Factors in R

- vector object used to specify a discrete classification (grouping) of the components of other vectors of the same length
- default way of storing character data in data frames
- used in formulas in R
- used in `tapply` function

Example

```r
> help(state, package="datasets")
state package:datasets R Documentation
US State Facts and Figures

Description:
Data sets related to the 50 states of the United States of America.

Usage:
state.abb
state.area
state.center
state.division
state.name
state.region
state.x77

Details:
R currently contains the following "state" data sets. Note that all data are arranged according to alphabetical order of the state names.

'state.abb': character vector of 2-letter abbreviations for the state names.

'state.area': numeric vector of state areas (in square miles).

'state.center': list with components named 'x' and 'y' giving the approximate geographic center of each state in negative longitude and latitude. Alaska and Hawaii are placed just off the West Coast.

'state.division': factor giving state divisions (New England, Middle Atlantic, South Atlantic, East South Central, West South Central, East North Central, West North Central, Mountain, and Pacific).

'state.name': character vector giving the full state names.

'state.region': factor giving the region (Northeast, South, North Central, West) that each state belongs to.

'state.x77': matrix with 50 rows and 8 columns giving the following statistics in the respective columns.

'Population': population estimate as of July 1, 1975

'Income': per capita income (1974)

'Area': land area in square miles

Source:
U.S. Department of Commerce, Bureau of the Census (1977) _County and City Data Book_.

References:
Functions operating on factors

```r
> is.factor(statedf[,"div"])
[1] TRUE

> levels(statedf[,"div"])
[1] "New England"  "Middle Atlantic" "South Atlantic"
[4] "East South Central"  "West South Central" "East North Central"
[7] "West North Central"  "Mountain" "Pacific"
```

Using factors in formulas for plotting and model fitting

```r
> boxplot( Population ~ div, data = statedf )
> boxplot( Population ~ div, data = statedf, pars=list(cex.axis=0.75))
> dev.copy2eps( file="~/166/lects2005/boxplotstatepop.ps", horizontal=T)
```

<table>
<thead>
<tr>
<th>State</th>
<th>Abbreviation</th>
<th>Division</th>
<th>Region</th>
<th>Population</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alabama</td>
<td>AL</td>
<td>East South Central</td>
<td>South</td>
<td>3615</td>
<td>50708</td>
</tr>
<tr>
<td>Alaska</td>
<td>AK</td>
<td>Pacific</td>
<td>West</td>
<td>365</td>
<td>566432</td>
</tr>
<tr>
<td>Arizona</td>
<td>AZ</td>
<td>Mountain</td>
<td>West</td>
<td>2212</td>
<td>113417</td>
</tr>
<tr>
<td>Arkansas</td>
<td>AR</td>
<td>West South Central</td>
<td>South</td>
<td>2110</td>
<td>51945</td>
</tr>
<tr>
<td>California</td>
<td>CA</td>
<td>Pacific</td>
<td>West</td>
<td>21198</td>
<td>156361</td>
</tr>
<tr>
<td>Colorado</td>
<td>CO</td>
<td>Mountain</td>
<td>West</td>
<td>2541</td>
<td>103766</td>
</tr>
<tr>
<td>Connecticut</td>
<td>CT</td>
<td>New England</td>
<td>Northeast</td>
<td>5100</td>
<td>4862</td>
</tr>
<tr>
<td>Delaware</td>
<td>DE</td>
<td>South Atlantic</td>
<td>South</td>
<td>579</td>
<td>1982</td>
</tr>
<tr>
<td>Florida</td>
<td>FL</td>
<td>South Atlantic</td>
<td>South</td>
<td>8277</td>
<td>54090</td>
</tr>
<tr>
<td>Georgia</td>
<td>GA</td>
<td>South Atlantic</td>
<td>South</td>
<td>4931</td>
<td>58073</td>
</tr>
<tr>
<td>Hawaii</td>
<td>HI</td>
<td>Pacific</td>
<td>West</td>
<td>868</td>
<td>6425</td>
</tr>
<tr>
<td>Idaho</td>
<td>ID</td>
<td>Mountain</td>
<td>West</td>
<td>813</td>
<td>82677</td>
</tr>
<tr>
<td>Illinois</td>
<td>IL</td>
<td>East North Central</td>
<td>North Central</td>
<td>11197</td>
<td>55748</td>
</tr>
<tr>
<td>Indiana</td>
<td>IN</td>
<td>East North Central</td>
<td>North Central</td>
<td>5313</td>
<td>36097</td>
</tr>
<tr>
<td>Iowa</td>
<td>IA</td>
<td>West North Central</td>
<td>North Central</td>
<td>2861</td>
<td>55941</td>
</tr>
</tbody>
</table>
Graphics in R

Plotting functions in base R:

- High-level plotting functions create a new plot on the graphics device, possibly with axes, labels, titles and so on.
- Low-level plotting functions add more information to an existing plot, such as extra points, lines and labels.
- Interactive graphics functions allow you interactively add information to, or extract information from, an existing plot, using a pointing device such as a mouse.

Example of high-level function: Plot

plot is a generic plotting function whose behavior is determined by the class of the object(s) to which it is applied.

- argument is factor: bar graph of counts of each level
  > plot( statedf["div"], cex.axis=0.75,
         + main = "Number of States per Division" )
- arguments are two numeric vectors: scatterplot with first vector on x-axis
  > plot( statedf["Area"], statedf["Population"],
         + xlab = "Area in Square Miles", ylab = "Population in thousands")
- argument is a data frame: scatterplot matrix
  > plot( statedf )
- plotting one object against each object in an expression
  - object to left of "~" will be on y-axis
  > par(mfrow=c(1,2) )
  > plot( Population ~ Area + reg, data = statedf )
**Low-level plotting functions**

- add extra information (such as points, lines or text) to the current plot.

- `points(x,y)`
- `lines(x,y)`
- `text(x,y,labels,...)`

```r
> attach(statedf)
> plot( Area, Population, type="n" )
> text( Area, Population, abb)
> legend(x, y, legend, ...)
```

**Interactive graphic functions**

- `locator(n, type)`
  - Waits for the user to select `n` locations on the current plot using the left mouse button.
  - returns the locations of the points selected as a list with two components `x` and `y`.

```r
> text( locator(2), "Outlier")
```