ggplot2

Dr. Jennifer (Jenny) Bryan
Department of Statistics and Michael Smith Laboratories
University of British Columbia

Digression: R’s formula syntax

use in another intro?

http://cran.r-project.org/doc/manuals/R-intro.html#Formulae-for-statistical-models

```
y ~ x
```

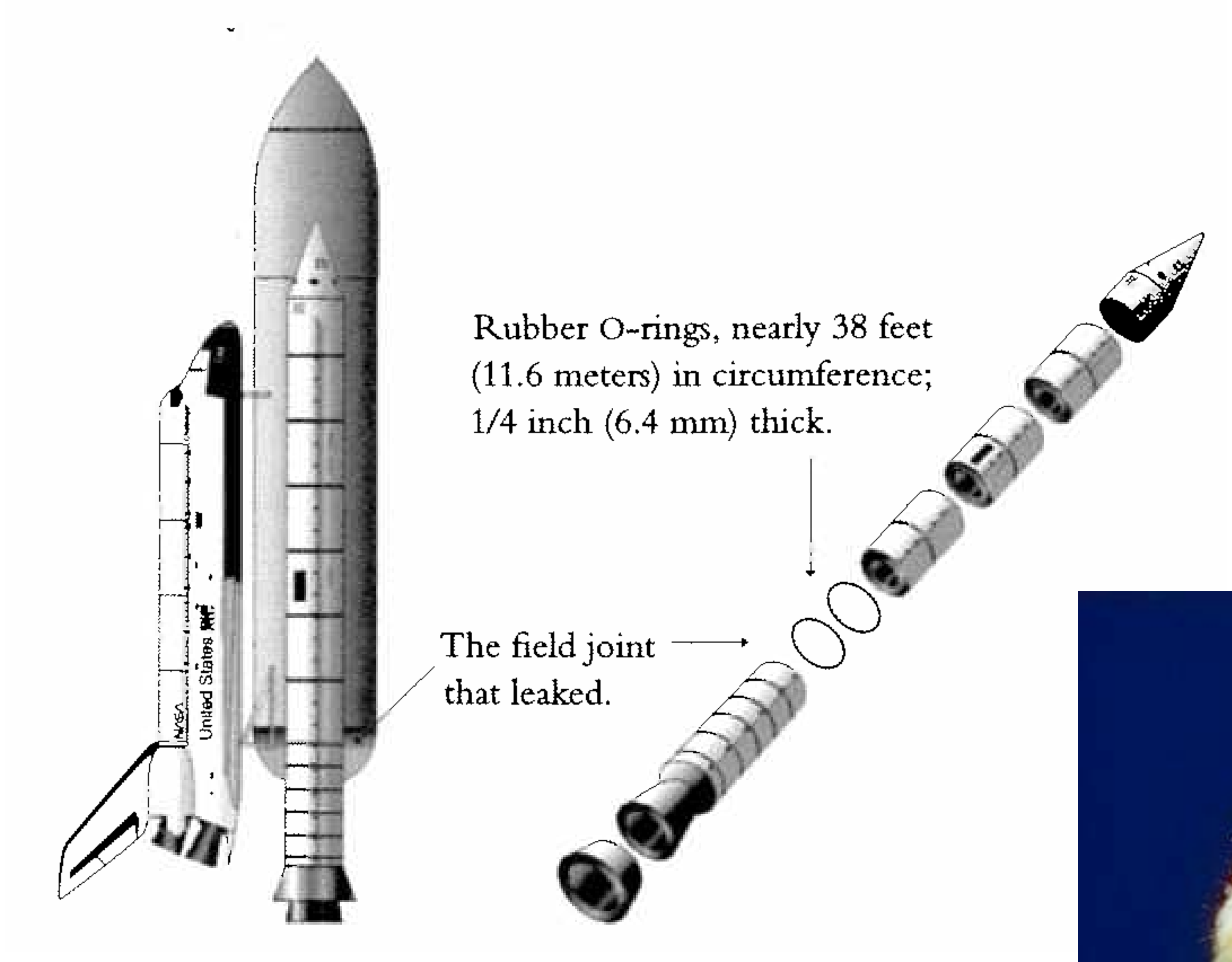
“y twiddle x”

In modelling functions, says y is response or dependent variable and x is the predictor or covariate or independent variable. More generally, the right-hand side can be much more complicated.

In many plotting functions, esp. lattice, this says to plot y against x.

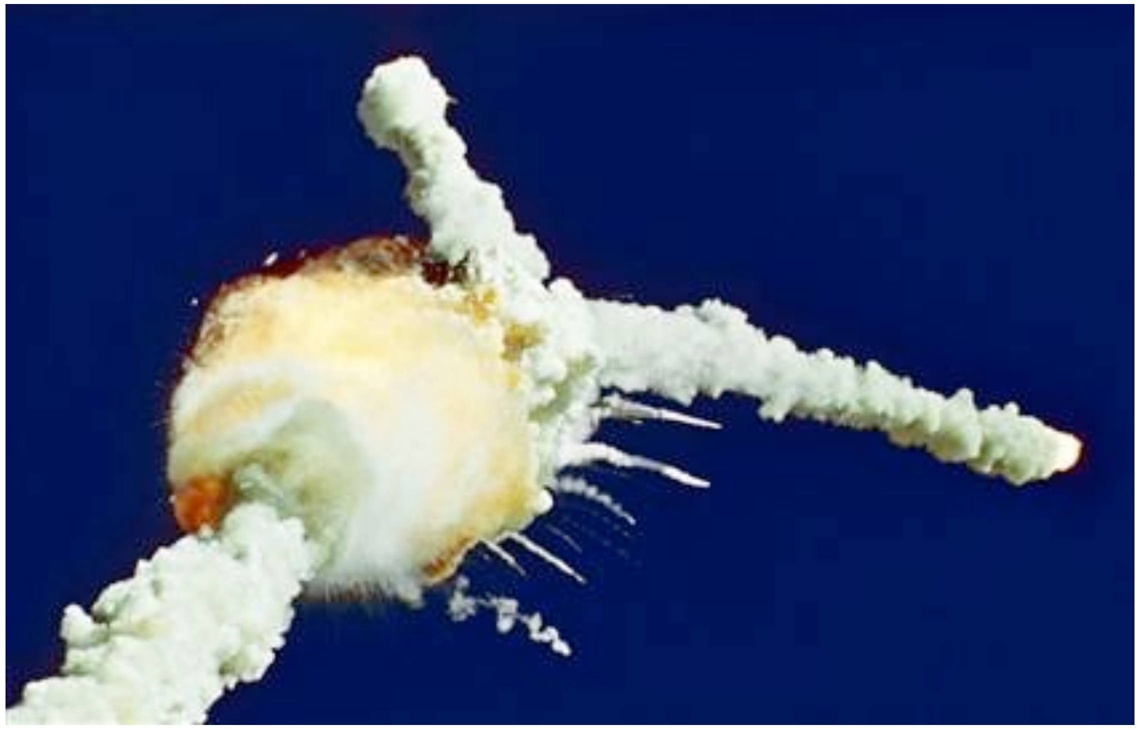
“A picture is worth
a thousand words”

1986 Challenger space shuttle disaster
Favorite example of Edward Tufte



Rubber O-rings, nearly 38 feet (11.6 meters) in circumference; 1/4 inch (6.4 mm) thick.

The field joint that leaked.



http://msnbcmedia1.msn.com/j/msnbc/Components/Photos/050709/050609_columbia_hmed_6p.hmedium.jpg

TEMPERATURE CONCERN ON SRM JOINTS

27 JAN 1986

HISTORY OF O-RING DAMAGE ON SRM FIELD JOINTS

			Cross Sectional View			Top View		
		SRM No.	Erosion Depth (in.)	Perimeter Affected (deg)	Nominal Dia. (in.)	Length Of Max Erosion (in.)	Total Heat Affected Length (in.)	Clocking Location (deg)
Oct 30, 1985	61A LH Center Field**	22A	None	None	0.280	None	None	36° --66°
	61A LH ~~CENTER~~ AFT FIELD**	22A	NONE	NONE	0.280	NONE	NONE	338°-18°
85	51C LH Forward Field**	15A	0.010	154.0	0.280	4.25	5.25	163
	51C RH Center Field (prim)***	15B	0.038	130.0	0.280	12.50	58.75	354
	51C RH Center Field (sec)***	15B	None	45.0	0.280	None	29.50	354
	41D RH Forward Field	13B	0.028	110.0	0.280	3.00	None	275
	41C LH Aft Field*	11A	None	None	0.280	None	None	--
	41B LH Forward Field	10A	0.040	217.0	0.280	3.00	14.50	351
July	STS-2 RH Aft Field	2B	0.053	116.0	0.280	--	--	90

*Hot gas path detected in putty. Indication of heat on O-ring, but no damage.
**Soot behind primary O-ring.
***Soot behind primary O-ring, heat affected secondary O-ring.

Clocking location of leak check port - 0 deg.

OTHER SRM-15 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY AND NO SOOT NEAR OR BEYOND THE PRIMARY O-RING.

SRM-22 FORWARD FIELD JOINT HAD PUTTY PATH TO PRIMARY O-RING, BUT NO O-RING EROSION AND NO SOOT BLOWBY. OTHER SRM-22 FIELD JOINTS HAD NO BLOWHOLES IN PUTTY.

