An Introduction to R

Lattice Graphics

Deepayan Sarkar

Indian Statistical Institute, Delhi

October 2011

R graphics

- R has two largely independent graphics subsystems
  - Traditional graphics
    - Available in R from the beginning
    - Rich collection of tools
    - Not very flexible
  - Grid graphics
    - Relatively recent (2000)
    - Low-level tool, highly flexible
- Grid forms the basis of two high-level graphics systems:
  - lattice: based on Trellis graphics (Cleveland)
  - ggplot2: inspired by *"Grammar of Graphics"* (Wilkinson)

The lattice package

- Trellis graphics for R (originally developed in S)
- Powerful high-level data visualization system
- Provides common statistical graphics with conditioning
  - Emphasis on multivariate data
  - Sufficient for typical graphics needs
  - Flexible enough to handle most nonstandard requirements
- Traditional user interface:
  - Collection of high-level functions: `xyplot`, `dotplot`, etc.
  - Interface based on formula and data source

High-level functions in lattice

Function	Default Display
`histogram()`	Histogram
`densityplot()`	Kernel Density Plot
`qqmath()`	Theoretical Quantile Plot
`qq()`	Two-sample Quantile Plot
`stripplot()`	Stripchart (Comparative 1-D Scatter Plots)
`bwplot()`	Comparative Box-and-Whisker Plots
`barchart()`	Bar Plot
`dotplot()`	Cleveland Dot Plot
`xyplot()`	Scatter Plot
`splom()`	Scatter-Plot Matrix
`contourplot()`	Contour Plot of Surfaces
`levelplot()`	False Color Level Plot of Surfaces
`wireframe()`	Three-dimensional Perspective Plot of Surfac
`cloud()`	Three-dimensional Scatter Plot
`parallel()`	Parallel Coordinates Plot

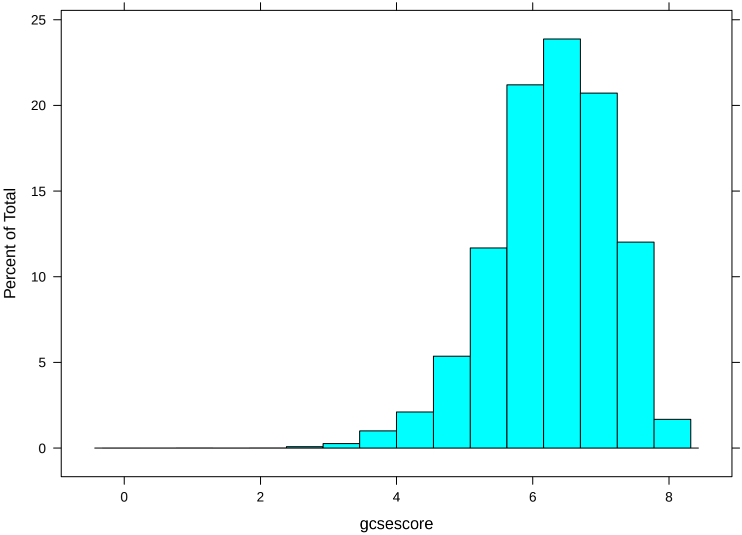
The `Chem97` dataset

1997 A-level Chemistry examination in Britain

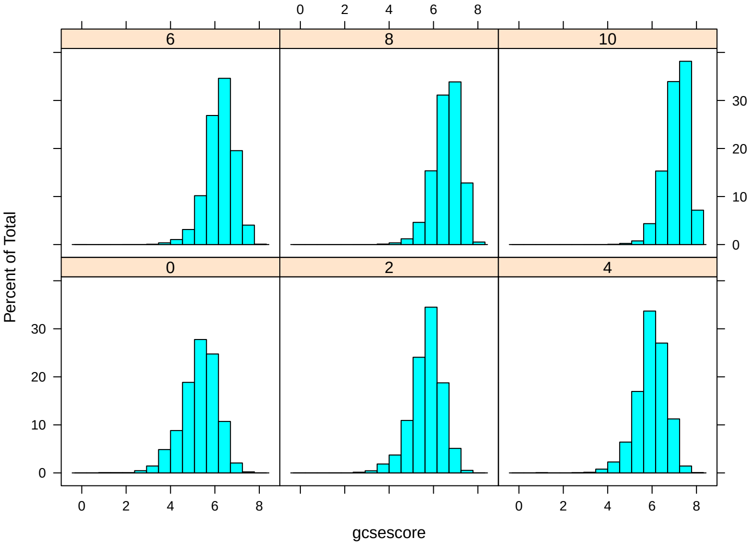
```
> data(Chem97, package = "mlmRev")
> head(Chem97[c("score", "gender", "gcsescore")])
```

```
  score gender gcsescore
1     4      F     6.625
2    10      F     7.625
3    10      F     7.250
4    10      F     7.500
5     8      F     6.444
6    10      F     7.750
```

