University of BRISTOL

Advanced Graphics Using R

Osama Mahmoud

Website: http://osmahmoud.com
E-mail: o.mahmoud@bristol.ac.uk

Goals

In this sesseion, you will be learned:

- What `ggplot2` is.
- How to use `ggplot2` in R to produce advanced graphics.
- What the building-blocks of `ggplot2` graphs are.

Contents

Graphics in R: Background

Installing packages in R is straightforward. For example:

```
> install.packages("ggplot2")
```

Then, you can simply load it to your R session whenever needed:

```
> library("ggplot2")
```

Types of R graphics

Types of R graphics

- Base graphics.
- Grid graphics.
- Lattice graphics.
- ggplot2 graphics.

Overview of ggplot2

Installing Course R package: BristolVis

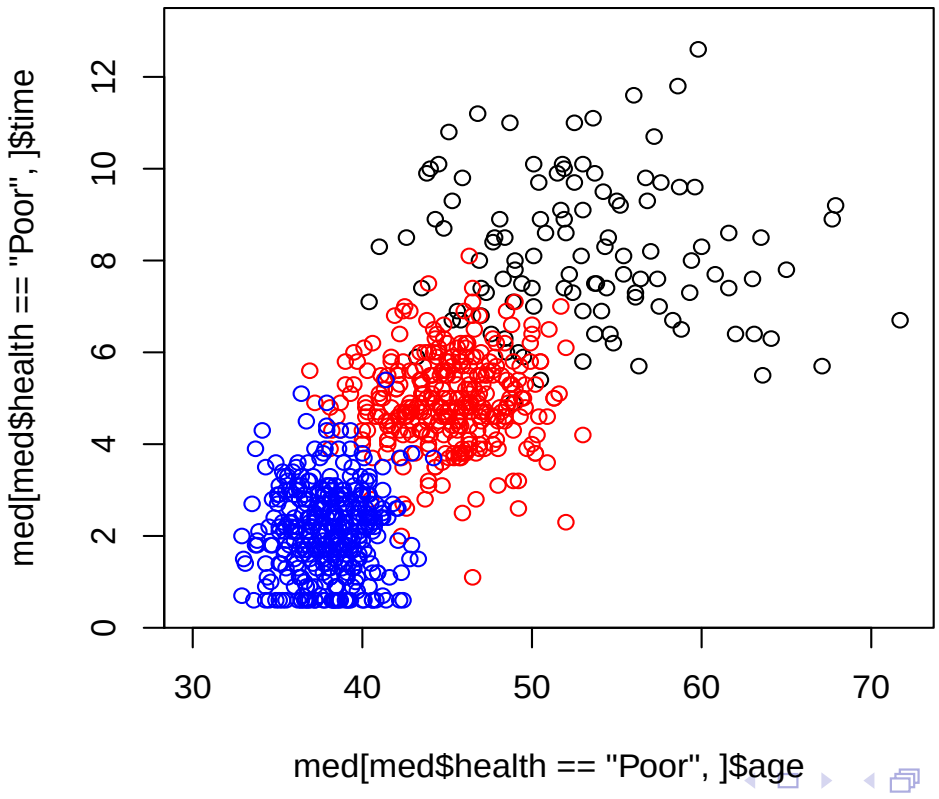
```
> install.packages("drat")
```

```
> drat::addRepo("statcourses")
```

```
> install.packages("BristolVis")
```

Basic plots using base graphics

```
> plot(med[med$health=="Poor",]$age, med[med$health=="Poor",]$time,
xlim=c(30,72), ylim=c(0.5,13))
> points(med[med$health=="Fair",]$age, med[med$health=="Fair",]$time,
col=2)
> points(med[med$health=="Good",]$age, med[med$health=="Good",]$time,
col=4)
```



