Introduction to ggplot2

Dennis Prangle
(University of Lancaster)

2nd May 2013

Overview

- `ggplot2` is an R graphics package
- In this talk I will:
  - Describe the background to the package
  - Give examples of its use
  - Give my take on its pros and cons
- n.b. I am not an expert-level user!

Part I

Background

R graphics

- `base` graphics
  - R's default graphics system
  - Functions like `plot`, `hist` etc.
  - A "pen-on-paper" approach
- `grid` graphics
  - Alternative flexible *low-level* graphics system
  - Underlies `lattice` and `ggplot2`
- `lattice` package
  - Package for trellis graphics
- And others (see CRAN task view on graphics)

ggplot2 history

- "The Grammar of Graphics" is an idea of Leyland Wilkinson
  - (see book the same name)
- Data variables are mapped to graphical concepts e.g. position, color, shape
- `ggplot2` is an implementation for R
- Written by Hadley Wickham 2005-present
- (originally called `ggplot` - the "2" was added after a major revision)

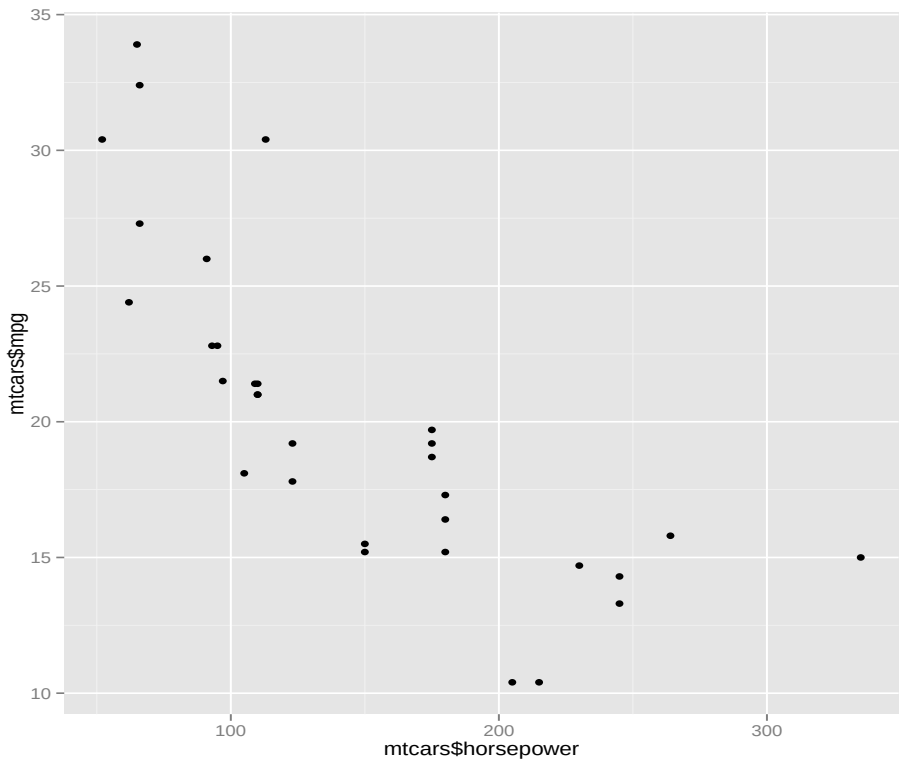
Part II

Using ggplot2

qplot as a replacement for plot

- `qplot` ("quick plot") has a similar synax to `plot`
- Good place to start

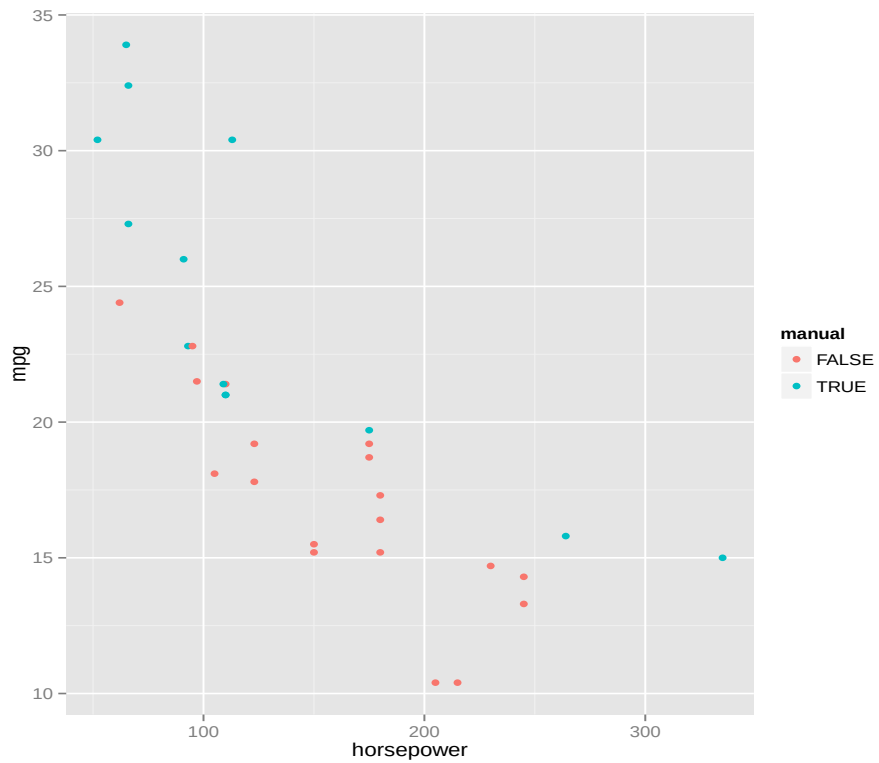
```
qplot(x=mtcars$horsepower, y=mtcars$mpg)
```



Standard qplot use

- ggplot2 expects input as a dataframe e.g.

```
qplot(x=horsepower, y=mpg, colour=manual, data=mtcars)
```



Some ggplot2 features

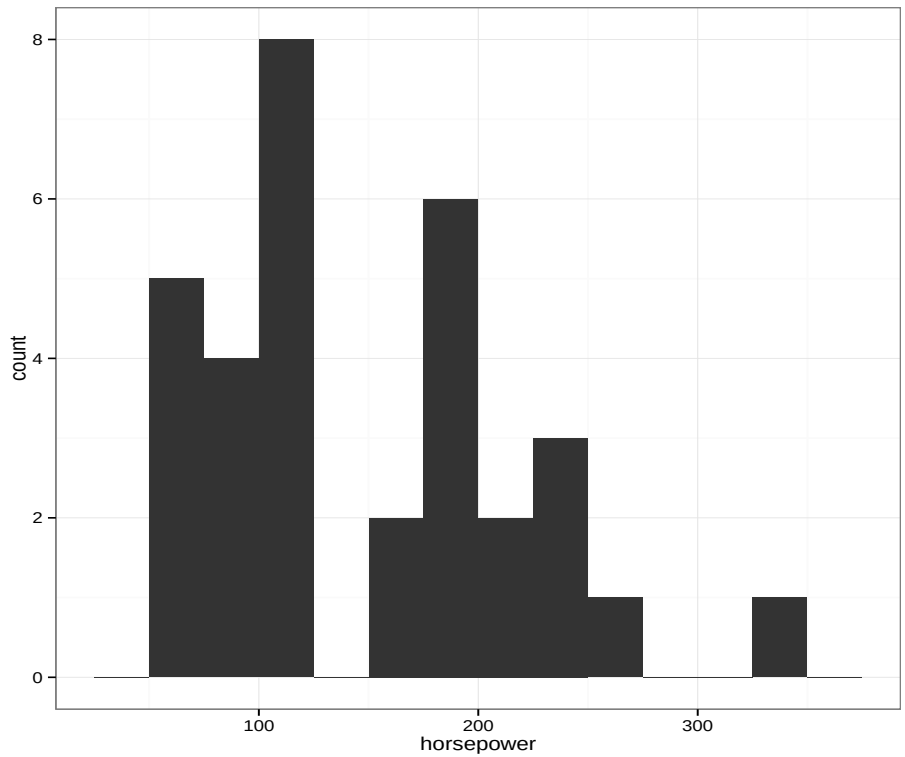
- Colours chosen automatically (with nice defaults)
- Automatic legend
- UK or US spelling of "colour" / "color" accepted
- Default background is grey. To change:

```
theme_set(theme_bw())
```