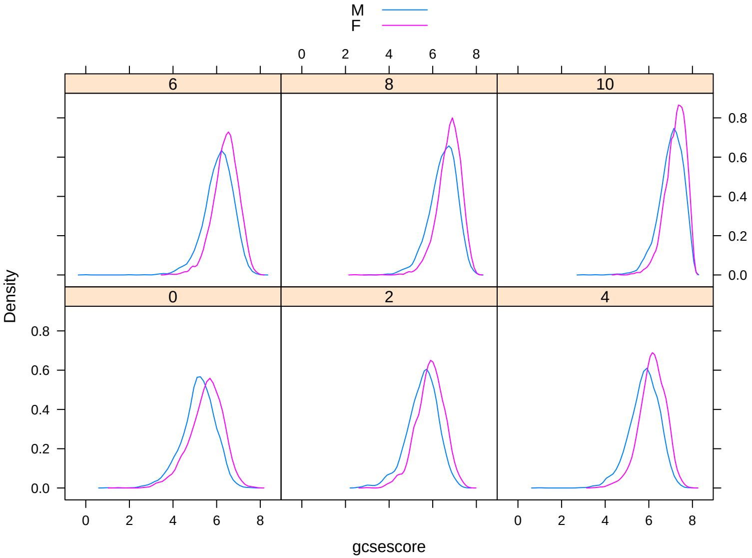
```
> histogram(~ gcsescore, data = Chem97)
```



```
> histogram(~ gcsescore | factor(score), data = Chem97)
```



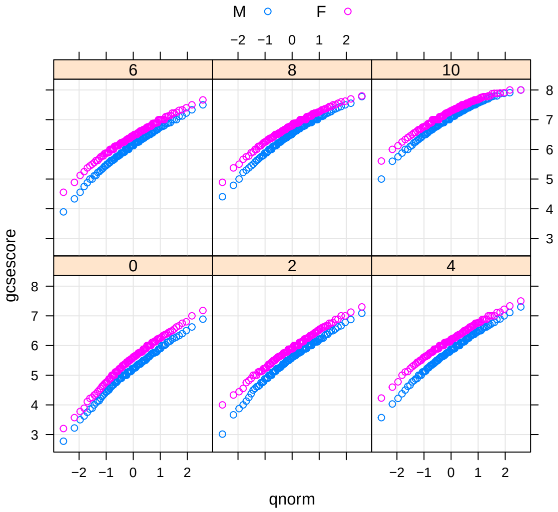
```
> densityplot(~ gcsescore | factor(score), Chem97,
              plot.points = FALSE,
              groups = gender, auto.key = TRUE)
```



