University of BRISTOL

Advanced Graphics Using R

Osama Mahmoud

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

Structure

In this sesseion, I aim to cover:

- What is data visualisation?
- Why do we need visualisation?
- How to use `ggplot2` in R.
- What are the building-blocks of `ggplot2` graphs
- How to use facets to produce multiple panels in same plot
- How to use themes to control the appearance of a plot.

Contents

Graphics reveal data

"*The greatest value of a picture is when it forces us to notice what we never expected to see.*"

JOHN W. TUKEY (1915 - 2000)

Why do we need visualisations?

I		II		III	
x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46
8.0	6.95	8.0	8.14	8.0	6.77
13.0	7.58	13.0	8.74	13.0	12.74
9.0	8.81	9.0	8.77	9.0	7.11
11.0	8.33	11.0	9.26	11.0	7.81
14.0	9.96	14.0	8.10	14.0	8.84
6.0	7.24	6.0	6.13	6.0	6.08
4.0	4.26	4.0	3.10	4.0	5.39
12.0	10.84	12.0	9.13	12.0	8.15
7.0	4.82	7.0	7.26	7.0	6.42
5.0	5.68	5.0	4.74	5.0	5.73

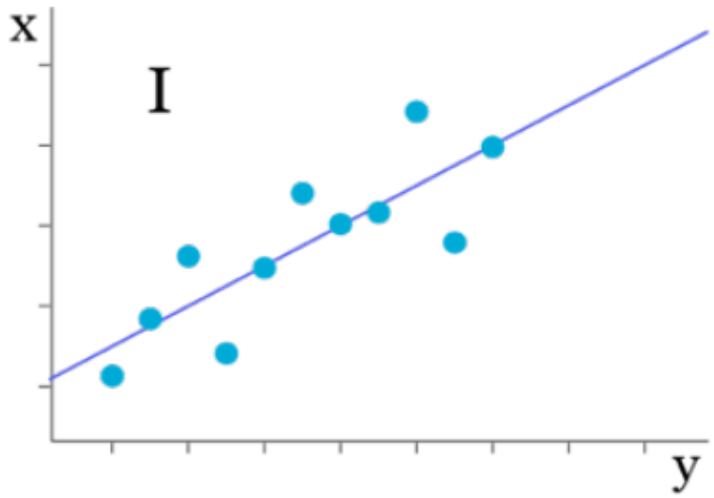
n = 11

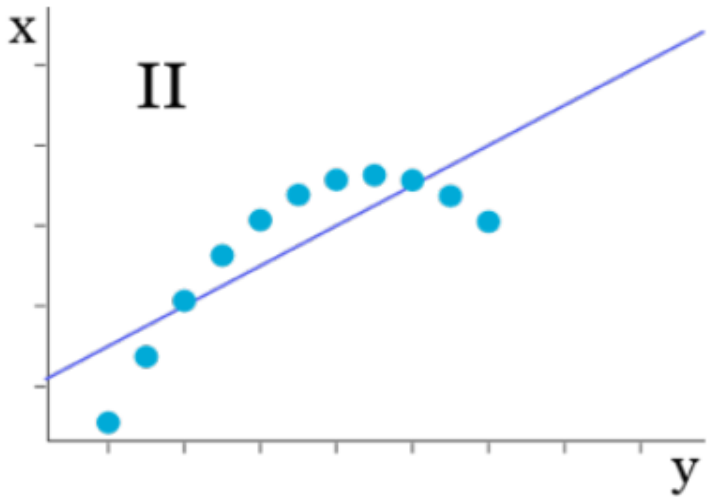
mean x = 9.0
mean y = 7.5

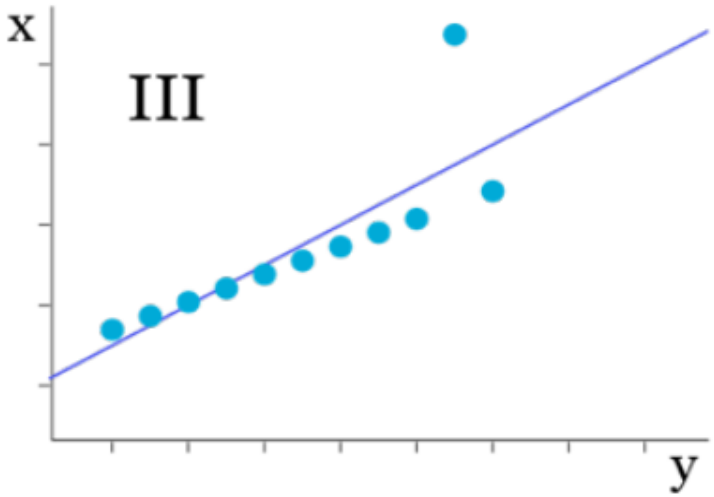
variance x = 11.0
variance y = 4.12

correlation x & y = 0.816
regression line: y = 3+0.5x

Why do we need visualisations?