BLOW BY HISTORY

SRM-15 WORST BLOW-BY
- ○ 2 CASE JOINTS (80°), (110°) ARC
- ○ MUCH WORSE VISUALLY THAN SRM-22

SRM 22 BLOW-BY
- ○ 2 CASE JOINTS (30-40°)

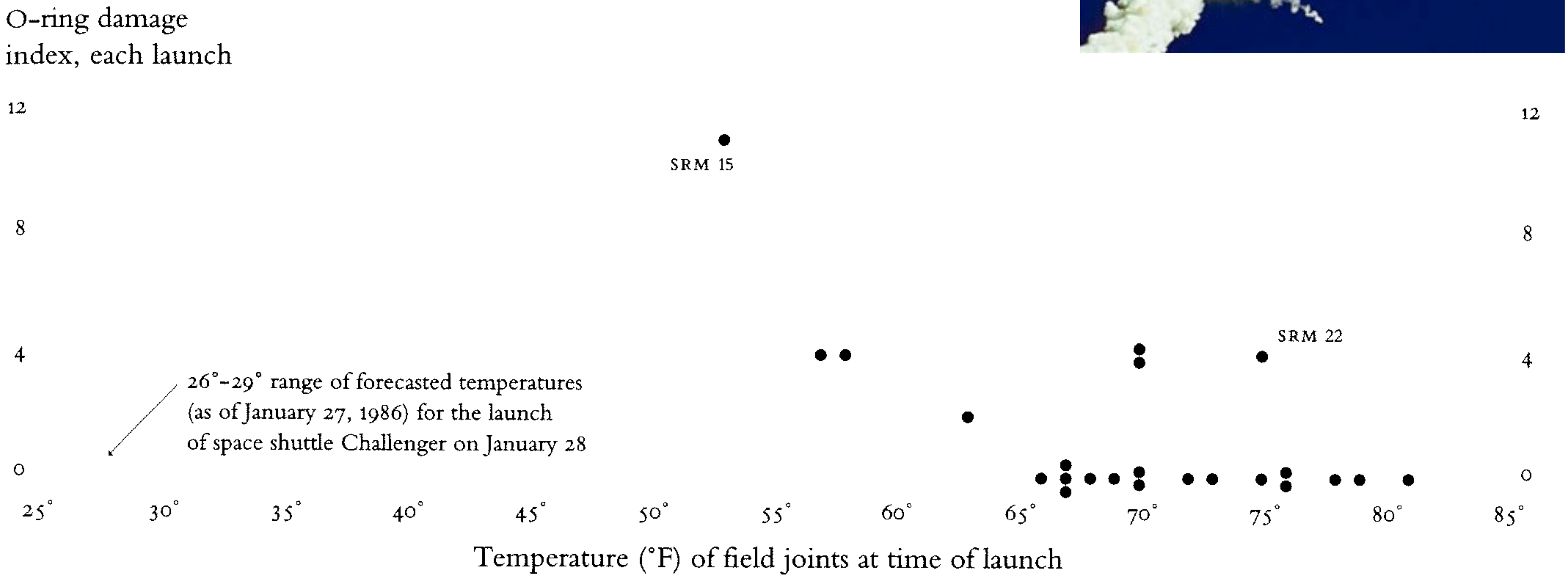
SRM-13A, 15, 16A, 18, 23A 24A
- ○ NOZZLE BLOW-BY

HISTORY OF O-RING TEMPERATURES (DEGREES - F)

MOTOR	MBT	AMB	O-RING	WIND
DM-4	68	36	47	10 MPH
DM-2	76	45	52	10 MPH
QM-3	72.5	40	48	10 MPH
QM-4	76	48	51	10 MPH
SRM-15	52	64	53	10 MPH
SRM-22	77	78	75	10 MPH
SRM-25	55	26	29 27	10 MPH 25 MPH

MOTOR	O-RING
DM-4	47
DM-2	52
QM-3	48
QM-4	51
SRM-15	53
SRM-22	75
SRM-25	29 27

“A picture is worth a thousand words”


O-ring damage
index, each launch
12
8
4
0
12
8
4
0
SRM 15
SRM 22
26°-29° range of forecasted temperatures
(as of January 27, 1986) for the launch
of space shuttle Challenger on January 28
25°
30°
35°
40°
45°
50°
55°
60°
65°
70°
75°
80°
85°
Temperature (°F) of field joints at time of launch

“A picture is worth a thousand words”

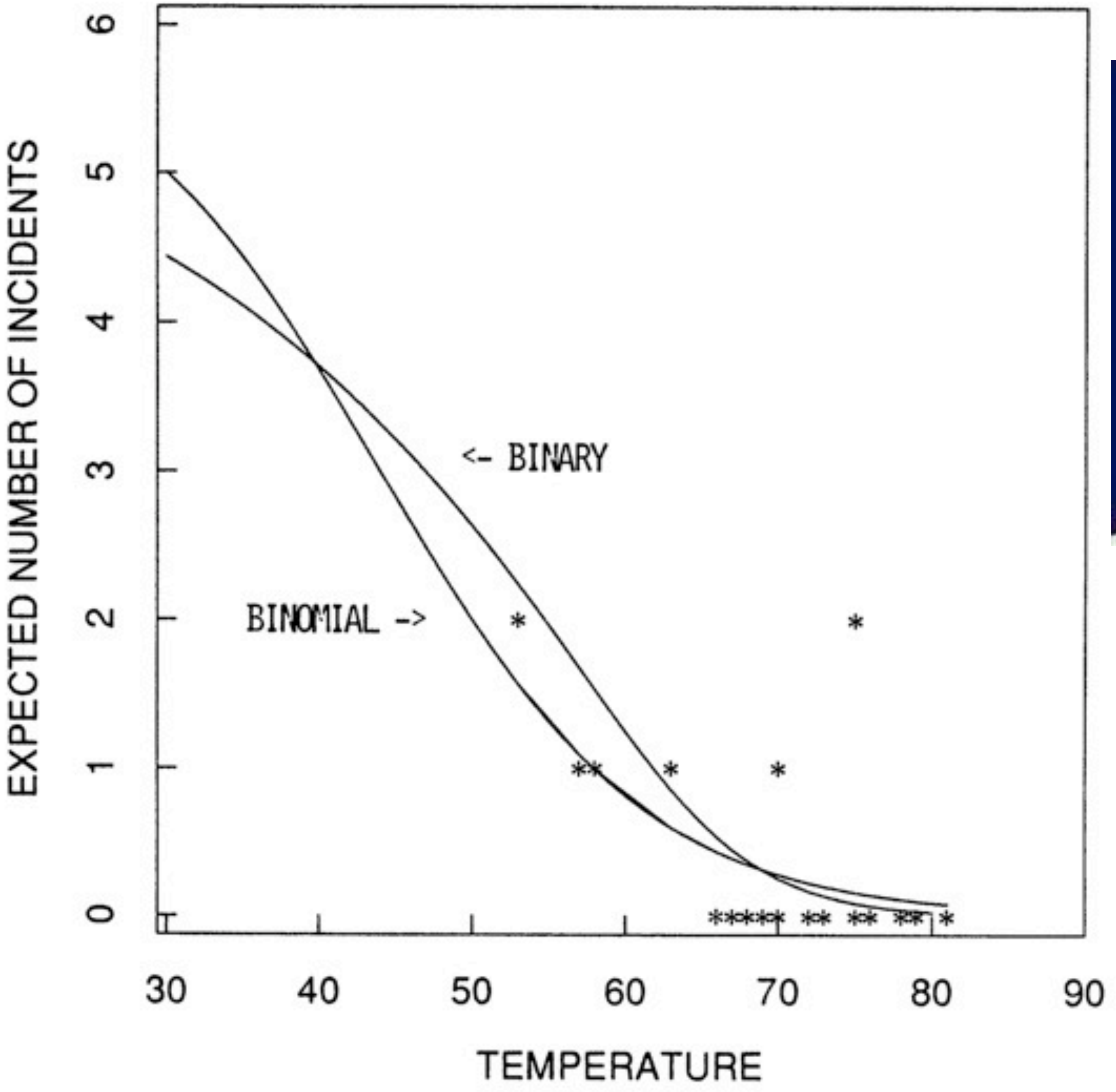


Figure 4. O-Ring Thermal-Distress Data: Field-Joint Primary O-Rings, Binomial-Logit Model, and Binary-Logit Model.

Siddhartha R. Dalal; Edward B. Fowlkes; Bruce Hoadley. Risk Analysis of the Space Shuttle: Pre-Challenger Prediction of Failure. JASA, Vol. 84, No. 408 (Dec., 1989), pp. 945-957. Access via JSTOR.

Edward Tufte
http://www.edwardtufte.com

BOOK:
Visual Explanations: Images and Quantities, Evidence and Narrative

Ch. 5 deals with the Challenger disaster
That chapter is available for $7 as a downloadable booklet:
http://www.edwardtufte.com/tufte/books_textb

“A picture is worth a thousand words”

Always, always, always plot the data.

Replace (or complement) ‘typical’ tables of data or statistical results with figures that are more compelling and accessible.

