Graphics in R

Part 1: High-level graphics functions

We'll be working in this section with many of R's built-in data sets. To see a list of them, just type

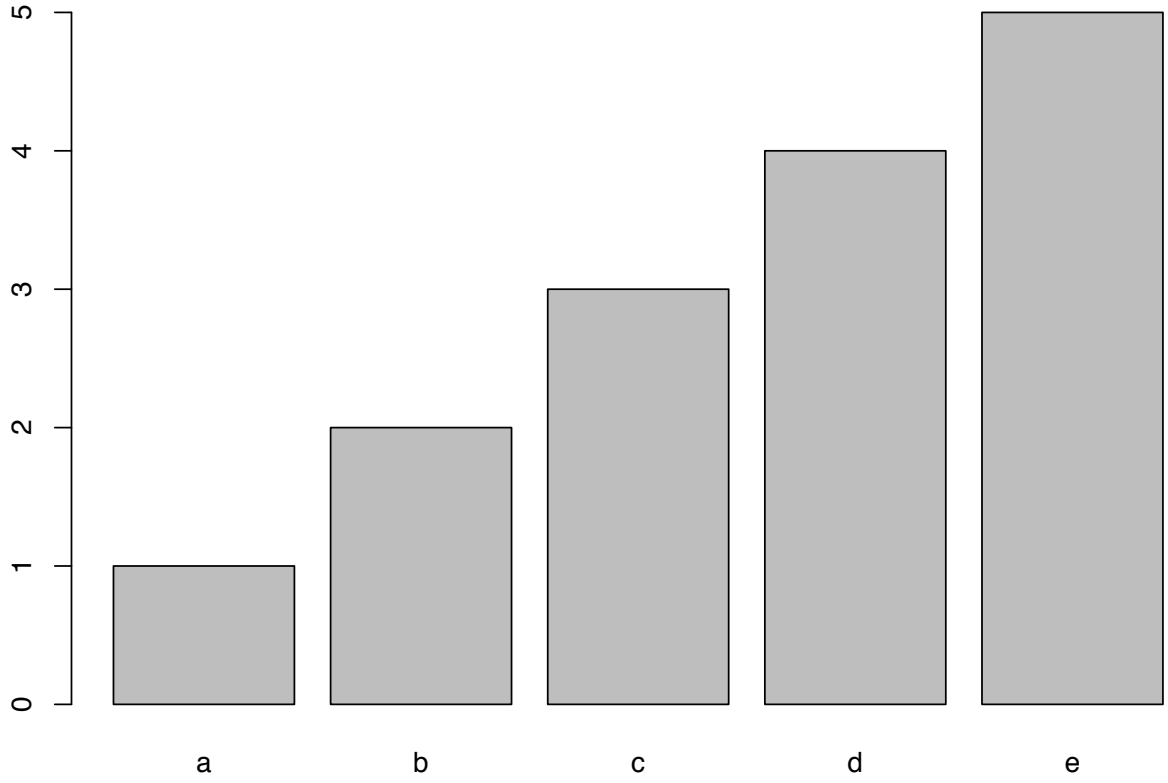
```
> data()
Data sets in package 'datasets':

AirPassengers             Monthly Airline Passenger Numbers
                          1949-1960
BJsales                   Sales Data with Leading Indicator
BJsales.lead (BJsales)
                          Sales Data with Leading Indicator
BOD                       Biochemical Oxygen Demand
CO2                       Carbon Dioxide uptake in grass
                          plants
ChickWeight               Weight versus age of chicks
                          different diets
```

. . . many more

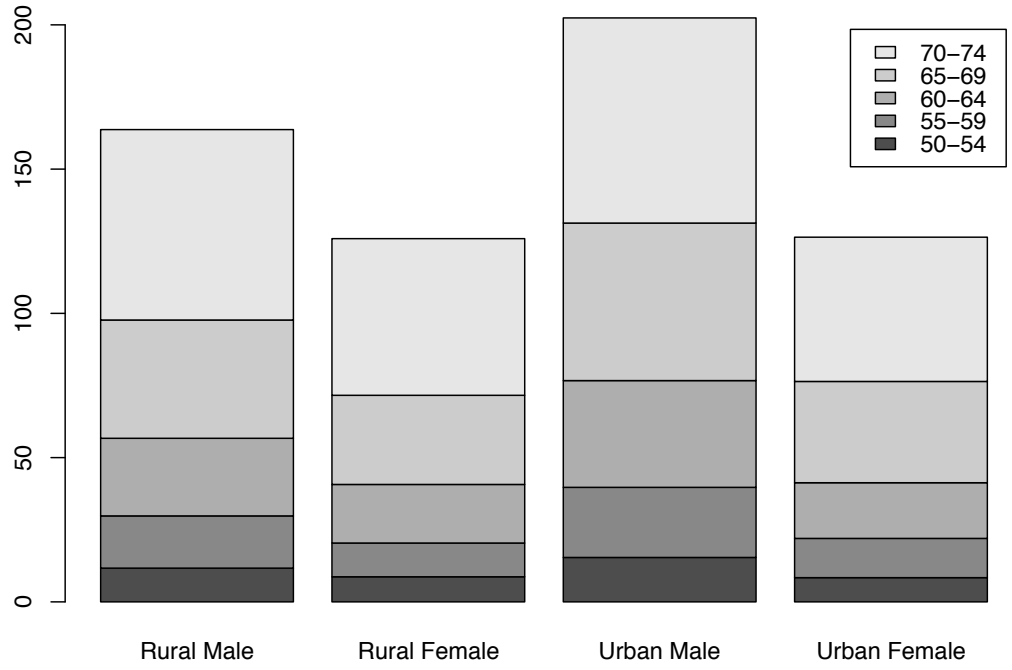
Also, `help(AirPassengers)` and so on will give you information about each one.

1. Barplots

```
> x <- 1:5; names(x) <- letters[1:5]
> x
a b c d e
1 2 3 4 5
> barplot(x)
```



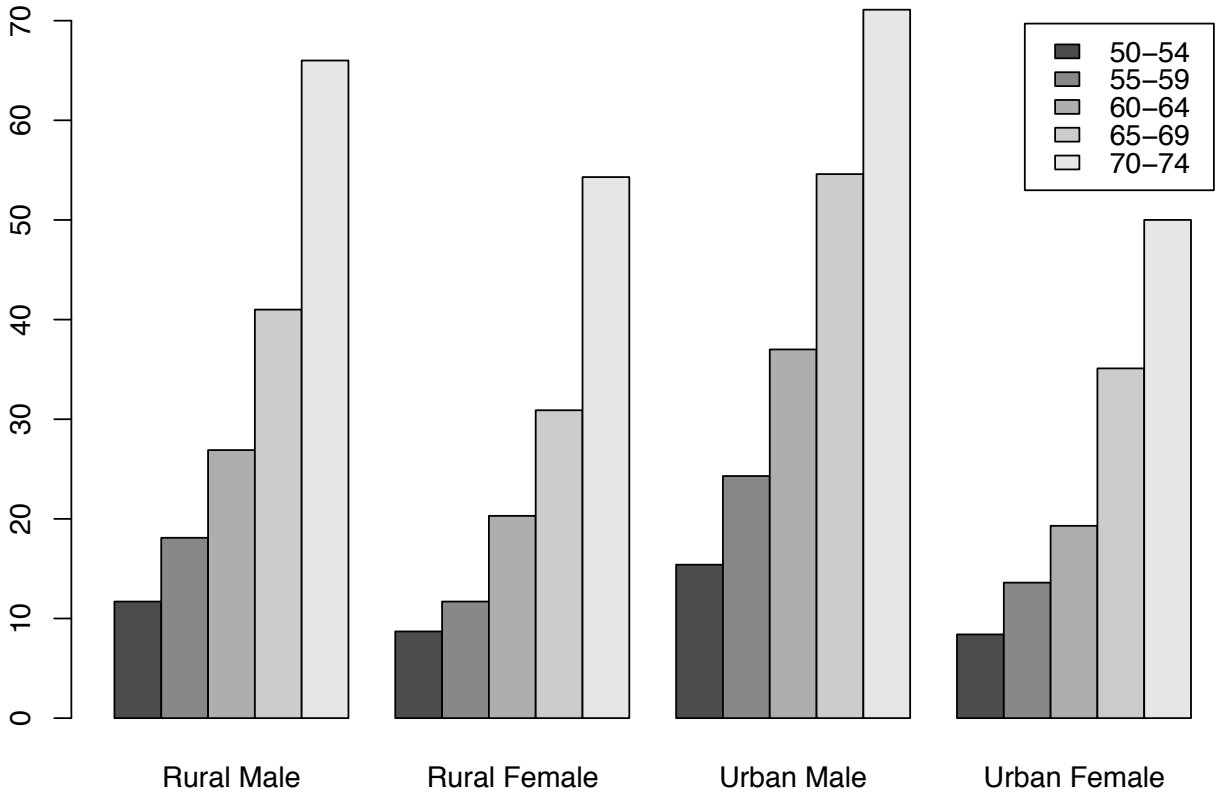
```
> VADeaths
      Rural Male Rural Female Urban Male Urban Female
50-54       11.7          8.7       15.4          8.4
55-59       18.1         11.7       24.3         13.6
60-64       26.9         20.3       37.0         19.3
65-69       41.0         30.9       54.6         35.1
70-74       66.0         54.3       71.1         50.0
> barplot(VADeaths, legend = TRUE)
```



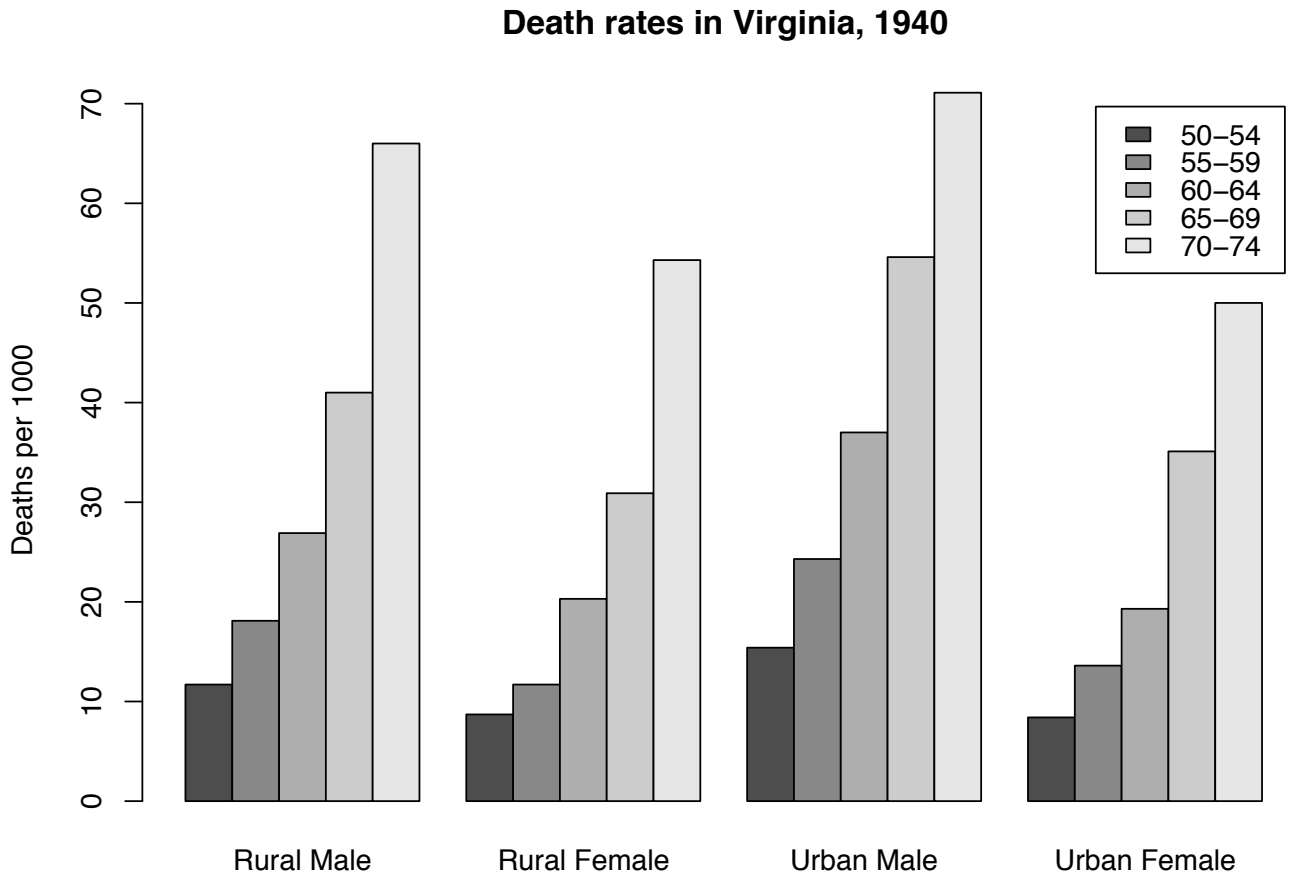
This stacked barplot makes it hard to read anything but the bottom category and the total.

Making a good plot in R is often a matter of iterative improvement.

```
> barplot(VADeaths, beside = TRUE, legend = TRUE)
```



```
> barplot(VADeaths, beside = TRUE, legend = TRUE,
+         ylab = "Deaths per 1000",
+         main = "Death rates in Virginia, 1940")
```



Aside: Saving your plots as graphics files

If you call a high-level plot command, R will automatically start a graphics *device* or window.

To save the contents of the already open device to a file, use `dev.print`.

```
> barplot(VADeaths, legend = TRUE)
> dev.print(device = pdf, file = "mybar.pdf",
+           height = 5, width = 6)              # Inches
> dev.print(device = jpeg, file = "mybar.jpeg",
+           height = 500, width = 600)          # Pixels
```

See `help(device)` for a list of other graphics formats.

Alternatively, if you are using the R GUI, you can click on the plot to highlight that window, then go to File > Save to save the image as a pdf file. However, it's nice to use the code version, so you can easily rerun your whole analysis, including generating the plots.

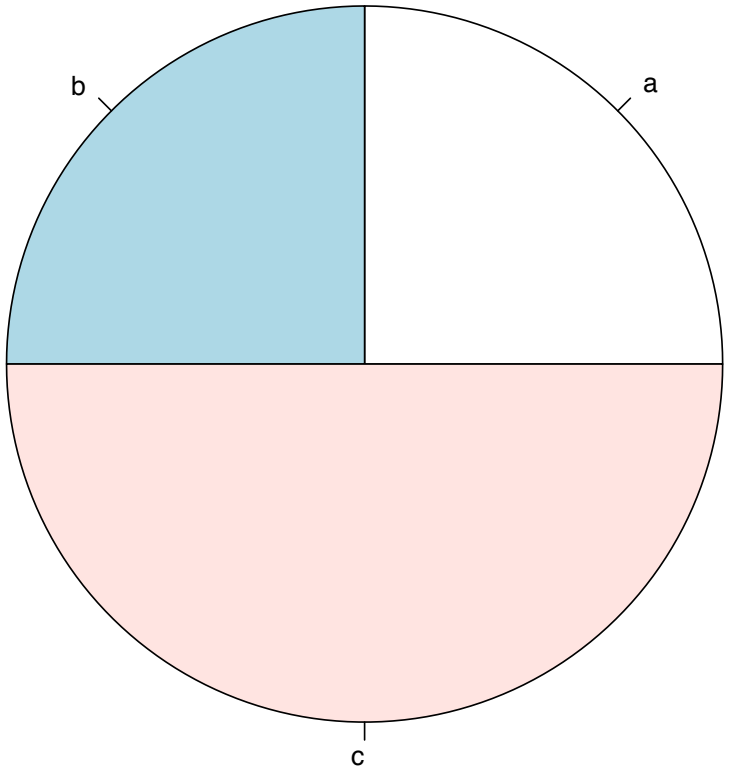
To close the device (shut the window), type

```
> dev.off()
```

To open multiple devices at once, use `x11()` or click on the little graphics icon in the GUI.