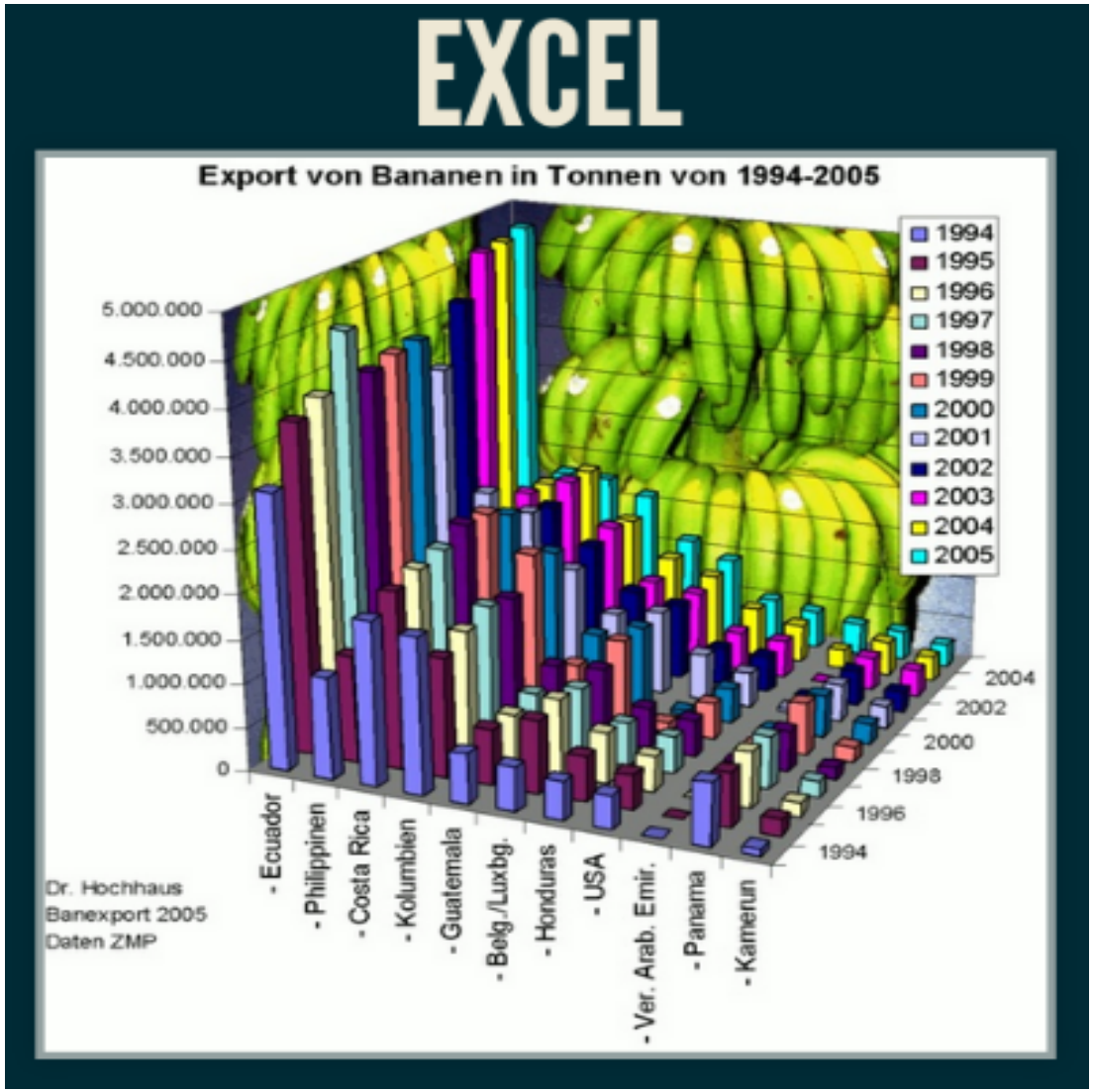




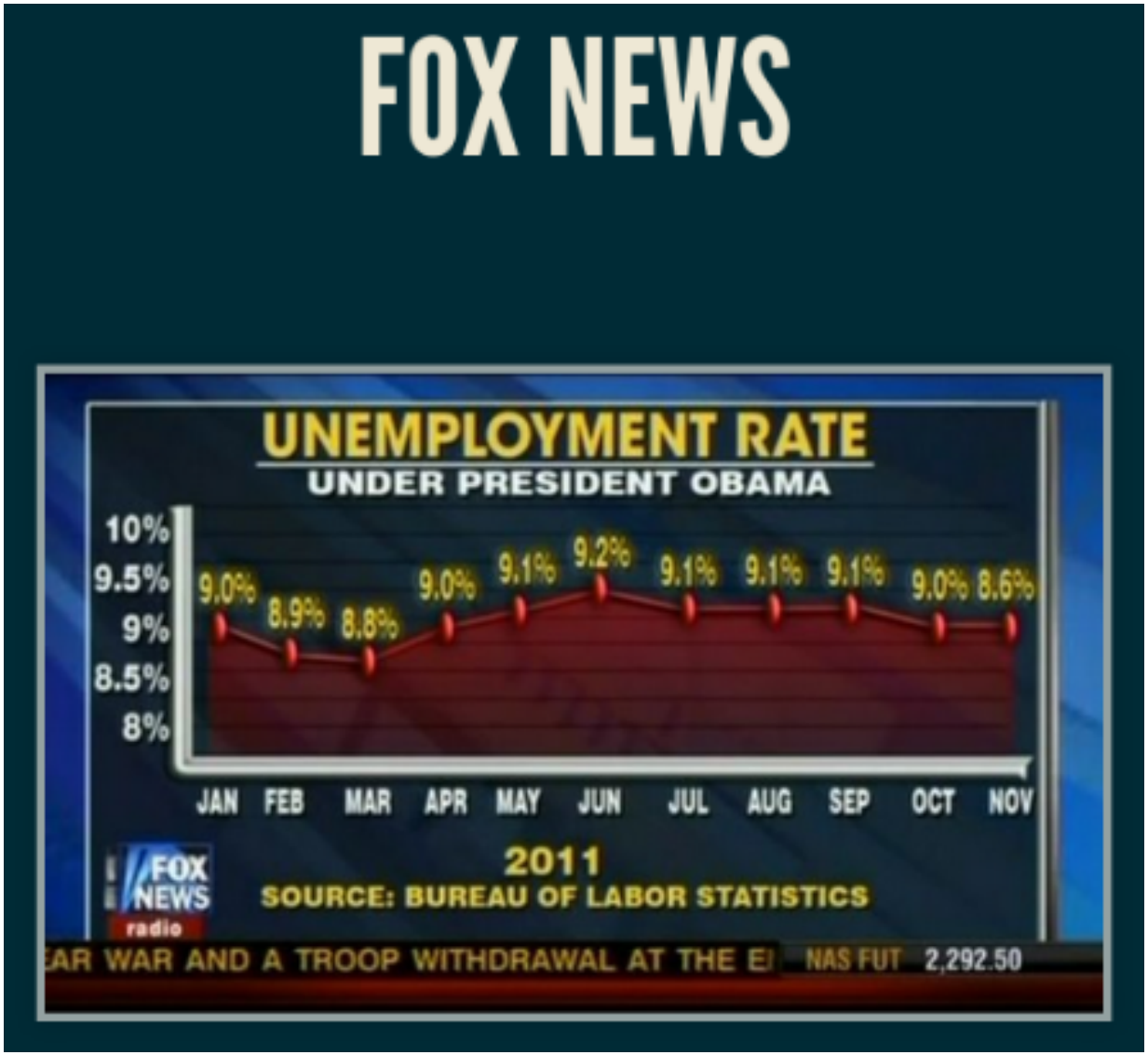


THE UGLY

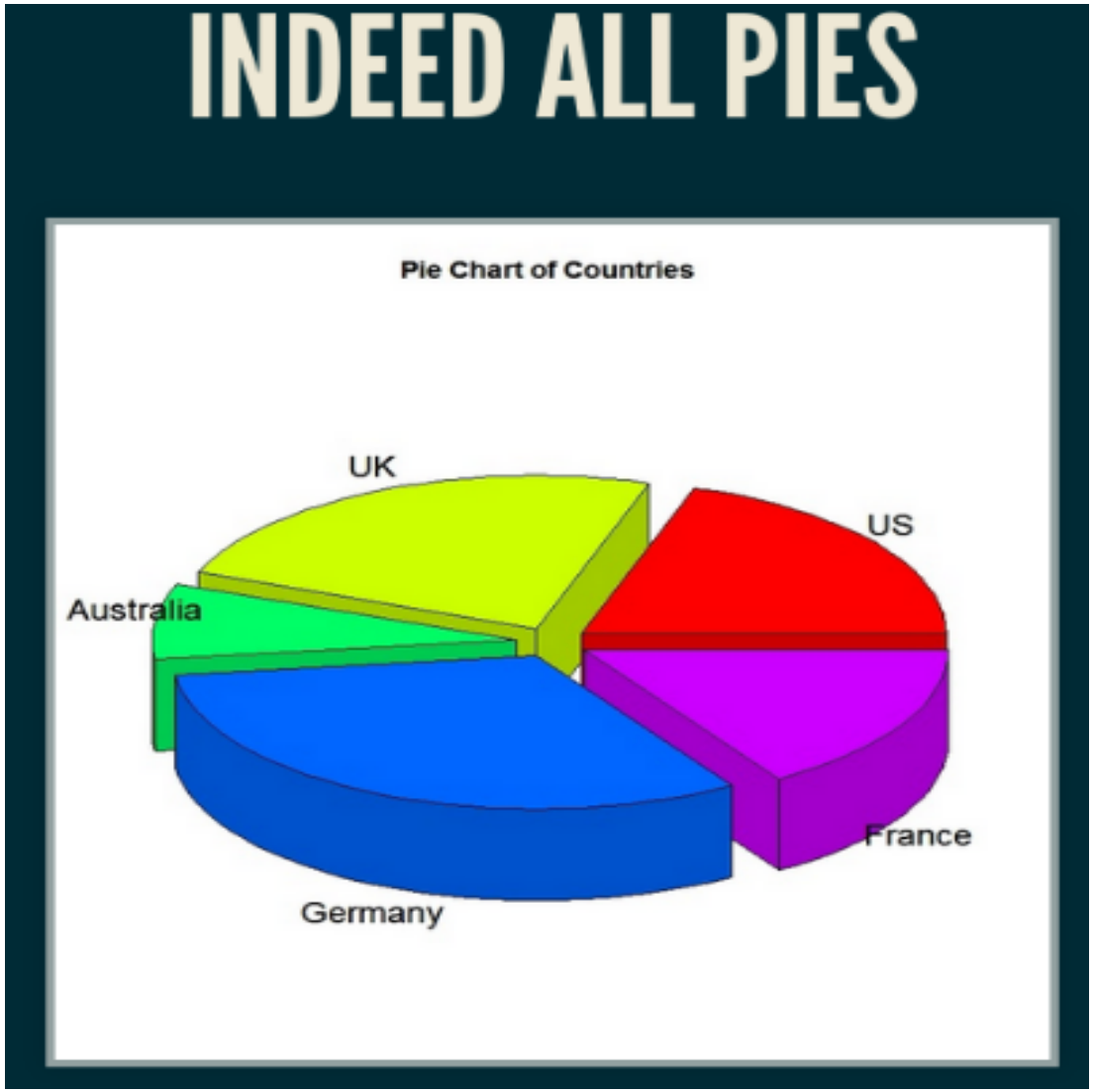
The ugly


EXCEL
Export von Bananen in Tonnen von 1994-2005
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
5.000.000
4.500.000
4.000.000
3.500.000
3.000.000
2.500.000
2.000.000
1.500.000
1.000.000
500.000
0
2004
2002
2000
1998
1996
1994
- Ecuador
- Philippinen
- Costa Rica
- Kolumbien
- Guatemala
- Belg./Luxbg.
- Honduras
- USA
- Ver. Arab. Emir.
- Panama
- Kamerun
Dr. Hochhaus
Banexport 2005
Daten ZMP

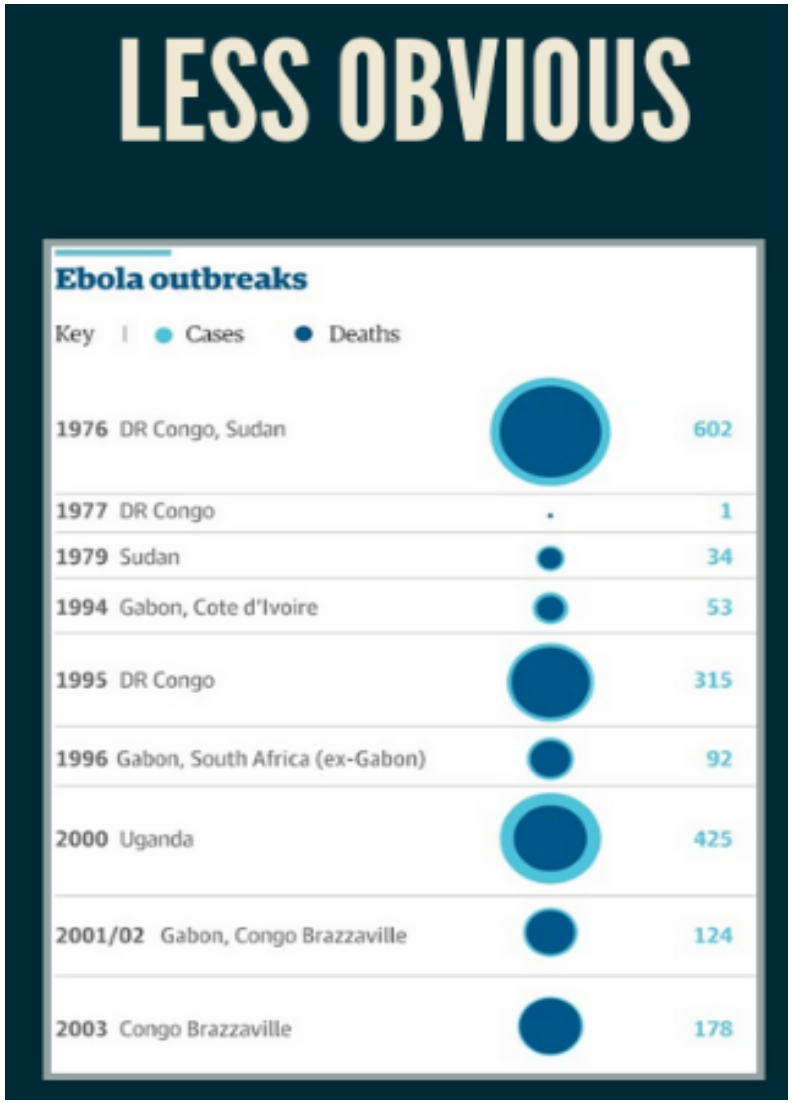
The ugly


FOX NEWS
UNEMPLOYMENT RATE
UNDER PRESIDENT OBAMA
10%
9.5%
9%
8.5%
8%
9.0%
8.9%
8.8%
9.0%
9.1%
9.2%
9.1%
9.1%
9.1%
9.0%
8.6%
JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV
2011
SOURCE: BUREAU OF LABOR STATISTICS
FOX NEWS radio
AR WAR AND A TROOP WITHDRAWAL AT THE E
NAS FUT 2,292.50

The ugly


INDEED ALL PIES
Pie Chart of Countries
UK
US
Australia
France
Germany

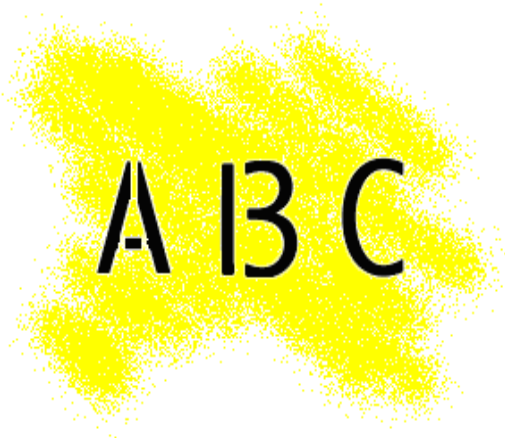
The ugly


LESS OBVIOUS
Ebola outbreaks
Key
Cases
Deaths
1976 DR Congo, Sudan
602
1977 DR Congo
1
1979 Sudan
34
1994 Gabon, Cote d'Ivoire
53
1995 DR Congo
315
1996 Gabon, South Africa (ex-Gabon)
92
2000 Uganda
425
2001/02 Gabon, Congo Brazzaville
124
2003 Congo Brazzaville
178

FEW ISSUES

Few issues

- Human perception can be ...

