36-350: Data Mining

Lab 5
Date: September 26, 2003 **Due: end of lab**

Interspersed throughout this lab are questions that you will have to answer at check-off.

1. Download the files for this lab from the course web page to the desktop:

   `http://www.stat.cmu.edu/~minka/courses/36-350/lab/`

2. Open a Word or Notepad document to record your work.

Start R

3. `Start -> All Programs -> Class software -> R 1.7.0`

4. Load the special functions for this lab:

   `File -> Source R code...`

   Browse to the desktop and pick `lab5.r` (it may have been renamed to `lab5.r.txt` when you downloaded it). Another window will immediately pop up for you to pick the `mining.zip` file you downloaded.

The dataset

5. The dataset is the 50 states, each described by seven demographic statistics:

Income	per capita income (1974)
Illiteracy	percent of population illiterate (1970)
Life.Exp	life expectancy in years (1969–71)
Homocide	murder rate per 100,000 population (1976)
HS.Grad	percent high-school graduates (1970)
Frost	mean number of days with minimum temperature below freezing (1931–1960)
Density	Population density per square mile as of July 1, 1975

   Load this data via

   `data(States)`

   This defines a matrix called `States`.

Standardizing

6. Make a histogram of the state variables. *Two of the variables should be transformed with a logarithm. Which two are they?*

7. Transform the variables and make another histogram to check that it worked. Then standardize the variables to have zero mean and unit variance.

Scatterplots

8. Make a scatterplot matrix involving only the four variables `Income`, `Illiteracy`, `HS.Grad`, and `Density`. Keep a copy for doing the homework.

9. Make a scatterplot of `Illiteracy` versus `HS.Grad`, including state names and a trend line on top. *Some states are far from the trend—they have an unusually high illiteracy rate given the amount of high school graduates they have. Which states are these?*

PCA projection

10. Project the state data into two dimensions using PCA. Note the R^2 value. Make one plot with dots and another with state names, axis arrows on both. Scale the names so they are readable. Keep a copy for the homework *and hand in your plots with the homework.*

11. You can now get checked off.

Histograms If `x` is a matrix then

```
hist(x)
```

will show histograms of each column. If `x` is a vector, it shows one histogram.

Transformation Column `i` of a matrix `x` can be transformed using one of the following:

```
x[,i] = x[,i]^2
x[,i] = log(x[,i])
```

The first line squares all values, the second line takes the logarithm. `i` can be the name or number of a column, or a vector of names or numbers (to transform several columns at once).

Standardizing If `x` is a matrix,

```
x = scale(x)
```

will change each column to have zero mean and unit variance.

Scatterplot matrix If `x` is a matrix,

```
pairs(x)
```

shows a matrix of pairwise scatterplots.

Scatterplot with labels To make an individual scatterplot of a column named `v2` versus `v1`, use one of the following:

```
plot(v2 ~ v1, x)
text.plot(v2 ~ v1, x)
text.plot(v2 ~ v1, x, cex=0.6, asp=1)
```

The first uses dots, the second uses labels. The parameter `cex` be can given to either command, to specify how big to make the dots or labels. Also you can give the parameter `asp` to specify the aspect ratio (**required** for PCA projections.)

Trend line A model for `v2` versus `v1` can be estimated via one of:

```
fit = lm(v2 ~ v1, x)
fit = smooth(v2 ~ v1, x)
```

The first fits a straight line, the second fits a smooth curve. Either fit can be plotted via one of:

```
model.plot(fit)
model.plot(fit,add=T)
```

The first makes a new plot while the second adds to an existing plot.

PCA If `x` is a matrix, the PCA combination weights can be computed via

```
w = pca(x,k)
```

where k is the desired number of dimensions. To actually project the data using these weights (i.e. a matrix multiply):

```
px = project(x,w)
```

`px` is the new data matrix, with k columns. `px` can be plotted using `plot` or `text.plot`, but you must use `asp=1` to make it look right. To overlay the original axes as arrows:

```
plot.axes(w)
```