2. Pie charts

```
> pie(c(1, 1, 2), labels = letters[1:3])
```



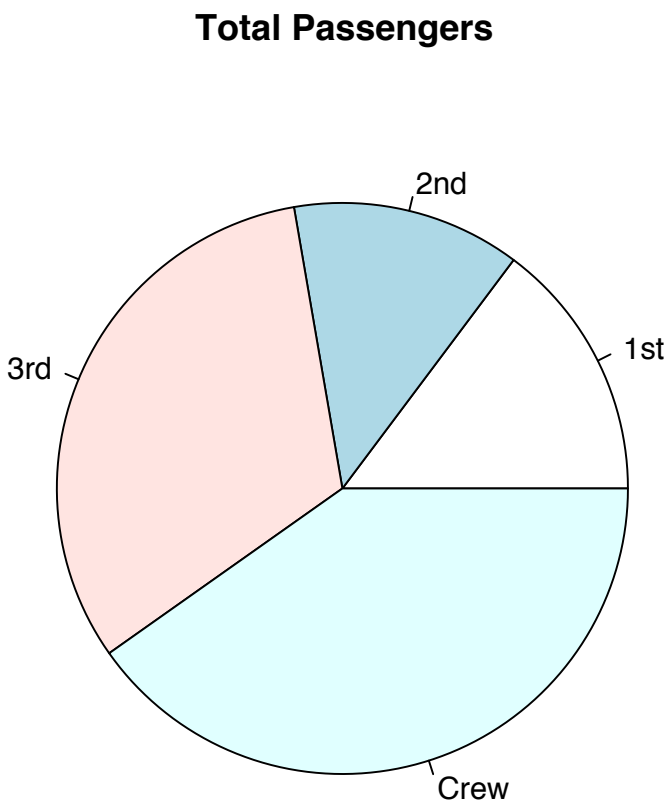
Note that elements of the vector are normalized by their sum, so that the total gives 100% of the pie.

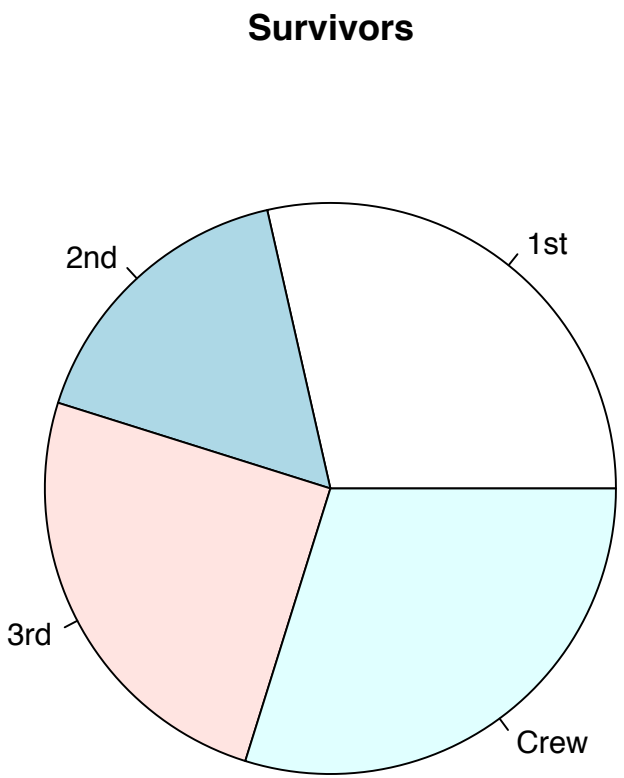
```
> Titanic
, , Age = Child, Survived = No
      Sex
Class  Male Female
  1st     0      0
  2nd     0      0
  3rd    35     17
  Crew    0      0
, , Age = Adult, Survived = No
      Sex
Class  Male Female
  1st   118      4
  2nd   154     13
  3rd   387     89
  Crew  670      3
```

. . . two more matrices not printed here, with survivors

Did all groups have an equal survival rate?

```
> apply(Titanic, 1, sum) # Total passengers, each class
 1st  2nd  3rd Crew
 325  285  706  885
> pie(apply(Titanic, 1, sum), main = "Total Passengers")
> pie(apply(Titanic[,,,"Yes"], 1, sum),
+     main = "Survivors")
```





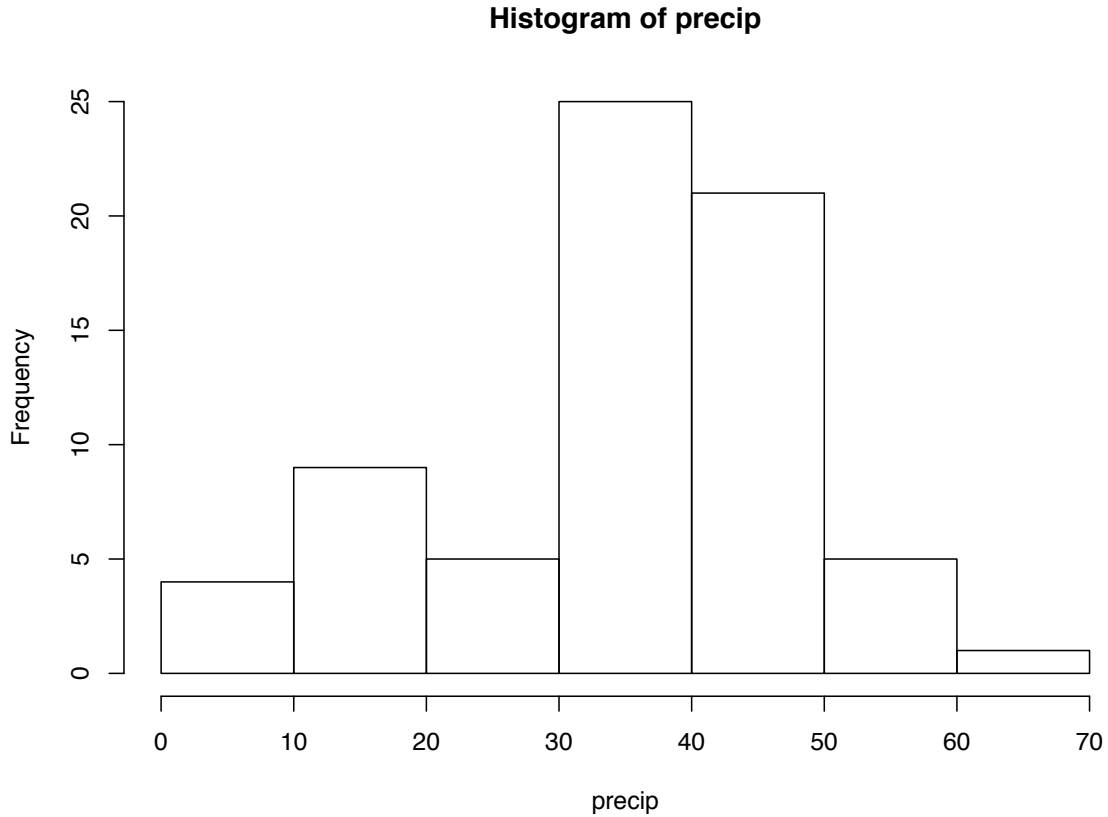
Studies of human perception show we are not very good at comparing areas, volumes, or angles.

- When making bar plots, start the axis at zero and keep all bars the same width, so that length and area are proportional.

- Try to avoid pie charts for anything requiring a precise comparison.

3. Histograms

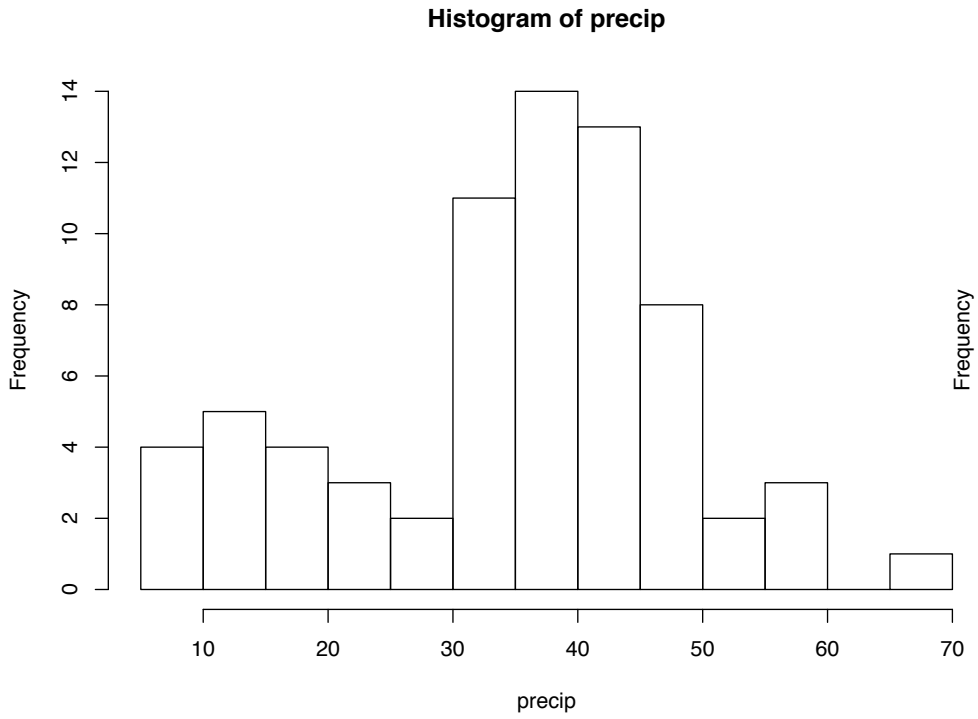
```
> precip[1:4] # Average annual precipitation in cities
    Mobile      Juneau     Phoenix Little Rock
      67.0        54.7         7.0        48.5
> hist(precip)
```

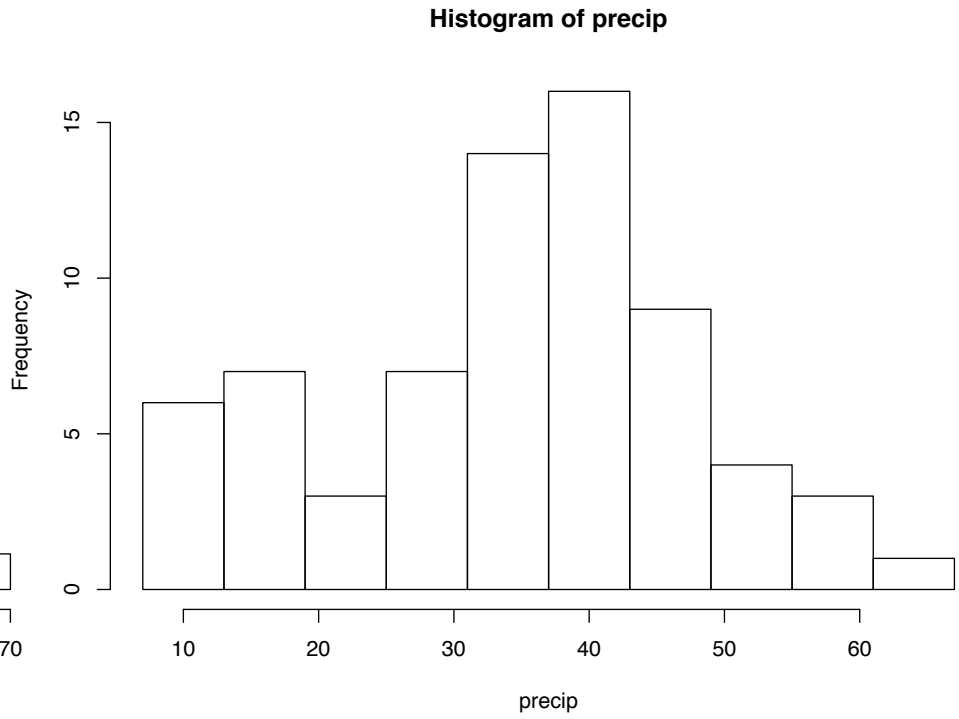


The height of the bars shows the number of observations falling into each bin.

There are several ways to change the cutoff points.

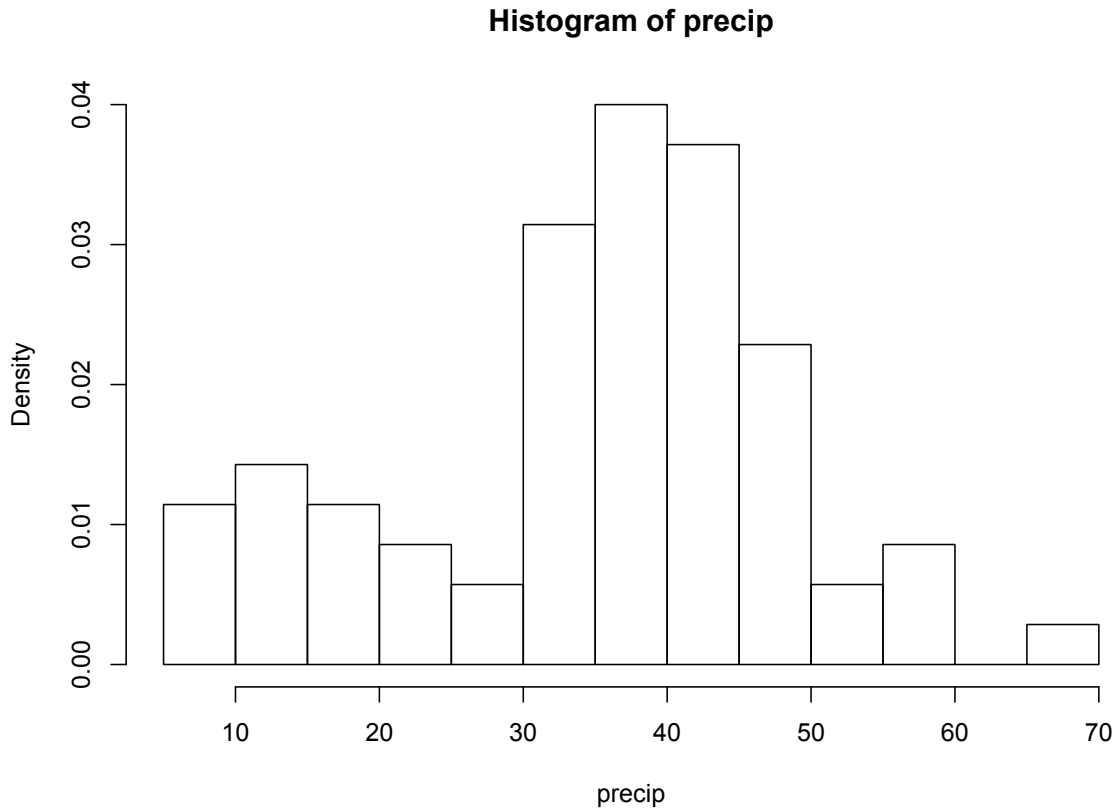
```
> hist(precip, breaks = 10) # Only a suggestion to R
> hist(precip, breaks = seq(min(precip), max(precip),
+                           length = 11)) # Force it
```





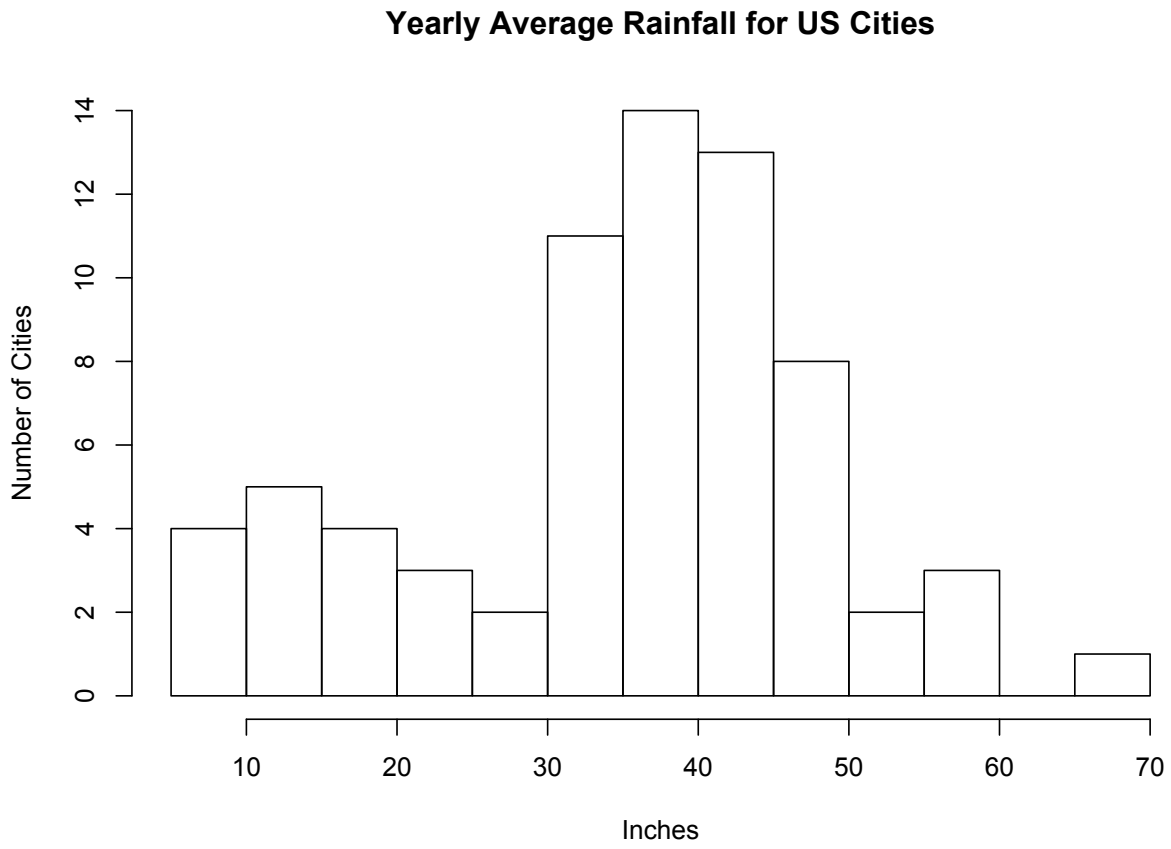
We can also change the y-axis to reflect probabilities rather than densities. Under the rescaled axis, the *areas of the bars* will sum to 1.

```
> hist(precip, breaks = 10, freq = FALSE)
```