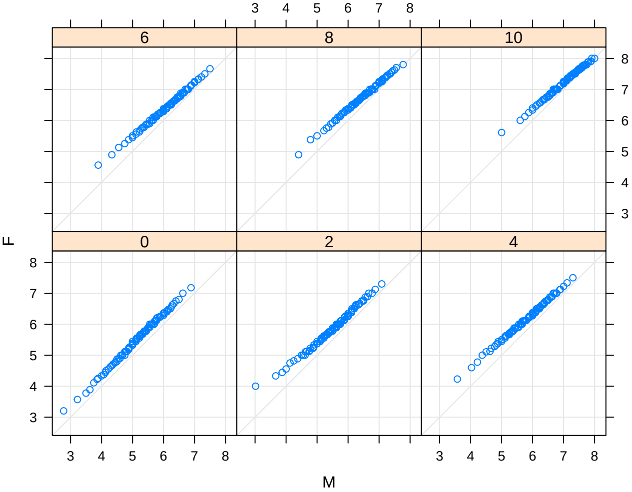
Trellis Philosophy: Part I

- Display specified in terms of
  - Type of display (histogram, densityplot, etc.)
  - Variables with specific roles
- Typical roles for variables
  - Primary variables: used for the main graphical display
  - Conditioning variables: used to divide into subgroups and juxtapose (multipanel conditioning)
  - Grouping variable: divide into subgroups and superpose
- Primary interface: high-level functions
  - Each function corresponds to a display type
  - Specification of roles depends on display type
    - Usually specified through the formula and the `groups` argument

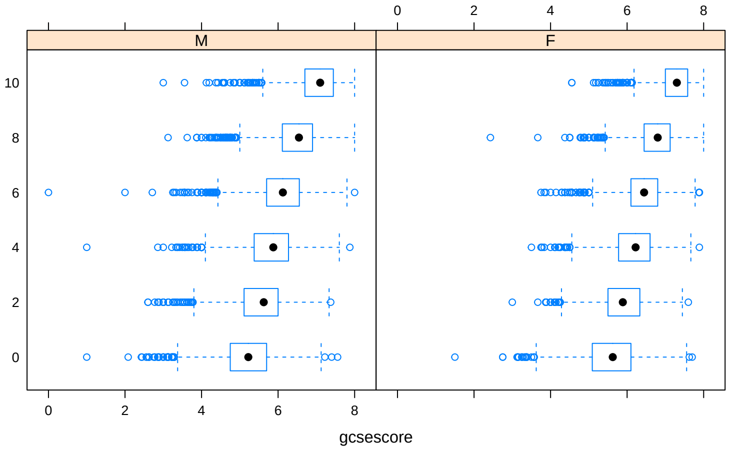
```
> qqmath(~ gcsescore | factor(score), Chem97,
         groups = gender, auto.key = list(columns = 2),
         f.value = ppoints(100),
         type = c("p", "g"), aspect = "xy")
```



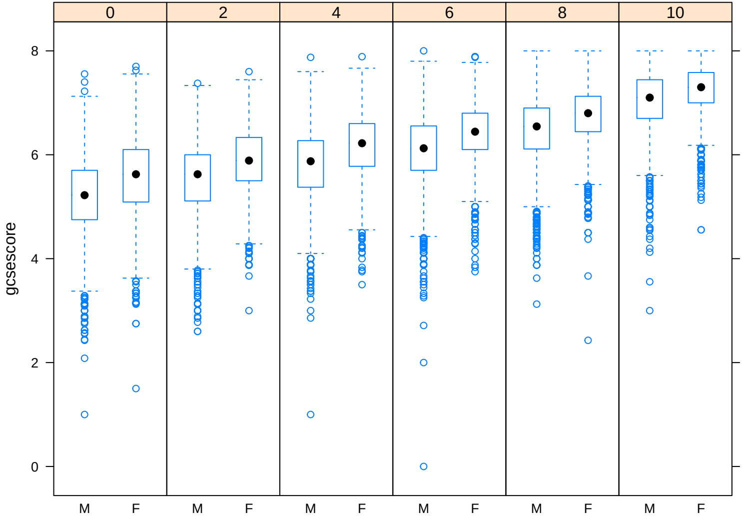
```
> qq(gender ~ gcsescore | factor(score), Chem97,
     f.value = ppoints(100), type = c("p", "g"),
     aspect = 1)
```



