36-350: Data Mining

Lab 5
Date: September 27, 2002 **Due: end of lab**

---

1 Introduction

This lab teaches you the steps involved in visualizing multivariate data.
There are 5 questions. For each one, submit your commands and a response from R demonstrating that they work. (Only hand in commands relevant to the question.) To submit a plot, click on the plot window and select

```
File -> Save as -> Postscript...
```

This saves the plot to a file which can be printed, incorporated into a Word document, or mailed to us as an attachment.

2 Starting R

Start R as in lab 1. On the class web page, go to "computer labs" and download the files for lab 5 into your work folder. Read the special functions into your running R application via the commands

```
source("lab5.r")
```

If this fails, check that the files were downloaded correctly.

3 The data

The dataset used in this lab is the 50 states, each described by seven numerical 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) in capital or large city

Density Population density per square mile as of July 1, 1975

Load this data via

```
load("States.rda")
```

This defines a variable called `states`, which can be accessed like a matrix. For example,

```
states[1:4,]
```

is the first 4 rows, and

```
states[,1:4]
```

is the first 4 columns.

4 Standardizing

Notice that the variables are all measured on different scales. So the first thing to do is transform and standardize the variables so that they are more comparable. A histogram of each variable will tell you which need to be transformed. The command for this is `hist.data.frame`:

```
hist.data.frame(states)
```

> **Question 1:** Two of the variables have a skewed distribution and need to be transformed with a logarithm to make them more symmetric. Which two are they? Submit code to do the transformation. Let the variable `i` contain the names of the two variables. For example, if they were Income and Homocide, then `i` would be
>
> ```
> i <- c("Income","Homocide")
> ```
>
> The data can then be transformed via
>
> ```
> states[,i] <- log(states[,i])
> ```
>
> (You can check the result by making another histogram. The distributions should be more symmetric.)

After transformation, the variables need to be standardized with zero mean and unit variance. This is done with one command, `scale`:

```
sx <- scale(states)
```

Now `sx` contains the data you should work with.

5 Scatterplots

The command `pairs` displays all pairwise scatterplots:

```
pairs(sx)
```

This gives a quick idea of which variables are correlated, and which are not. (You can use it to check your homework answers.) Several of these scatterplots have interesting features. See for example `Illiteracy` versus `Density` and `Illiteracy` versus `HS.Grad`.

> **Question 2:** Use `pairs` to make a figure of pairwise scatterplots involving only the four variables `Income`, `Illiteracy`, `HS.Grad`, and `Density`. (Hint: Start by setting `i` to their names.) Turn in this plot and keep a copy for doing the homework.

An individual scatterplot can be made using the `plot` command you used in lab 3. The usage is

```
plot(x,y,xlab="x label",ylab="y label")
```

where `x` and `y` are vectors of values.
Instead of dots, the states can be plotted with their names, using the similar command `text.plot`. The only difference from `plot` is that you also provide labels and a character expansion factor `cex`:

```
text.plot(x,y,labels,cex=0.75,xlab="x label",ylab="y label")
```

Here `x`, `y`, and `labels` are vectors of the same length. `x` and `y` are numeric vectors, describing position, and `labels` is a vector of text strings. (Recall the function `rownames`.)
A new command for this lab is `lowess`, which adds a trend line to the plot. It is used as follows:

```
lines(lowess(x,y),col=2)
```

where `x` and `y` are the same as in `plot` or `text.plot`.

> **Question 3:** (a) Use `text.plot` to make a properly labeled, readable scatterplot of `Illiteracy` versus `HS.Grad`, using state names and a trend line. (b) 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? (Extra credit, if you've finished the rest of the lab: Give a plausible explanation. Are high schools graduating illiterates?)

6 PCA projection

The PCA projection for a dataset `sx` is computed via

```
w <- pca(sx,k)
```

This gives a matrix of variable weights. Here `k` is the number of dimensions you want to project the data onto (should be 2 for visualization).

> **Question 4:** What variable has the most weight in the first dimension (`h1`)? What variable has the most weight in the second dimension (`h2`)?

The matrix `w` is then applied to the data via

```
px <- project(sx,w)
```

giving a reduced dataset `px` with only `k` dimensions. The projected data can be plotted with

```
plot(px[,1],px[,2],asp=1)
```

The special argument `asp=1` makes the aspect ratio equal to 1, so that angles are properly represented. The variable axes are then plotted with

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

Instead of dots, the states can be plotted with their names, by using `text.plot` instead of `plot`. (Ignore any warnings about "asp".)

> **Question 5:** Plot the projected data with state names and axes. Scale the names so they are readable. Turn in the plot and keep a copy for the homework.