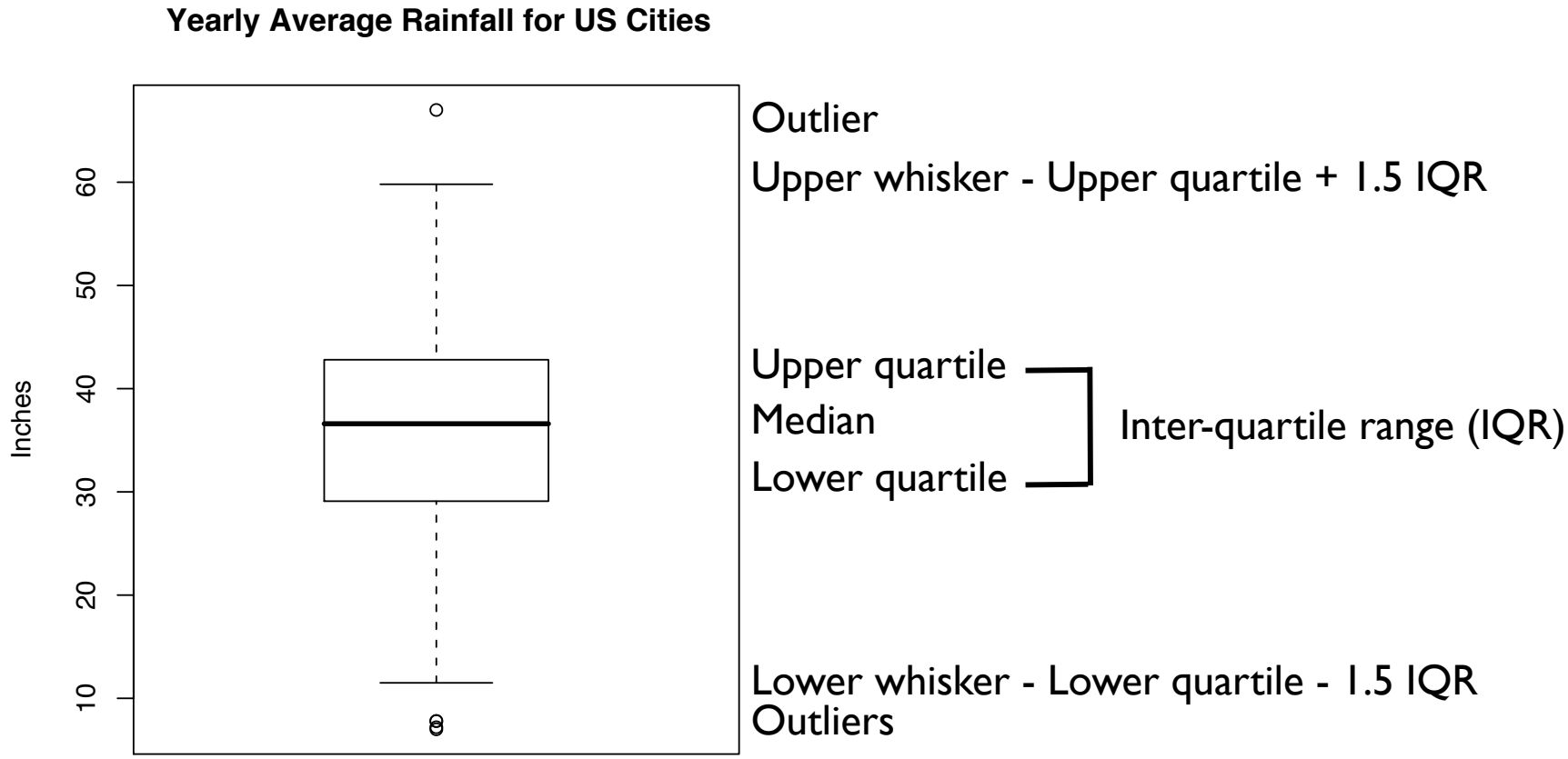
Again, let's add meaningful axis labels and a title.

```
> hist(precip, breaks = 10, xlab = "Inches",
+      ylab = "Number of Cities",
+      main = "Yearly Average Rainfall for US Cities")
```

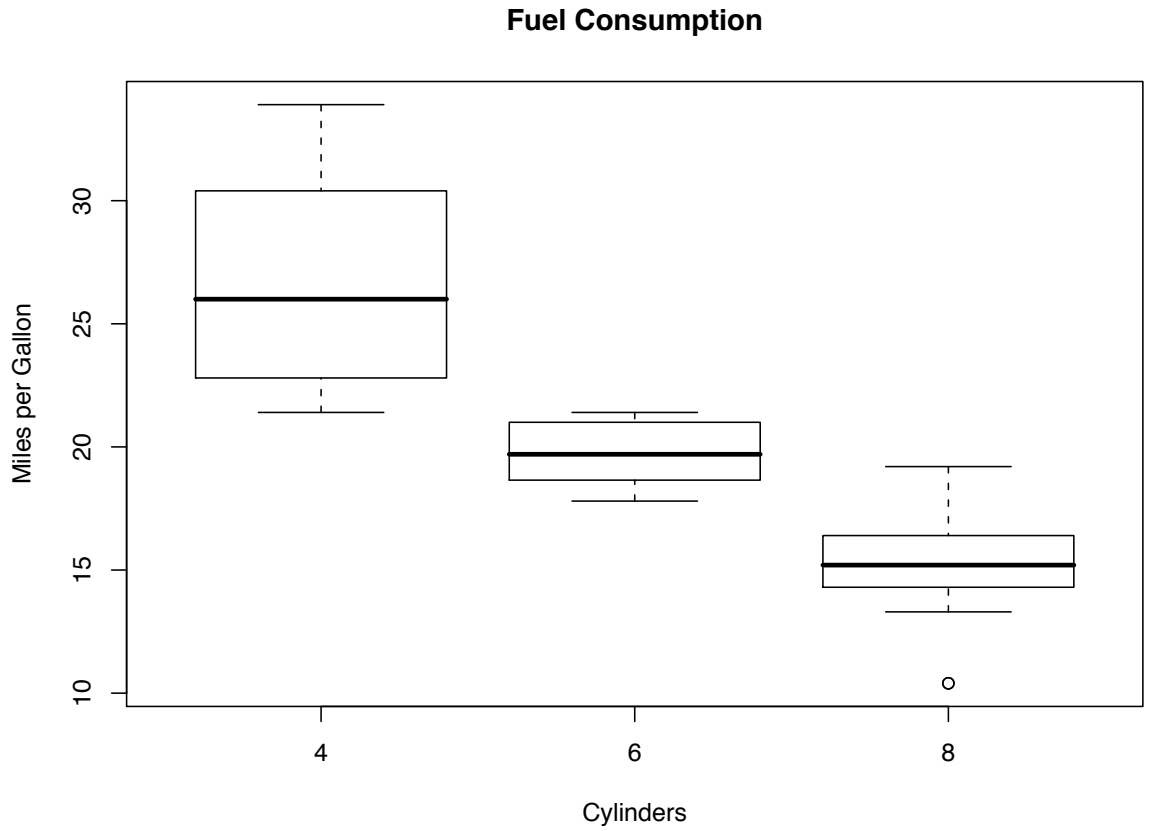


4. Boxplots

```
> boxplot(precip, ylab = "Inches",
+         main = "Yearly Average Rainfall for US Cities")
```

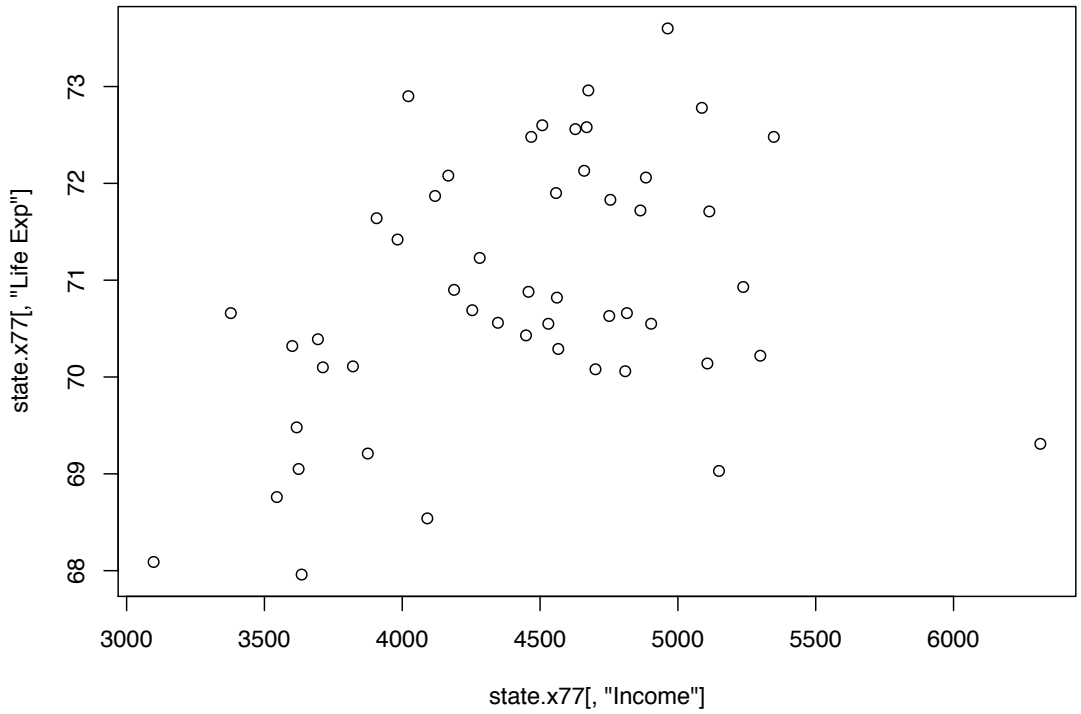


```
> mtcars[1:2,1:5]
              mpg cyl disp  hp drat
Mazda RX4      21   6  160 110  3.9
Mazda RX4 Wag  21   6  160 110  3.9
> boxplot(mpg~cyl, data = mtcars, xlab = "Cylinders",
+         ylab = "Miles per Gallon",
+         main = "Fuel Consumption")
```

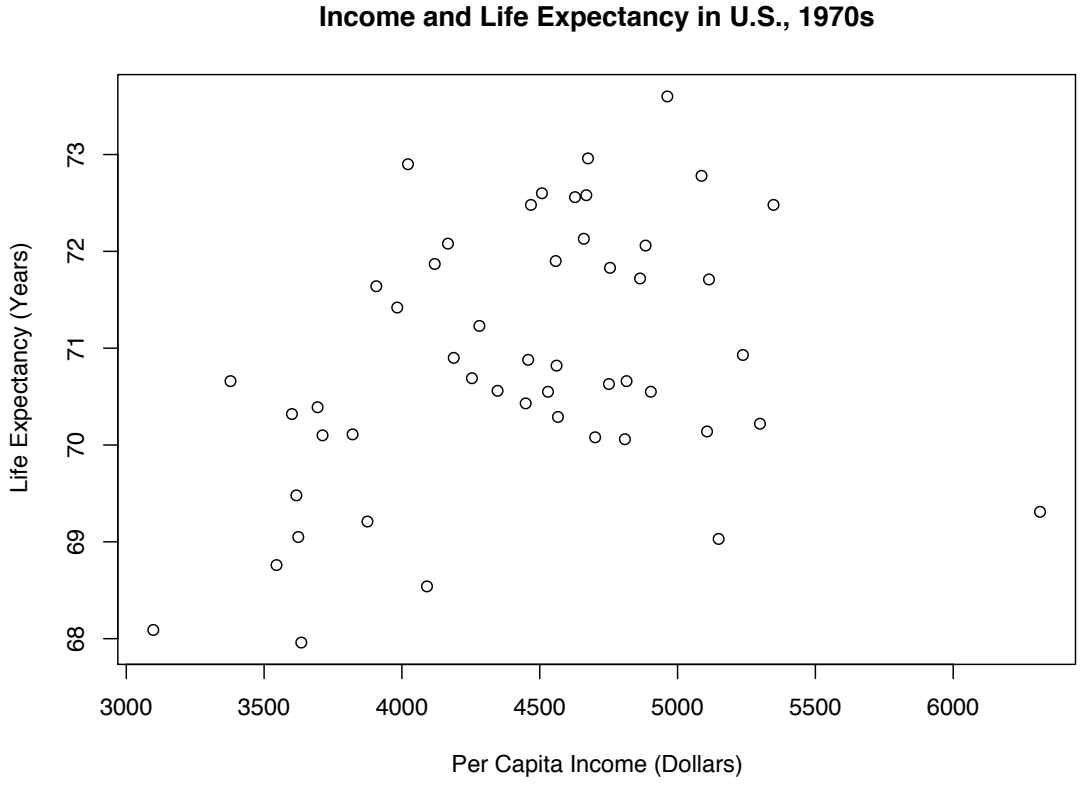


5. Scatterplots

```
> state.x77[1:2,1:4]
           Population Income Illiteracy Life Exp
Alabama          3615   3624        2.1    69.05
Alaska            365   6315        1.5    69.31
> plot(state.x77[,"Income"], state.x77[,"Life Exp"])
```



```
> plot(state.x77[,"Income"], state.x77[,"Life Exp"],
+      xlab = "Per Capita Income (Dollars)",
+      ylab = "Life Expectancy (Years)",
+      main = "Income and Life Expectancy in U.S., 1970s")
```



We can label the interesting cases with `identify`.

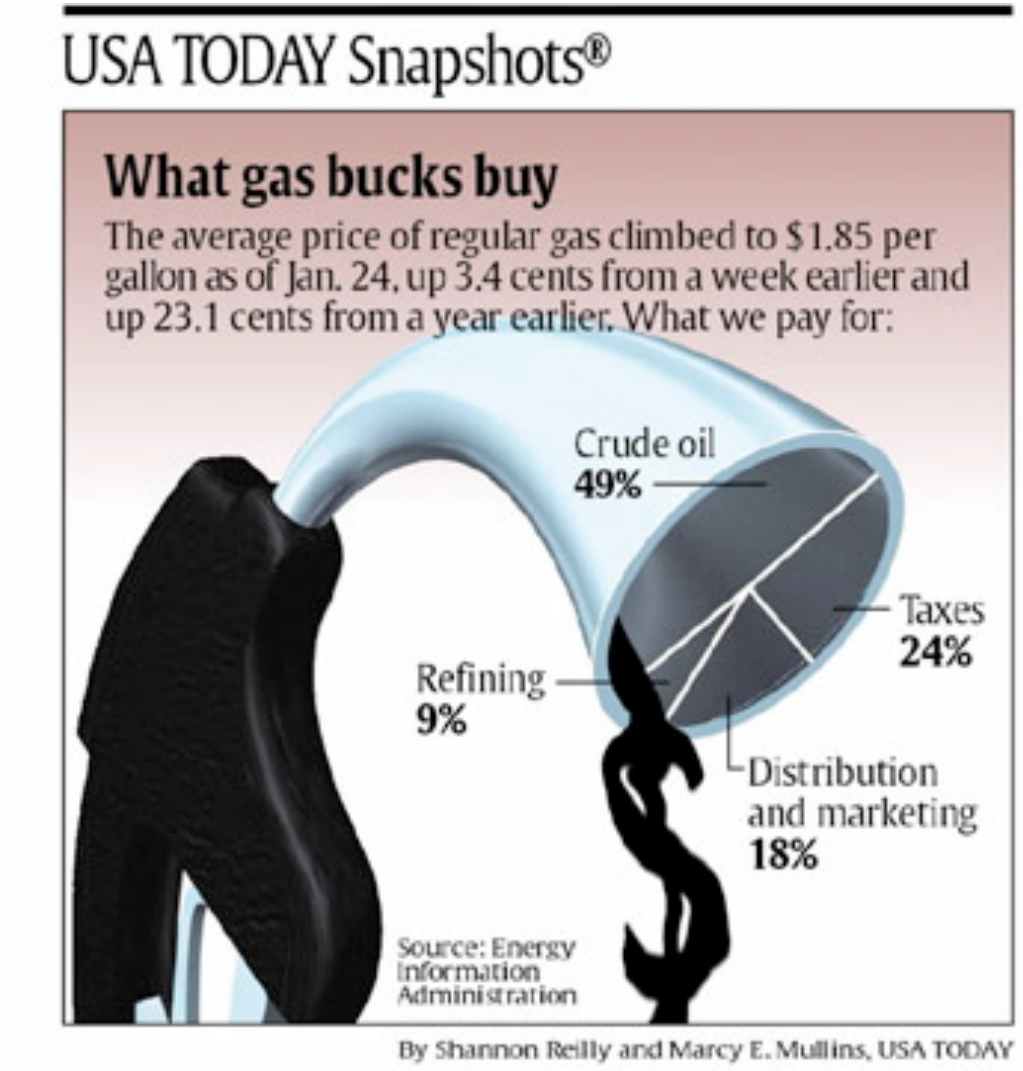
Designing Good Graphics

(or, avoiding "The Dirty Dozen")

Inspired by Wainer, H. (1984) "How to Display Data Badly." The American Statistician, 38, 137-147.
Some additional images from Tufte, E. The Visual Display of Quantitative Information and online news sources.