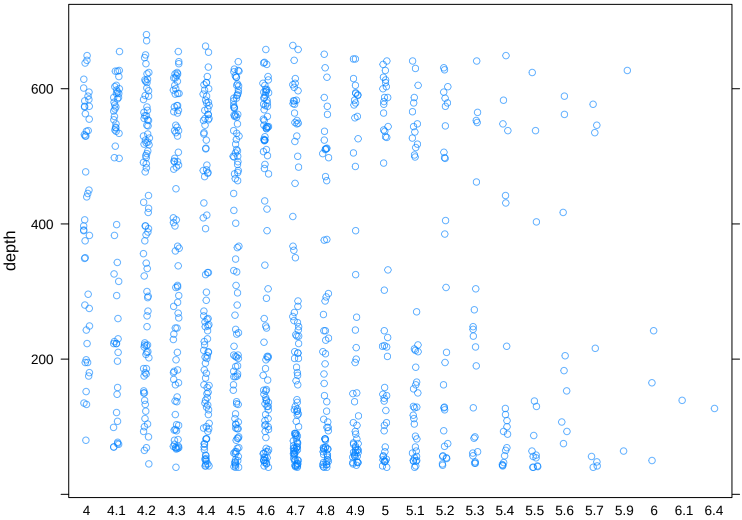
```
> bwplot(factor(score) ~ gcsescore | gender, Chem97)
```



```
> bwplot(gcsescore ~ gender | factor(score), Chem97,
         layout = c(6, 1))
```



```
> stripplot(depth ~ factor(mag), data = quakes,
            jitter.data = TRUE, alpha = 0.6)
```

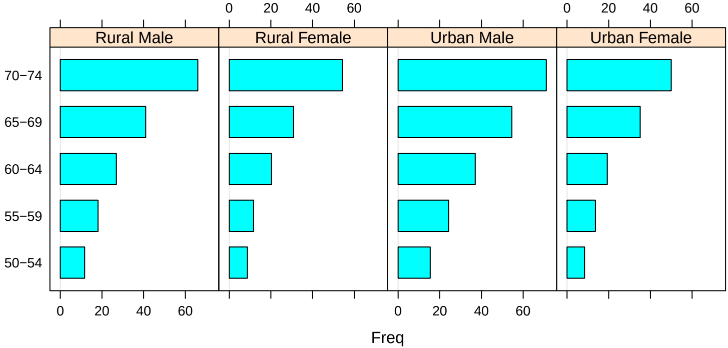


The `VADeaths` dataset

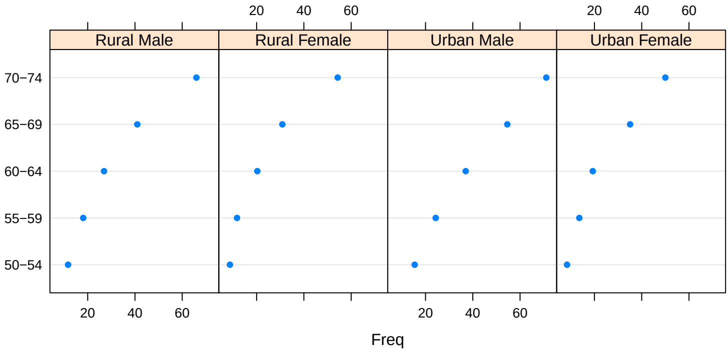
- Death rates in Virginia, 1941, among different population subgroups

```
> VADeaths

      Rural Male Rural Female Urban Male
50-54       11.7          8.7       15.4
55-59       18.1         11.7       24.3
60-64       26.9         20.3       37.0
65-69       41.0         30.9       54.6
70-74       66.0         54.3       71.1
      Urban Female
50-54          8.4
55-59         13.6
60-64         19.3
65-69         35.1
70-74         50.0
```

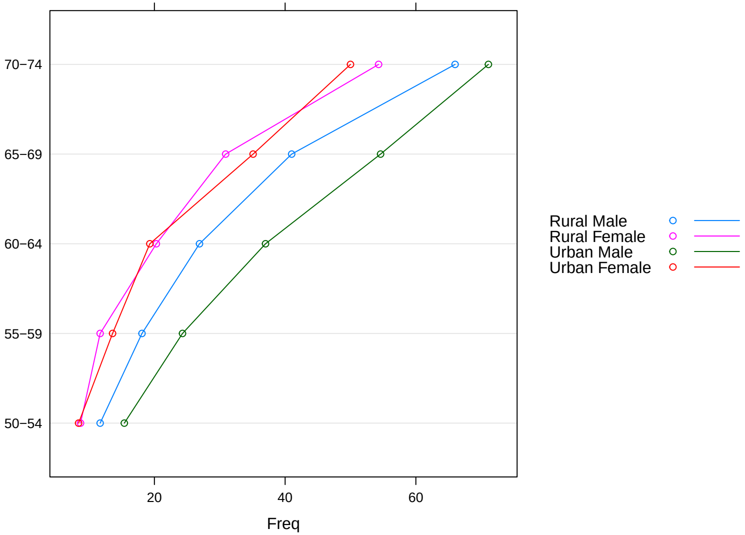
```
> barchart(VADeaths, groups = FALSE, layout = c(4, 1))
```