Whenever possible, generate figures that overlay / juxtapose observed data and analytical results, e.g. the ‘fit’.

base or traditional graphics

vs

`lattice` package

ships with R, but must load with `library(lattice)`

vs

`ggplot2` package

must be installed and loaded

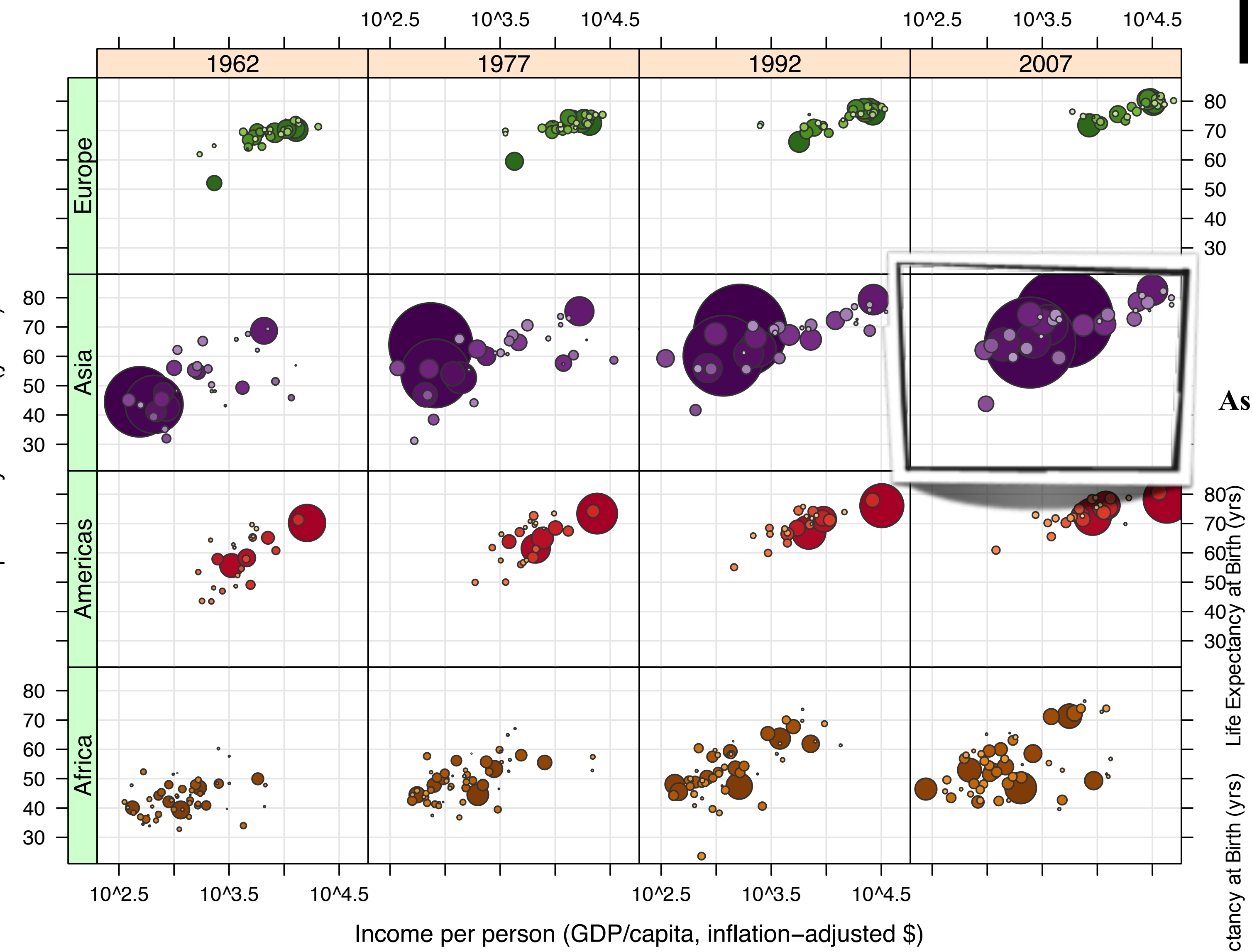
```
install.packages(“ggplot2”, dependencies = TRUE)
library(ggplot2)
```

Two main goals for statistical graphics

- To facilitate comparisons.
- To identify trends.

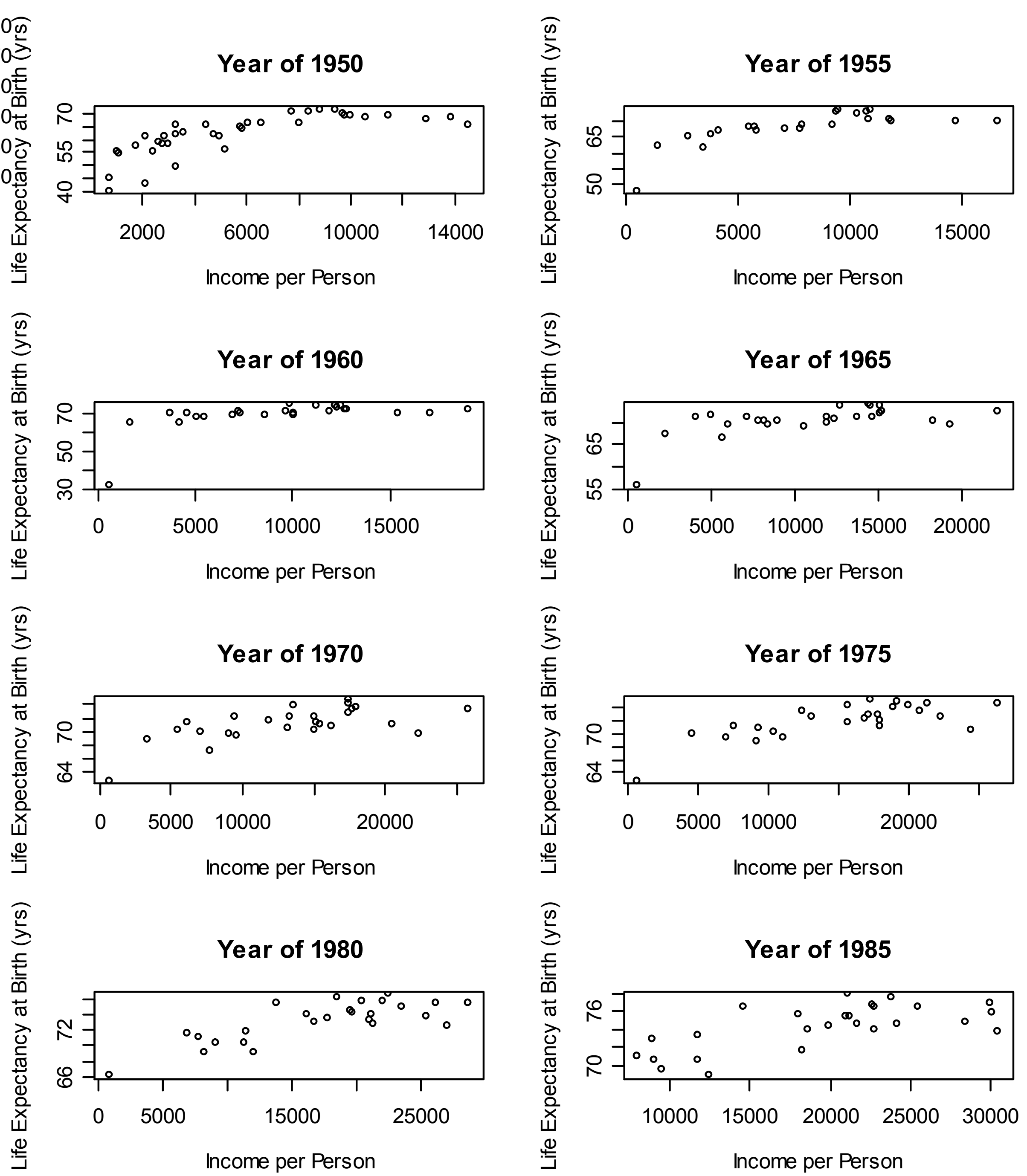
lattice and ggplot2 graphics are simply better than traditional graphics for achieving these goals

