

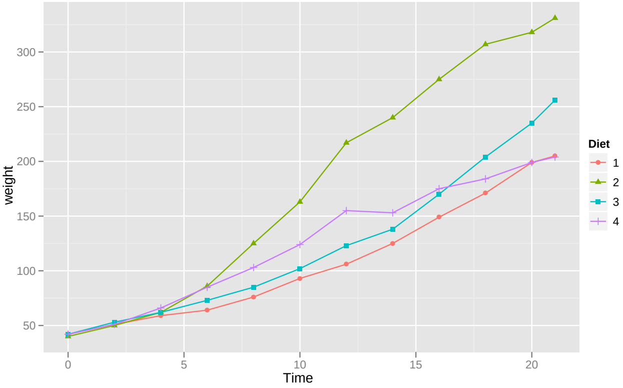




Layers

Much of the flexibility comes from being able to combine different geoms in layers

```
> ggplot(cw2) + geom_point(aes(x = Time,
+     y = weight, colour = Diet,
+     shape = Diet)) + geom_line(aes(x = Time,
+     y = weight, colour = Diet))
```


weight
300
250
200
150
100
50
0
5
10
15
20
Time
Diet
1
2
3
4

Default aesthetics

Multiple geoms using the same set of aesthetics show the utility of default aesthetics

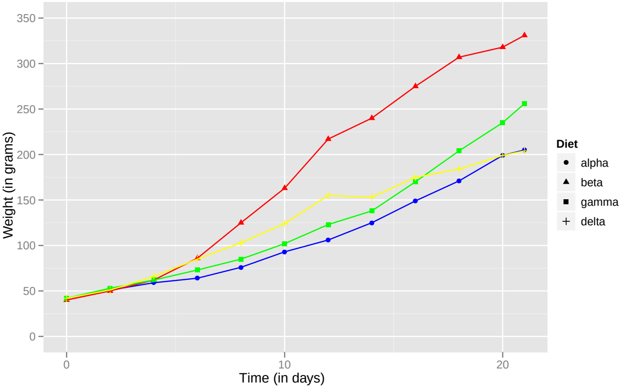
```
> ggplot(cw2, aes(x = Time,
+       y = weight, colour = Diet)) +
+       geom_point(aes(shape = Diet)) +
+       geom_line()
```

Aesthetics vs. scales

- Aesthetics define which variable is associated with a property
- Scales define
  - Which particular color/shape/etc. is associated with which data value
  - Which data values are part of the graph
  - How the data values are displayed in the legend
  - How the legend is titled

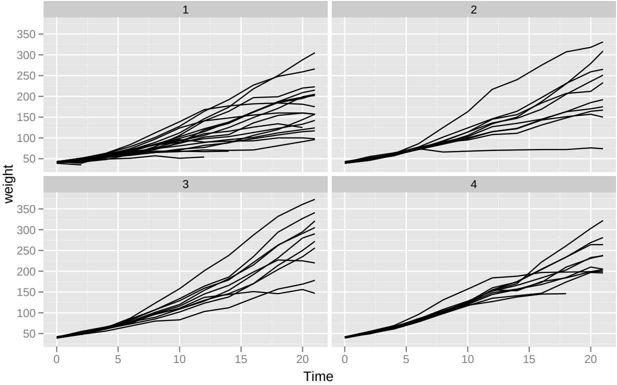
- Previous example had 4 scales
  - x position
  - y position
  - colour
  - shape
- Note that even though both the points and lines used colour, there is just one common scale for both
- In this case
  - both positions are continuous
  - colour and shape are discrete

```
> ggplot(cw2, aes(x = Time,
+     y = weight, colour = Diet)) +
+     geom_point(aes(shape = Diet)) +
+     geom_line() + scale_x_continuous("Time (in days)",
+     breaks = c(0, 10, 20)) +
+     scale_y_continuous("Weight (in grams)",
+         limits = c(0, 350)) +
+     scale_colour_manual(breaks = 1:4,
+         values = c("blue",
+             "red", "green",
+             "yellow"),
+         legend = FALSE) +
+     scale_shape(breaks = 1:4,
+         labels = c("alpha",
+             "beta", "gamma",
+             "delta"))
```


