36-350: Data Mining

Handout 8
September 22, 2003

Visualizing general numeric data

This week's dataset is 93 cars from the 1993 model year, each described by 11 attributes:

Type	Small, Sporty, Compact, Midsize, Large, or Van
Price	Midrange Price (in $1,000)
MPG.highway	Highway miles per gallon by EPA rating
EngineSize	Engine size (liters)
Passengers	Passenger capacity (persons)
Length	Length (inches)
Wheelbase	Wheelbase (inches)
Width	Width (inches)
Weight	Weight (pounds)

The first five rows:

```
                Type Price MPG.highway EngineSize Horsepower
Acura Integra Small  15.9          31        1.8        140
Dodge Colt    Small   9.2          33        1.5         92
Dodge Shadow  Small  11.3          29        2.2         93
Eagle Summit  Small  12.2          33        1.5         92
Ford Festiva  Small   7.4          33        1.3         63
              Passengers Length Wheelbase Width Turn.circle Weight
Acura Integra          5    177       102    68          37   2705
Dodge Colt             5    174        98    66          32   2270
Dodge Shadow           5    172        97    67          38   2670
Eagle Summit           5    174        98    66          36   2295
Ford Festiva           4    141        90    63          33   1845
```

We want to "mine" this data for patterns, such as:

- Attribute values which tend to appear together (trends and clusters).
- Attribute values which tend *not* to appear together (voids and anomalies).

As with text and images, the first task is to develop an appropriate distance measure between cars. We will use **invariance** as a guide.

Standardizing

We want our results to be **invariant** to the units used to represent the attributes (pounds, inches, etc.). In other words, we want invariance to scaling. Similarly, we want to be invariant to any simple transformation of an attribute, like length vs. area vs. volume, miles per gallon vs. gallons per mile. This is done by **standardizing** the attributes to have similar distributions.

Some different ways to standardize data:

Rank conversion Replace all values with their rank in the dataset. This is invariant to any monotonic transformation, including scaling.

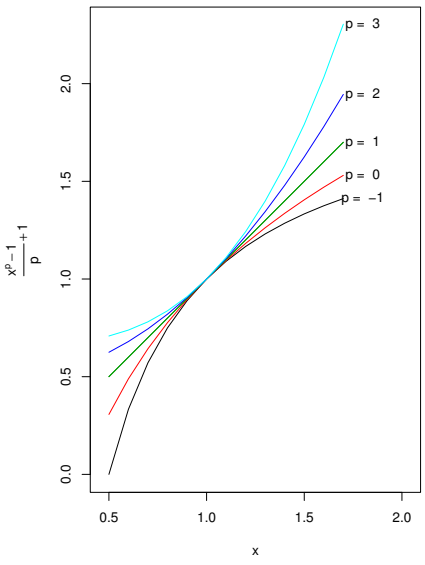
Transform for symmetry Raise each attribute to the power which makes its distribution symmetric. This can handle most monotonic transformations, without being as extreme as rank conversion. Also known as Box-Cox transformation.

Scale to equalize variance Divide all attributes by their standard deviation. This is invariant to changes in units but not other transformations.

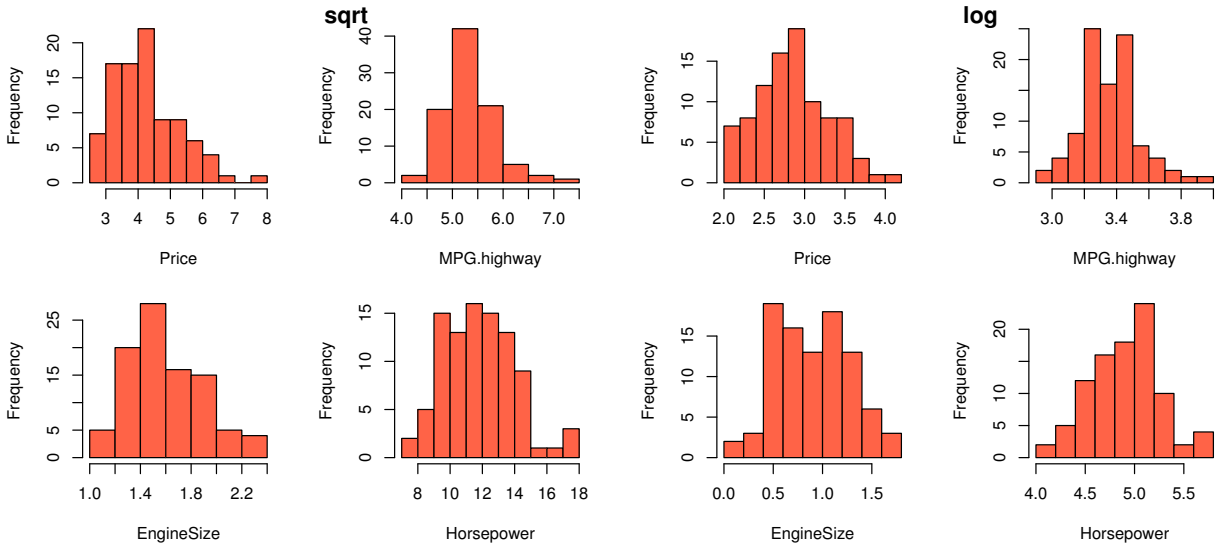
Whitening Scale and subtract attributes from each other to make the variances 1 and covariances 0. This is invariant to taking linear combinations of the attributes.