lattice



"multi-panel conditioning"

```
lifeExp ~ gdpPercap | continent * year
```



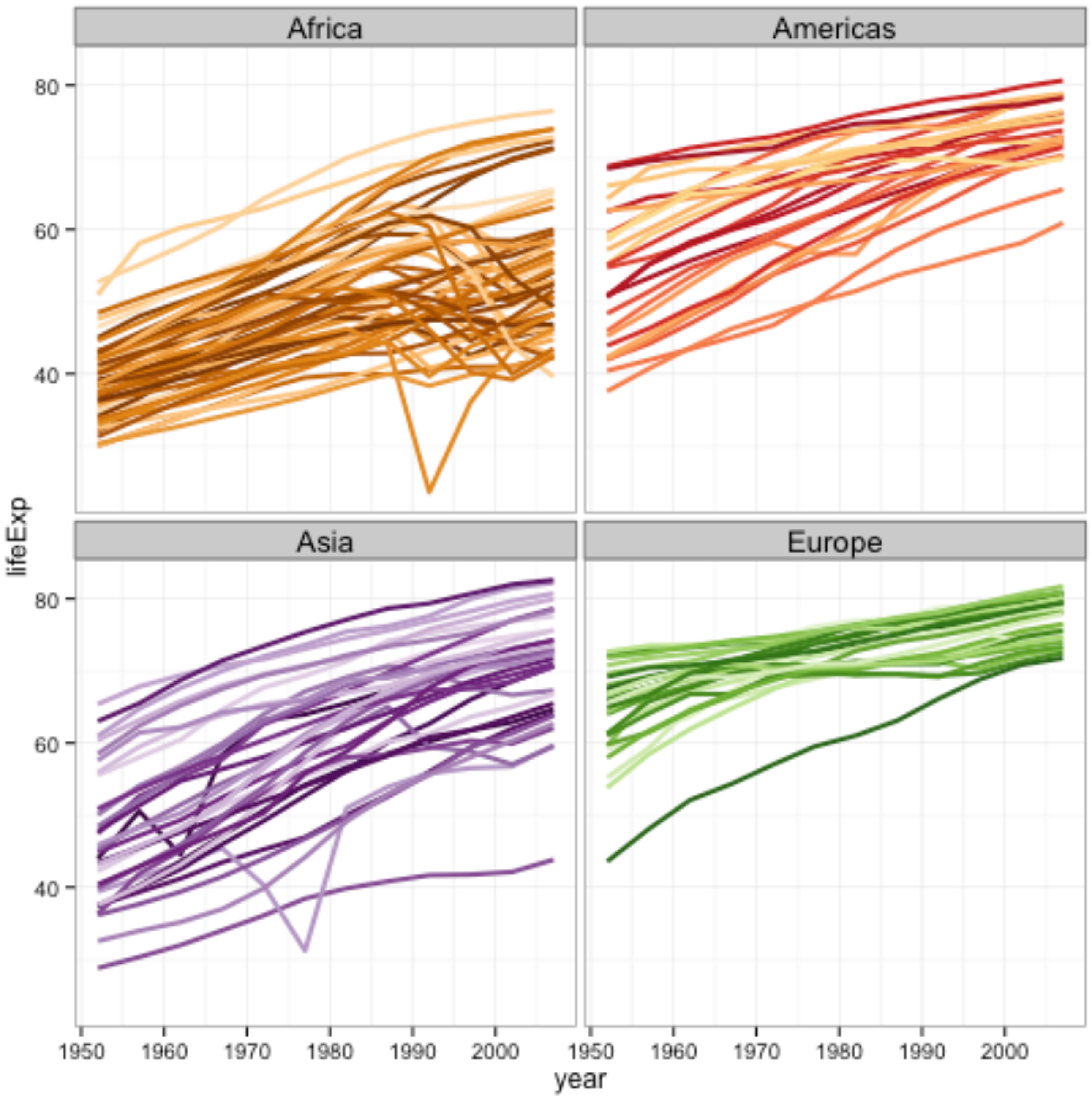
Assignment 1: Best Set of Graphs

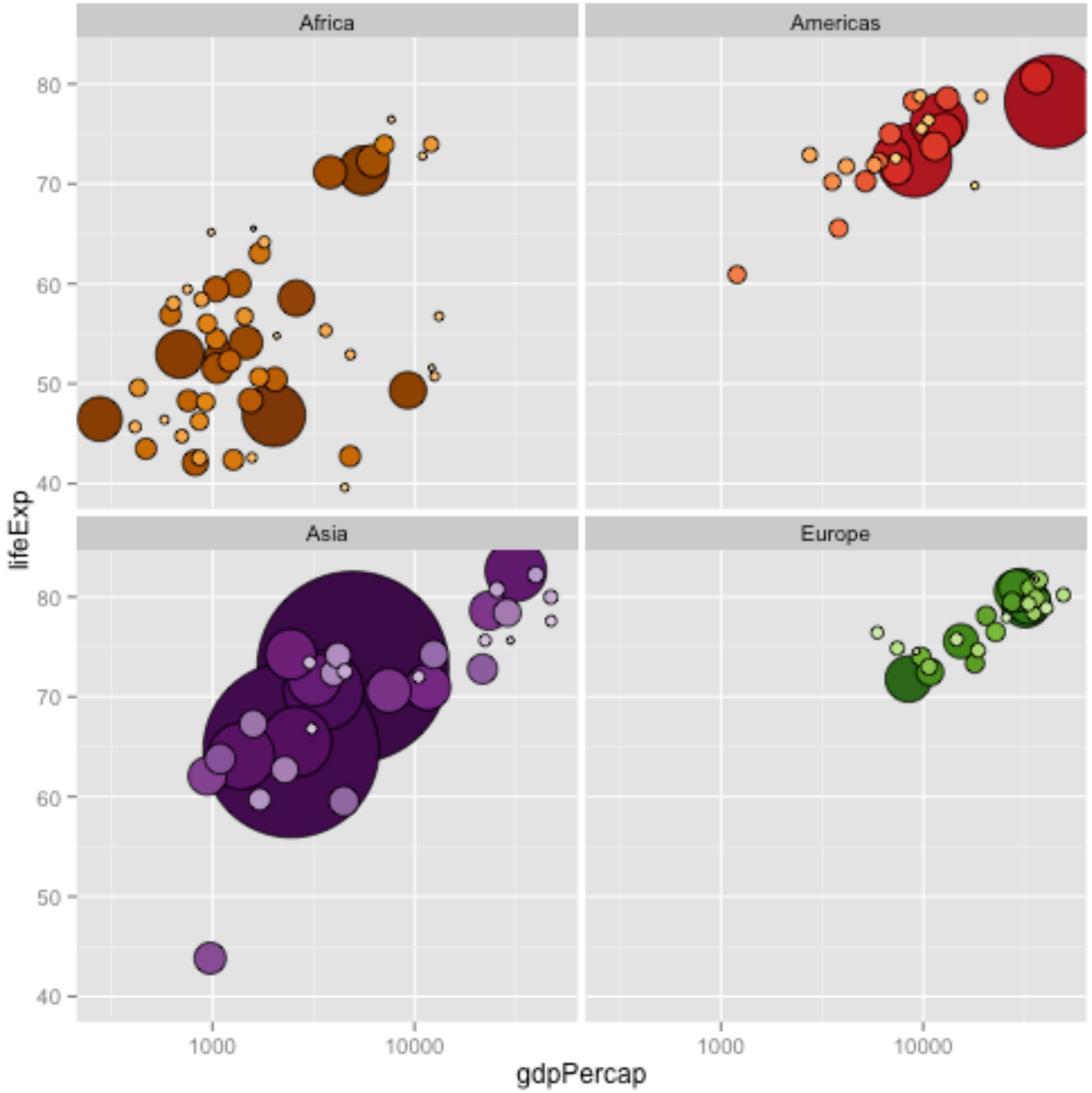
base

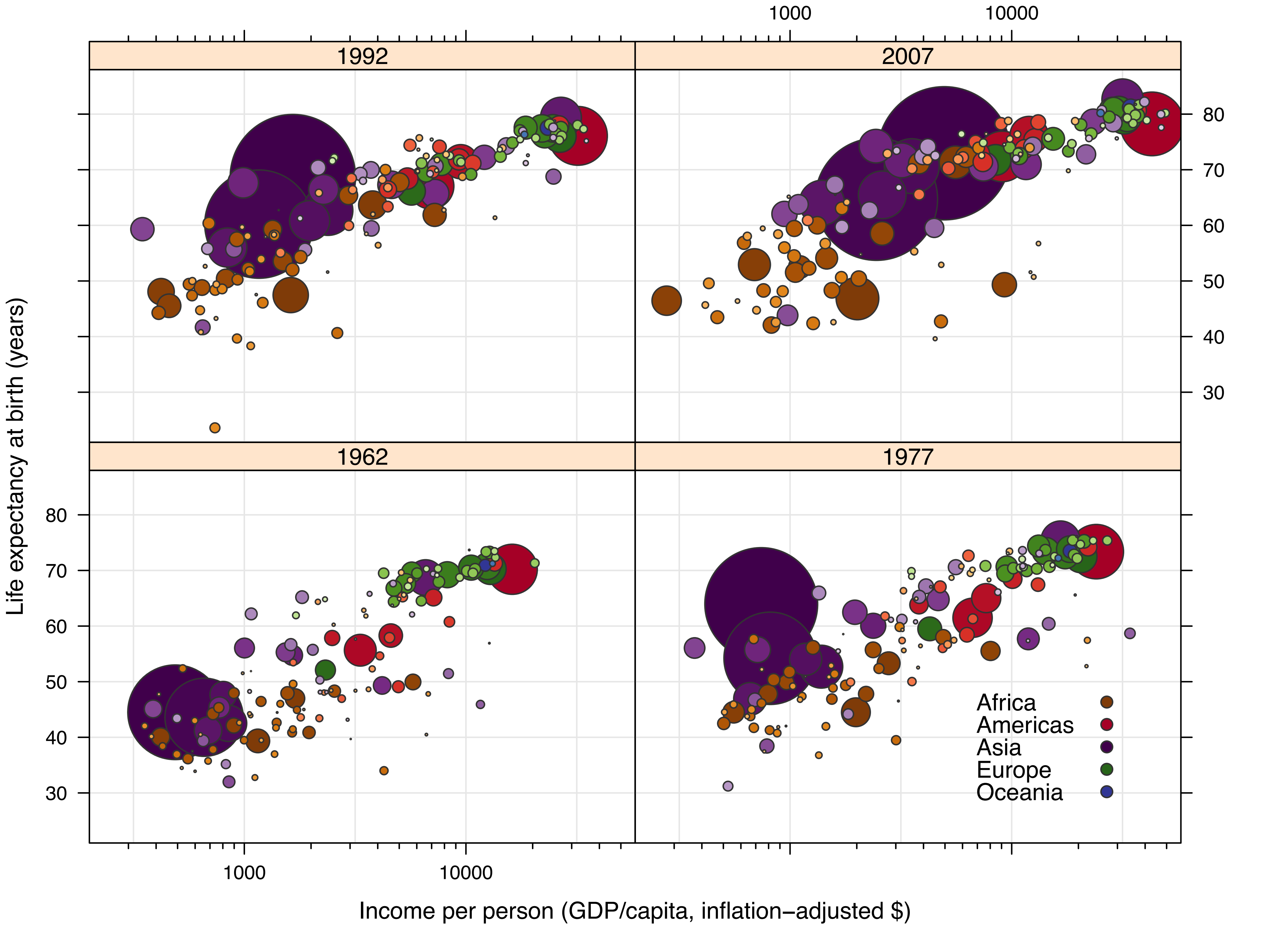