Basic plots using base graphics

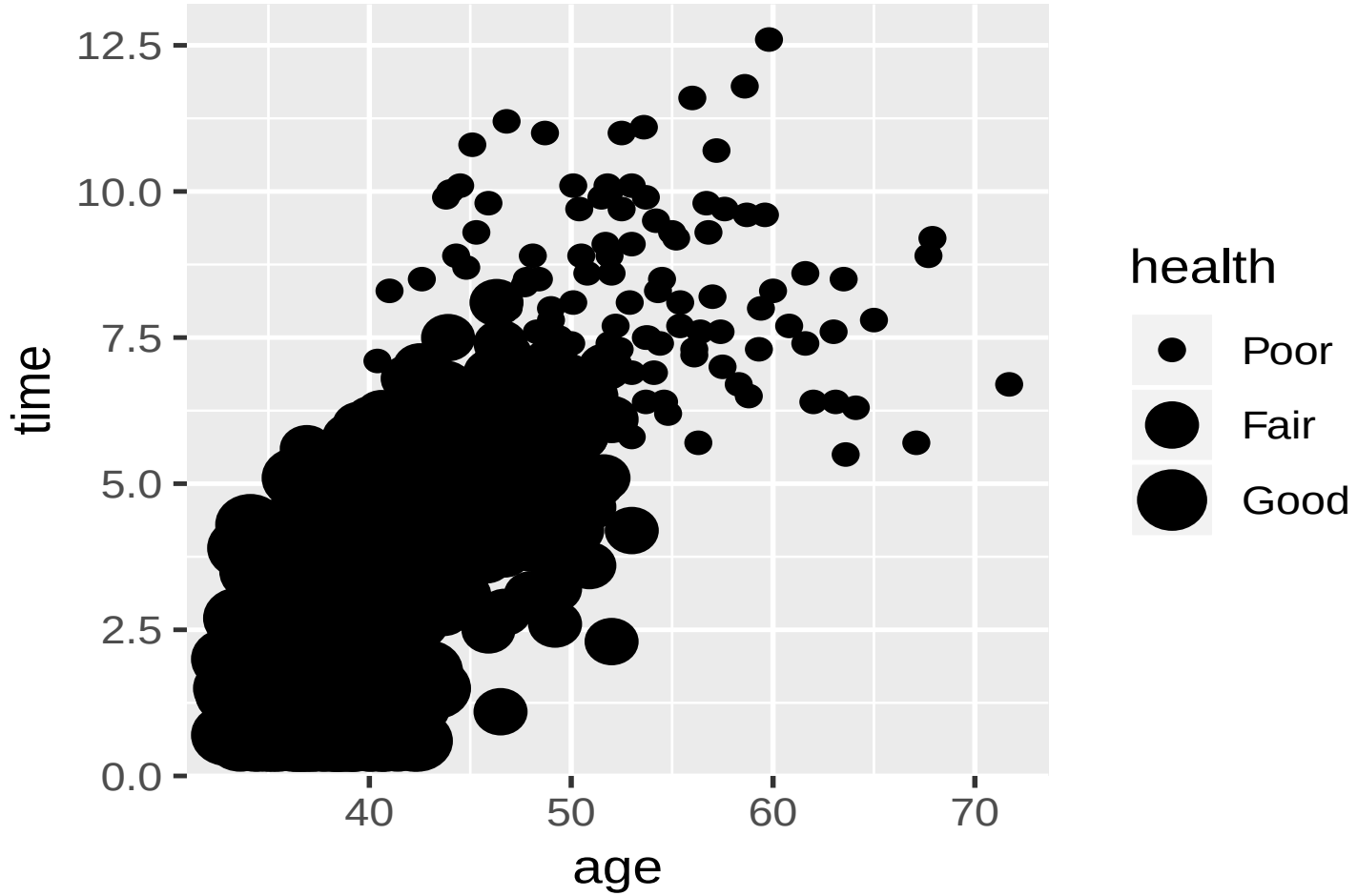
- We had to manually set the scales using the `xlim` and `ylim` parameters.
- We had not created a legend. We would need to use the `legend` function to create one.
- The default axis labels were terrible!

Equivalent graphics using ggplot2

```
> library(ggplot2)
> g = ggplot(data=med, aes(x=age, y=time))
> g + geom_point(aes(colour=health))
```