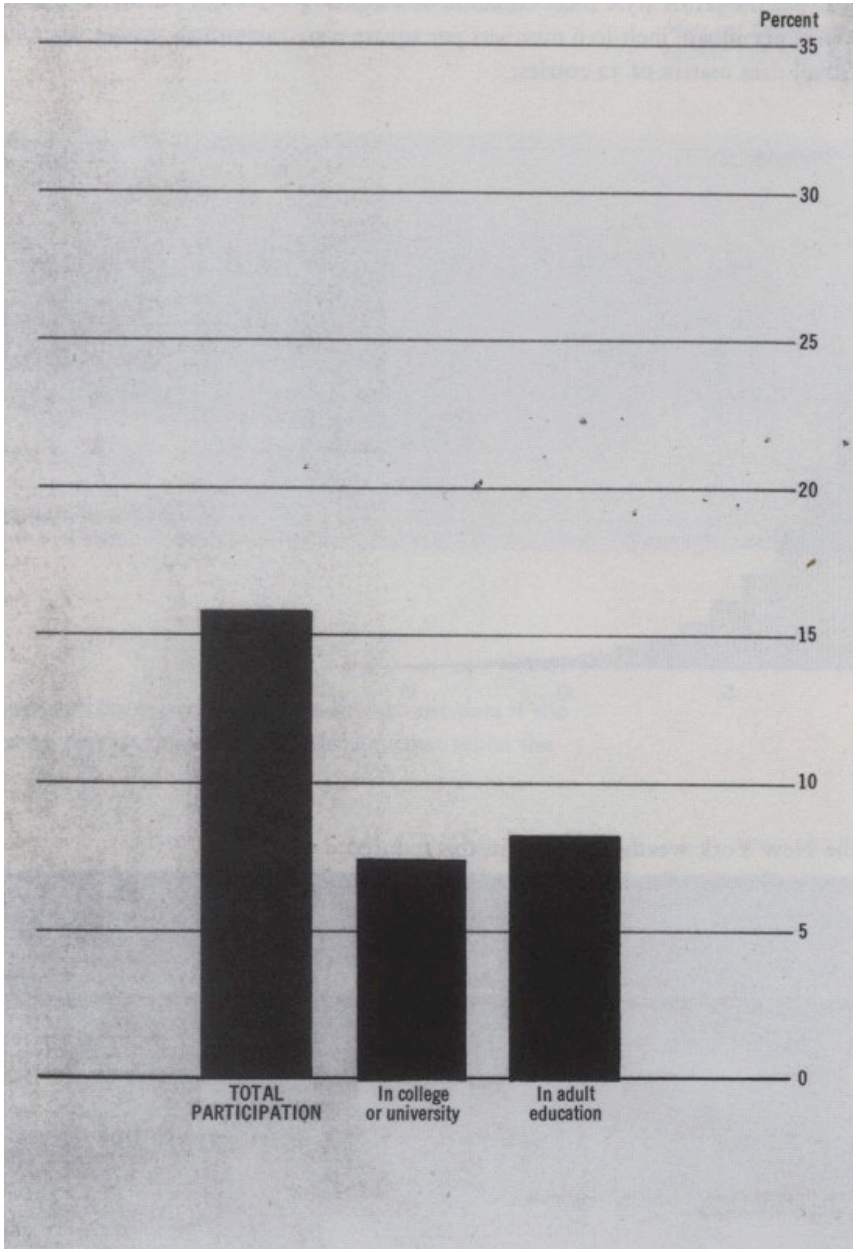
1. Show as few data as possible.

An example with lots of "chart junk," not to mention visual distortion

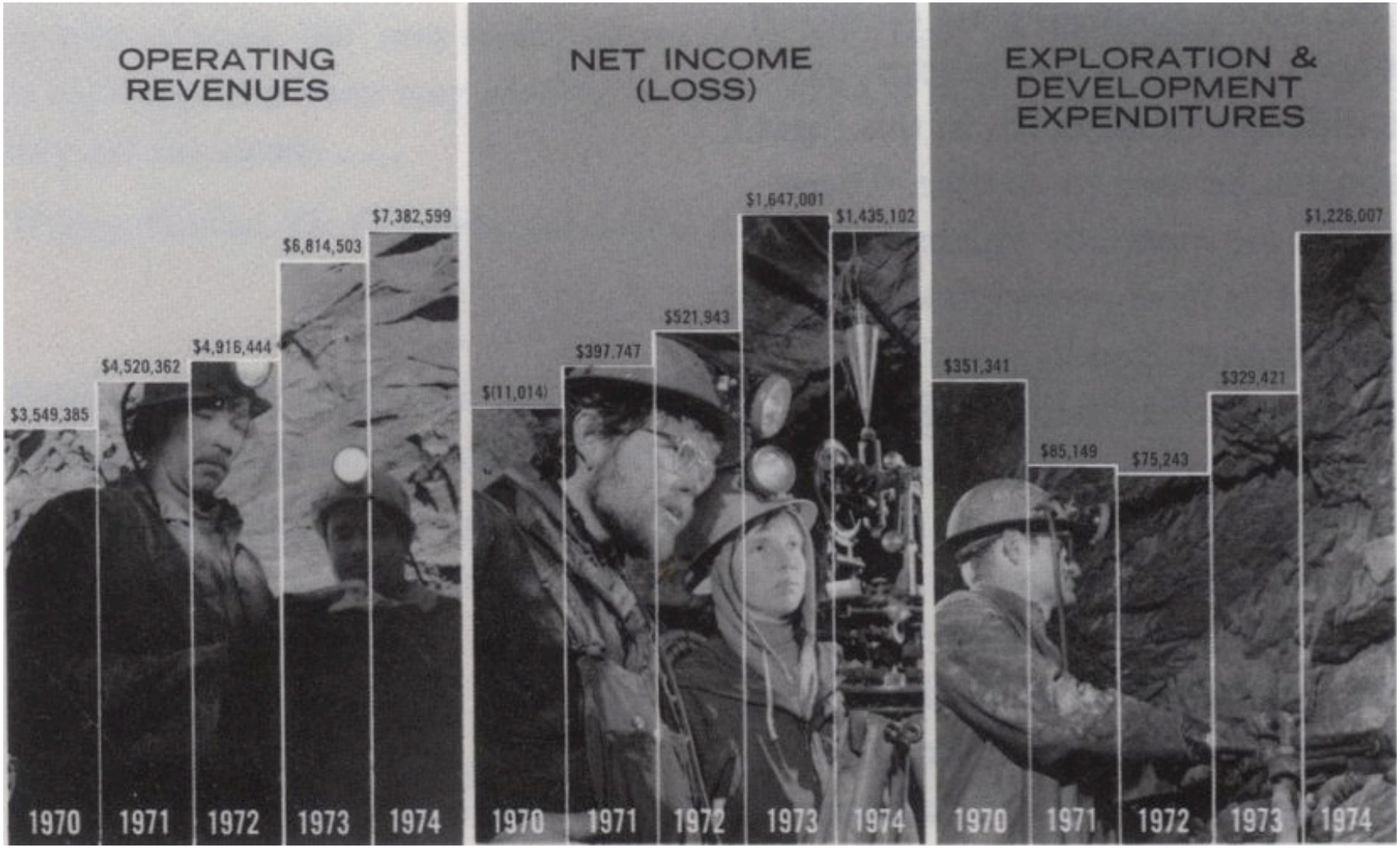


How many data points?

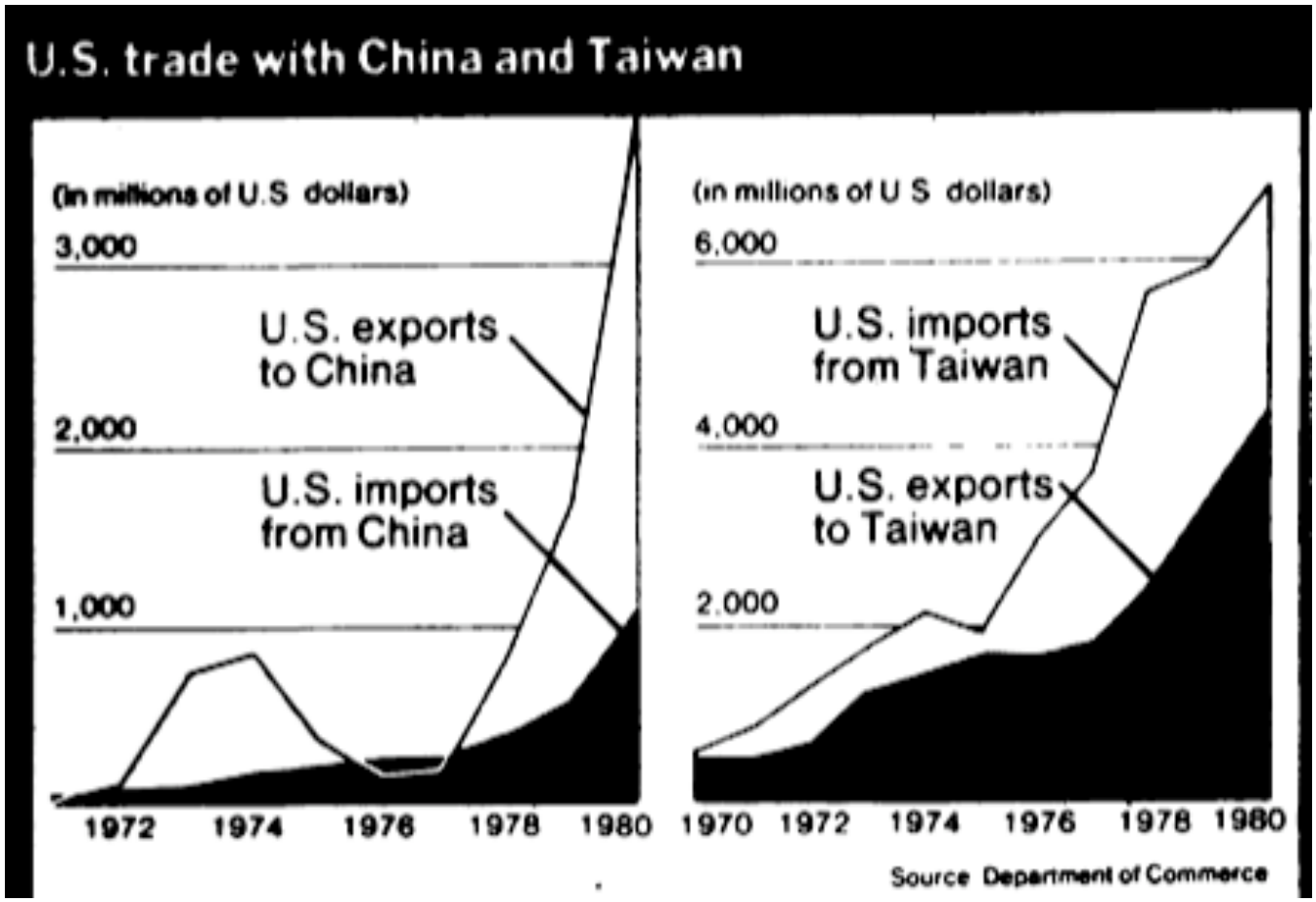
Tufte's "data to ink ratio"



2. Hide what data you do show.

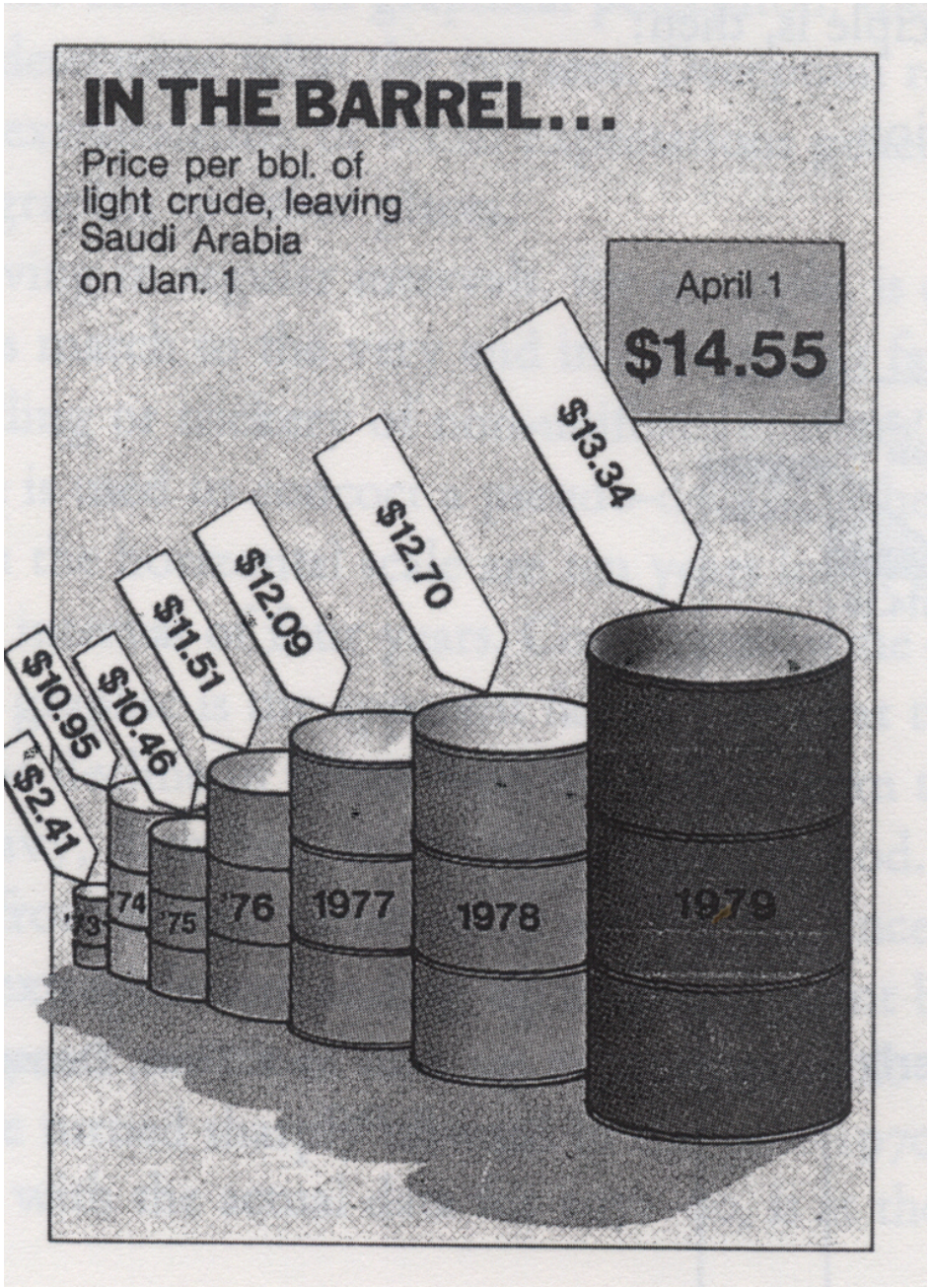


3. Ignore the visual metaphor, or reverse it mid-graph.



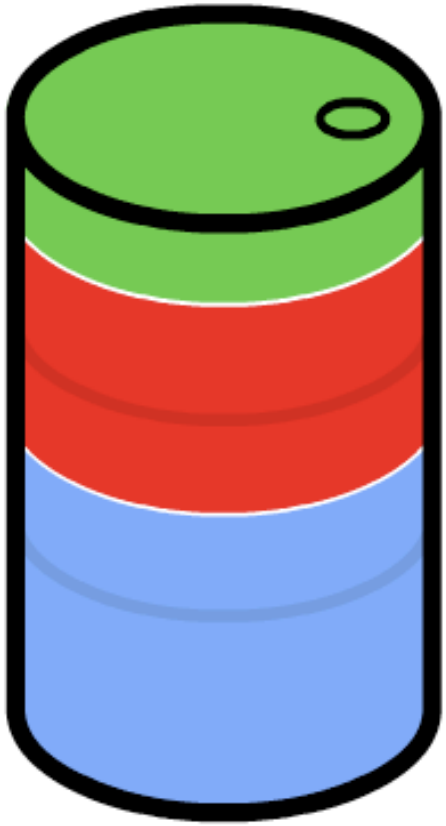
4. Only order matters.