Weight (in grams)
350
300
250
200
150
100
50
0
0
10
20
Time (in days)
Diet
● alpha
▲ beta
■ gamma
+ delta

Facets, or small multiples, are sets of graphs showing subsets of data on common x and/or y axes for comparison purposes

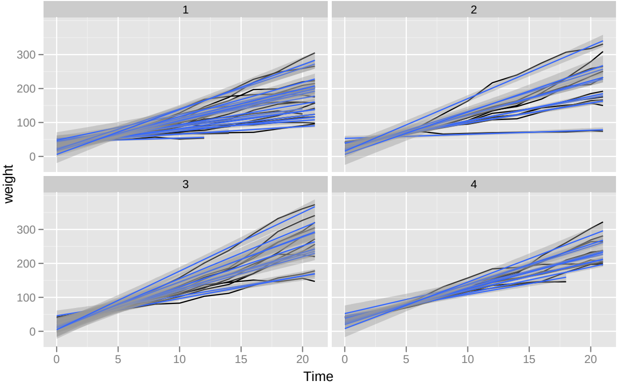
```
> ggplot(ChickWeight, aes(x = Time,
+       y = weight, group = Chick)) +
+       geom_line() + facet_wrap(~Diet)
```


1
2
3
4
weight
350
300
250
200
150
100
50
0
5
10
15
20
Time

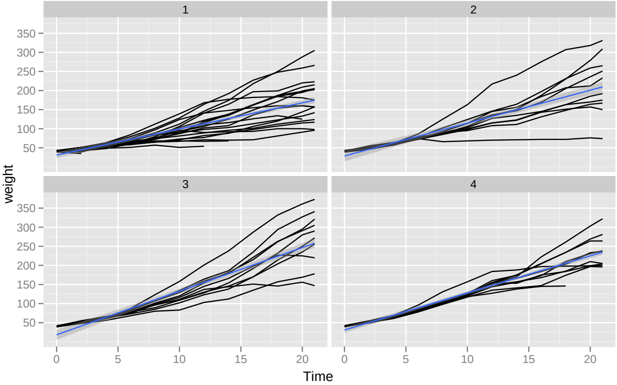
- By default, a separate group is created for each combination of categorical (factor or character) variables that are mapped to aesthetics
- Can be overridden with the group aesthetic
- Many things done on a one-per-group basis (line, path, ribbon, stat)

- Transformation (summarization) of data by group
    - Binning (1 or 2 dimensional)
    - Quartiles
    - Mean and standard deviation
    - Smoothing/regression
- Results of the transformations are mapped to aesthetics

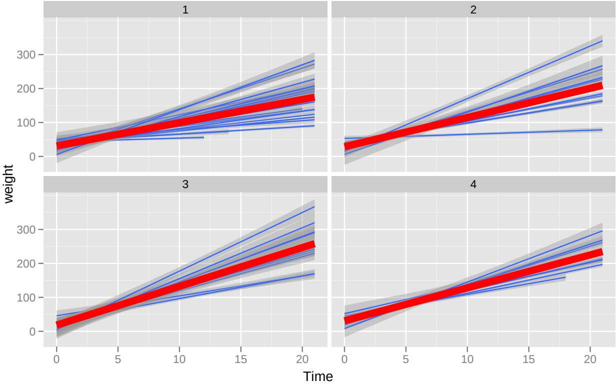
```
> ggplot(ChickWeight, aes(x = Time,
+      y = weight, group = Chick)) +
+      geom_line() + stat_smooth(method = "lm") +
+      facet_wrap(~Diet)
```


1
2
3
4
weight
Time
0
100
200
300
0
5
10
15
20

```
> ggplot(ChickWeight, aes(x = Time,
+   y = weight, group = Chick)) +
+   geom_line() + stat_smooth(aes(group = Diet),
+   method = "lm") + facet_wrap(~Diet)
```


1
2
3
4
weight
Time
350
300
250
200
150
100
50
0
5
10
15
20

```
> ggplot(ChickWeight, aes(x = Time,
+     y = weight, group = Chick)) +
+     stat_smooth(method = "lm") +
+     stat_smooth(aes(group = Diet),
+         method = "lm",
+         colour = "red",
+         size = 3) + facet_wrap(~Diet)
```


1
2
3
4
weight
300
200
100
0
0
5
10
15
20
Time