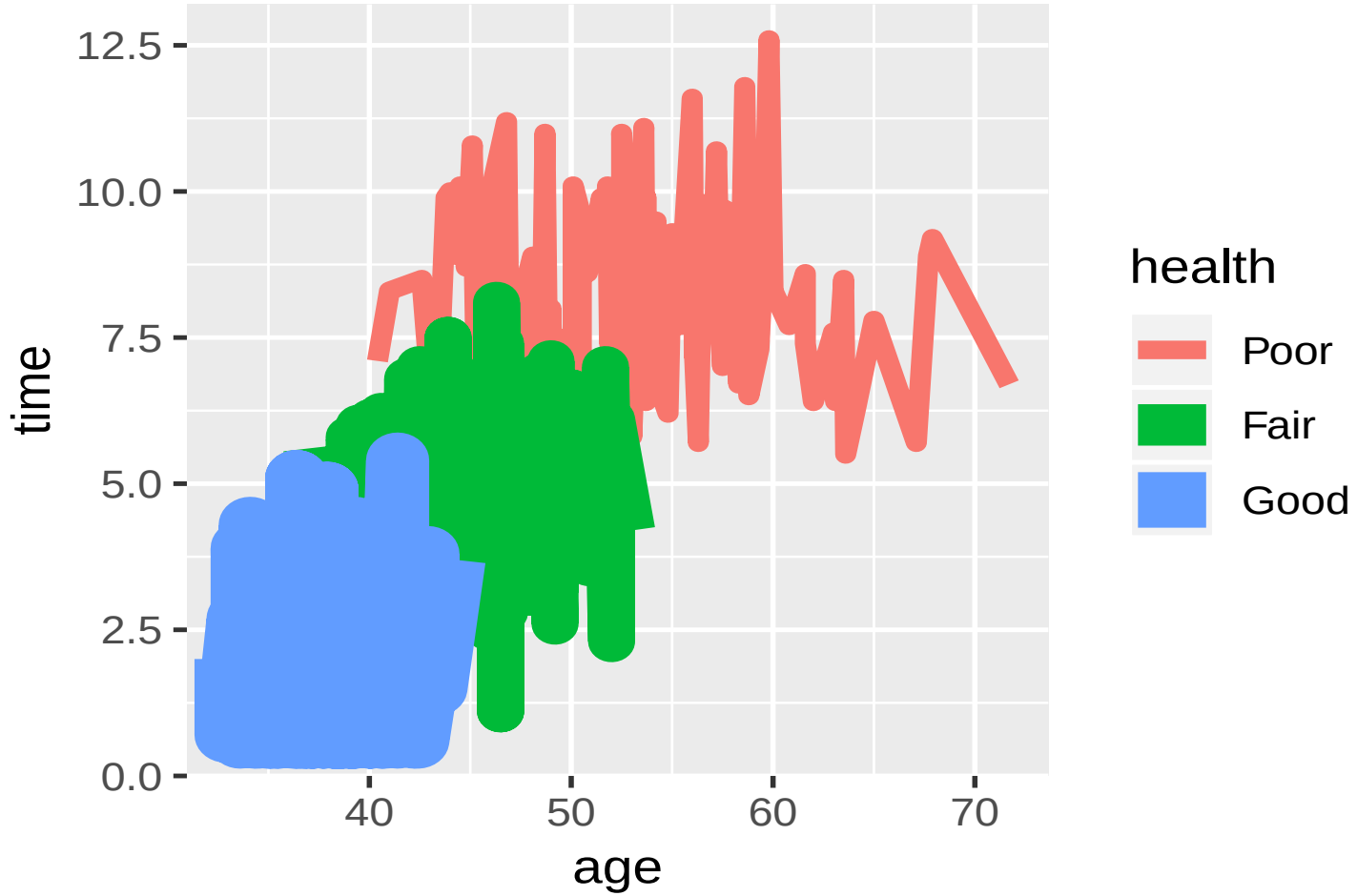
Factor of point sizes

```
g + geom_point(aes(size=health))
```



Example of a line chart

```
g + geom_line(aes(colour=health, size = health)))
```