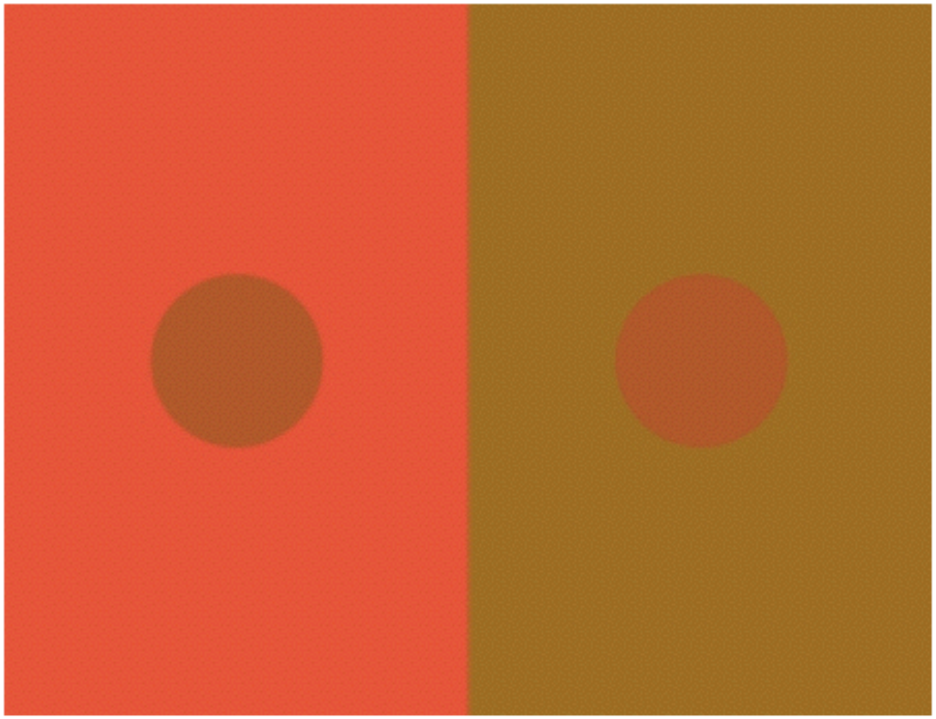


Few issues

- Human perception can be deceiving

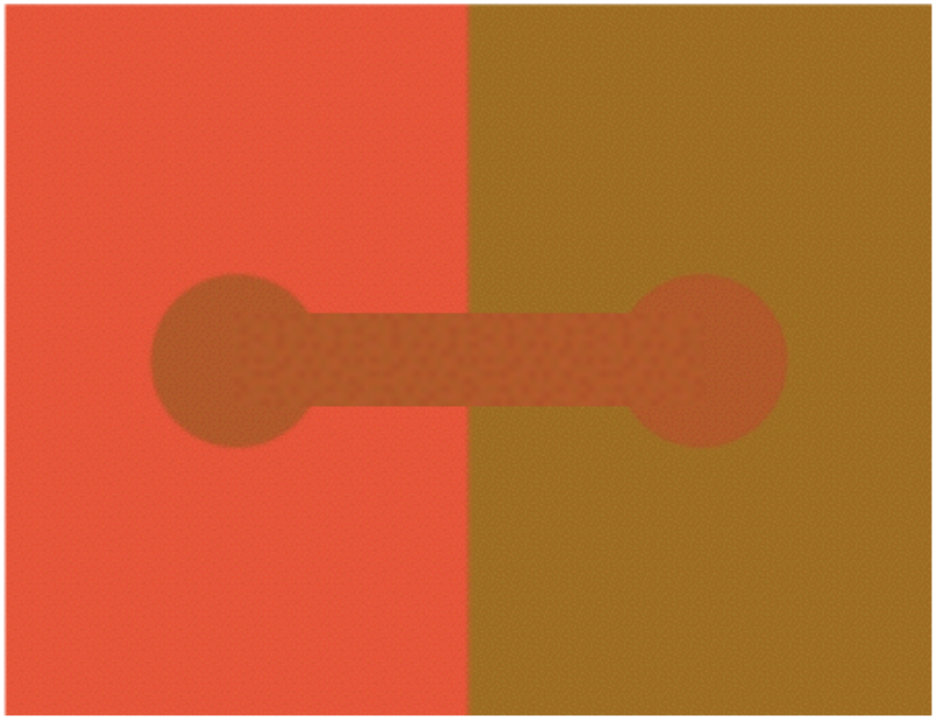
Few issues

- Another example:



Few issues

- Another example:



Few issues

- Selective attention

Selective attention test

Graphics in R

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

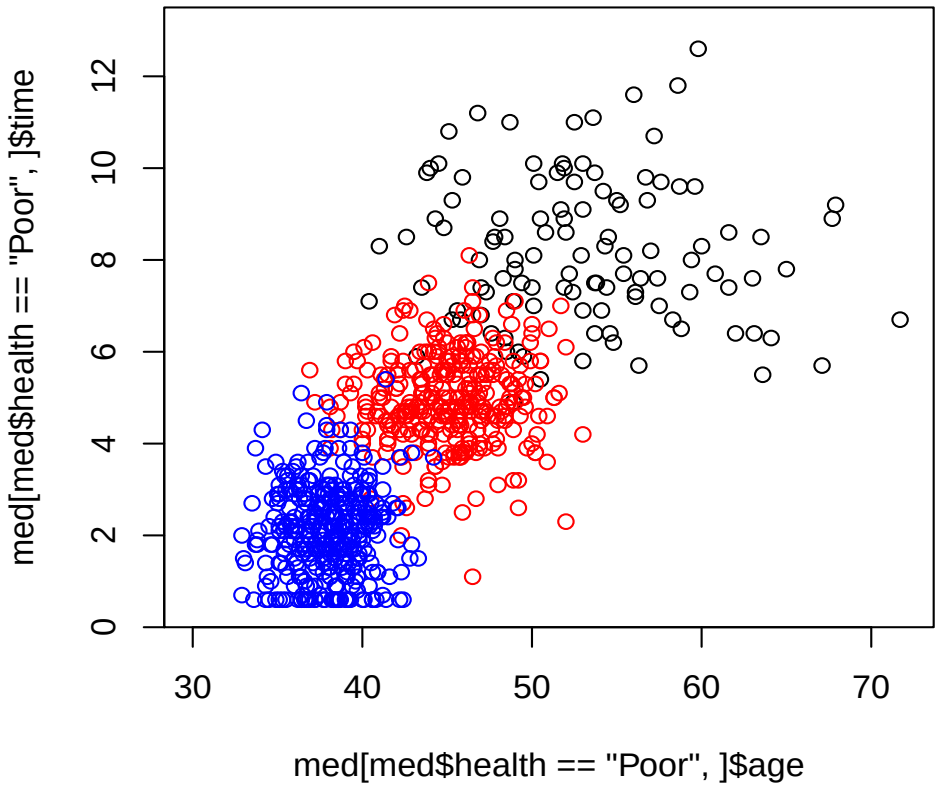
```
> install.packages("drat")
```

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

```
> install.packages("BristolVis", type="source")
```

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

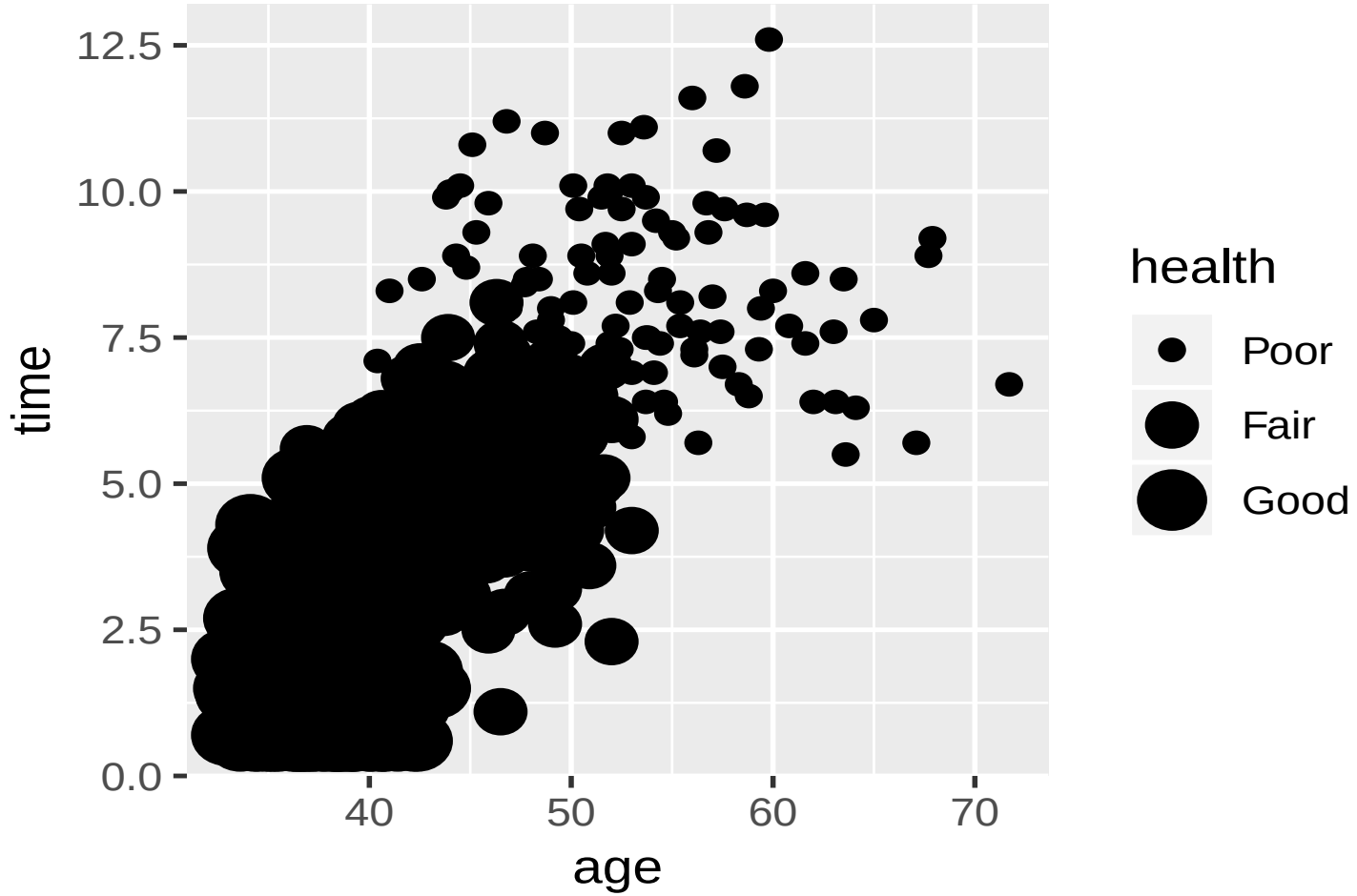
- We had to manually set the scales using the `xlim` and `ylim` parameters.
- We had not created a legend. We would need to use the `legend` function to create one.
- The default axis labels were terrible!