ggplot2

“facetting”

```
ggplot(...) + ... +
  facet_wrap(~ continent)
```







lattice

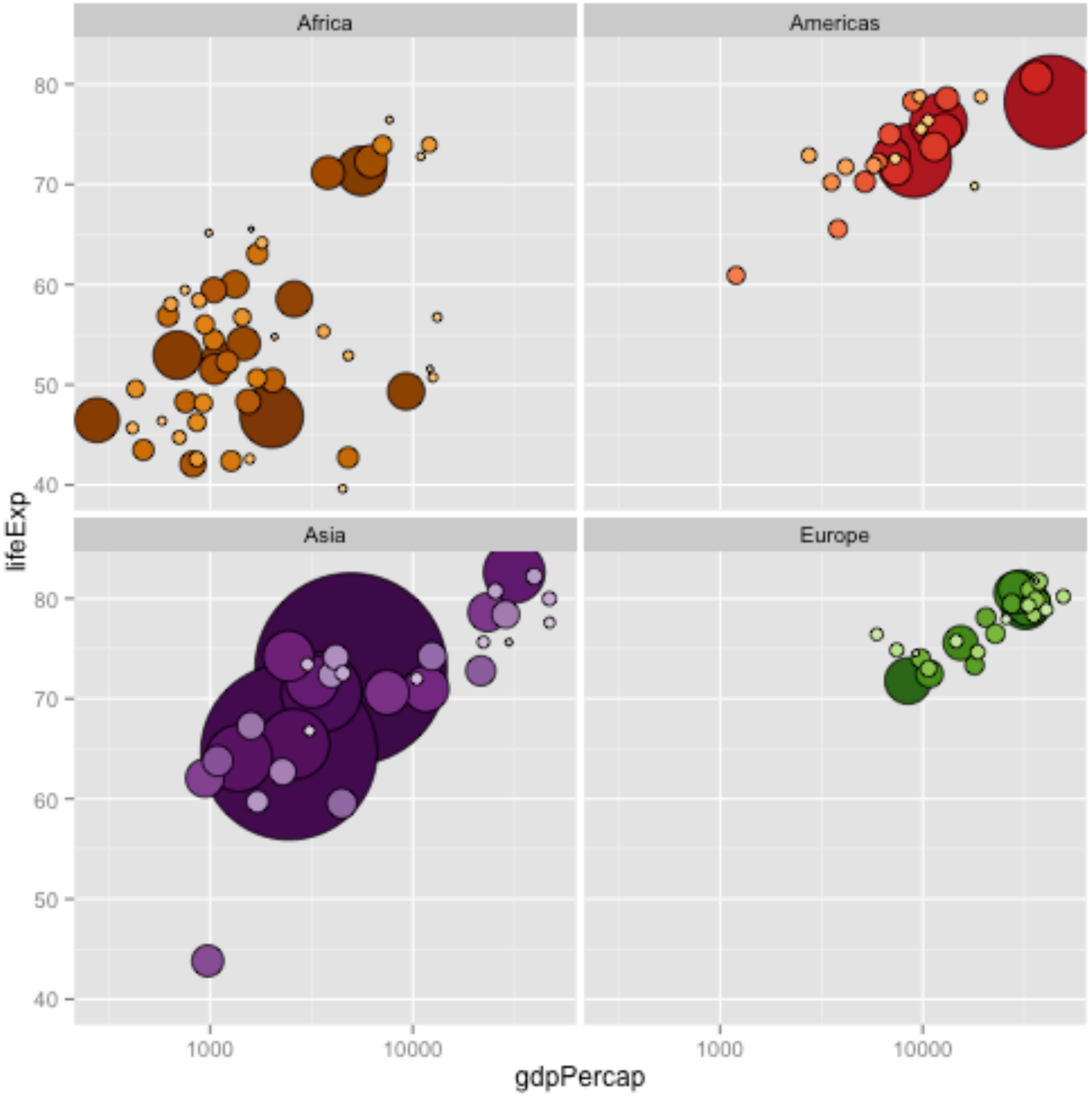
“groups and superposition”

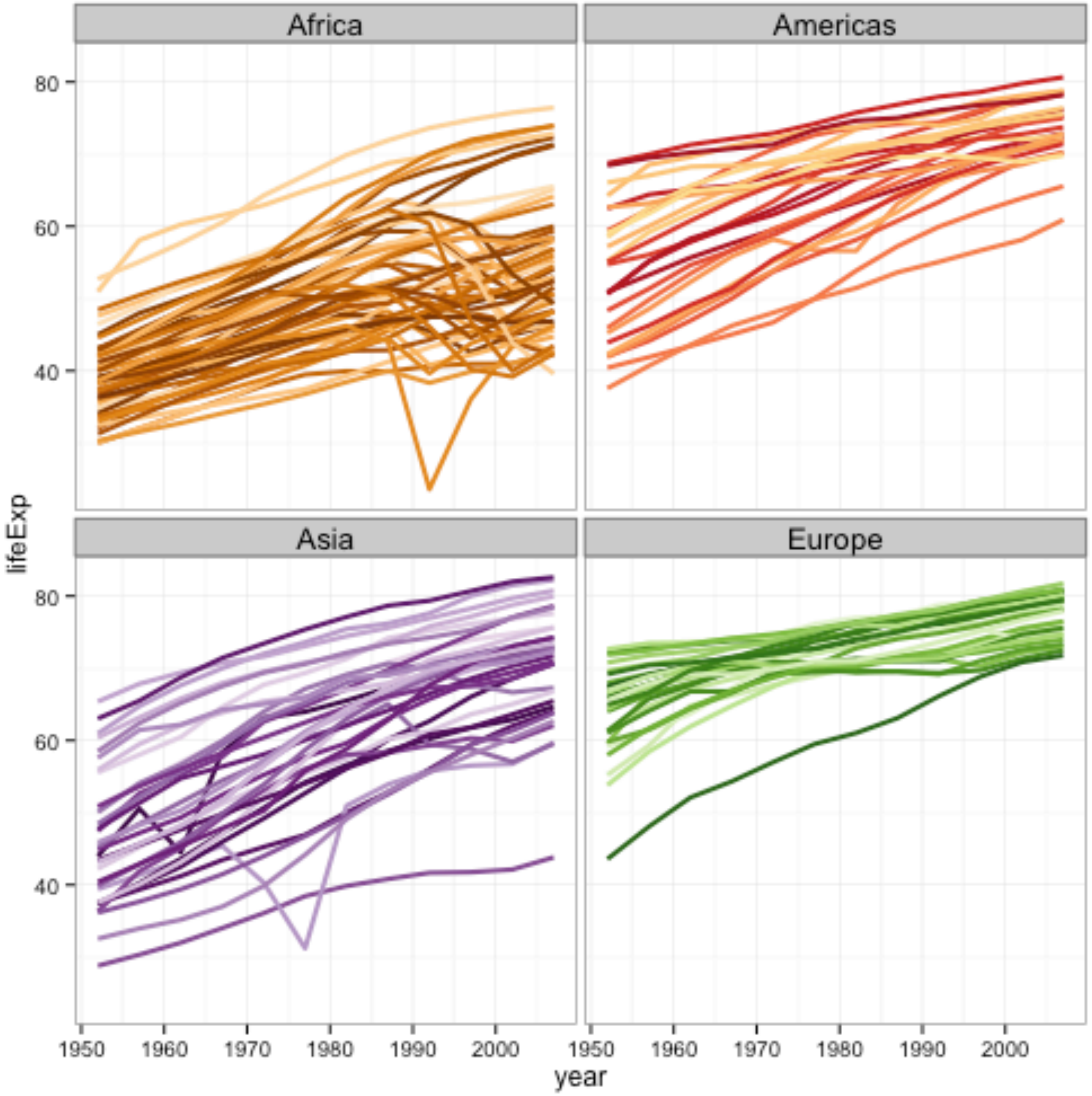
```
lifeExp ~ gdpPercap | year, group = country
```