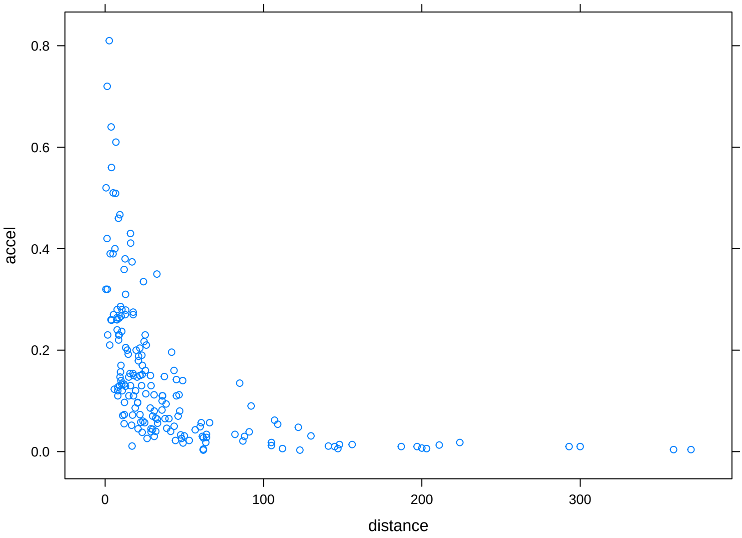
```
> dotplot(VADeaths, groups = FALSE, layout = c(4, 1))
```



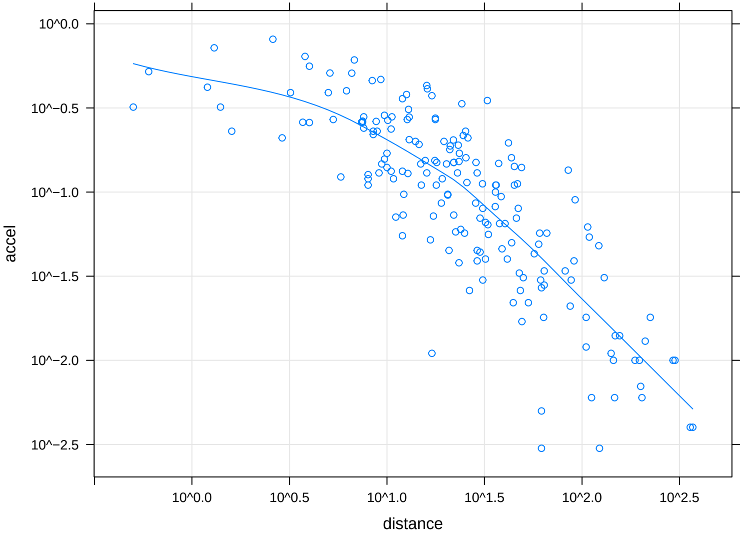
```
> dotplot(VADeaths, type = "o",
          auto.key = list(points = TRUE, lines = TRUE,
                          space = "right"))
```



```
> data(Earthquake, package = "nlme")
> xyplot(accel ~ distance, data = Earthquake)
```