Geometric objects

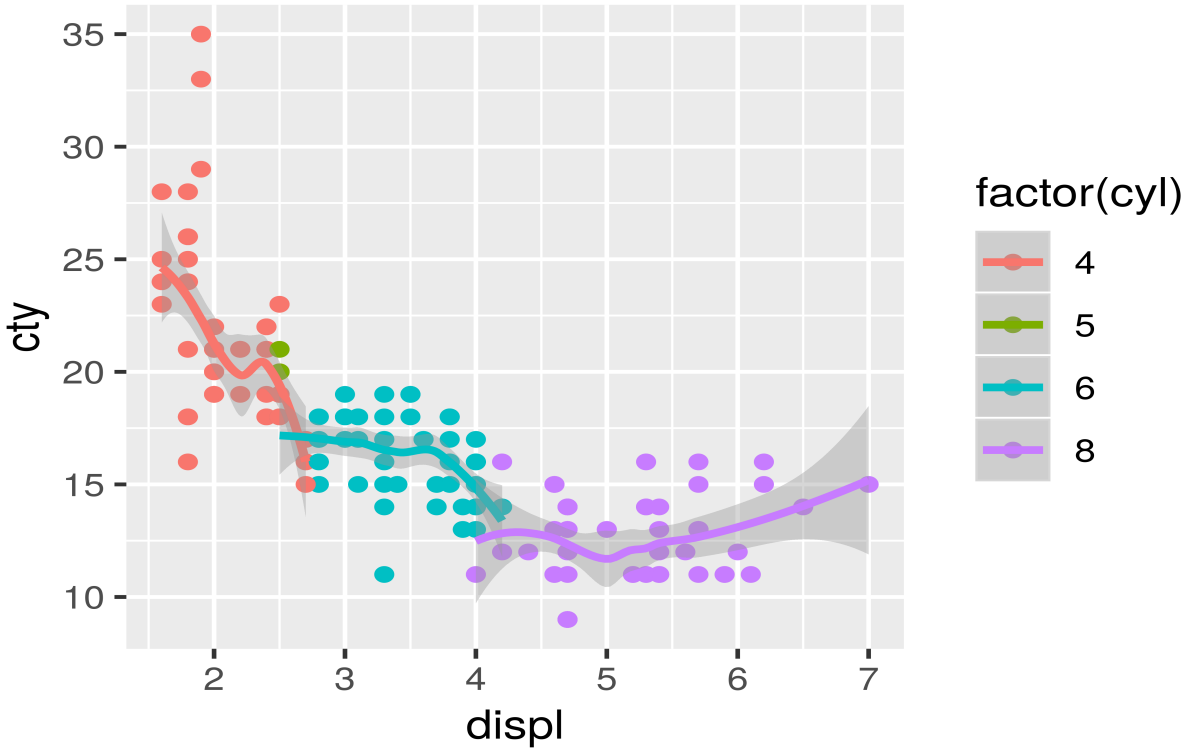
Points, bars and lines are examples of geom's. Some useful standard geoms and their equivalent base graphic counter part:

Plot Name	Geom	Base graphic
Bar chart	bar	`barplot`
Box-and-whisker	boxplot	`boxplot`
Histogram	histogram	`hist`
Line plot	line	`plot` and `lines`
Scatter plot	point	`plot` and `points`

More complicated functions

The idea of graphical layers, enables constructing more complicated functions, e.g.:

```
p = ggplot(mpg, aes(x = displ, y = cty)) +
geom_point(aes(colour=factor(cyl))) +
stat_smooth(aes(colour=factor(cyl)))
```



Understanding ggplot2 philosophy

Each `ggplot` command adds iteratively layers. A single layer may comprise of four elements:

- an aesthetic and data mapping;
- a geometric object (`geom`);
- a statistical transformation (`stat`);
- a position adjustment, i.e. how should overlapped objects be handled.

Understanding ggplot2 philosophy

For example, the command:

```
g + geom_point(aes(colour=health))
```

actually calls (in the background) the command:

```
g + layer(data = med, #inherited
mapping = aes(color=health), #x and y are inherited.
stat = "identity",
geom = "point",
position = "identity",
params = list(na.rm=FALSE))
```

Plot building-blocks

Initial plot object

An initial ggplot object, can be setup using the `ggplot()` function which has two arguments:

- `data` (*takes a data frame*)
- an aesthetic `mapping` (*creates default aesthetic attributes*)

```
g = ggplot(data=mpg, mapping=aes(x=displ, y=cty,
colour=factor(cyl)))
```

Or equivelently,

```
g = ggplot(mpg, aes(displ, cty, colour=factor(cyl)))
```

doesn't actually produce anything to be displayed, it just sets the intial plot object. We need to add layers for that to happen.