Equivalent graphics using ggplot2

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

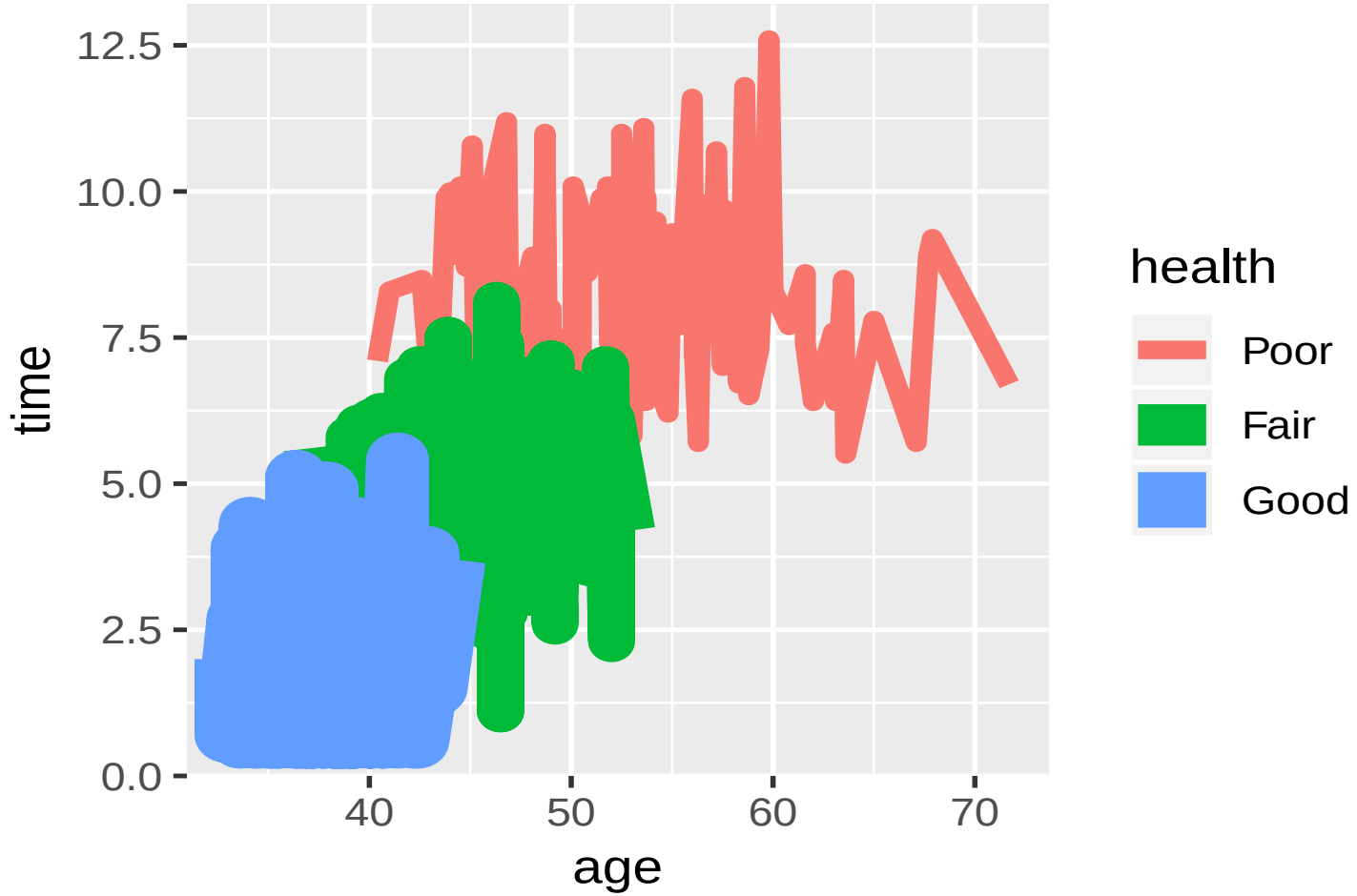
Factor of point sizes

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



Example of a line chart

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



Geometric objects

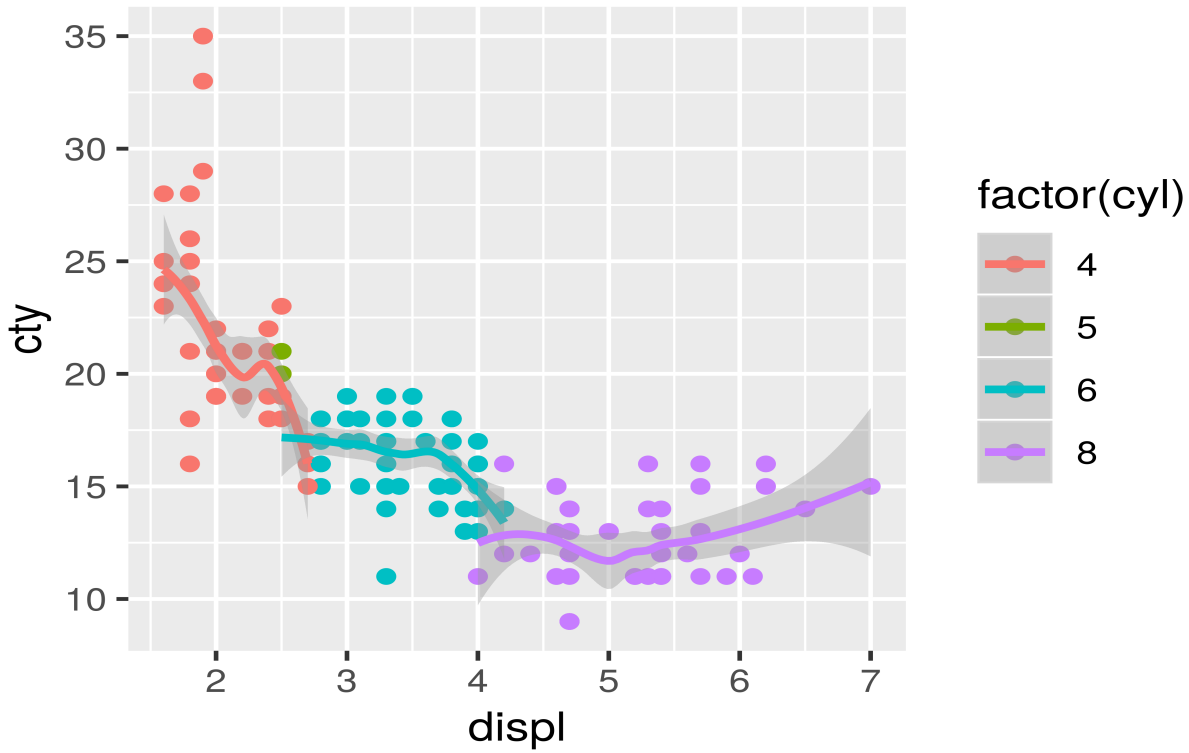
Points, bars and lines are examples of geom's. Some useful standard geoms and their equivalent base graphic counter part:

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:

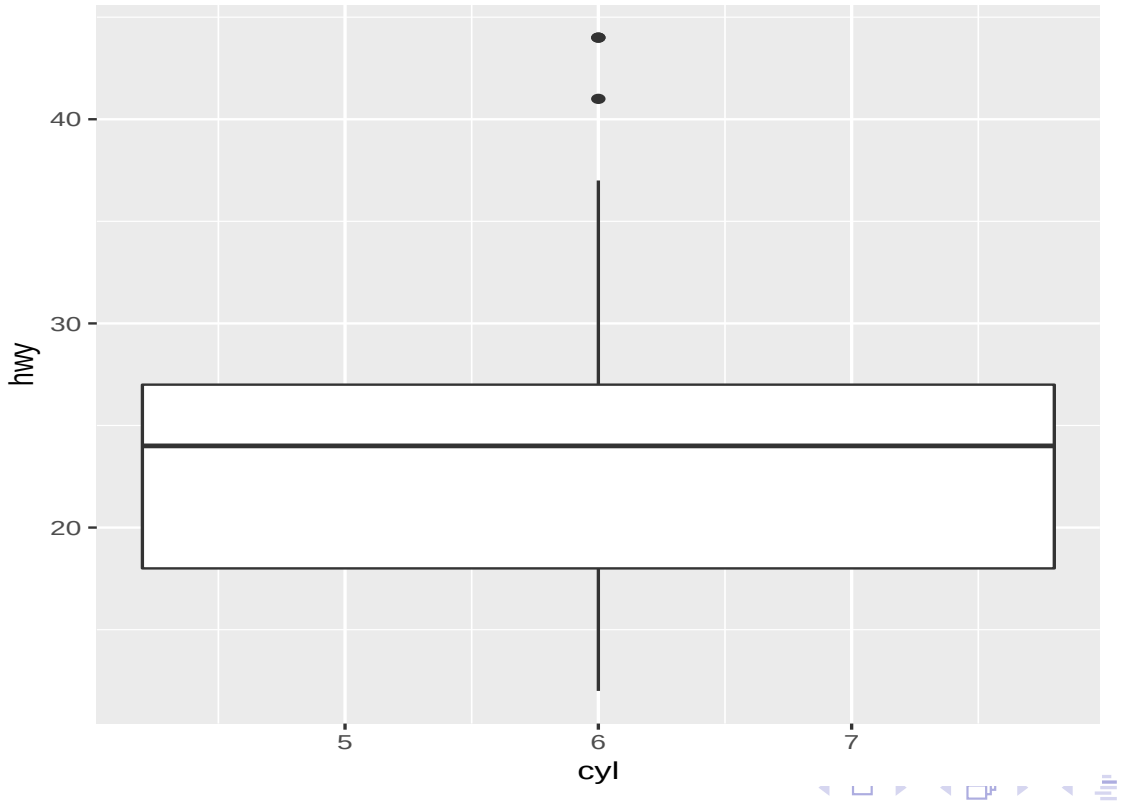
Name	Description
`abline`	Line, specified by slope and intercept
`boxplot`	Box and whiskers plot
`density`	Kernel density plot
`histogram`	Histograms
`jitter`	Individual points are jittered to avoid overlap
`step`	Connect observations by stairs

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.