ggplot2

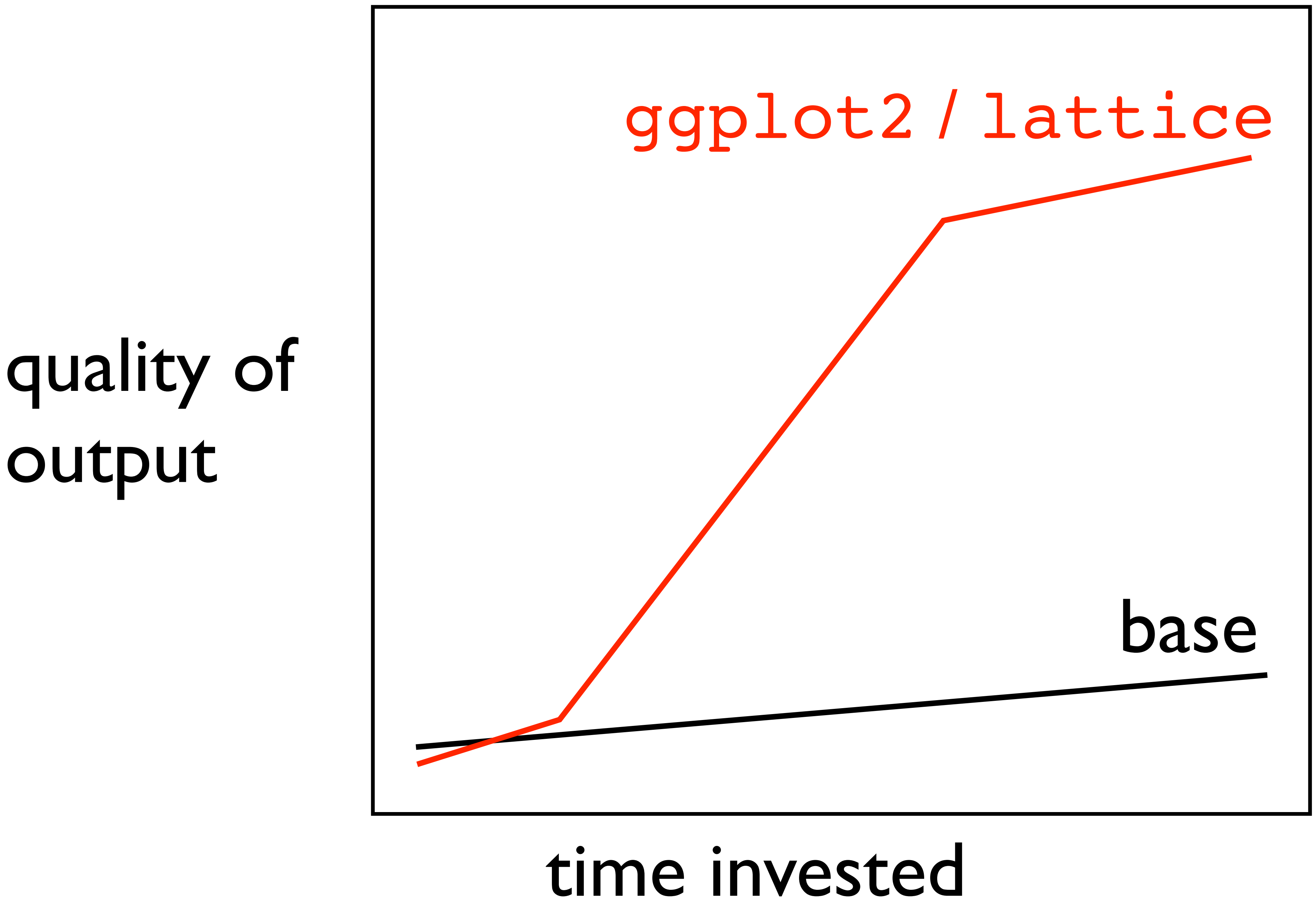
"aesthetic mapping"

```
ggplot(...) + ... +
  aes(fill = country)
```



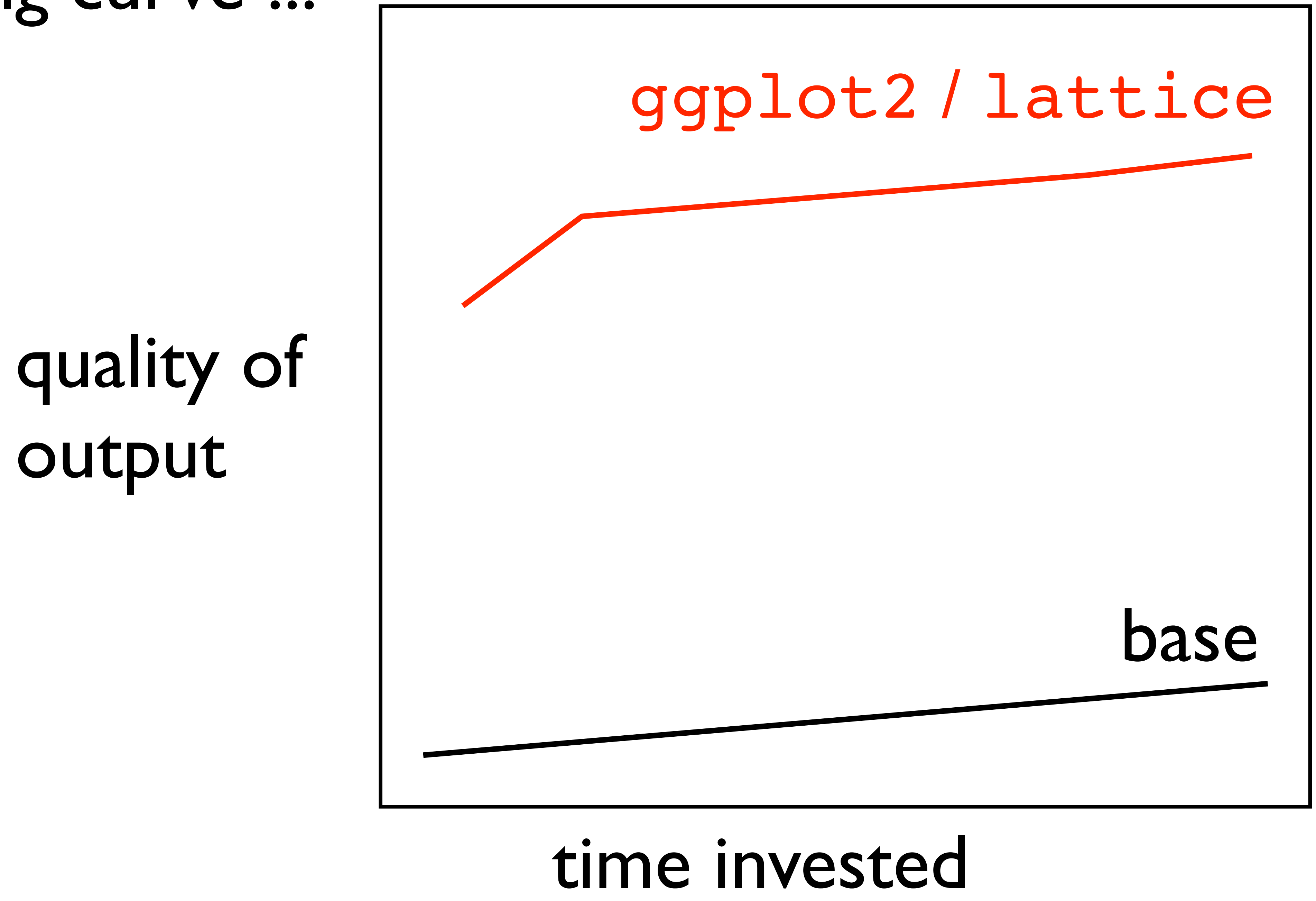


week one ....



* figure is totally fabricated but, I claim, still true

after you’ve climbed the steepest part of the learning curve ...



* figure is totally fabricated but, I claim, still true

Next few slides borrowed from here:

Data Visualization with R & ggplot2

Karthik Ram

September 2, 2013

Some housekeeping

Install some packages (make sure you also have recent copies of reshape2 and plyr)

```
install.packages("ggplot2", dependencies = TRUE)
```

Why ggplot2?

- Follows a grammar, just like any language.
- It defines basic components that make up a sentence. In this case, the grammar defines components in a plot.
- Grammar of graphics originally coined by Lee Wilkinson

Why ggplot2?

- Supports a continuum of expertise.
- Get started right away but with practice you can effortless build complex, publication quality figures.

Some terminology

- **ggplot** - The main function where you specify the dataset and variables to plot
- **geoms** - geometric objects
    - geom_point(), geom_bar(), geom_density(), geom_line(), geom_area()
- **aes** - aesthetics
    - shape, transparency (alpha), color, fill, linetype.
- **scales** Define how your data will be plotted
    - *continuous*, *discrete*, *log*

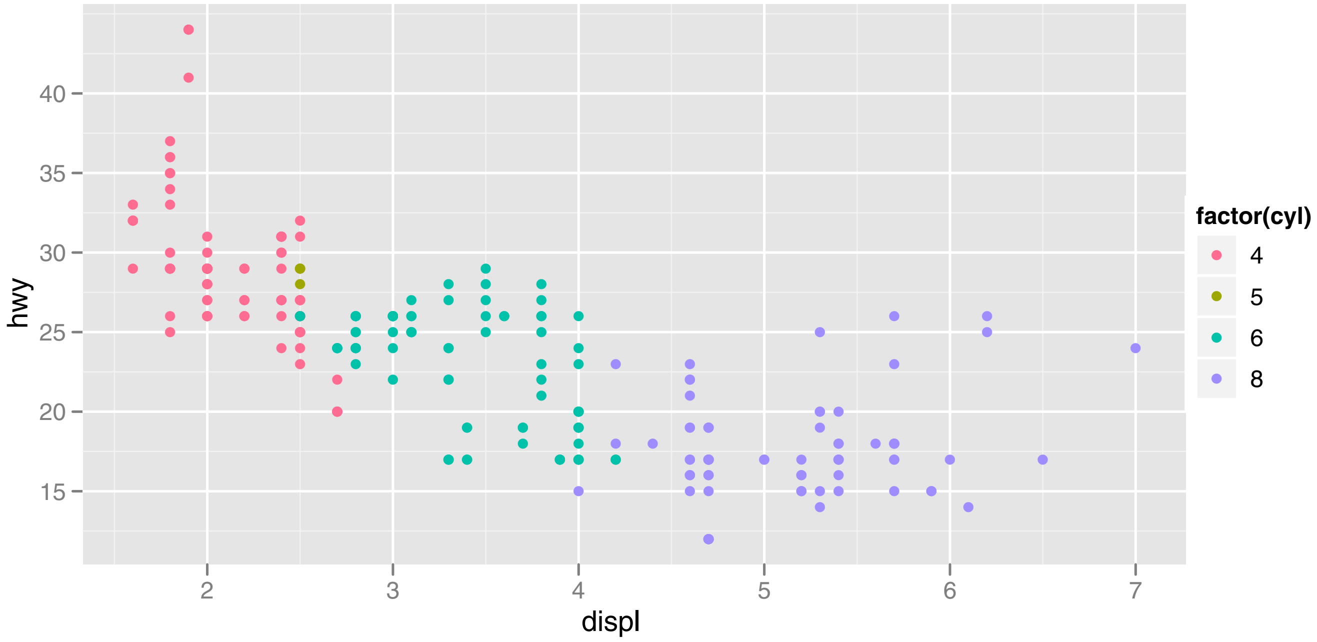


Fig. 3.1: A scatterplot of engine displacement in litres (displ) vs. average highway miles per gallon (hwy). Points are coloured according to number of cylinders. This plot summarises the most important factor governing fuel economy: engine size.