The geom_ functions

- The `geom_` functions perform the actual rendering in a plot, e.g. a line geom will create a line plot and a point geom creates a scatter plot.
- Each geom has a list of aesthetics that it accepts such as `x` , `y` , `colour` and `size`.
- However, some geoms have unique elements. For example, the `geom_errorbar` requires arguments ymax and ymin.
- The full list of aesthetics can be displayed by:

```
> ggplot2:::.all_aesthetics
```

The geom_ functions

- This table gives names and descriptions of some commonly used geoms:

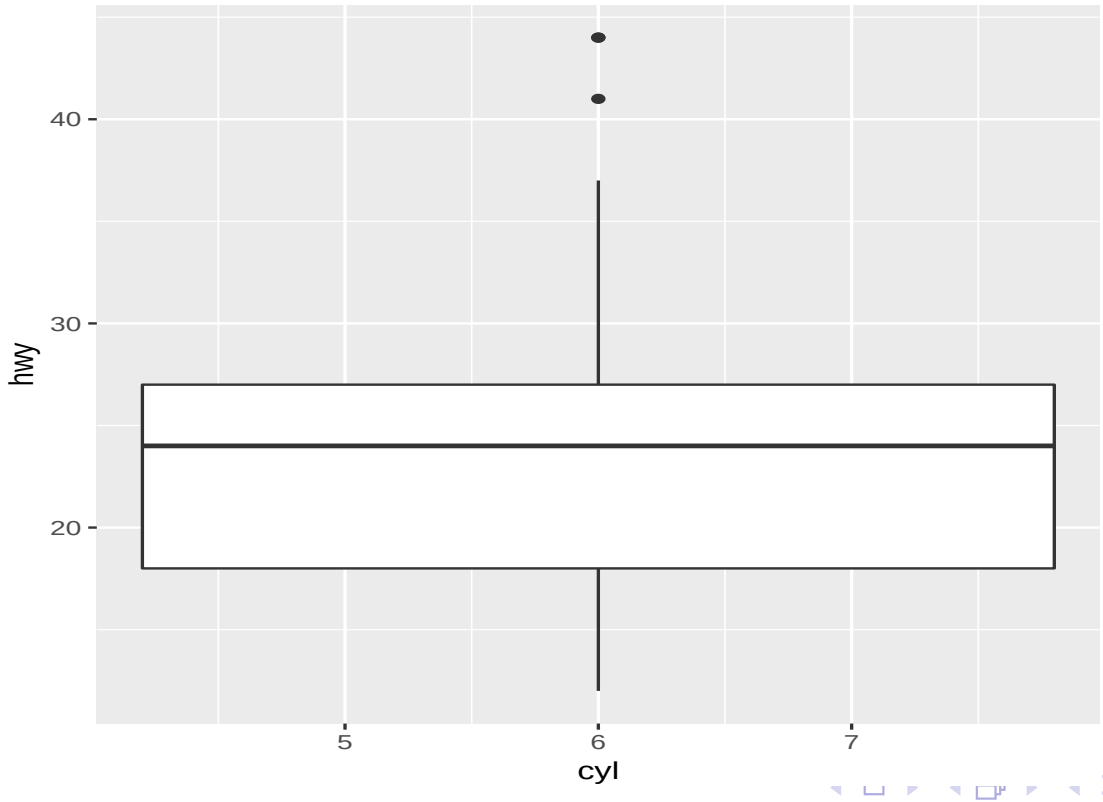
Name	Description
`abline`	Line, specified by slope and intercept
`boxplot`	Box and whiskers plot
`density`	Kernel density plot
`histogram`	Histograms
`jitter`	Individual points are jittered to avoid overlap
`step`	Connect observations by stairs

Combining geoms

I will show how to combine more that one geom function to produce a bit more complex plots. If we consider the mpg data set, a base ggplot object:

`g = ggplot(mpg, aes(x=factor(cyl), y=hwy))` will do nothing.