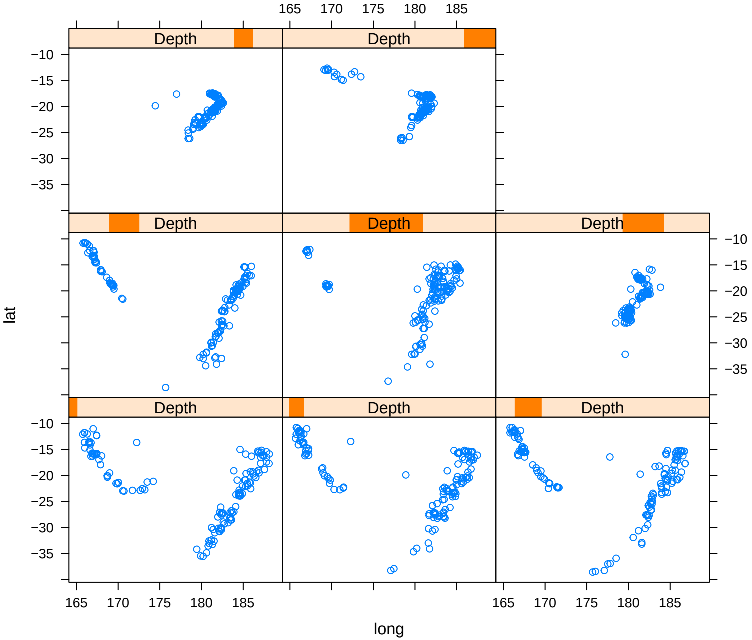


```r
> xyplot(accel ~ distance, data = Earthquake,
         scales = list(log = TRUE),
         type = c("p", "g", "smooth"))
```

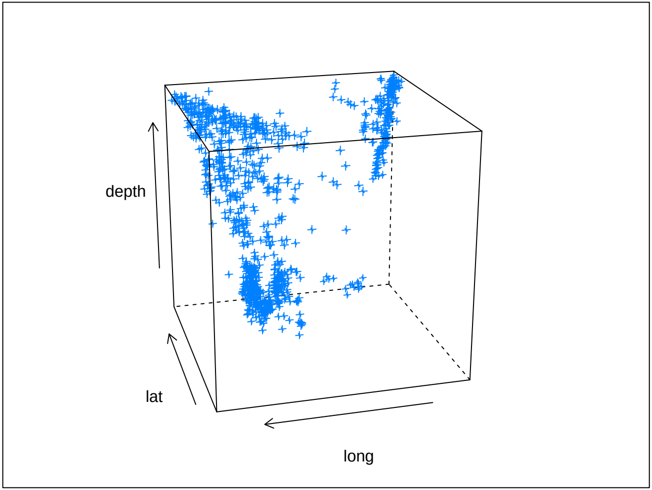


```
> Depth <- equal.count(quakes$depth, number = 8,
                       overlap = 0.1)
> summary(Depth)

Intervals:
    min   max count
1  39.5  63.5   138
2  60.5 102.5   138
3  97.5 175.5   138
4 161.5 249.5   142
5 242.5 460.5   138
6 421.5 543.5   137
7 537.5 590.5   140
8 586.5 680.5   137

Overlap between adjacent intervals:
[1] 16 14 19 15 14 15 15
```

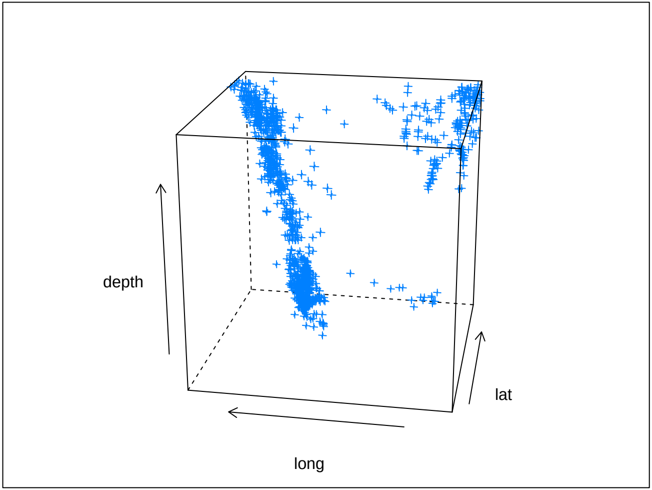
```
> xyplot(lat ~ long | Depth, data = quakes)
```



```
> cloud(depth ~ lat * long, data = quakes,
        zlim = rev(range(quakes$depth)),
        screen = list(z = 105, x = -70),
        panel.aspect = 0.75)
```



```
> cloud(depth ~ lat * long, data = quakes,
        zlim = rev(range(quakes$depth)),
        screen = list(z = 80, x = -70),
        panel.aspect = 0.75)
```