Box-Cox power family:

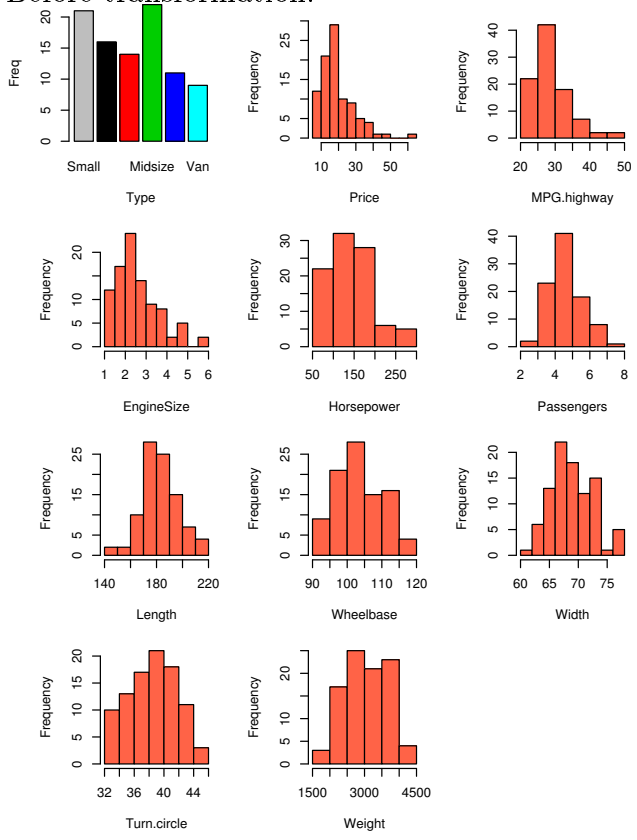
$$y = \begin{cases} \frac{x^p - 1}{p} + 1 & \text{if } p \neq 0 \\ \log(x) & \text{if } p = 0 \end{cases}$$



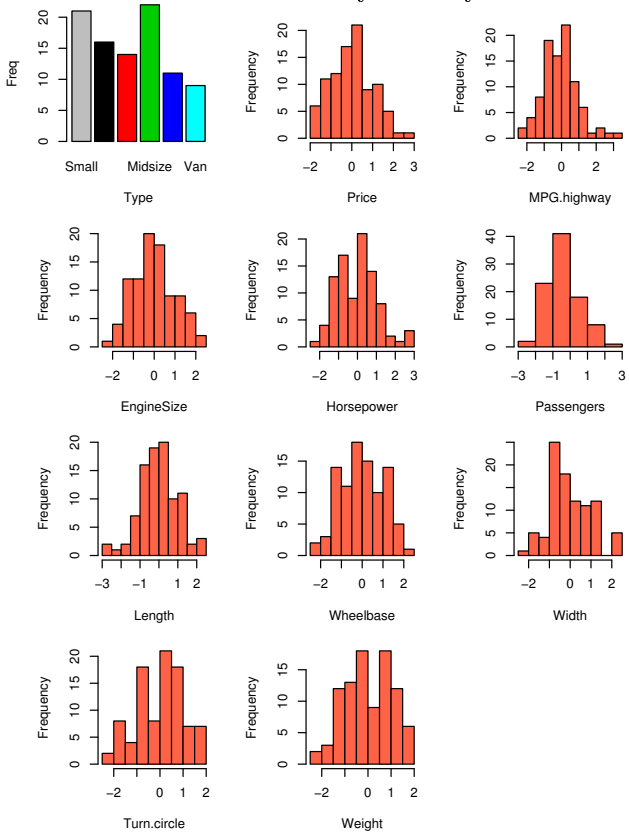
Skew rule—If the distribution is skewed right, use a smaller value of p (push down high values). If skewed left, use a larger value of p (push up high values).