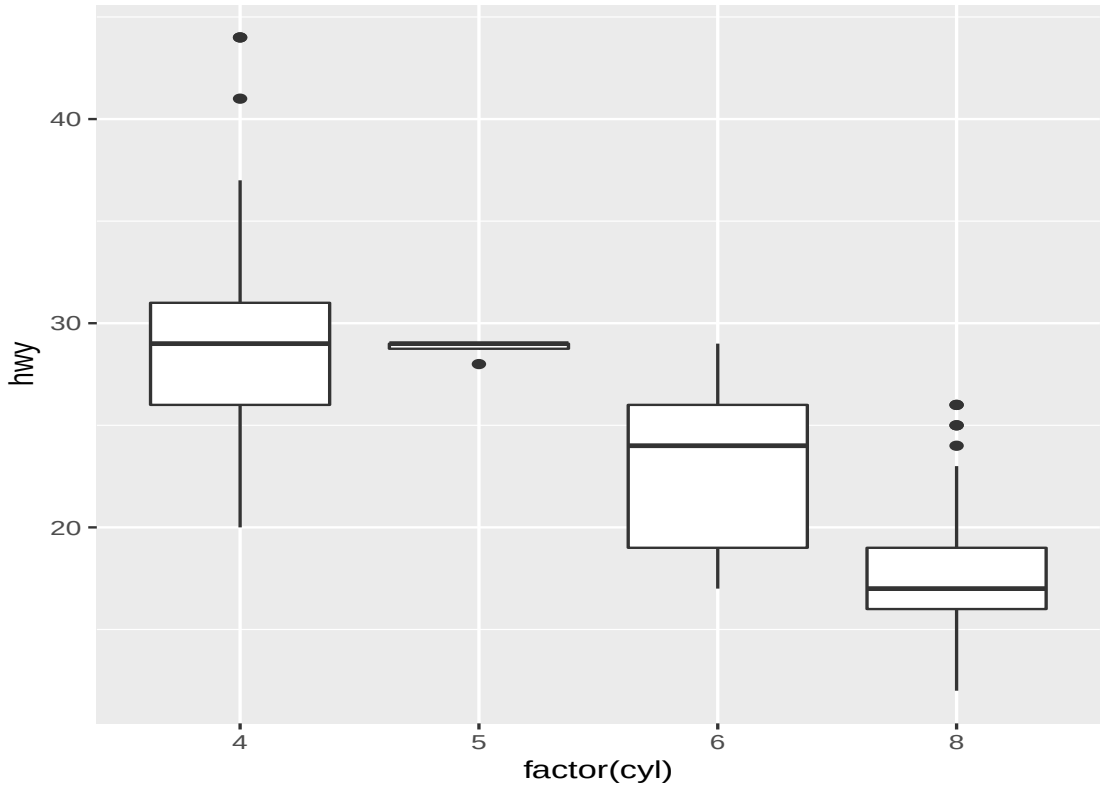
Now we'll create a boxplot: `(g1 = g + geom_boxplot())`



Combining geoms

Previous figure was a boxplot of all the `mpg` data, a more useful plot would be to have individual boxplots conditional on number of cylinders:

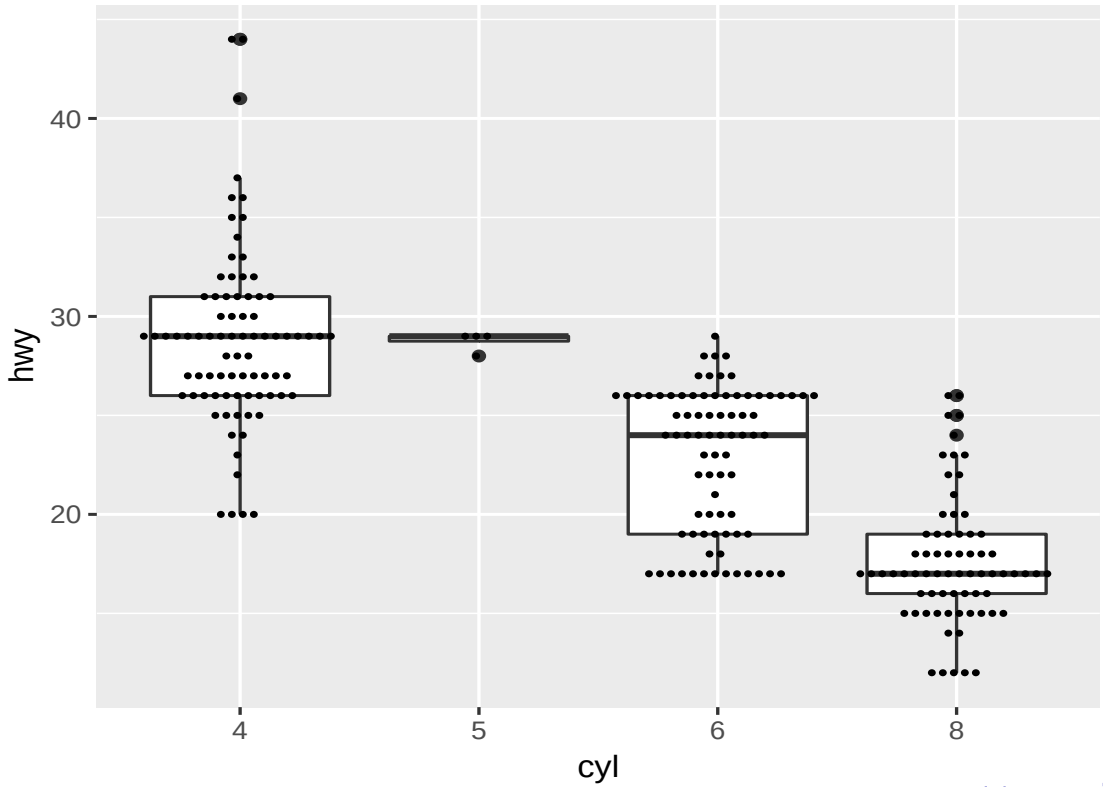
```
(g2 = g + geom_boxplot(aes(x=factor(cyl), group=cyl)))
```