Are we supposed to compare length, area, or volume?



Black gold at an all time high

Every $1 increase in the price of a barrel of crude equates to roughly 0.42p on a litre of petrol at the pump. At $150 a barrel, a litre of unleaded would cost £1.25 on the forecourt; at $200 that rises to £1.46 - or £80 to fill up the family runabout.



Forecourt and transport

8.9%

Oil and refining

32.4%

Tax

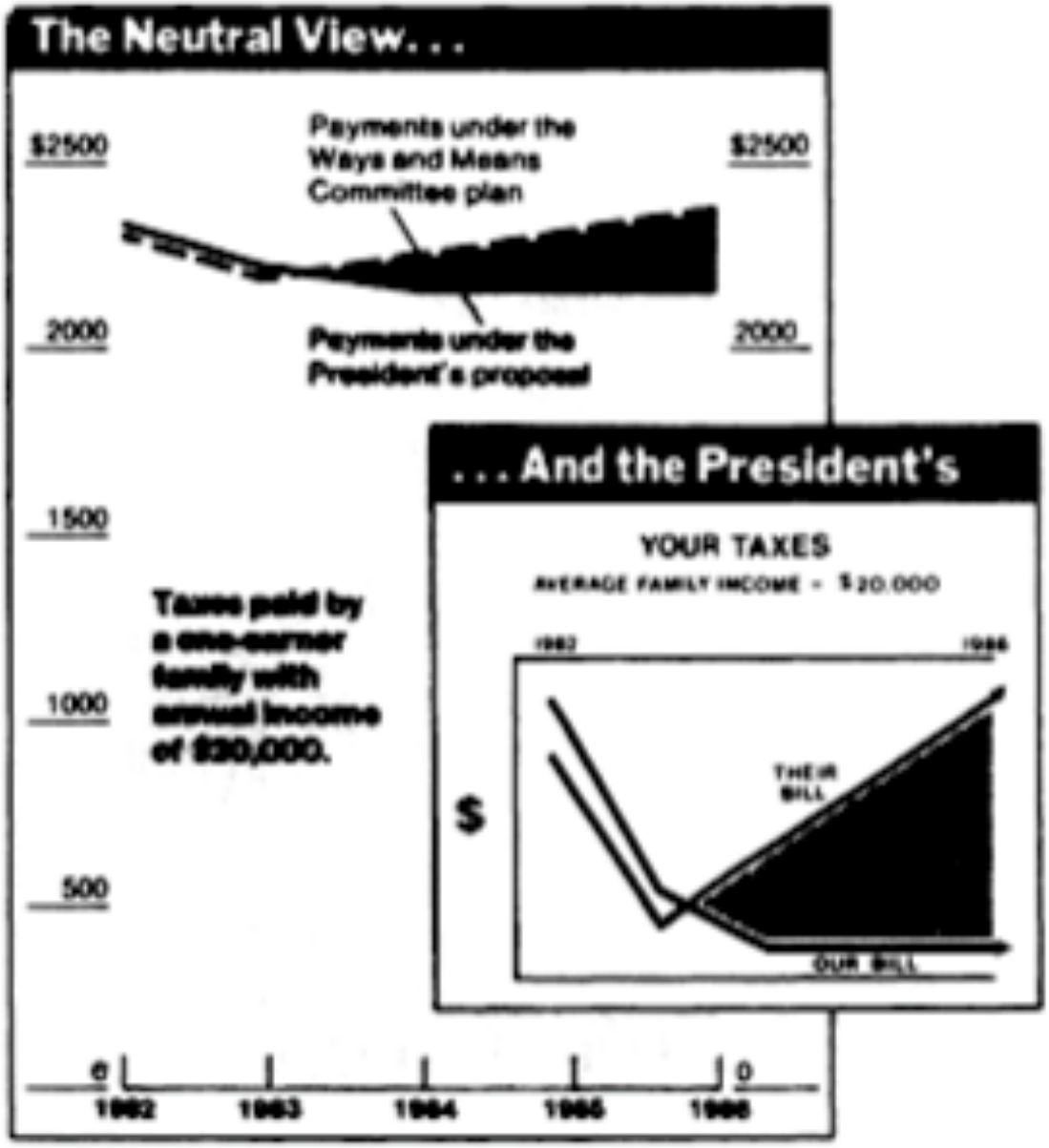
58.7%

Duty - 43.8%

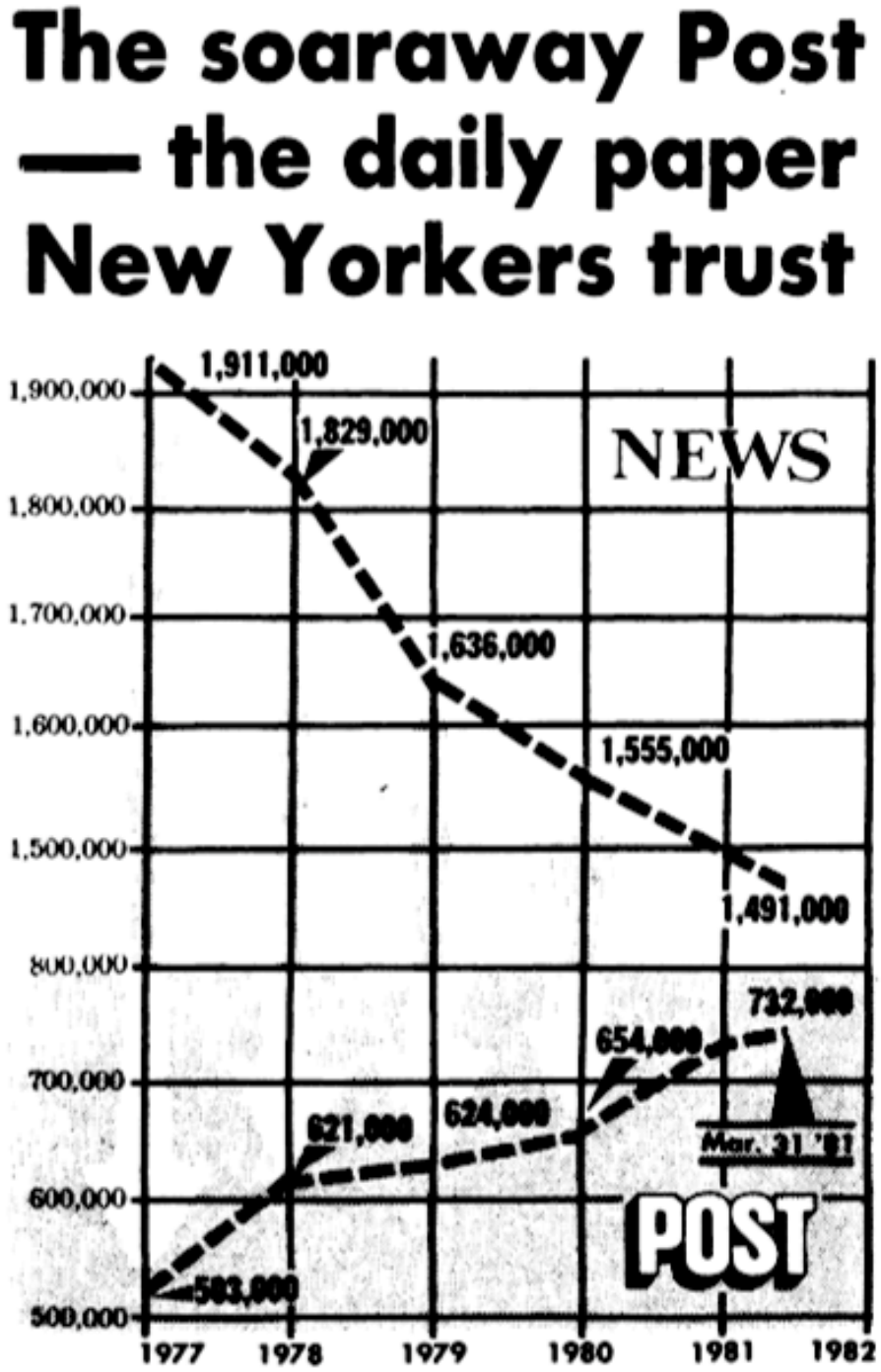
VAT - 14.9%

SOURCE: petrolprices.com

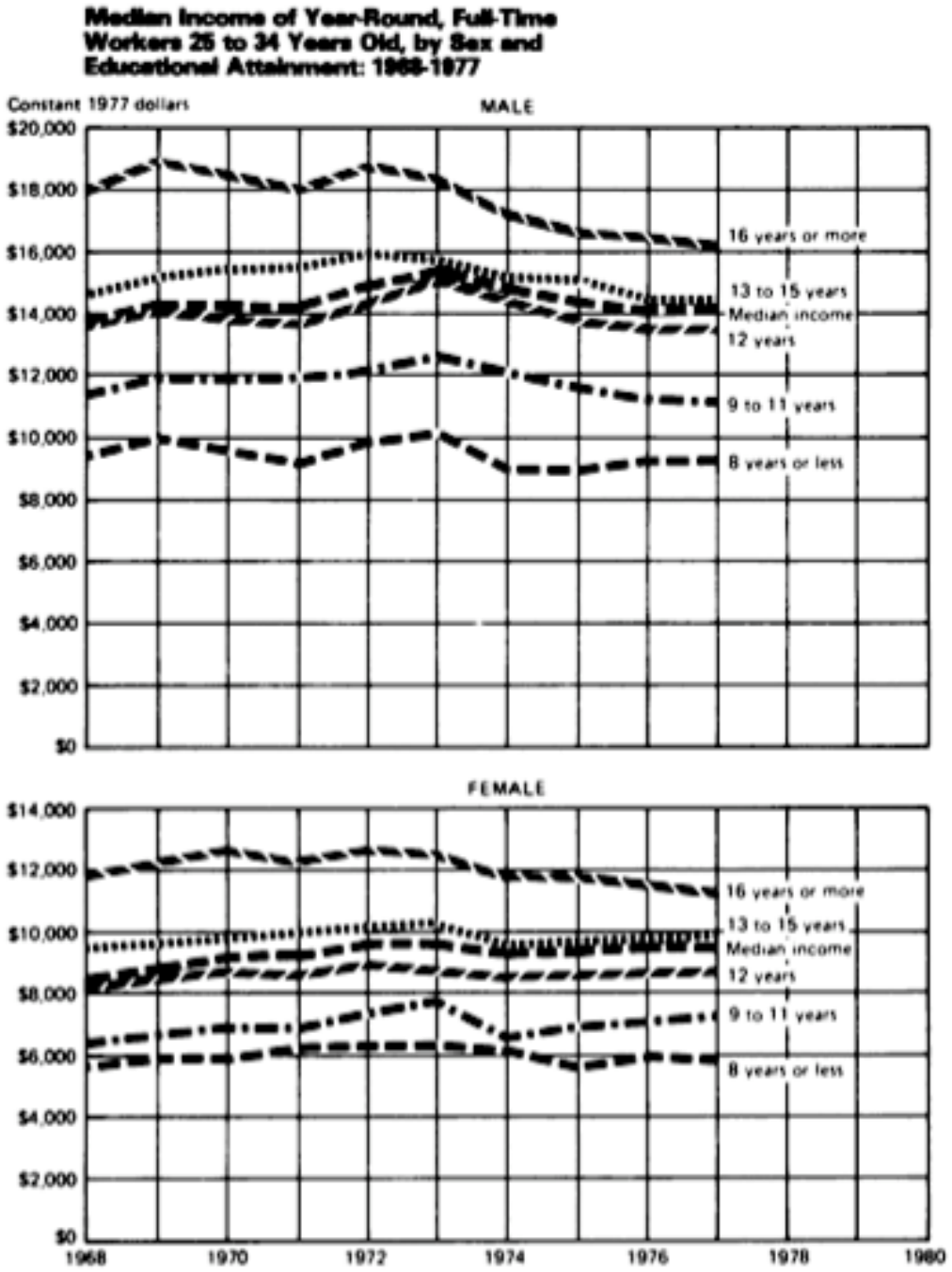
5. Graph data out of context.



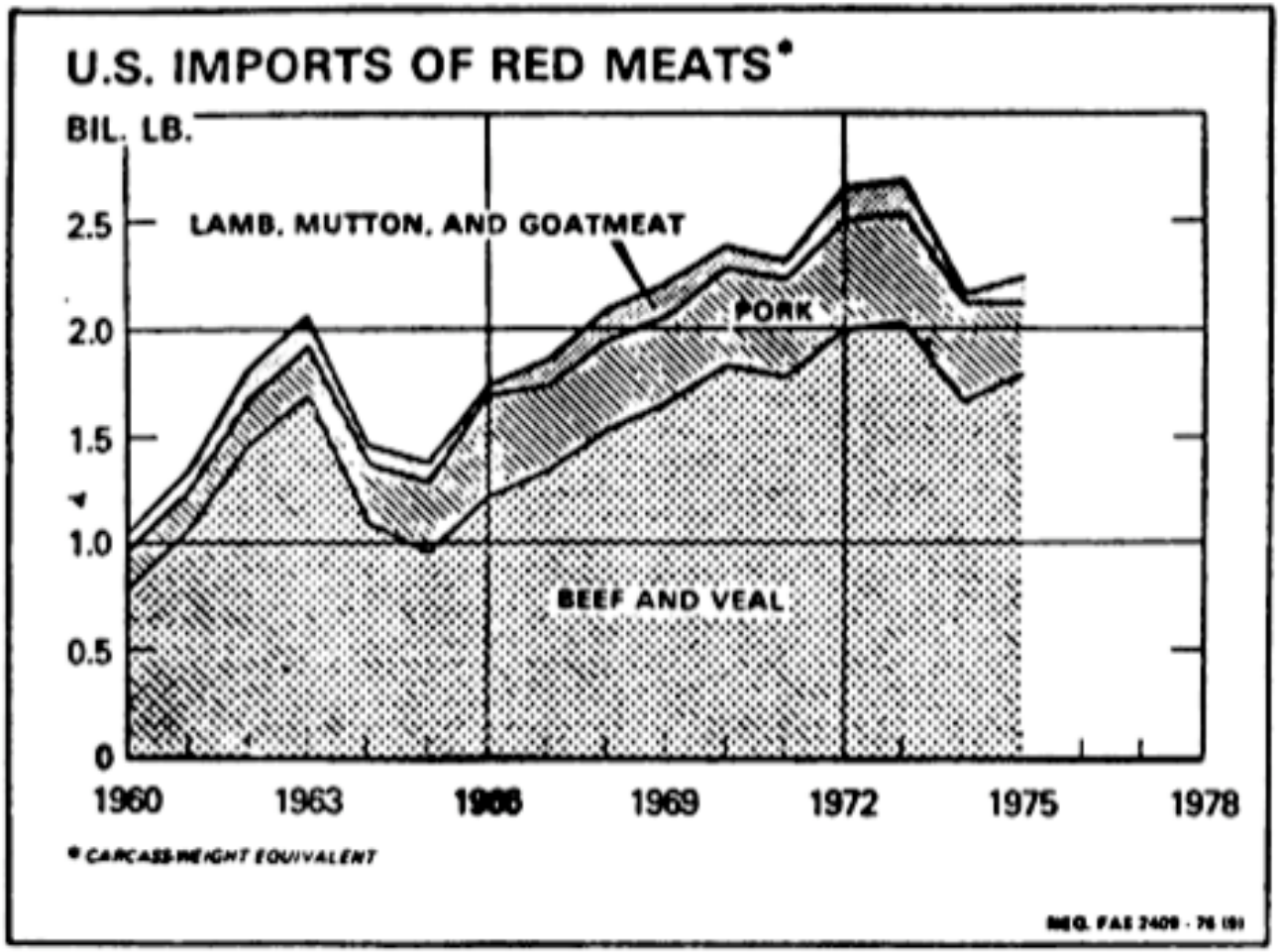
6. Change scales in mid-axis.