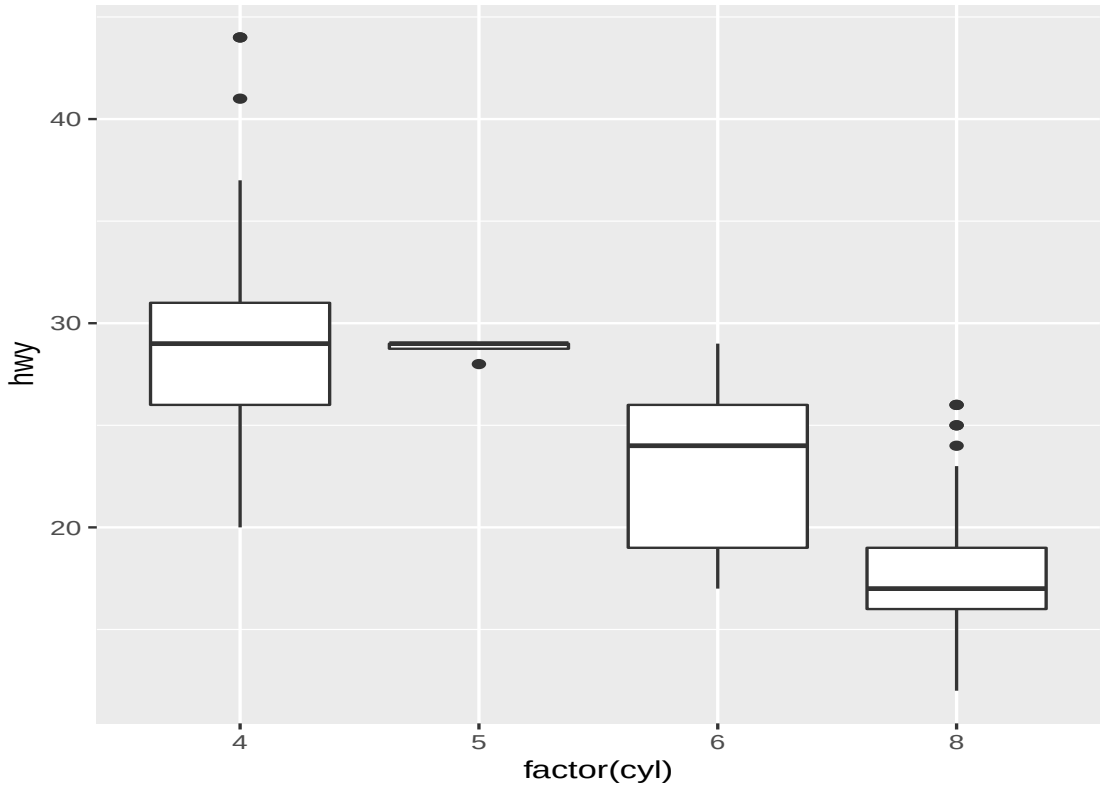
Now we'll create a boxplot: `(g1 = g + geom_boxplot())`



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:

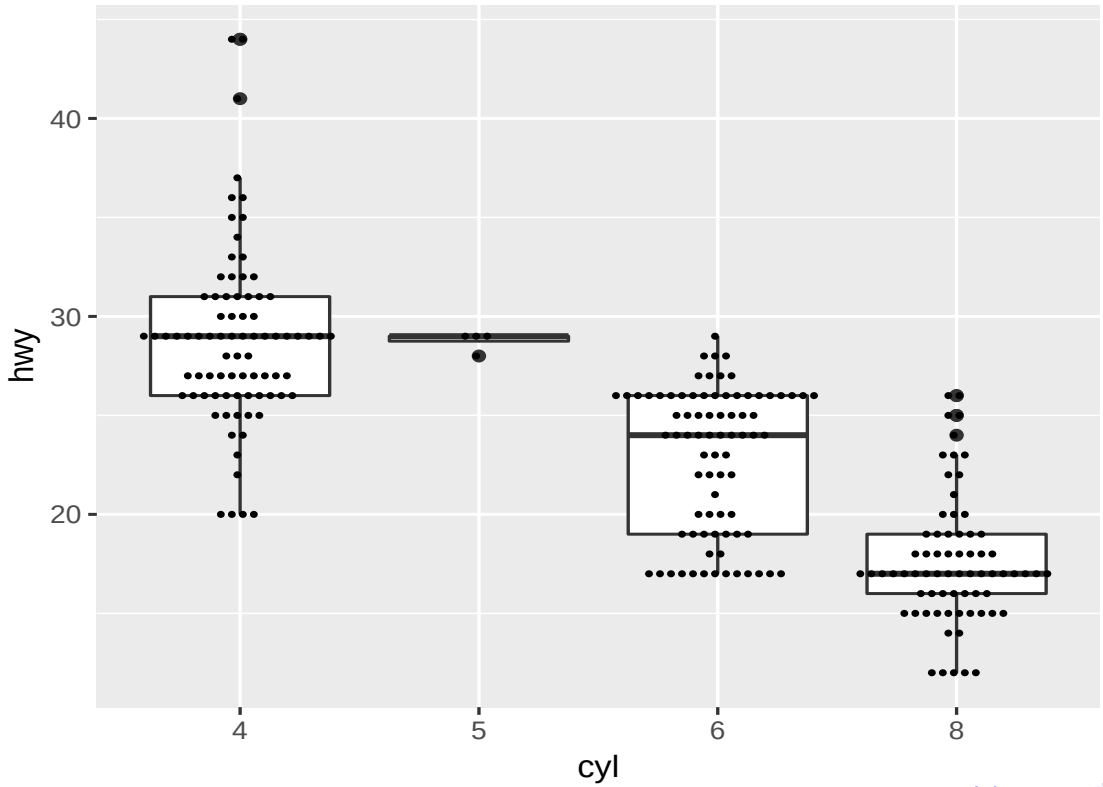
```
(g2 = g + geom_boxplot(aes(x=factor(cyl), group=cyl)))
```



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

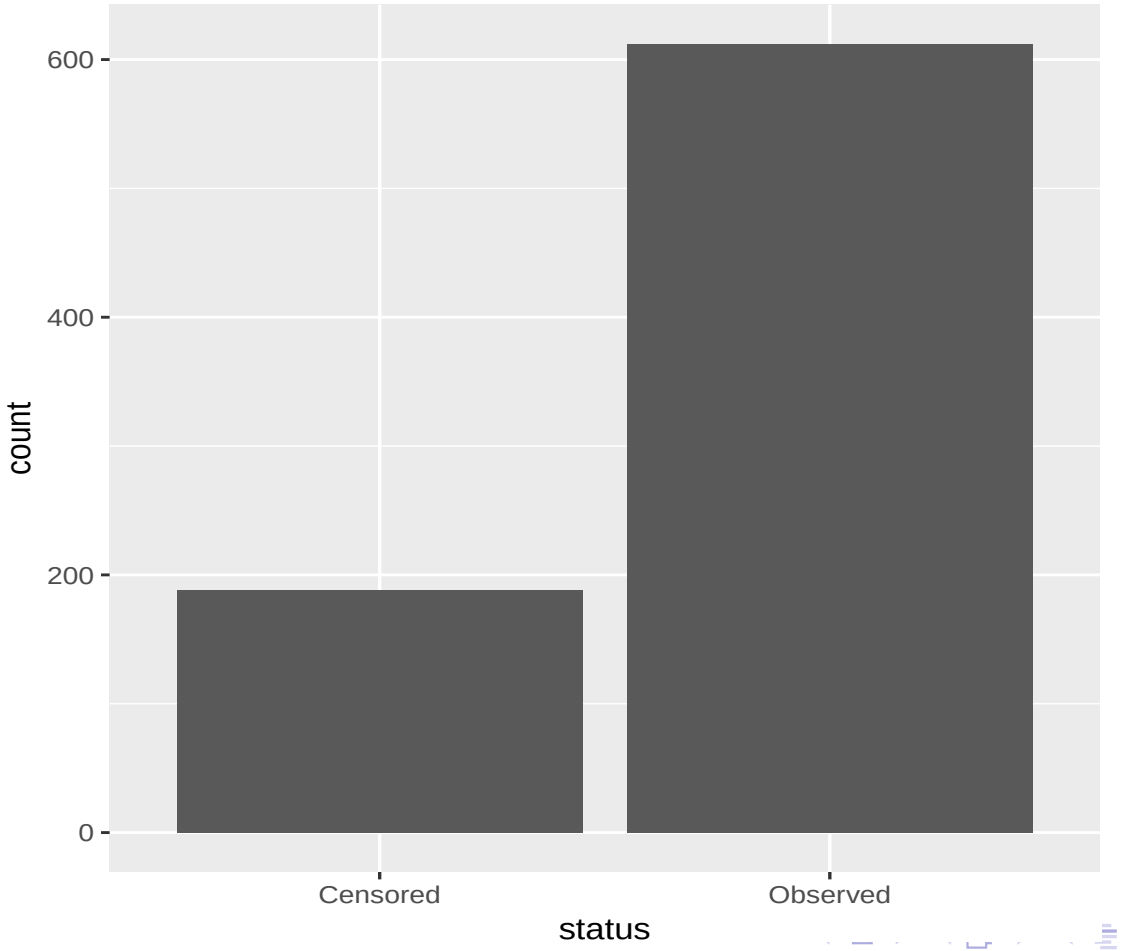
There are a few standard geom 's that are particular useful:

- 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:

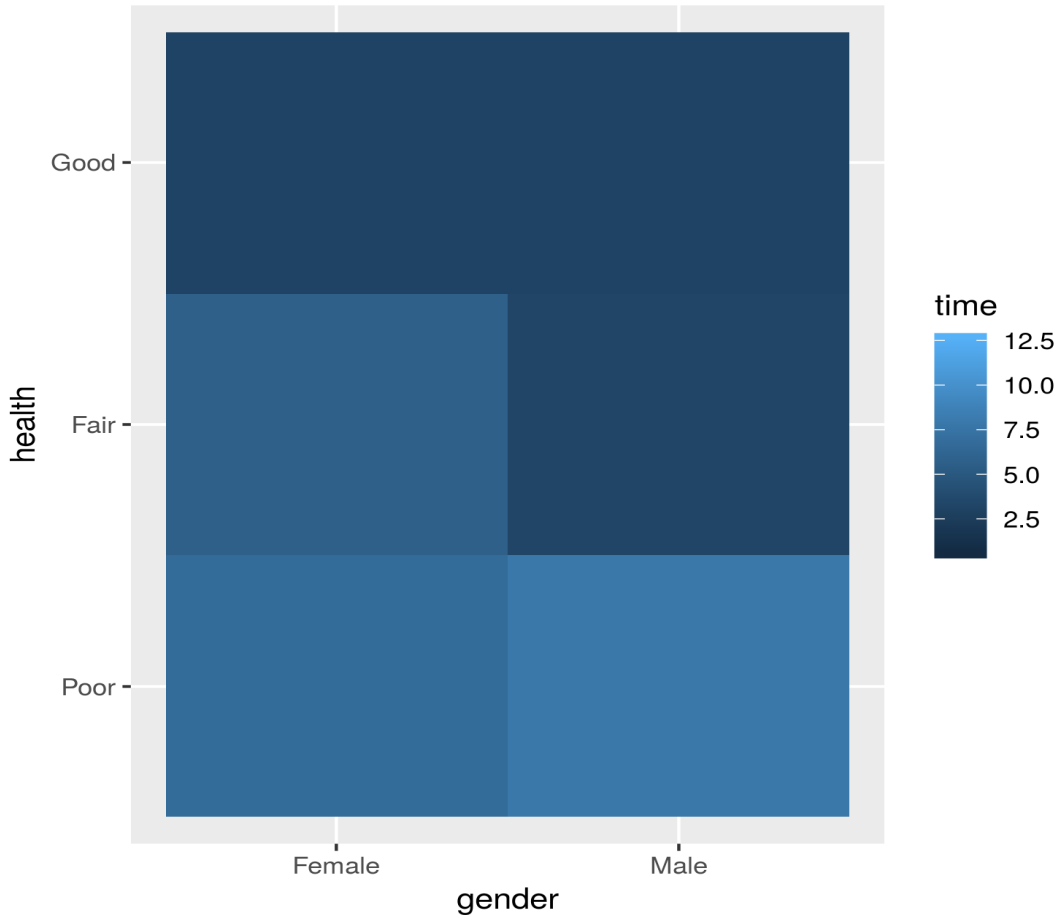
```
ggplot(med, aes(x=status)) + geom_bar()
```