Combining geoms

We are not restricted to a single geom. When data sets are reasonably small, it is useful to display the data on top of the boxplots:

```
(g3 = g2 + geom_dotplot(aes(x=factor(cyl), group=cyl),
binaxis="y", stackdir="center", binwidth=0.25,
stackratio=2))
```



Standard plots

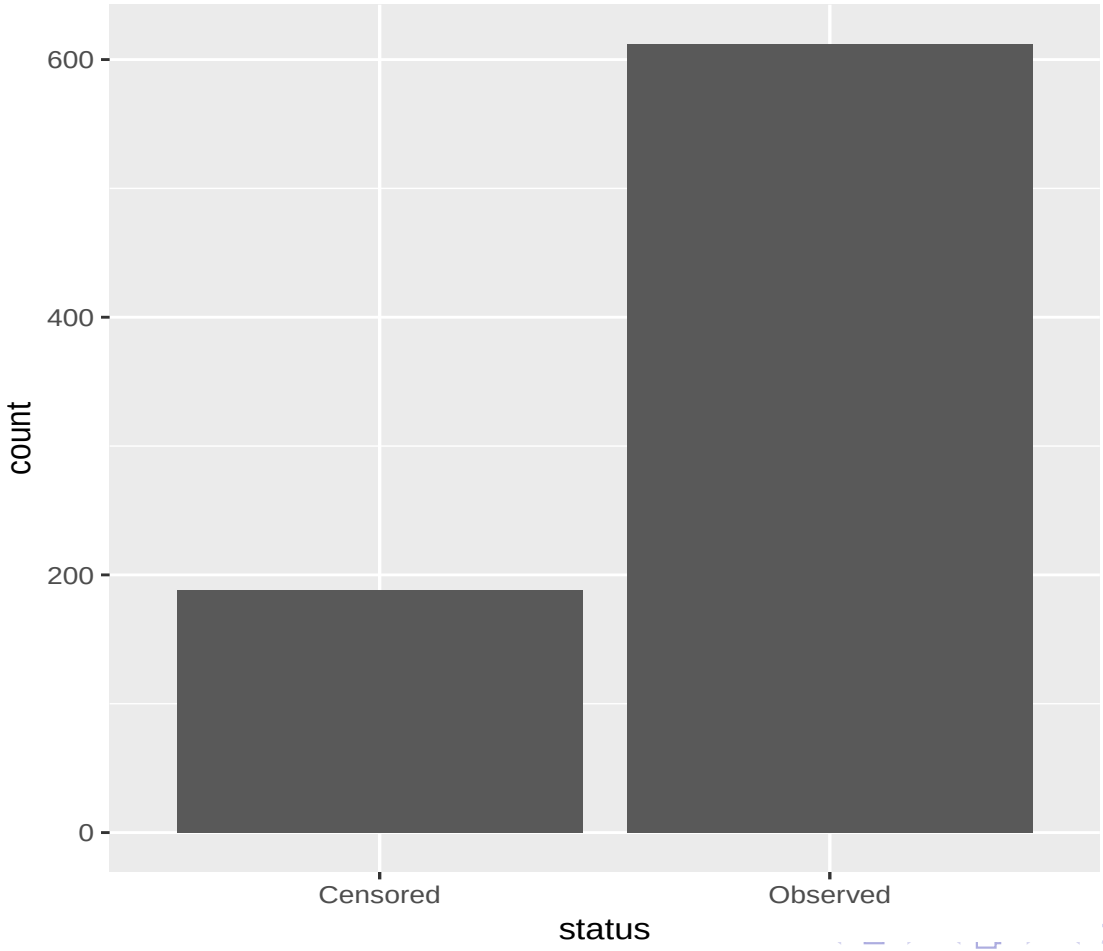
There are a few standard geom 's that are particular useful:

- geom_line : a line plot.
- geom_boxplot : produces a boxplot.
- geom_point : a scatter plot.
- geom_dotplot: a dot plot.
- geom_bar : produces a standard barplot that counts the x values.
- geom_text : adds labels to specified points (as `geom_point` but draw labels rather than points).
- geom_raster: Similar to `levelplot` (heatmap).

Standard plots

To generate a bar plot, e.g. for the status of effect observations in the `med` data set, the following code can be used:

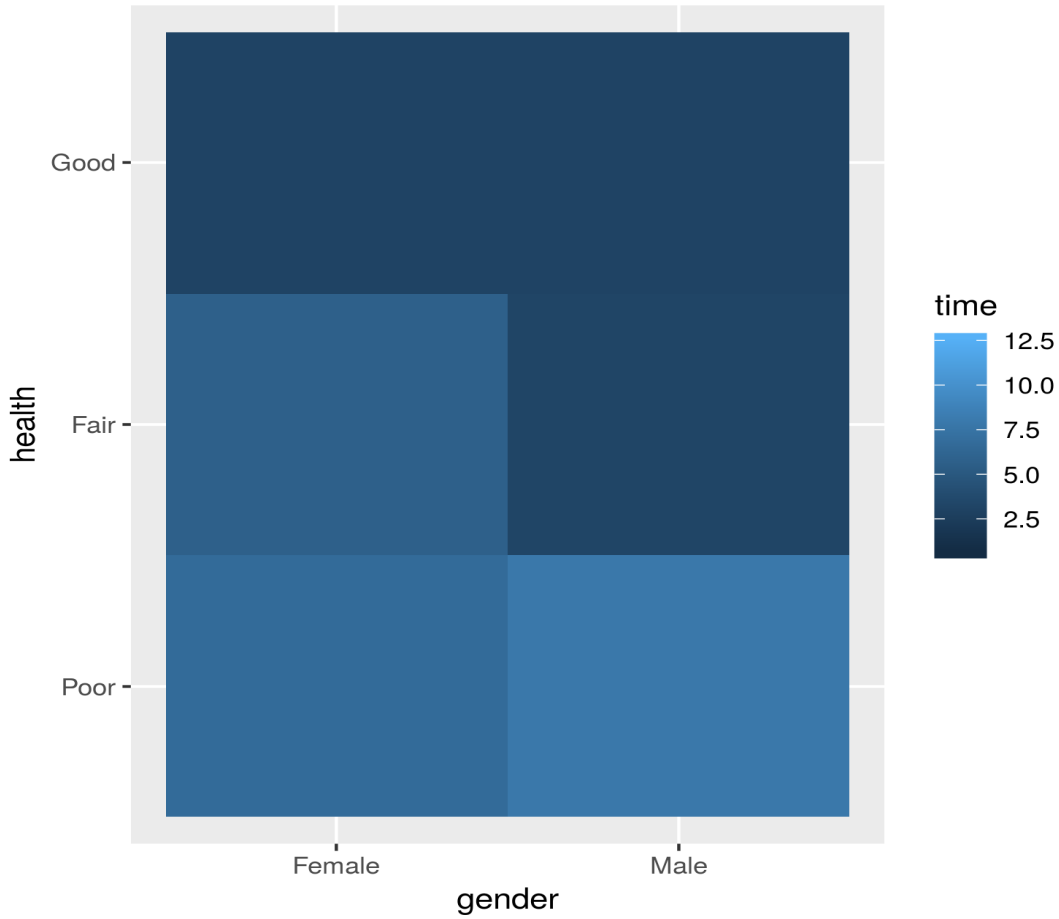
```
ggplot(med, aes(x=status)) + geom_bar()
```



Standard plots

An example of a `geom_raster`:

```
ggplot(med, aes(gender, health)) +
geom_raster(aes(fill=time))
```



Useful links

Github

Repository of course packages:
https://statcourses.github.io/

Web-page of the course

The BristolVis tool for learners of data visualisation using R:
https://github.com/statcourses/BristolVis

Thank You