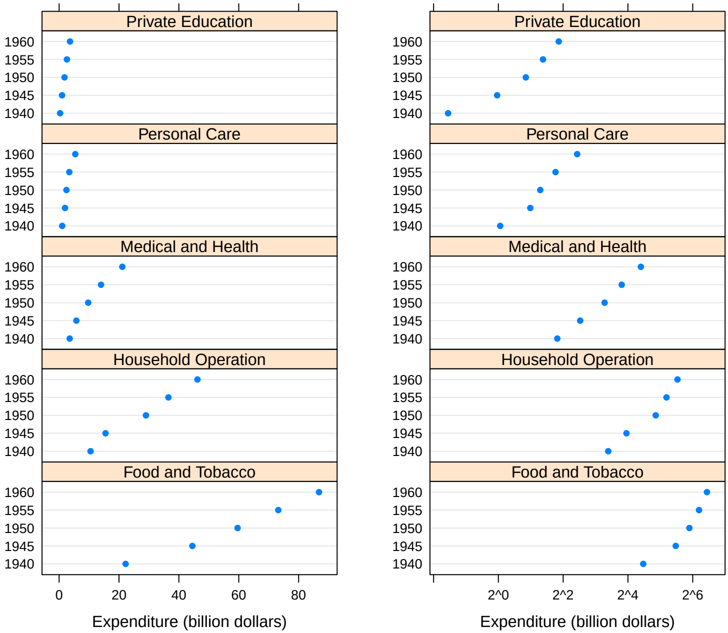
More high-level functions

- More high-level functions in lattice
  - Won't discuss, but examples in manual page
- Other Trellis high-level functions can be defined in other packages, e.g.,
  - `ecdfplot()`, `mapplot()` in the `latticeExtra` package
  - `hexbinplot()` in the `hexbin` package

The “trellis” object model

- One important feature of lattice:
  - High-level functions do not actually plot anything
  - They return an object of class “trellis”
  - Display created when such objects are `print()`-ed or `plot()`-ed
- Usually not noticed because of automatic printing rule
- Can be used to arrange multiple plots
- Other uses as well

```
> dp.uspe <-
      dotplot(t(USPersonalExpenditure),
              groups = FALSE, layout = c(1, 5),
              xlab = "Expenditure (billion dollars)")
> dp.uspe.log <-
      dotplot(t(USPersonalExpenditure),
              groups = FALSE, layout = c(1, 5),
              scales = list(x = list(log = 2)),
              xlab = "Expenditure (billion dollars)")
> plot(dp.uspe,     split = c(1, 1, 2, 1))
> plot(dp.uspe.log, split = c(2, 1, 2, 1),
       newpage = FALSE)
```



Private Education
Personal Care
Medical and Health
Household Operation
Food and Tobacco
1960
1955
1950
1945
1940
0
20
40
60
80
2^0
2^2
2^4
2^6
Expenditure (billion dollars)

Trellis Philosophy: Part I

- Display specified in terms of
  - Type of display (histogram, densityplot, etc.)
  - Variables with specific roles
- Typical roles for variables
  - Primary variables: used for the main graphical display
  - Conditioning variables: used to divide into subgroups and juxtapose (multipanel conditioning)
  - Grouping variable: divide into subgroups and superpose
- Primary interface: high-level functions
  - Each function corresponds to a display type
  - Specification of roles depends on display type
    - Usually specified through the formula and the `groups` argument

Trellis Philosophy: Part II

- Design goals:
  - Enable effective graphics by encouraging good graphical practice (e.g., Cleveland, 1985)
  - Remove the burden from the user as much as possible by building in good defaults into software
- Some obvious examples:
  - Use as much of the available space as possible
  - Encourage direct comparsion by superposition (grouping)
  - Enable comparison when juxtaposing (conditioning):
    - use common axes
    - add common reference objects (such as grids)
- Inevitable departure from traditional R graphics paradigms

Trellis Philosophy: Part III

- Any serious graphics system must also be flexible
- lattice tries to balance flexibility and ease of use using the following model:
    - A display is made up of various elements
    - Coordinated defaults provide meaningful results, but
    - Each element can be controlled independently
    - The main elements are:
        - the primary (panel) display
        - axis annotation
        - strip annotation (describing the conditioning process)
        - legends (typically describing the grouping process)

- The full system would take too long to describe
- Online documentation has details; start with `?Lattice`