7. Emphasize the trivial. (Ignore the important.)

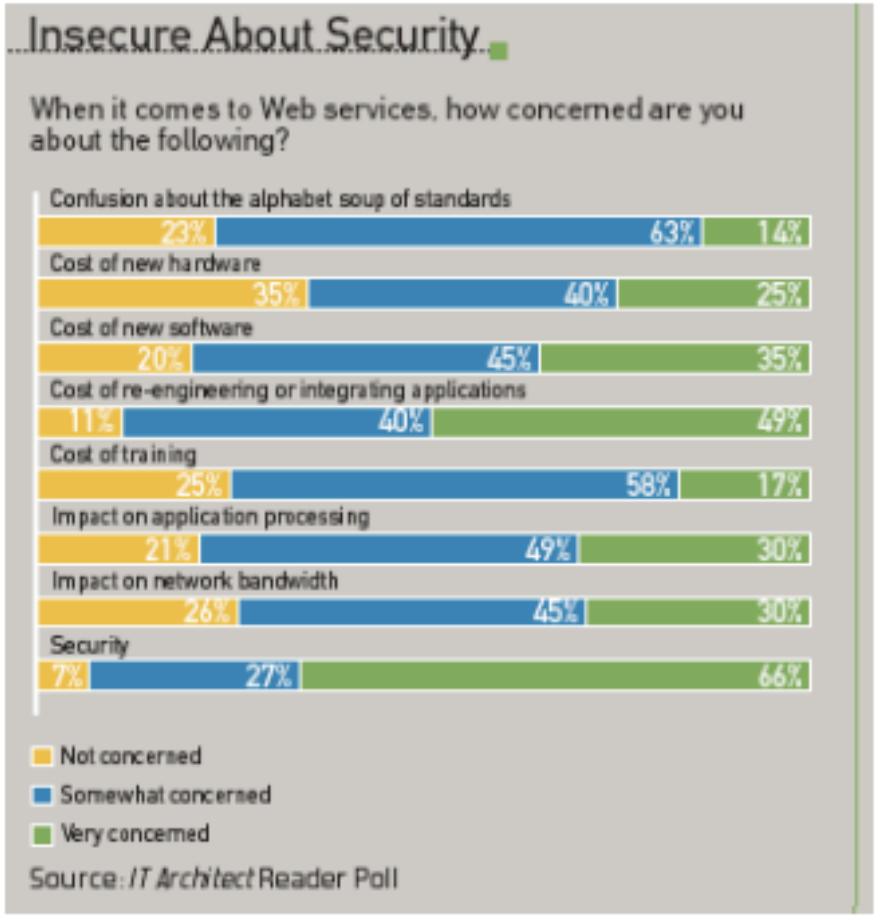


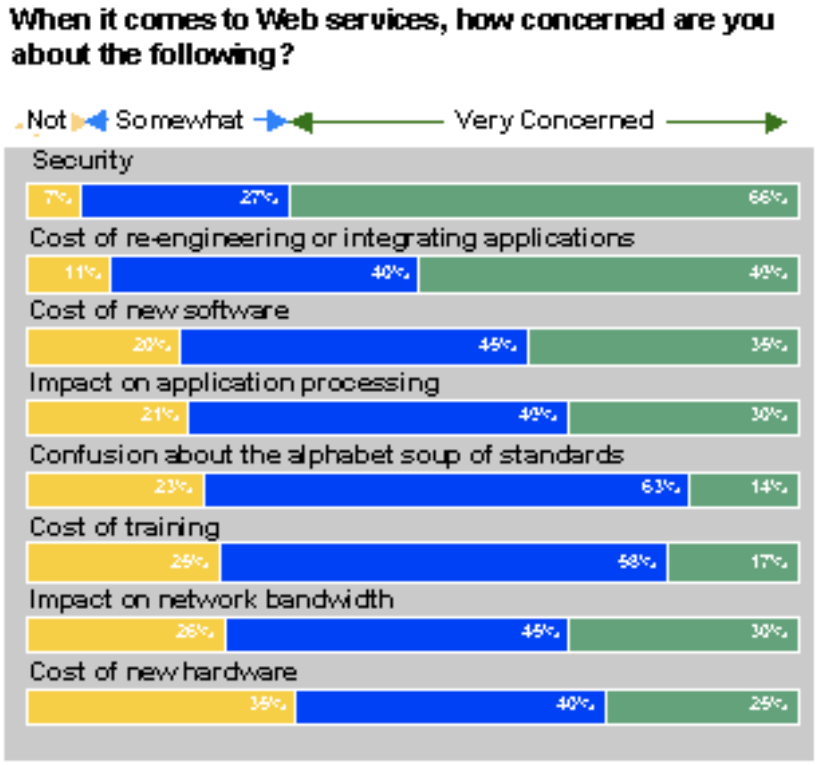
8. Jiggle the baseline.



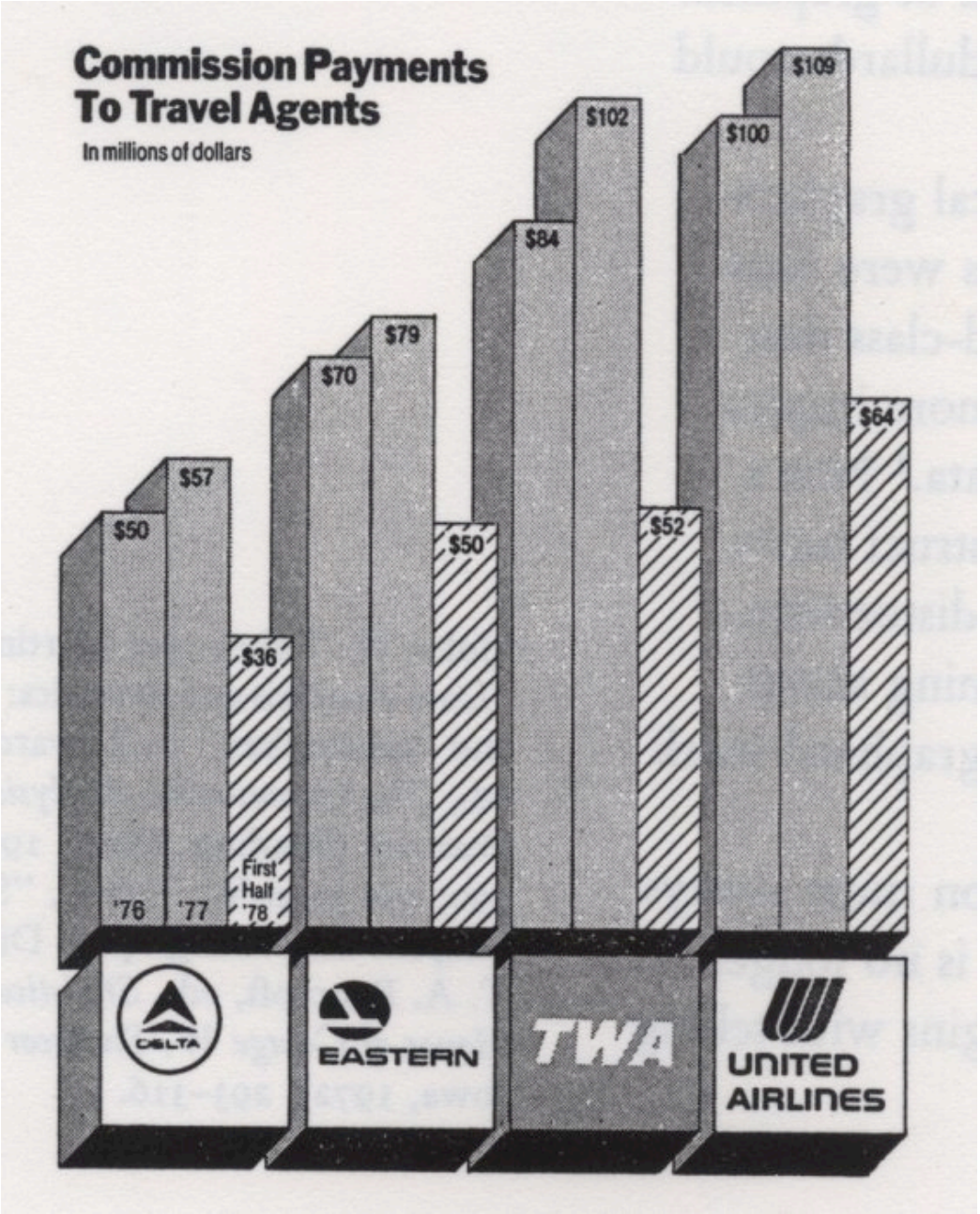
Sometimes varying the baseline is ok, if the main points of comparison are the first category and the total. This plot, however, is bizarre in other ways.

9. Austria first!

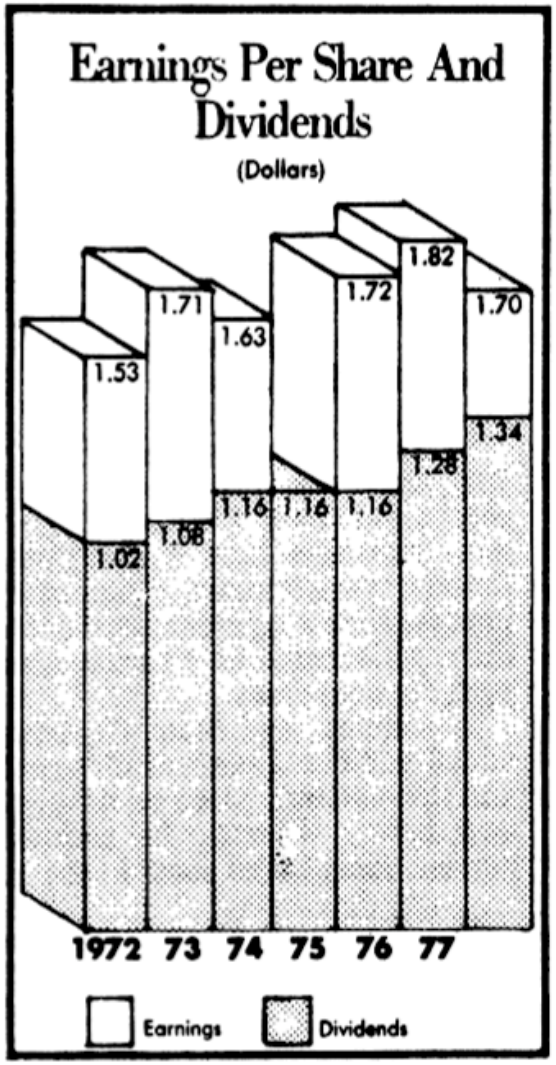


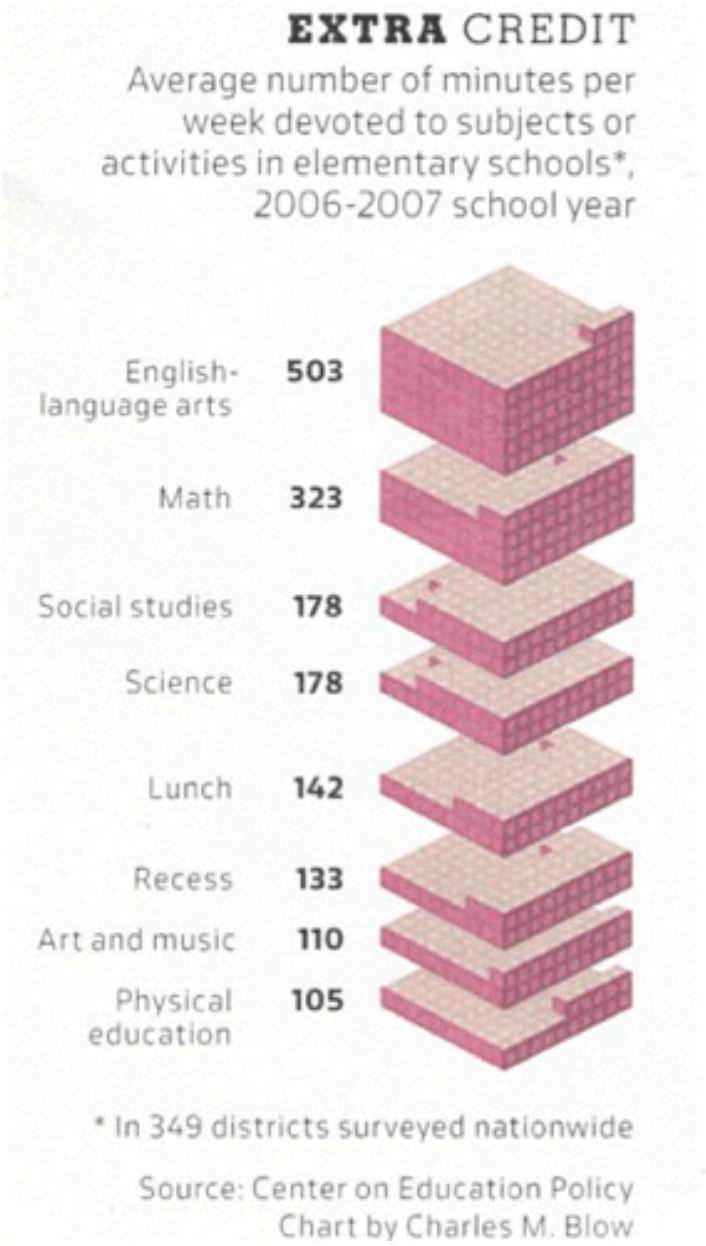


10. Label (a) Illegibly,
(b) Incompletely,
(c) Incorrectly, and
(d) Ambiguously.

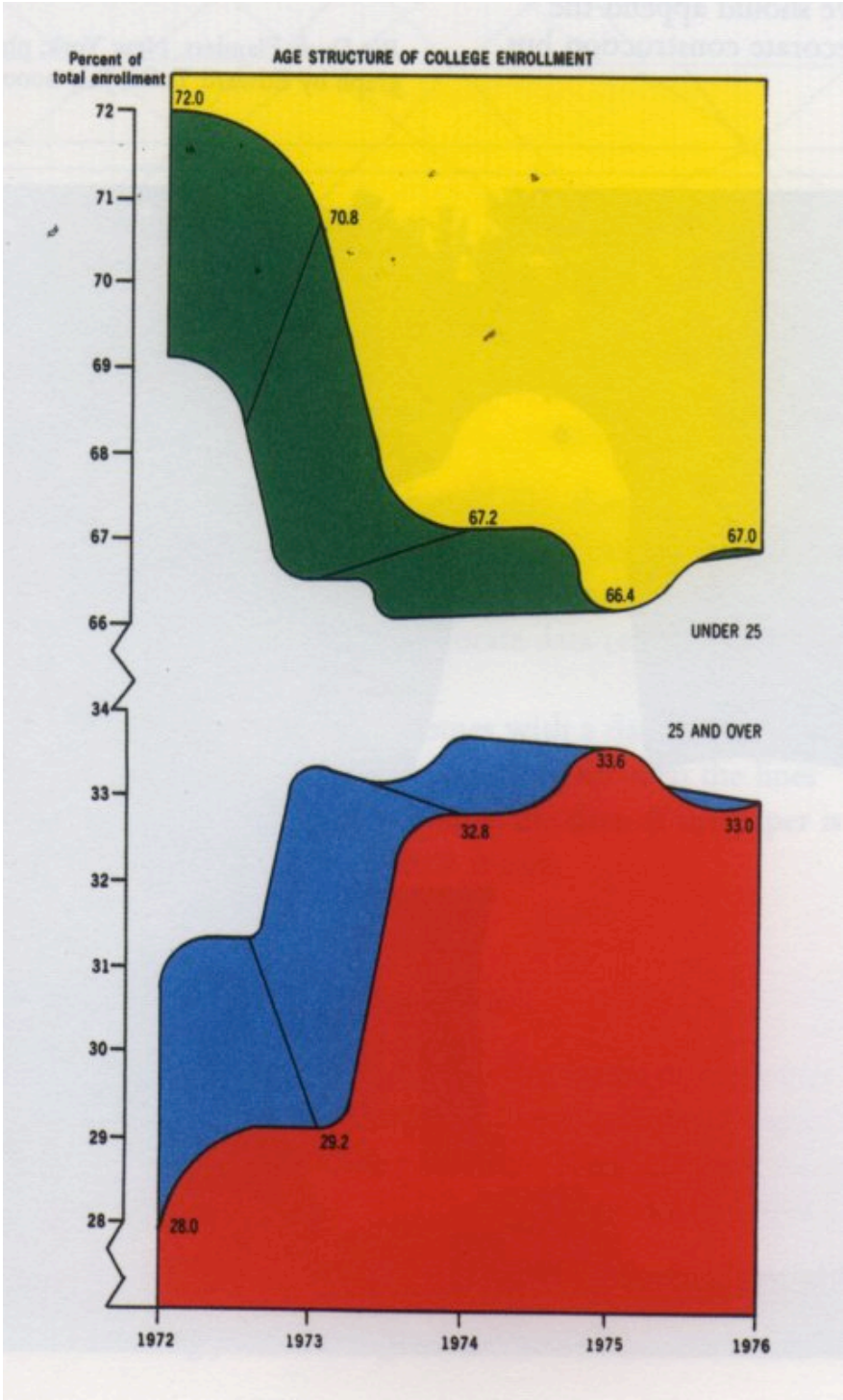


11. More (dimensions) is murkier.





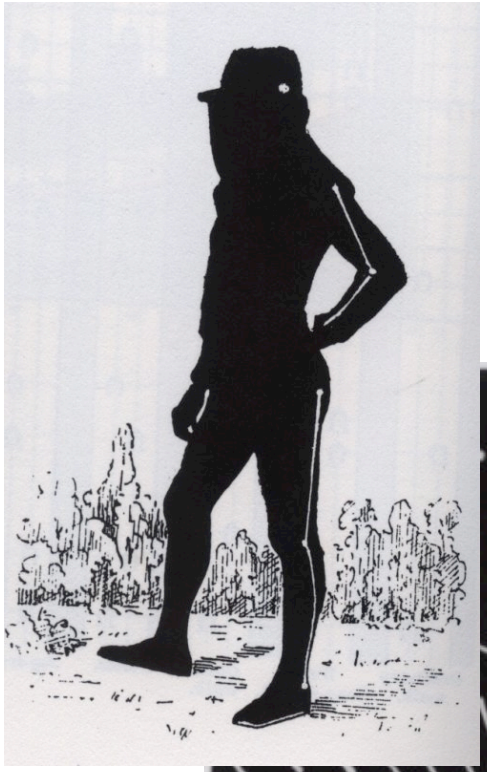
More dimensions AND colors!