Standard plots

An example of a geom_raster:

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



Facets

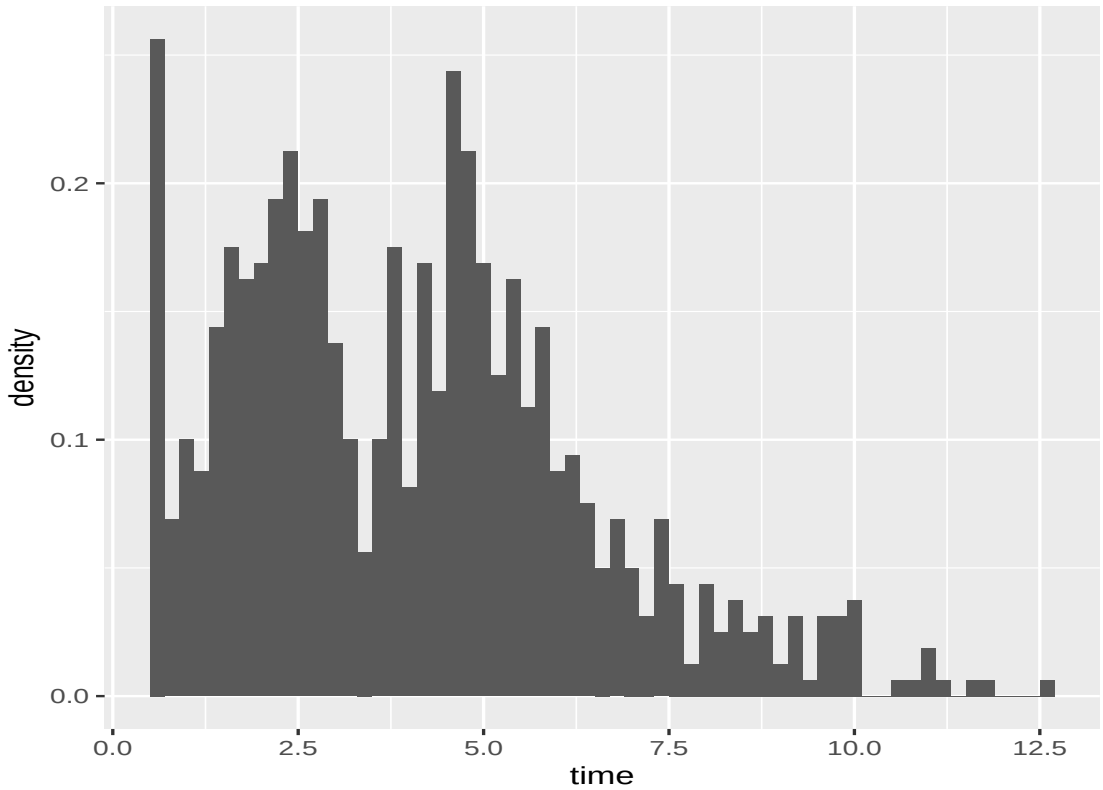
Facets

- Faceting is a mechanism for automatically laying out multiple plots on a page.
- The data is split into subsets, each subset onto a different panel.
- `ggplot2` has two types of faceting:
  - `facet_grid`: produces a 2d panel of plots where variables define rows and columns.
  - `facet_wrap`: produces a 1d ribbon of panels which can be wrapped into 2d.

Facet grid

Suppose we are interested in time to relief from the med data set. A first plot we could generate is a basic histogram:

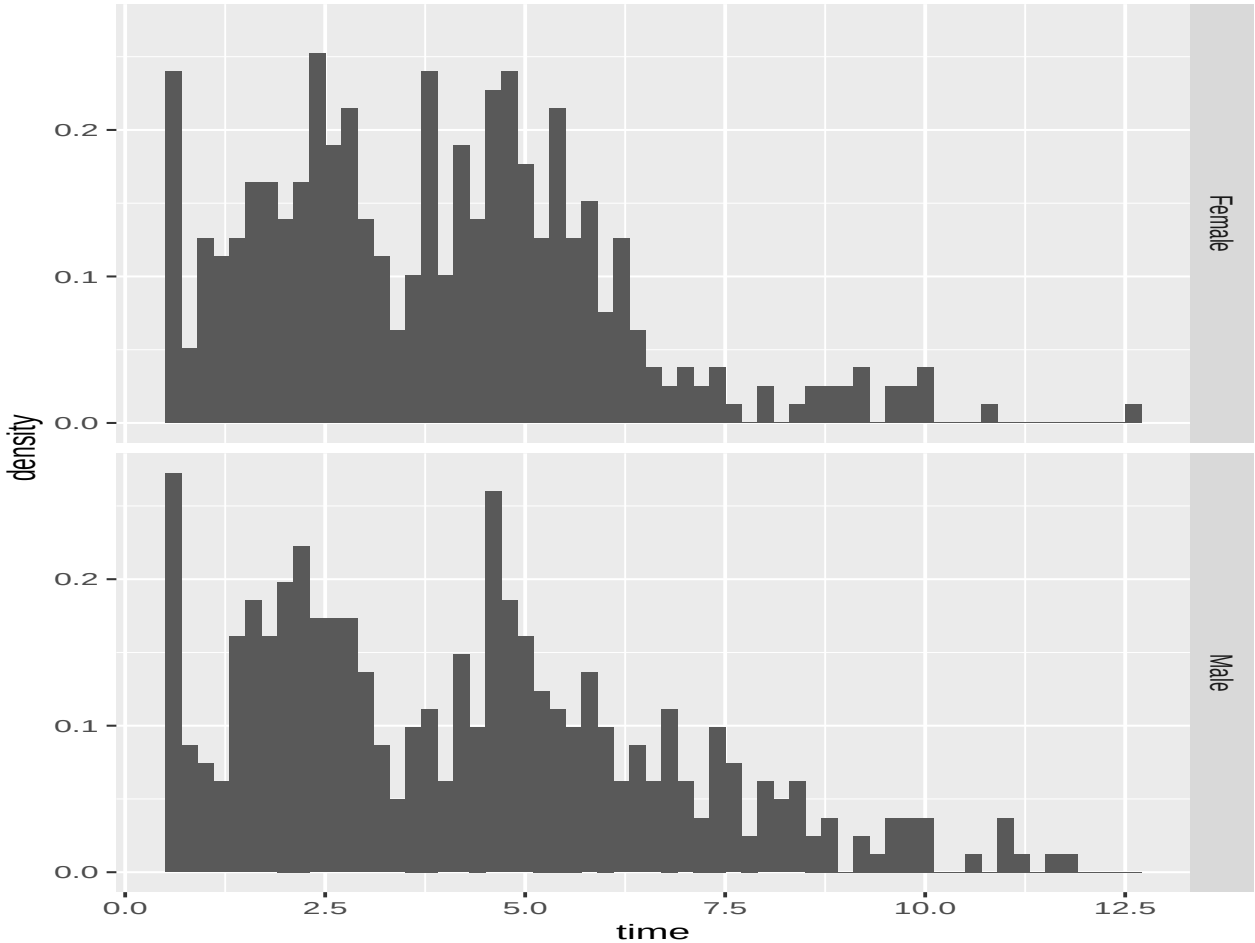
```
g = ggplot(med, aes(x=time)) +
geom_histogram(aes(y=..density..), binwidth=0.2))
```



Facet grid

We will now use faceting to explore the data further:

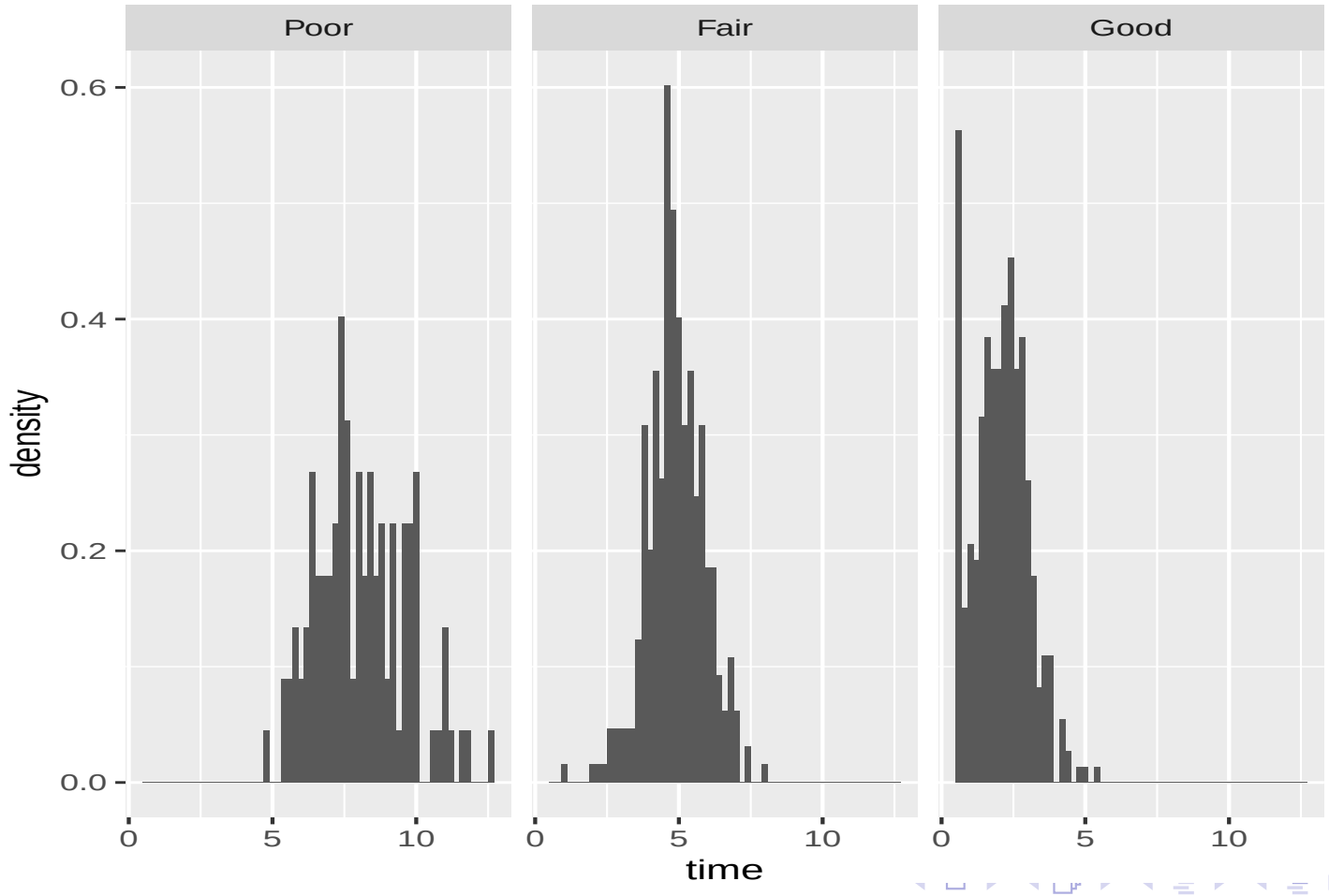
```
g + facet_grid(gender ~ .)
```



Facet grid

We could also split data by health condition in columns:

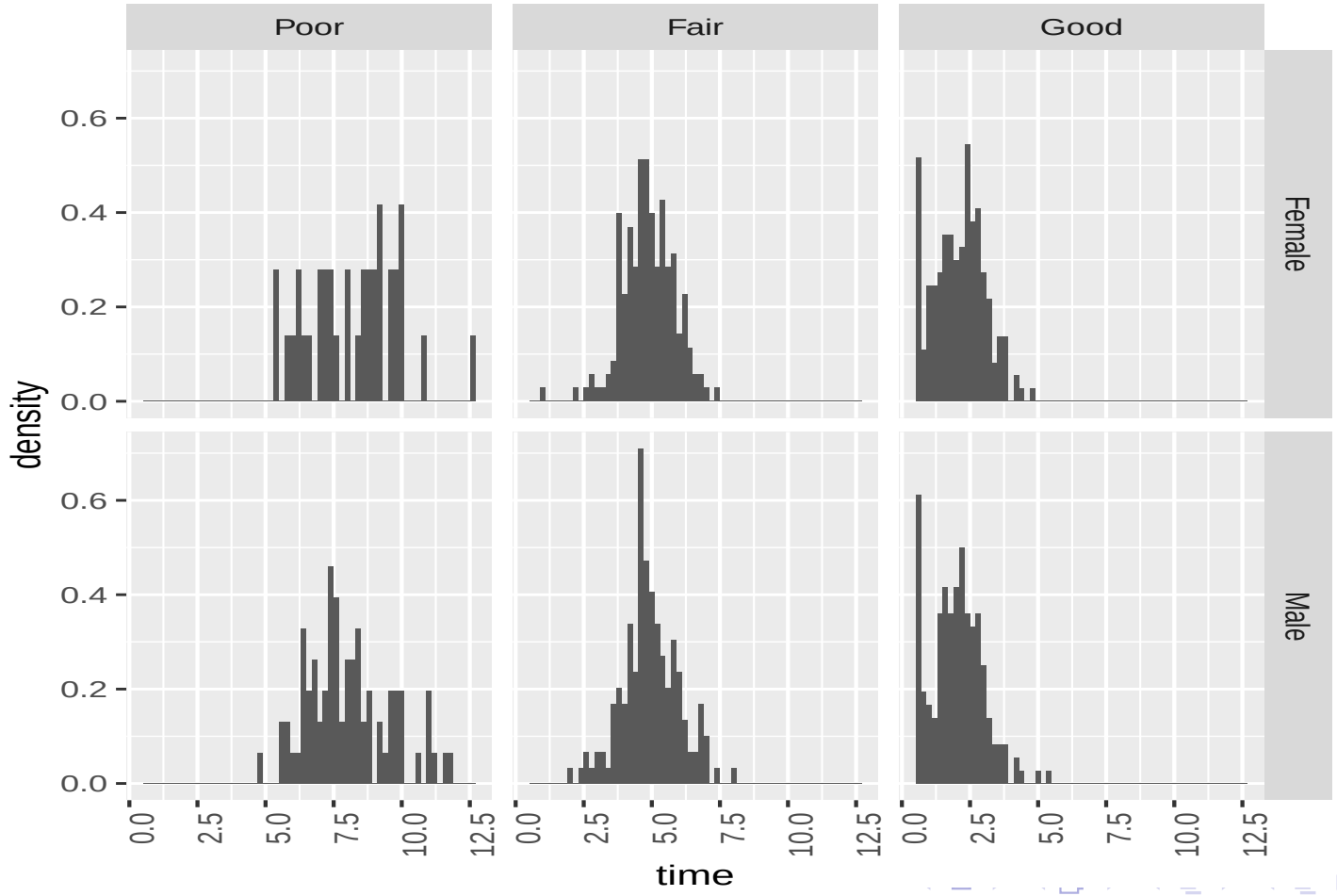
```
g + facet_grid(. ~ health) + scale_x_continuous(breaks = c(0, 5, 10))
```



Facet grid

Both rows and columns can be used to split data by:

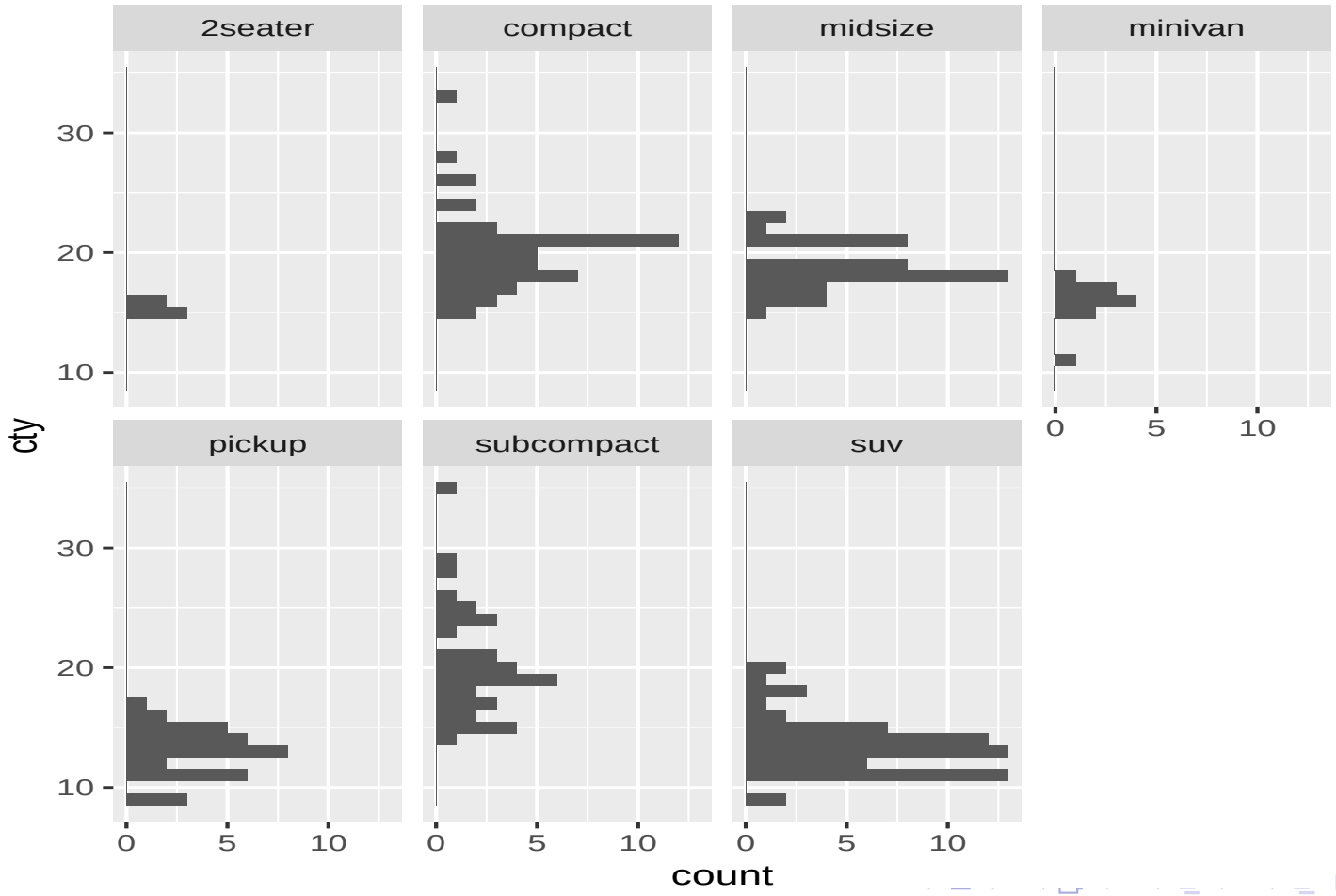
```
g + facet_grid(gender ~ health) + theme(axis.text.x =
element_text(angle = 90, hjust = 1))
```



Facet wrap

The `facet_wrap` function creates a 1d ribbon of plots:

```
ggplot(mpg, aes(x=cty)) + geom_histogram(binwidth=1) +
facet_wrap(~class, ncol=4) + coord_flip()
```



Themes

What are themes?

Themes are used in `ggplot2` to ease making consistent changes to all your plots. For example:

- change the font size;
- change background colour;
- control angels of axis label.
- etc.