Histograms

- `qplot` defaults to a scatter plot if `x` and `y` arguments present
- Or histogram if `x` present only
- (nb non-default binwidth used here to improve appearance)

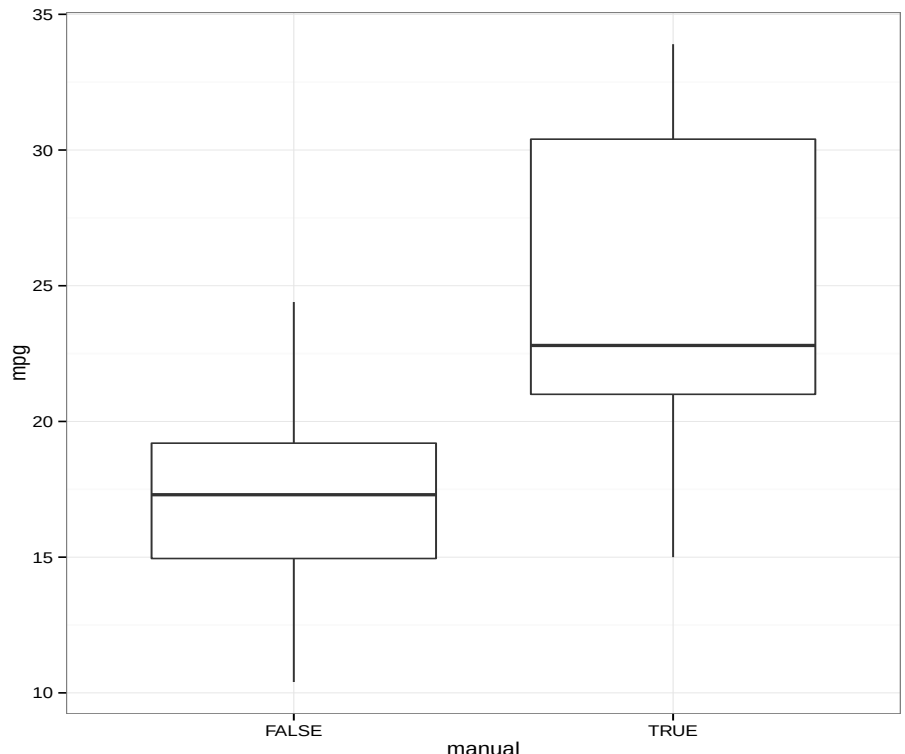
```
qplot(x=horsepower, data=mtcars, binwidth=25)
```



Other graphs

- Grammar of graphics approach maps variables to graph features
- Can be several ways to do this: different graph types
- Known as geoms in ggplot

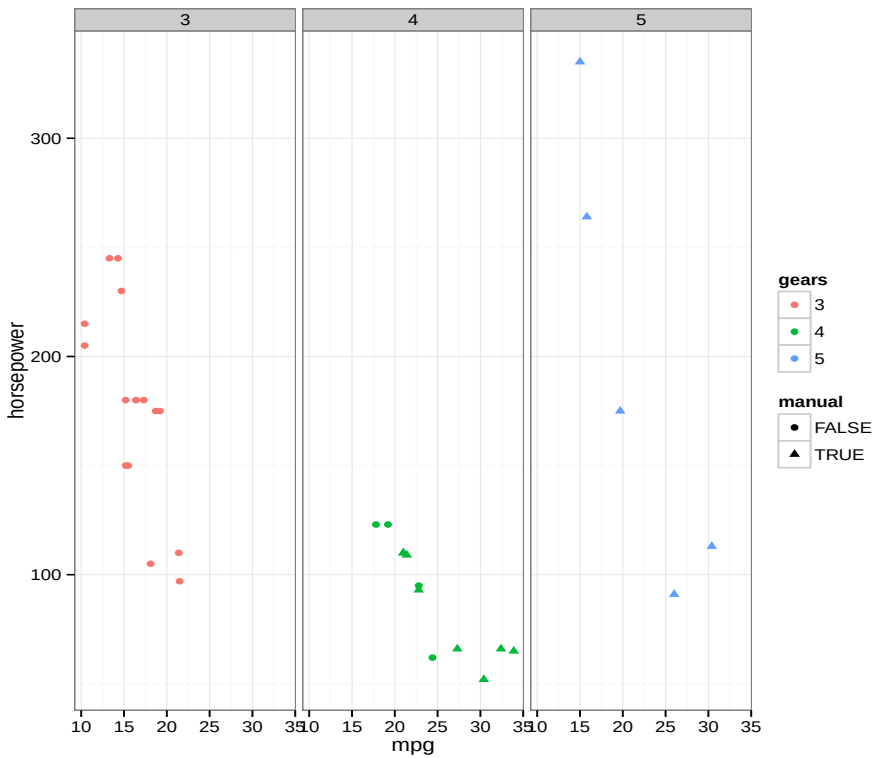
```
qplot(x=manual, y=mpg, geom="boxplot", data=mtcars)
```