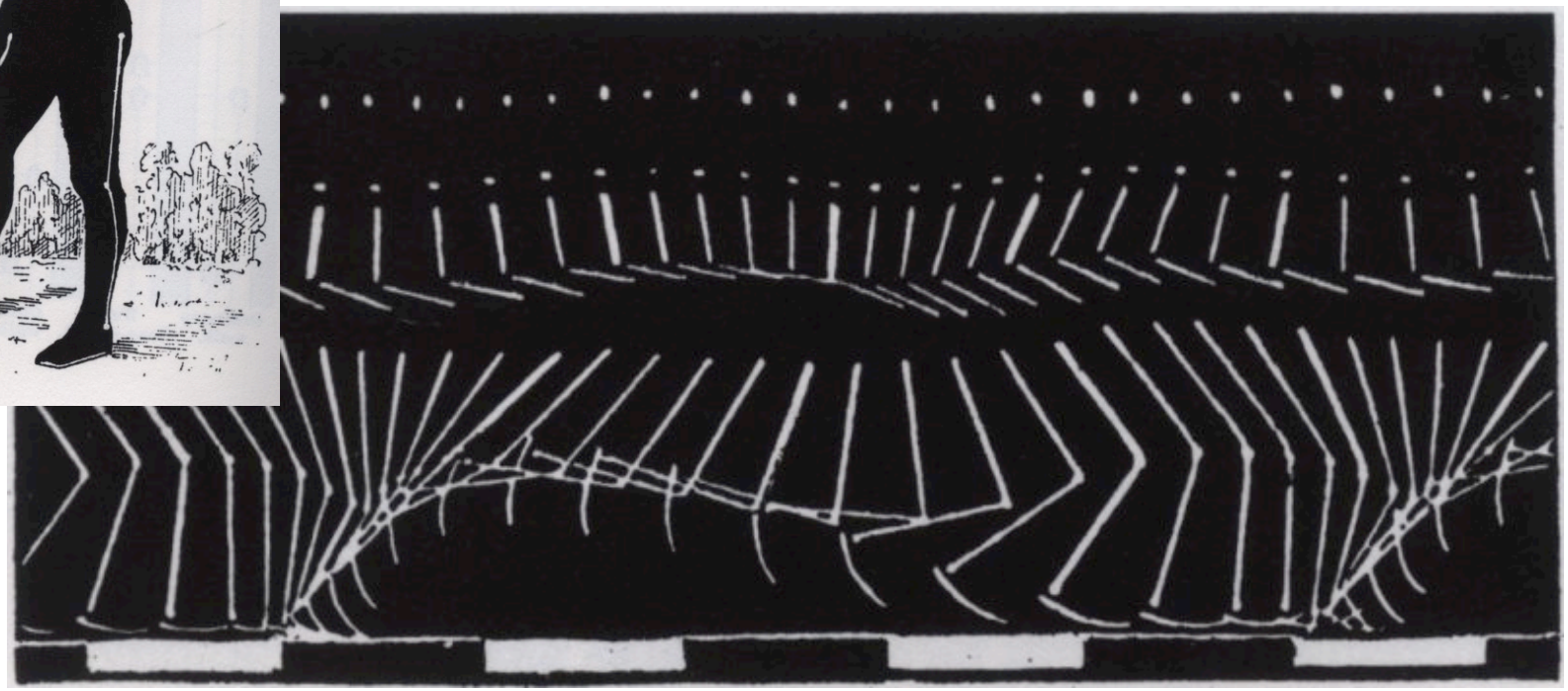


12. If it has been done well in the past, think of another way to do it.

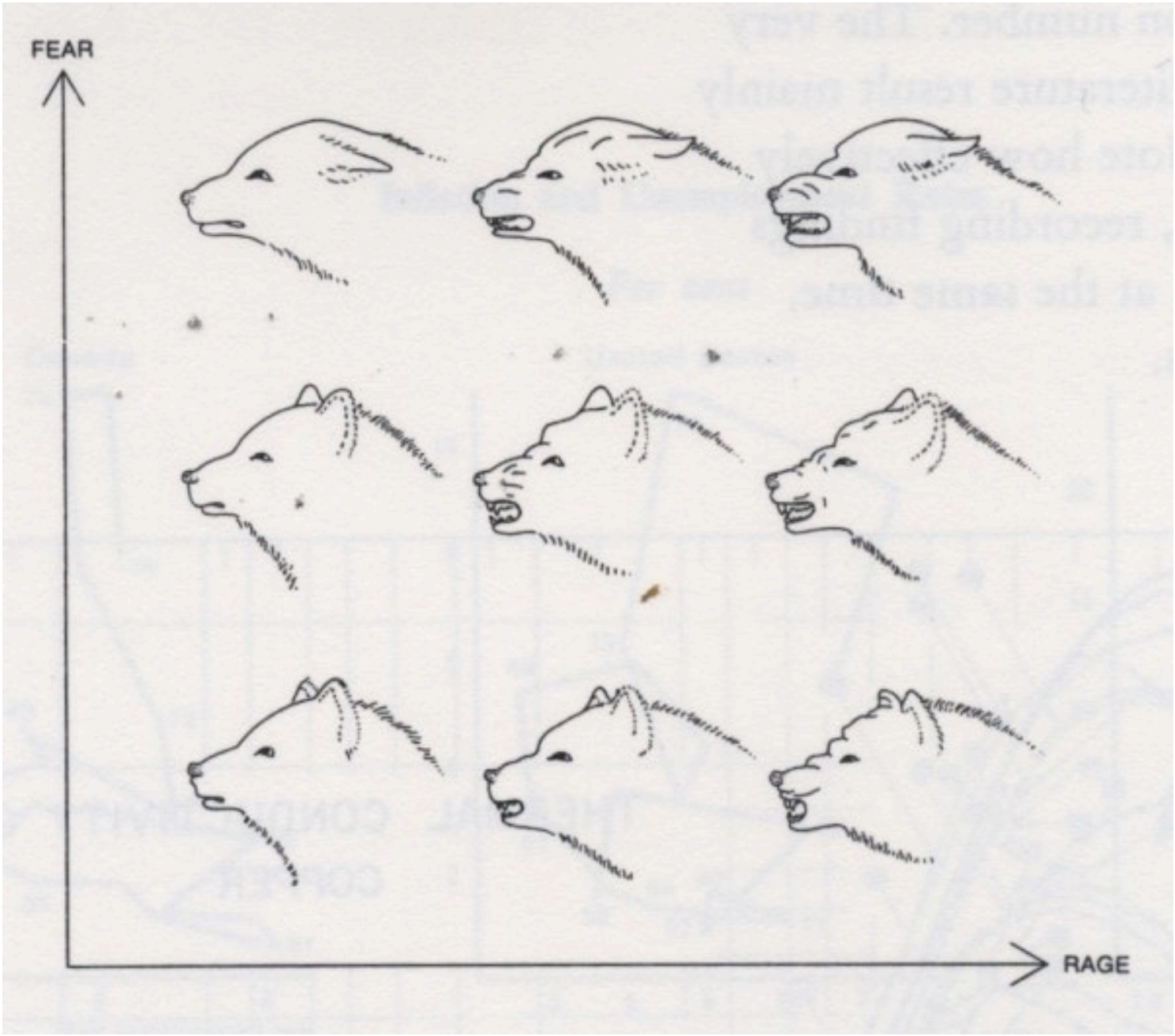
On the other hand, here are some creative plotting techniques you may want to consider.

1. Letting the data points represent another variable.



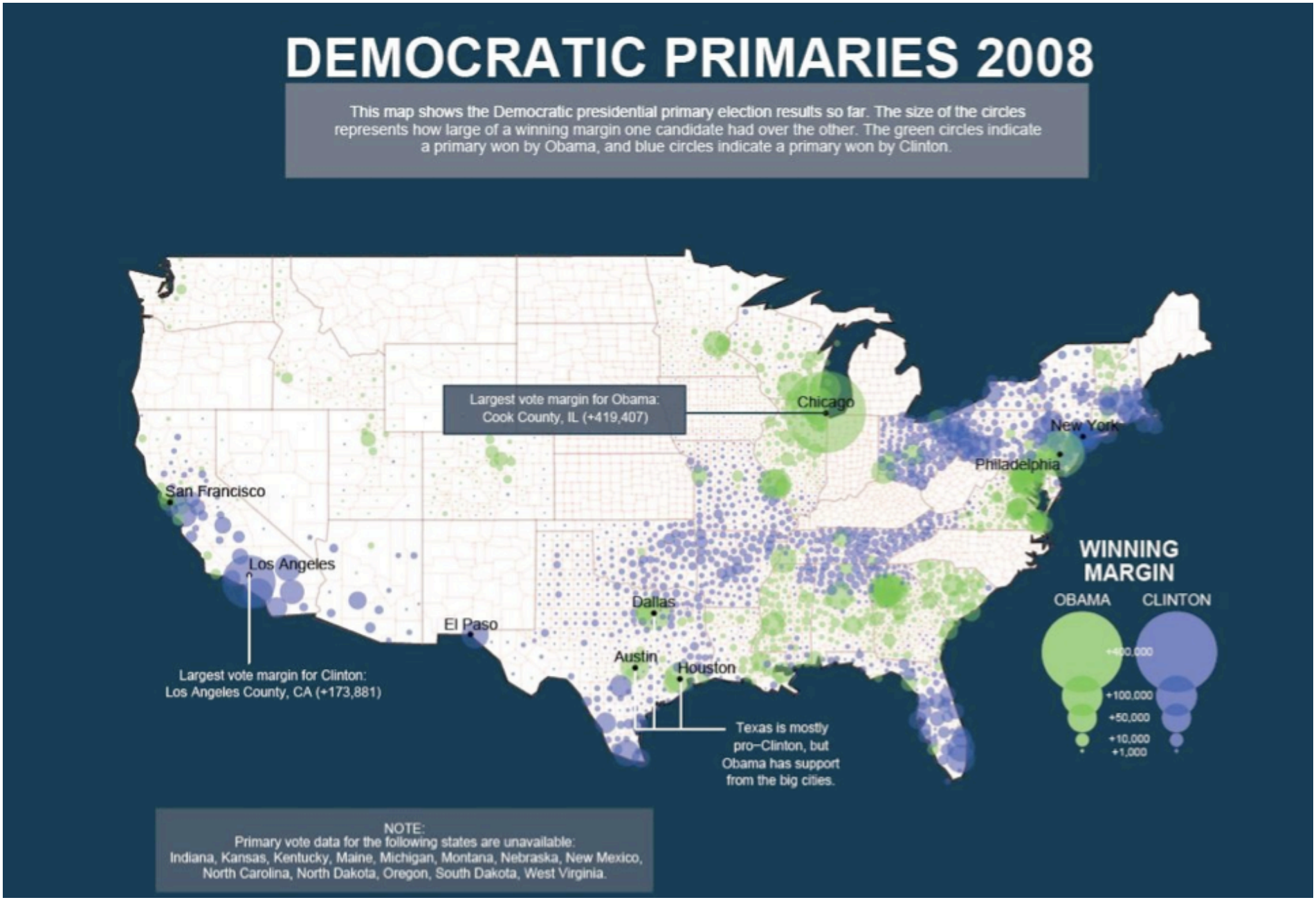


(E.J. Marey)


FEAR
RAGE

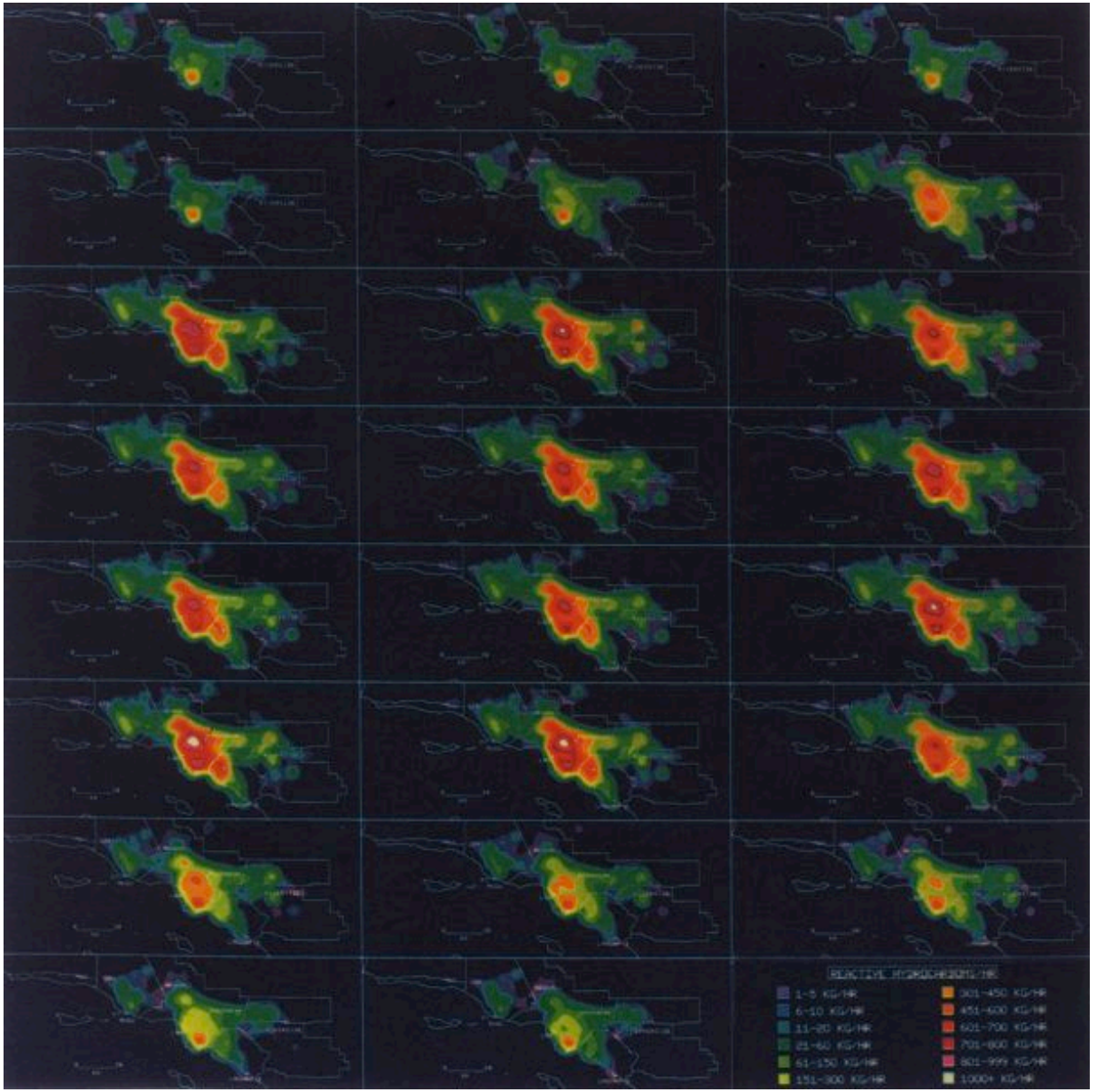
DEMOCRATIC PRIMARIES 2008

This map shows the Democratic presidential primary election results so far. The size of the circles represents how large of a winning margin one candidate had over the other. The green circles indicate a primary won by Obama, and blue circles indicate a primary won by Clinton.