Choice of themes

A few useful standard themes are available in the `ggthemes` package which can be loaded using:

```
require(ggthemes)
```

Once loaded, a range of standard themes would be available to control the plot appearance:

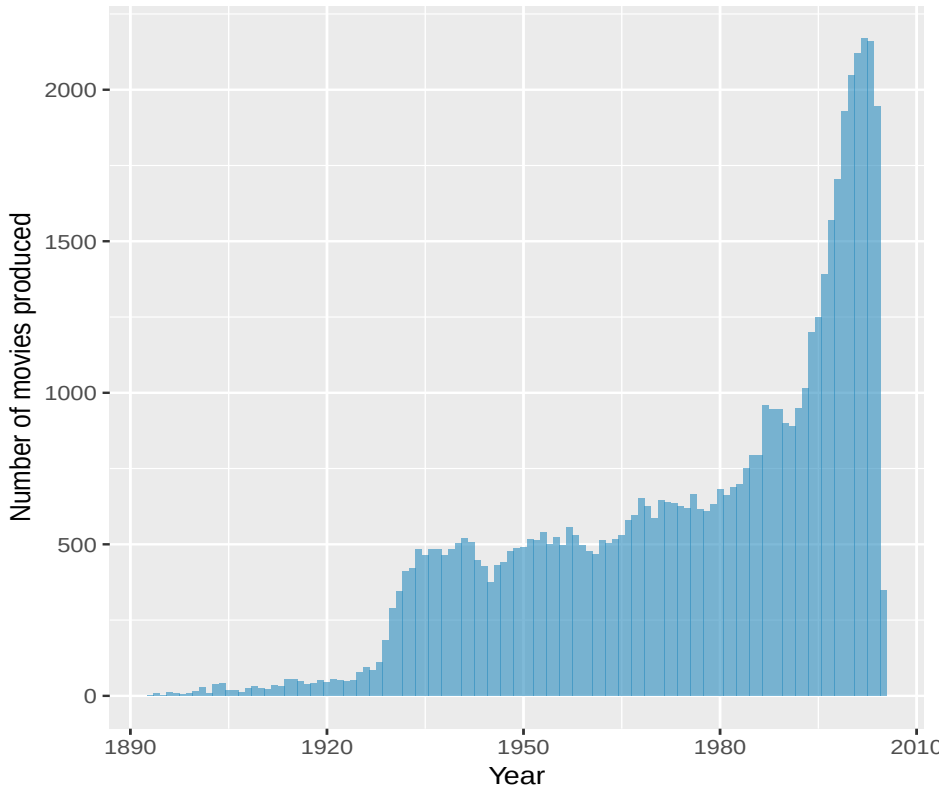
- `theme_bw()`
- `theme_economist()`
- `theme_stata()` - useful for stata users
- `theme_pander()`
- `theme_hc(style = "darkunica")`

Theme examples

I will use the movies data from the ggplot2movies to show few examples of standard themes:

```
data(movies, package="ggplot2movies")
```

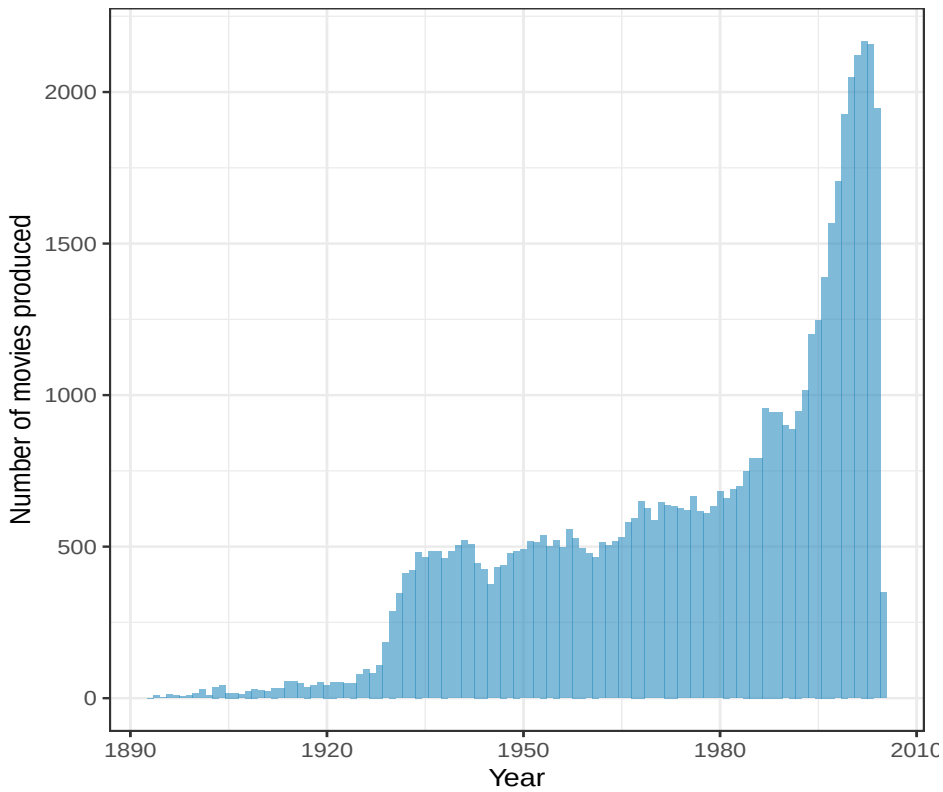
```
(h=
ggplot(movies, aes(year))+
geom_histogram(binwidth=1,
fill="#2b8cbe",alpha=0.6)+
xlab("Year")+ylab("Number
of movies produced"))
```



Theme examples

Using the `theme_bw()`, the appearance can be changed to:

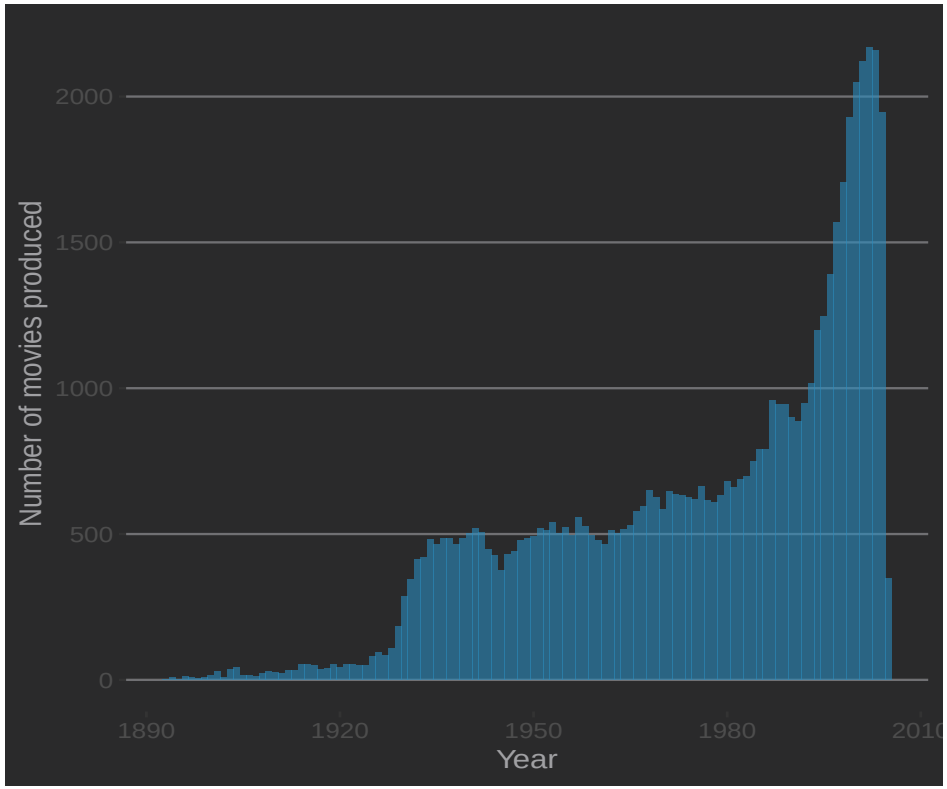
```
h + theme_bw()
```



Theme examples

Using the `theme_hc(style = "darkunica")`, the appearance can be changed to:

```
h + theme_hc(style =
"darkunica")
```



Define our own theme

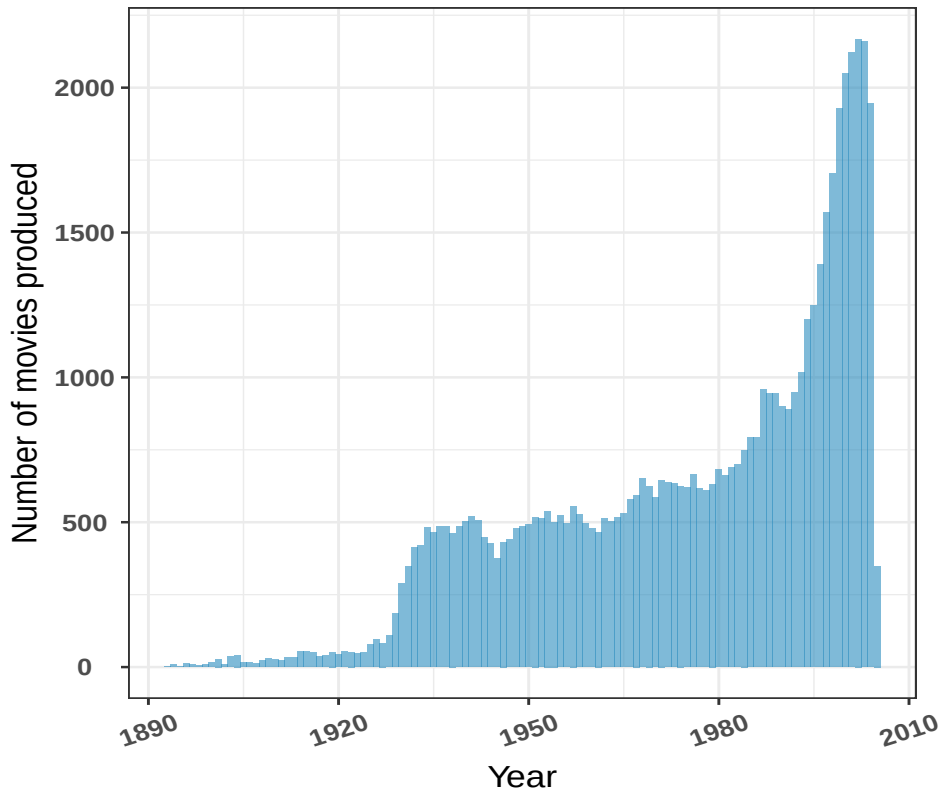
We can setup our own theme:

```
My_theme <-
theme(axis.text=element_text(size=10, face="bold"),
axis.text.x=element_text(angle=20, vjust=0.5),
axis.text.y=element_text(vjust=0.5, hjust = 0.5),
plot.margin = unit(c(0.5,0.5,0.5,0.5), "cm"),
axis.title=element_text(size = rel(1.2)),
axis.title.x=element_text(vjust=-0.5),
axis.title.y=element_text(vjust=1.5))
```

Define our own theme

Then, we can use it using:

```
h + theme_bw() + My_theme
```



Useful links

R Courses

Details and materials of a set of author's courses:
http://osmahmoud.com/R-courses/

Github

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

Shiny web-page

The RVis tool for learners of data visualisation using R:
http://bristol-medical-stat.bristol.ac.uk:3838/RVis/

Thank You