Before transformation:



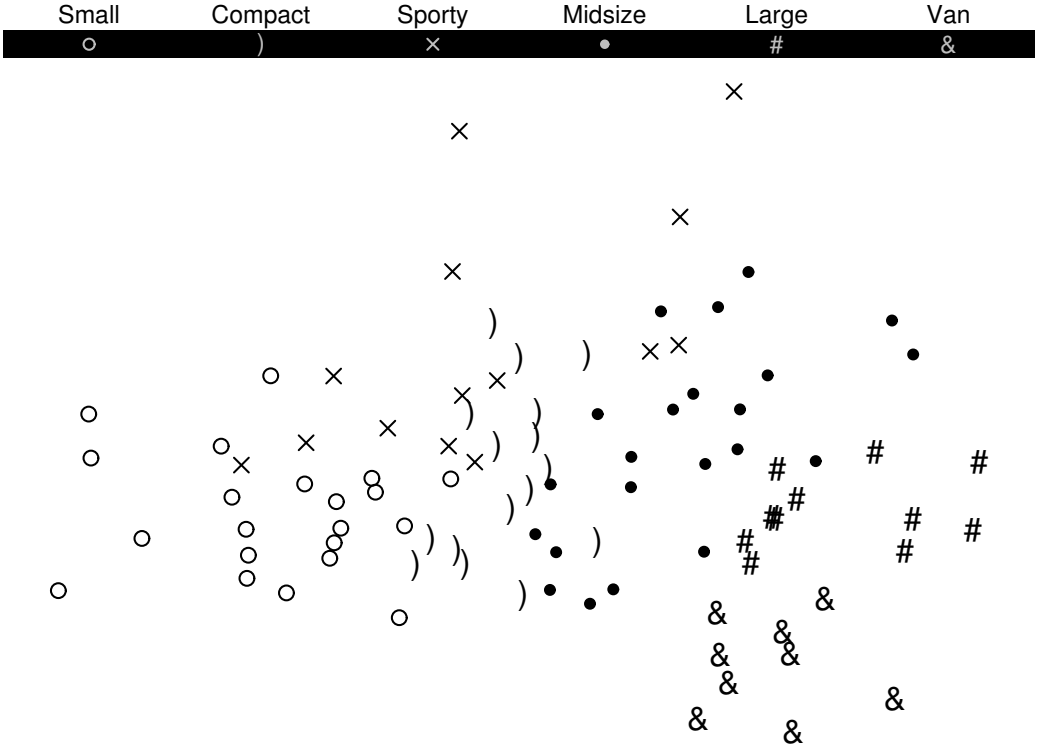
After transformation for symmetry:



Similarity searching errors over all 93 cars (want cars of same Type):

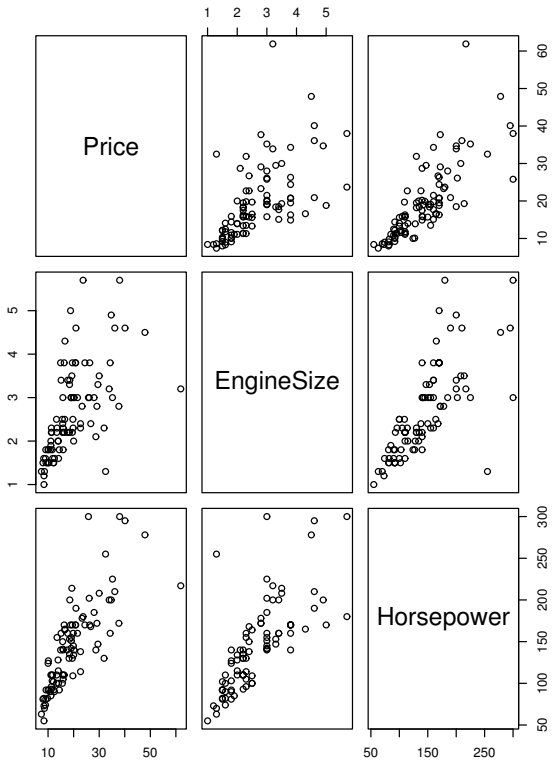
Standardization	Errors
None	42
Scaling only	19
Transformation	18
Ranks	16

Multidimensional scaling after transformation:

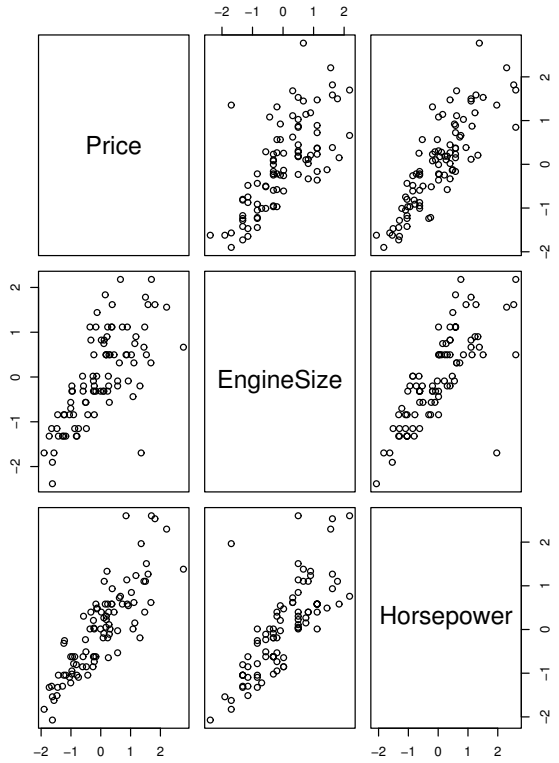


Transformations also tend to make plots easier to read.

Before transformation:



After transformation for symmetry:



The **scatterplot matrix** has consistent scales, so points can be tracked from panel to panel. For example, you can see a car (the Mazda RX-7) that has an unusually high Price for its EngineSize but a normal Price for its Horsepower. (It happens to have an unusually small engine.)

A larger scatterplot matrix shows that all of the variables are positively correlated, except for MPG which is negatively correlated with all others. The clusters and other structure that we want to find are not obvious from this visualization.

Projection—Reducing a dataset of many dimensions to fewer dimensions (usually two), by taking a weighted sum. Geometrically, it is like a photo of a high-dimensional point cloud onto two-dimensional 'film,' where the information perpendicular to the 'film' is squashed away. Picking two dimensions, as in the scatterplot matrix, is a special case of projection.

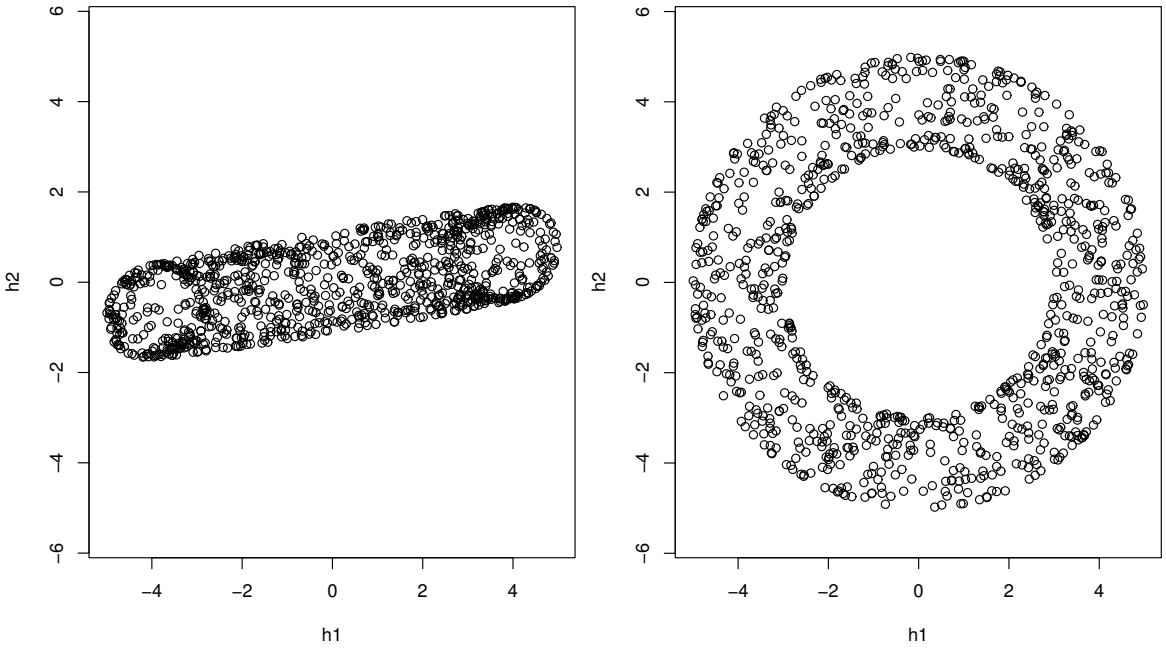
Mathematically, projection is a matrix multiply: (**w** is weights)