manufacturer	model	disp	year	cyl	cty	hwy	class
audi	a4	1.8	1999	4	18	29	compact
audi	a4	1.8	1999	4	21	29	compact
audi	a4	2.0	2008	4	20	31	compact
audi	a4	2.0	2008	4	21	30	compact
audi	a4	2.8	1999	6	16	26	compact
audi	a4	2.8	1999	6	18	26	compact
audi	a4	3.1	2008	6	18	27	compact
audi	a4 quattro	1.8	1999	4	18	26	compact
audi	a4 quattro	1.8	1999	4	16	25	compact
audi	a4 quattro	2.0	2008	4	20	28	compact

→

x	y	colour
1.8	29	4
1.8	29	4
2.0	31	4
2.0	30	4
2.8	26	6
2.8	26	6
3.1	27	6
1.8	26	4
1.8	25	4
2.0	28	4

→

x	y	colour	size	shape
0.037	0.531	#FF6C91	1	19
0.037	0.531	#FF6C91	1	19
0.074	0.594	#FF6C91	1	19
0.074	0.562	#FF6C91	1	19
0.222	0.438	#00C1A9	1	19
0.222	0.438	#00C1A9	1	19
0.278	0.469	#00C1A9	1	19
0.037	0.438	#FF6C91	1	19
0.037	0.406	#FF6C91	1	19
0.074	0.500	#FF6C91	1	19

mapping data
to aesthetics

scaling:
data units →
“computer” units

mapping data to aesthetics

```
ggplot(gDat,
       aes(x = gdpPercap, y = lifeExp))
```

```
ggplot(gDat,
       aes(x = gdpPercap, y = lifeExp,
           color = continent))
```

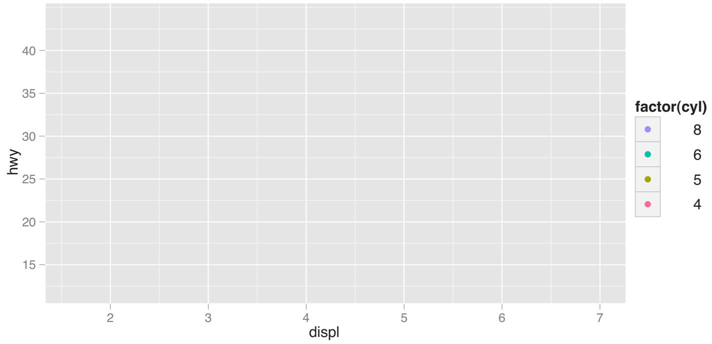


Fig. 3.5: Contributions from the scales, the axes and legend and grid lines, and the plot background. Contributions from the data, the point geom, have been removed.

the scales and coordinate system + plot annotations

+ “data, represented by the point geom”

“data, represented by the point geom”

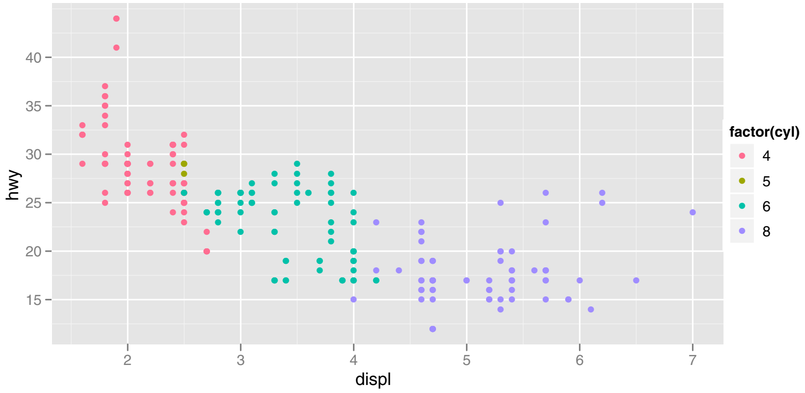


Fig. 3.1: A scatterplot of engine displacement in litres (displ) vs. average highway miles per gallon (hwy). Points are coloured according to number of cylinders. This plot summarises the most important factor governing fuel economy: engine size.

complete plot

facetting = multi-panel conditioning in lattice

layers = sort of like type = in lattice

the panels of the facets form a
2D grid and the layers extend
upwards in the 3rd dimension

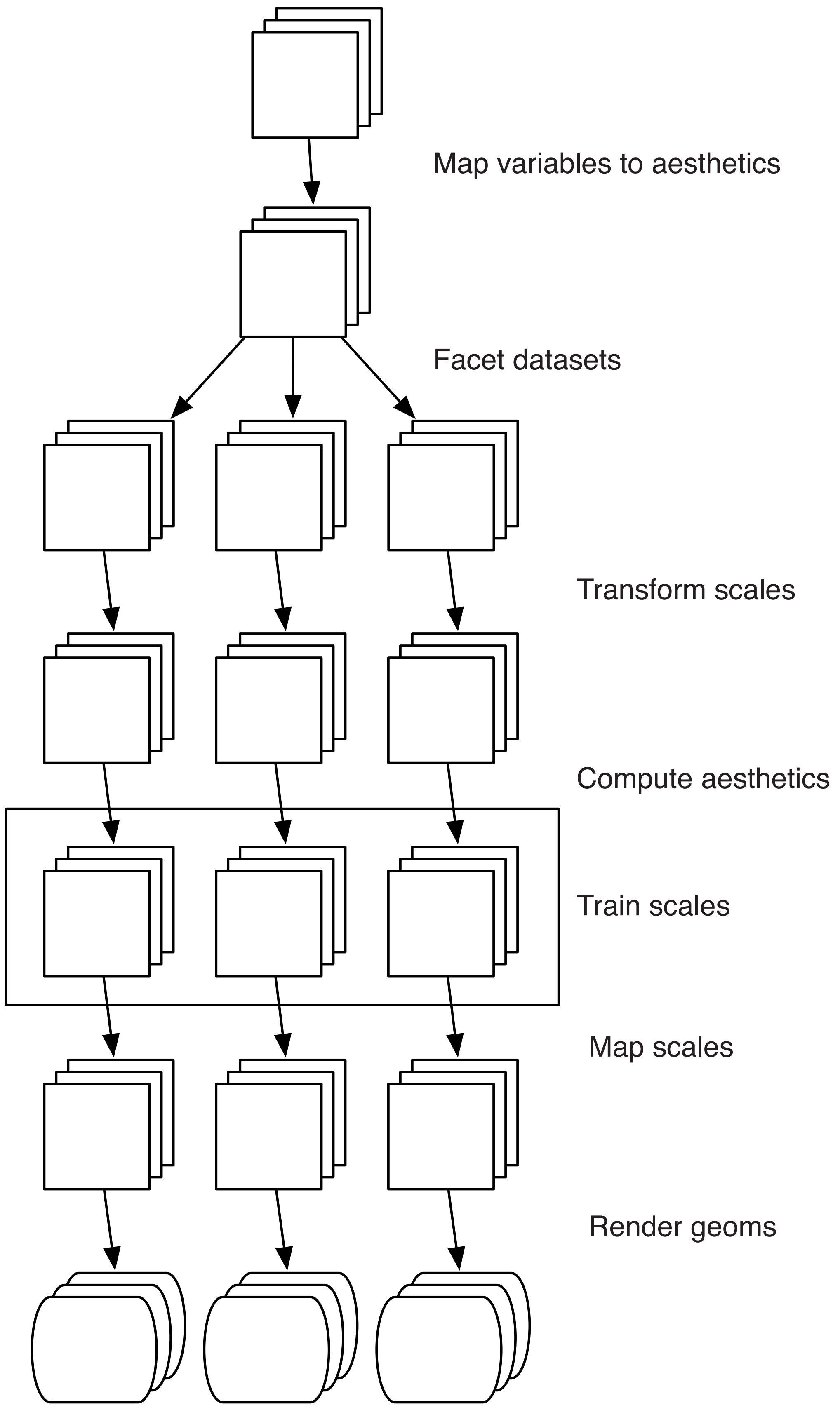


Fig. 3.7: Schematic description of the plot generation process. Each square represents a layer, and this schematic represents a plot with three layers and three panels. All steps work by transforming individual data frames, except for training scales which doesn't affect the data frame and operates across all datasets simultaneously.

one day (soon?) I will understand this

All together, the layered grammar defines a plot as the combination of:

- A default dataset and set of mappings from variables to aesthetics.
- One or more layers, each composed of a geometric object, a statistical transformation, and a position adjustment, and optionally, a dataset and aesthetic mappings.
- One scale for each aesthetic mapping.
- A coordinate system.
- The faceting specification.