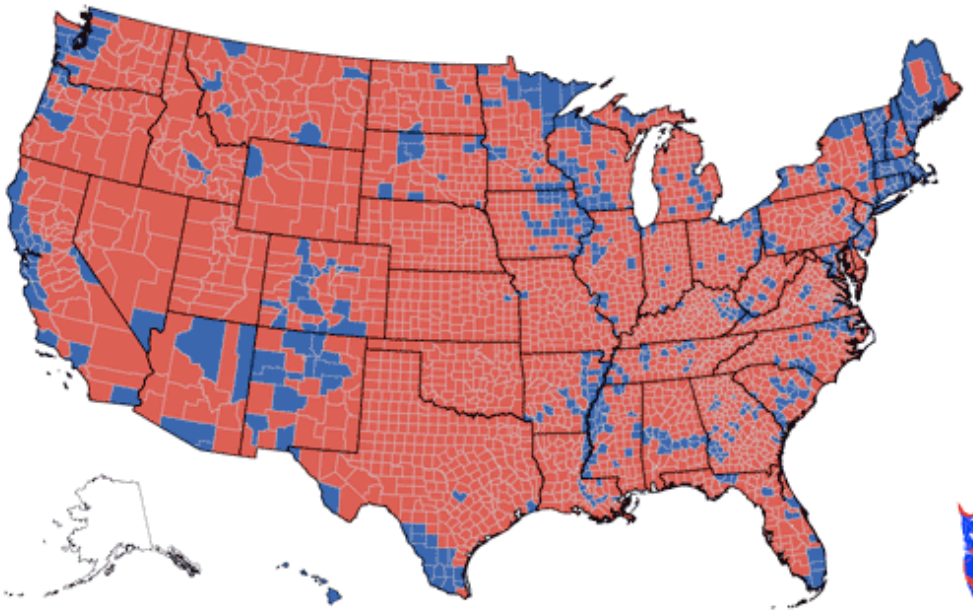


NOTE:
Primary vote data for the following states are unavailable:
Indiana, Kansas, Kentucky, Maine, Michigan, Montana, Nebraska, New Mexico, North Carolina, North Dakota, Oregon, South Dakota, West Virginia.

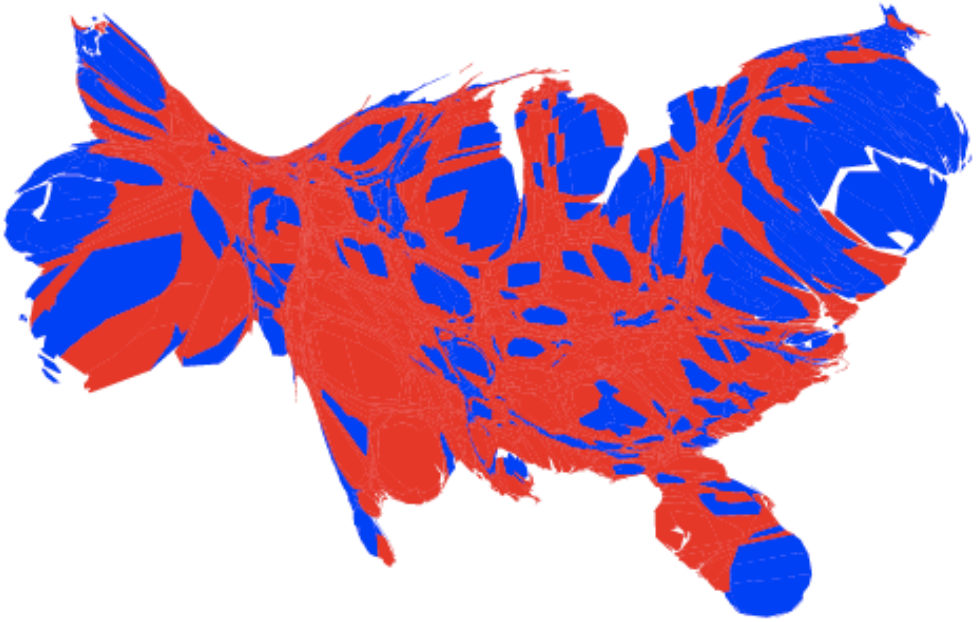
2. Using “small multiples”



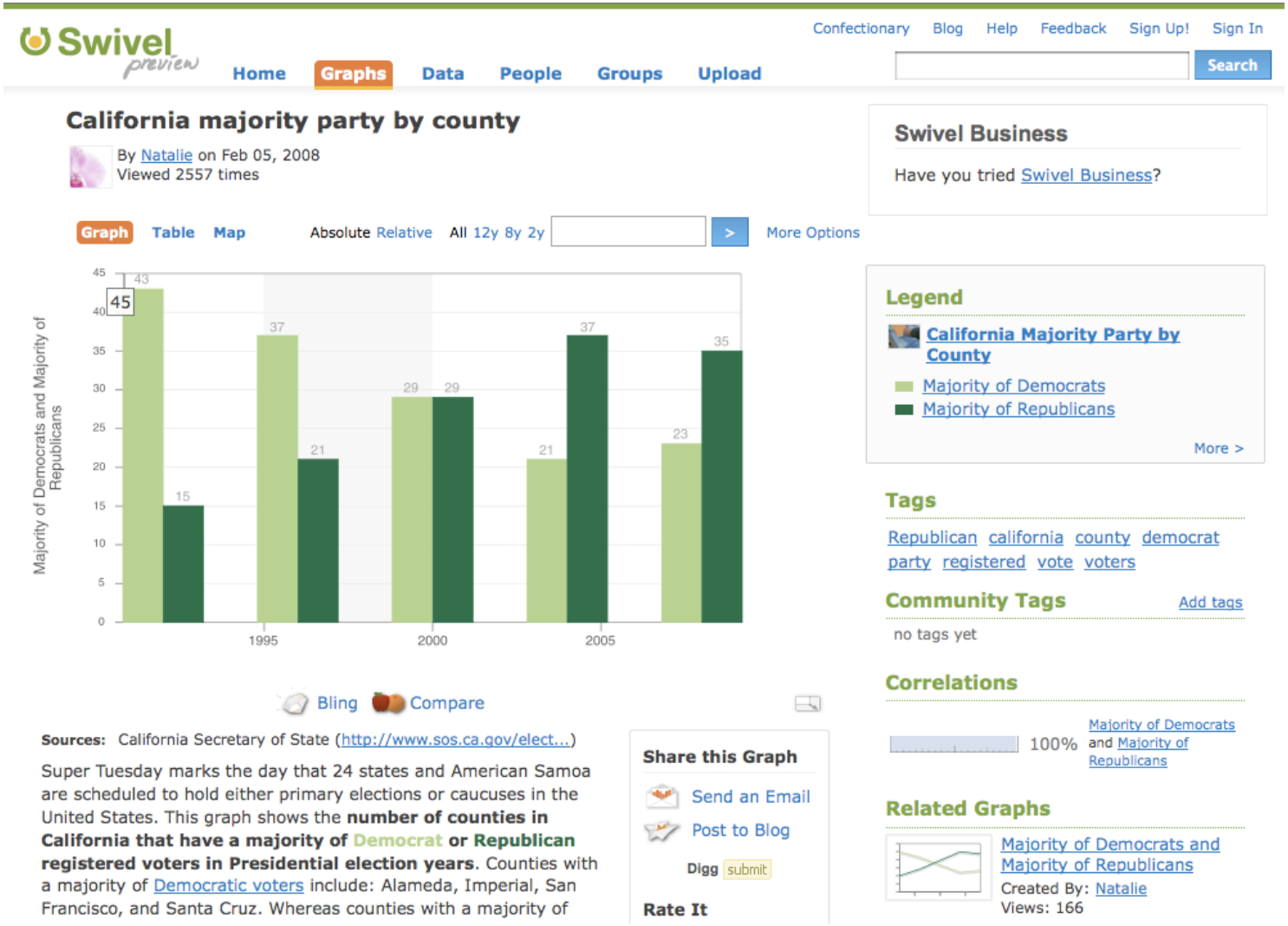
3. Letting deformation represent a variable.

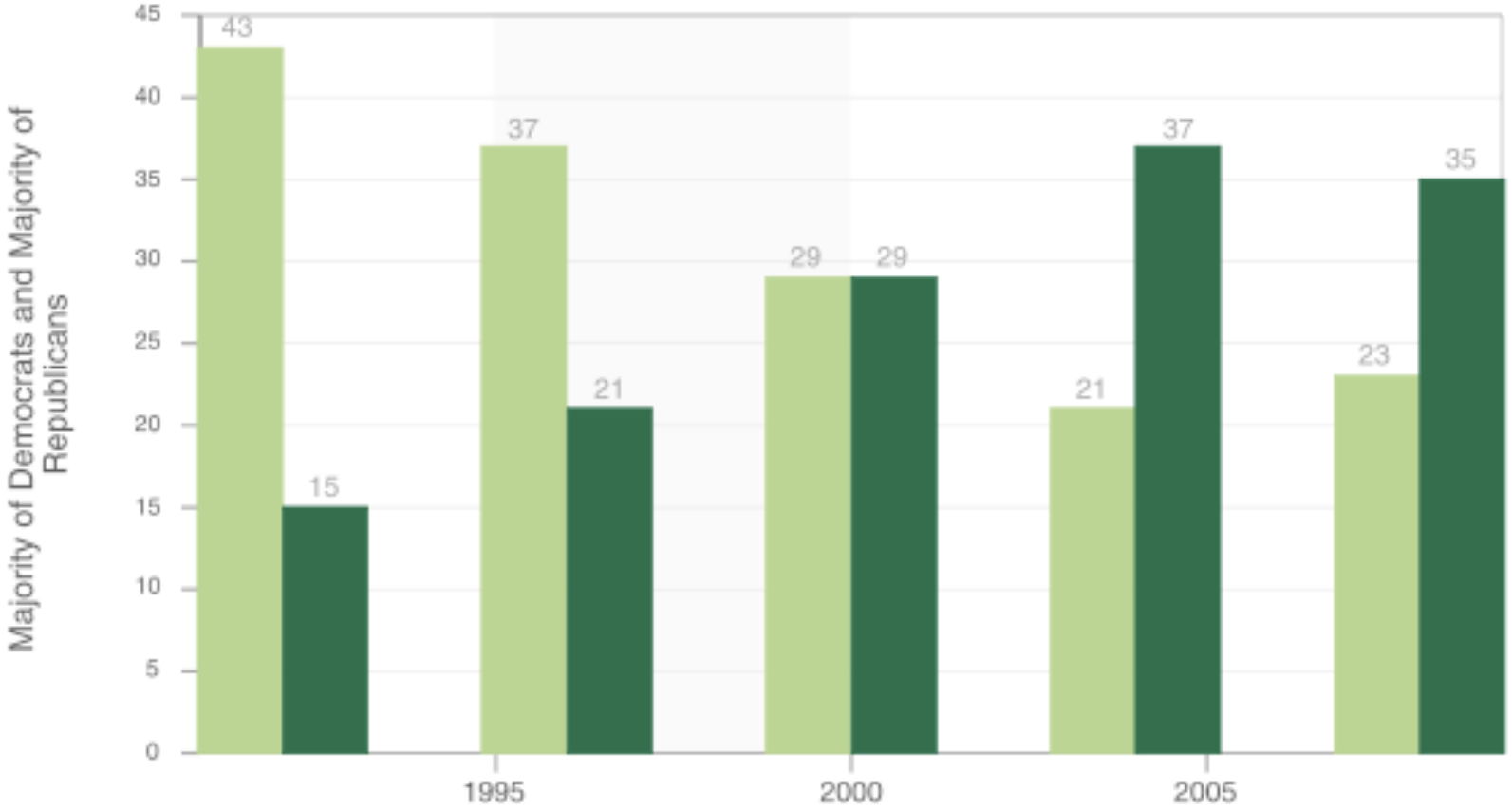


versus



An example from www.swivel.com



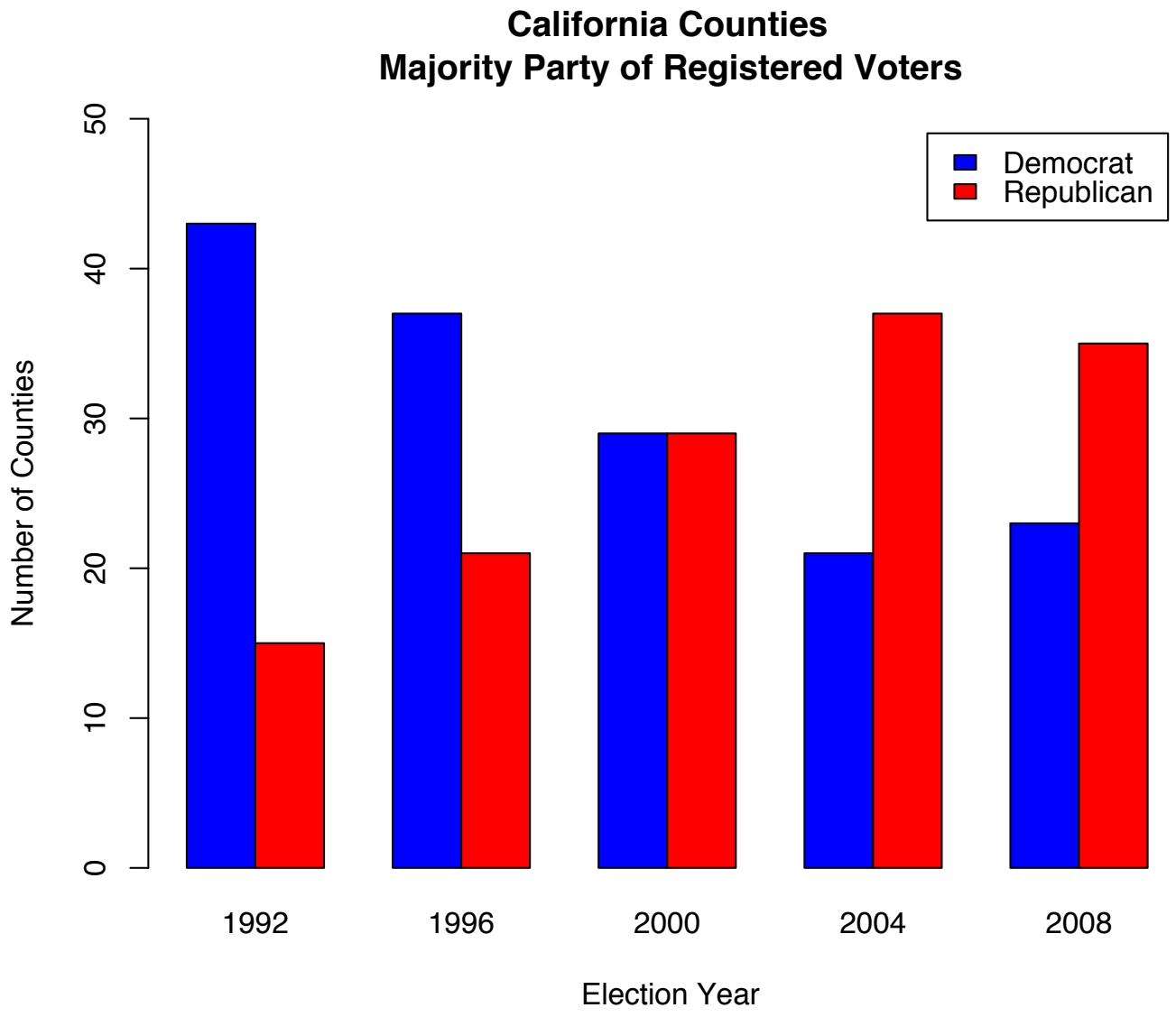


Critique:

x-axis labels poorly located - put them at election years

y-axis label misleading - these are numbers of counties

use of color could be improved (eg. red/blue)


California Counties
Majority Party of Registered Voters
Democrat
Republican
Number of Counties
0
10
20
30
40
50
1992
1996
2000
2004
2008
Election Year