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

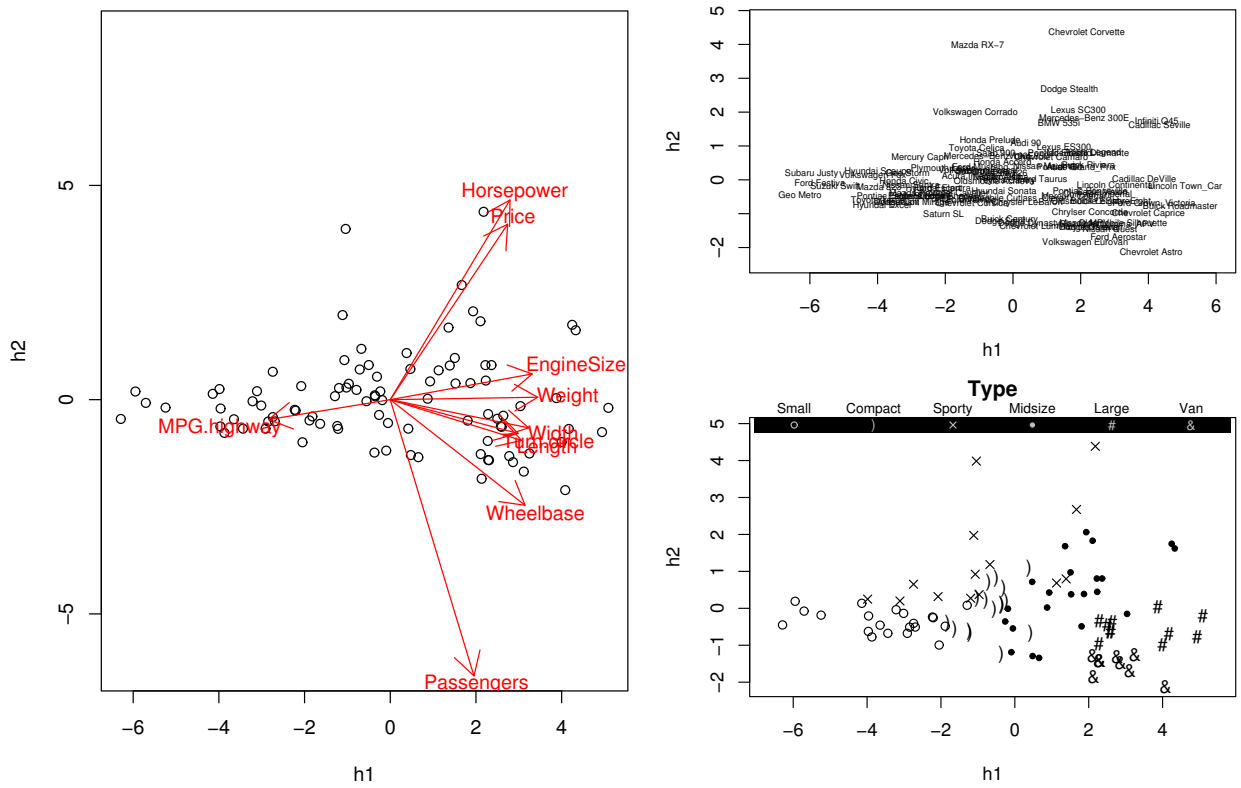
$$h_1 = x_{11}w_1 + x_{12}w_2 + x_{13}w_3$$

$$h_2 = x_{21}w_1 + x_{22}w_2 + x_{23}w_3$$

Two different projections of a 3D torus:



Principal Component Analysis (PCA) is a projection that maximizes the variance of the projected data. This tends to give much more information than pairwise scatterplots, especially when attributes are highly correlated.



The weight matrix for this projection:

	h1	h2
Price	0.29	0.43
MPG.highway	-0.30	-0.05
EngineSize	0.35	0.06
Horsepower	0.30	0.49
Passengers	0.21	-0.68
Length	0.33	-0.10
Wheelbase	0.33	-0.26
Width	0.34	-0.07
Turn.circle	0.32	-0.08
Weight	0.37	0.01

The result is very similar to MDS—why?

Here the dimensions have meaning, as a linear combination of the original attributes. The arrows are the projection of a unit vector pointing along each original axis. Changing the attribute value of a car will move its projection in that direction.

References

[1] John Chambers, William Cleveland, Beat Kleiner, and Paul Tukey. *Graphical Methods for Data Analysis.* Wadsworth, 1983.

[2] F. Mosteller and J. W. Tukey. *Data Analysis and Regression.* Addison-Wesley, 1977.