3.6 Data structures

This grammar is encoded into R data structures in a fairly straightforward way. A plot object is a list with components `data`, `mapping` (the default aesthetic mappings), `layers`, `scales`, `coordinates` and `facet`. The plot object has one other component we haven't discussed yet: `options`. This is used to store the plot-specific theme options described in Chapter 8.

described in the next chapter. Once you have a plot object, there are a few things you can do with it:

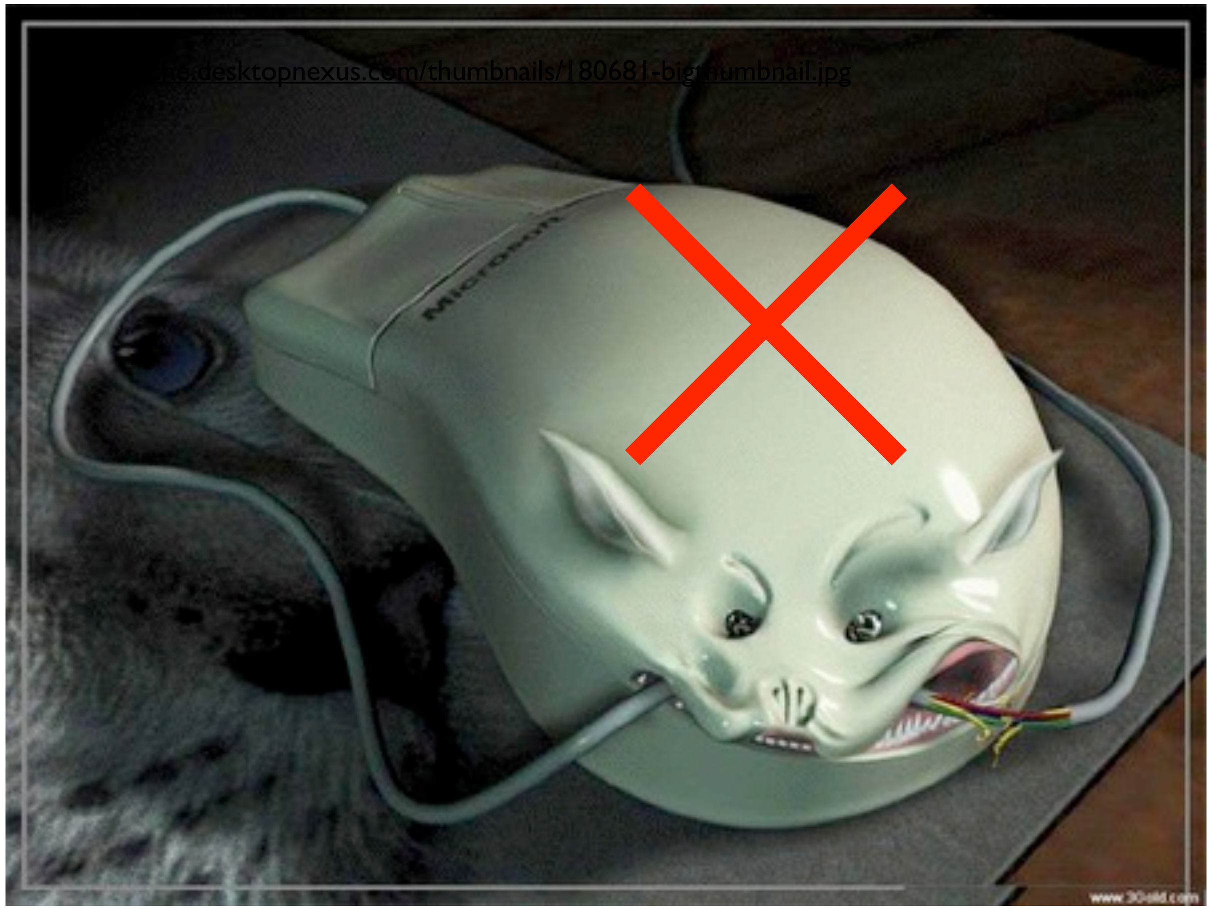
- Render it on screen, with `print()`. This happens automatically when running interactively, but inside a loop or function, you'll need to `print()` it yourself.
- Render it to disk, with `ggsave()`, described in Section 8.3.
- Briefly describe its structure with `summary()`.
- Save a cached copy of it to disk, with `save()`. This saves a complete copy of the plot object, so you can easily re-create that exact plot with `load()`. Note that data is stored inside the plot, so that if you change the data outside of the plot, and then redraw a saved plot, it will not be updated.

saving figures to file

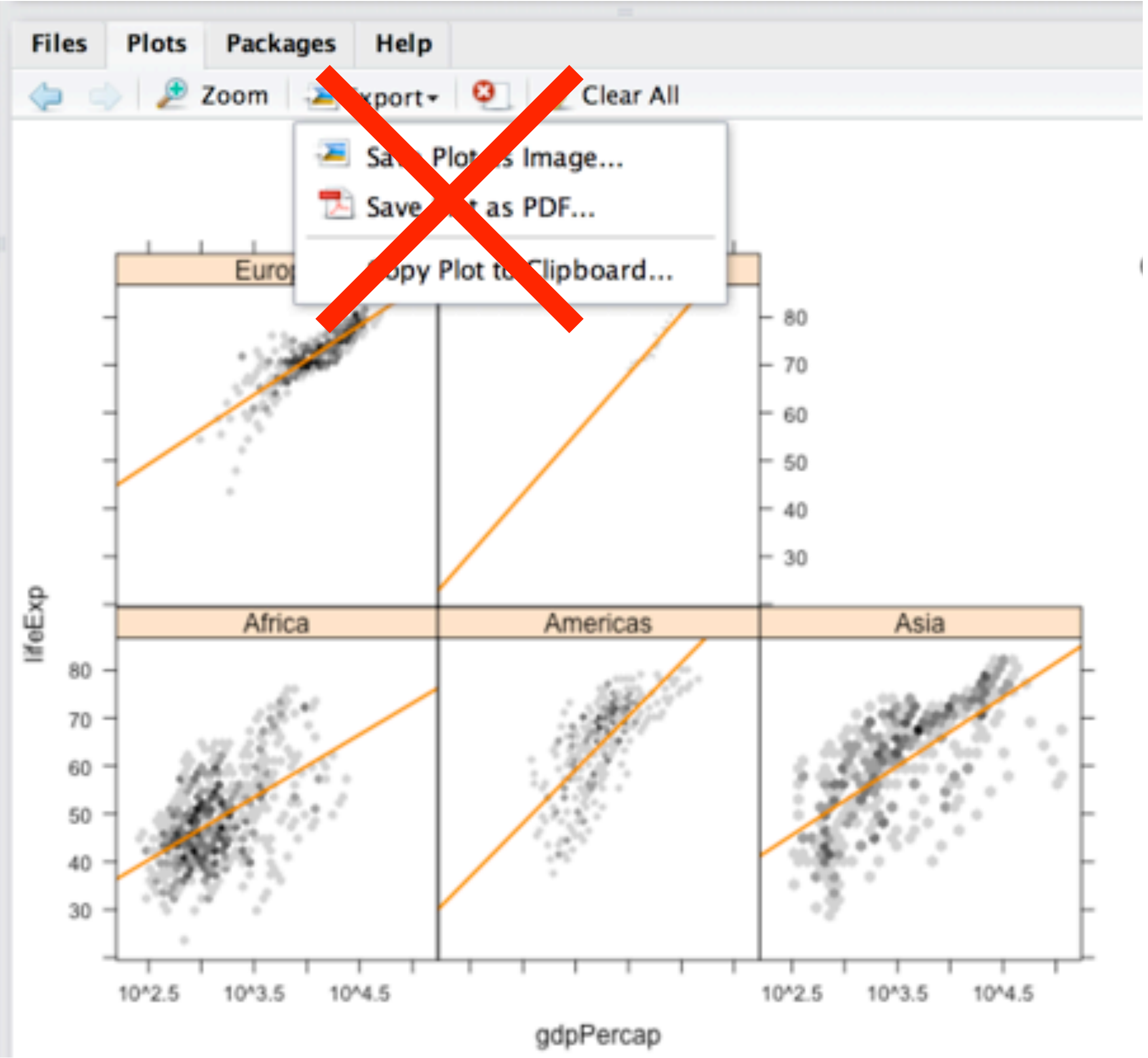
do not save figures mouse-y style

not self-documenting

not reproducible



he.desktopnexus.com/thumbnails/180681-bigthumbnail.jpg



most correct method:

```
pdf("awesome_figure.pdf")
plot(1:10)
dev.off()
```

`postscript(), svg(), png(), tiff(), ....`

fine for everyday use:

```
plot(1:10)
dev.print(pdf,"awesome_figure.pdf")
```

```
postscript(), svg(), png(), tiff(), ....
```

- If the plot is on your screen

```
ggsave("˜/path/to/figure/filename.png”)
```

- If your plot is assigned to an object

```
ggsave(plot1, file = "˜/path/to/figure/filename.png”)
```

- Specify a size

```
ggsave(file = "/path/to/figure/filename.png”, width = 6,
height =4)
```

- or any format (pdf, png, eps, svg, jpg)

```
ggsave(file = "/path/to/figure/filename.eps”)
ggsave(file = "/path/to/figure/filename.jpg”)
ggsave(file = "/path/to/figure/filename.pdf”)
```