Facets

- Easy to create facets (aka a trellis graph)

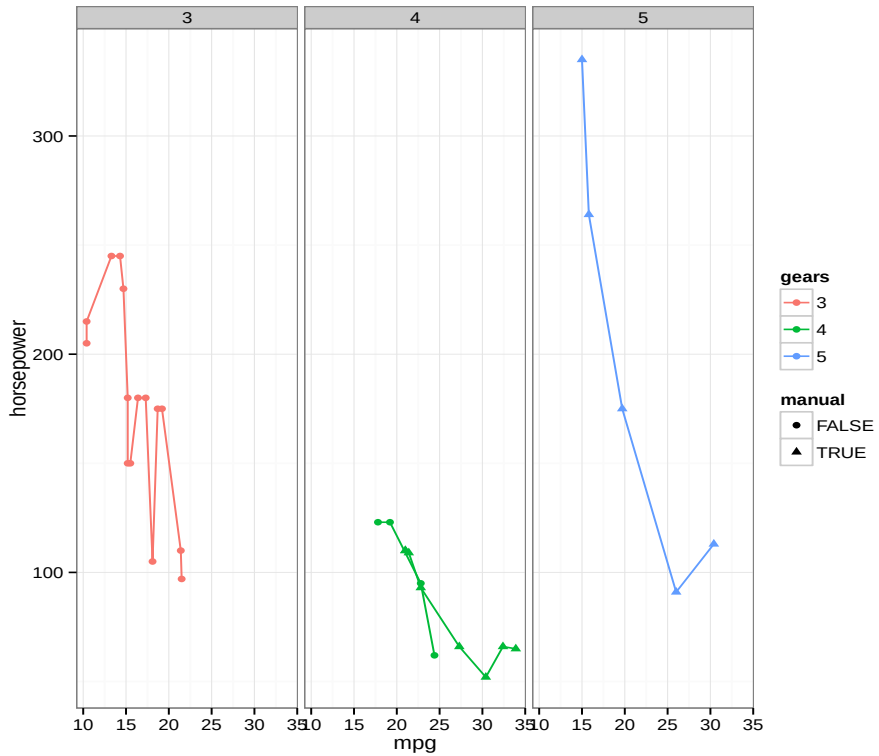
```
qplot(x=mpg, y=horsepower, facets=.~gears
  color=gears, shape=manual, data=mtcars)
```



Building graphs

- Use addition to modify a graph

```
last_plot() + geom_line()
```



- For complicated graphs start with `ggplot` command and add each element in turn

ggsave

- Easy to export the current graph with ggsave
- Detects suffix and saves to correct file type

```
ggsave("amazing_graph.pdf")
```

Part III

Conclusions (my opinion!)

Pros and cons

Pros:

- Easy to produce near-publication standard graphics with short commands
- Many annoying tasks under base graphics done automatically!
- Trellis graphics done easily

Cons:

- Requires `dataframe` input
- Hard to customise/extend
- Documentation poor - you need to buy the book

Conclusion

- A highly recommended tool
  1. For quick exploratory work
     - Light learning curve via `qplot`
  2. For publications
     - Customising graphs more complex
     - Need to learn full package in more detail
     - Best to work with `ggplot` to learn right syntax
- But some graphs still easier in base graphics

More resources

- "ggplot2: Elegant graphics for data analysis" - Hadley Wickham (2009) UseR series
- `http://ggplot2.org/` (official website)