# 36-350: Data Mining

**Lab 7**
**Date: October 11, 2002** **Due: end of lab**

## 1 Introduction

This lab teaches you the steps involved in visualizing the predictive relationship between variables in a dataset.
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 7 into your work folder. Read the special functions into your running R application via the commands

```
source("lab5.r")
source("lab6.r")
source("lab7.r")
```

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

## 3 The data

The dataset used in this lab is 506 neighborhoods in Boston, each described by 11 characteristics:

| | |
|---|---|
| Crime | per capita crime rate |
| Industry | proportion of non-retail business acres |
| Pollution | nitrogen oxides concentration (parts per 10 million) |
| Rooms | average number of rooms per dwelling |
| Old | proportion of owner-occupied units built prior to 1940 |
| Distance | weighted mean of distances to five Boston employment centres |
| Highway | index of accessibility to radial highways |
| Tax | full-value property-tax rate per $10,000 |
| Student.Teacher.Ratio | student-teacher ratio |
| Low.Status | percent of the population which is 'lower status' |
| Price | median value of owner-occupied homes in $1000 |

Load this data via

```
load("Housing.rda")
```

This defines a matrix called `housing`. Individual columns of this matrix can be accessed like so:

```
housing[,"Rooms"]
housing[,c("Price","Crime")]
```

## 4 Standardizing

As in lab 5, the first step is to standardize the variables. Histograms will tell you which variables need to be transformed:

```
hist.data.frame(housing)
```

> **Question 1:** The variables `Crime`, `Distance`, `Low.Status`, and `Pollution` all have skewed distributions and need to be transformed. Three require logarithm and one requires square root. Find out which ones and submit code to do the transformations. It should look something like this:
>
> ```
> x <- housing
> i <- c("A","B","C")
> x[,i] <- log(x[,i])
> i <- "D"
> x[,i] <- sqrt(x[,i])
> ```
>
> (You can check the result by making another histogram. The distributions should be more symmetric.)

After transformation, standardize the variables to have zero mean and unit variance:

```
sx <- scale(x)
```

Now `sx` has the standardized data you should work with.

## 5 PCA projection

Recall the commands for PCA projection:

```
w <- pca(sx,(final number of dimensions))
px <- project(sx,w)
plot(px[,1],px[,2],asp=1)
plot.axes(w,cex=0.8)
```

In this lab, you want to color the data according to `Price`. This is done with the command `color.plot` from lab 6:

```
color.plot(x,y,f,asp=1)
```

Here `x`, `y`, and `f` are vectors of the same length. `x` and `y` are numeric vectors, describing position, and `f` is a factor, describing subgroups to color.
Instead of a factor for `f`, you can also provide a numeric vector, such as a vector of prices. `color.plot` will automatically divide the numbers into quantiles and color them accordingly.

> **Question 2:** Make a PCA plot, with neighborhoods colored according to `Price`. Make sure the aspect ratio is 1 so that the arrows for variables have the correct angles. Turn in the plot and keep a copy for the homework. (Be careful about printing: color plots can be difficult to read when printed in black and white. You won't get credit for an unreadable plot.)

## 6 Contour plots

A color plot can be used to get a detailed understanding of how a response depends on two predictors. For example, if you want to understand how `Price` depends on `Rooms` and `Low.Status`, you would type

```
color.plot(sx[,"Rooms"],sx[,"Low.Status"],sx[,"Price"])
```

(This plot was used in class.) Note that the colors should always correspond to the response, not a predictor.
The boundaries between Price groups can be highlighted by adding surface contours to the plot. This is a two-step process: fit a surface to the data, then plot the contours. To fit a surface, use `loess`:

```
fit <- loess(Price ~ Rooms + Low.Status,sx)
```

The first argument to `loess` is what R calls a *formula*, and it describes what kind of model you want. In this case, we want to predict `Price` by some function of `Rooms` and `Low.Status`. The names in the formula are looked up the second argument, which here is `sx`. Here are some more examples of formulas:

```
Crime ~ Distance
Pollution ~ Distance + Highway + Industry
```

The first formula wants to model `Crime` as a function of `Distance`. The second wants to model `Pollution` as a function of `Distance`, `Highway`, and `Industry`. A formula can have any number of predictors. But for a contour plot, there should be exactly two.
`fit` now contains the fitted surface. It is given to `color.plot` to make a new plot with contours on top:

```
color.plot(fit,nlevels=4)
```

`nlevels` specifies how many contour lines to draw.

> **Question 3:** Make a contour plot which shows how `Price` depends on `Distance` and `Low.Status`. Use 8 contour lines. Turn in the plot and keep a copy for the homework.

An alternative to the contour plot is to make pairwise scatterplots with lowess curves on top. This command, similar to `loess`, will make those plots:

```
predict.plot(Price~Rooms+Low.Status,sx)
```

> **Question 4:** Make pairwise scatterplots which show how `Price` depends on `Distance` and `Low.Status`. Scale the window so that the plots are not too stretched out. Turn in the plot and keep a copy for the homework.

## 7 Discriminative projection

To see how Price depends on many predictors, you can take a projection of the predictors. The projection should be chosen so that Price groups are separated. This is done with the `projection` function from lab 6:

```
w <- projection(sx,f,k=2)
```

`sx` is the data to compute the projection from. Before, `f` was a factor describing the groups you want to separate. Now `f` will be a numeric vector, which is automatically divided according to quantiles. The result is a projection matrix, just like from `pca`.
Note that the projection should be computed using only the *predictors* in `sx`:

```
sx.pred <- sx[,1:10]
```

Otherwise, the projection will trivially pick `Price` for predicting `Price`.

> **Question 5:** (a) Project the predictors down to 2 dimensions, `h1` and `h2`. Turn in the projection matrix `w`, rounded for easy reading:
>
> ```
> round(w,1)
> ```
>
> (b) Make a contour plot showing how `h1` and `h2` predict `Price`. This can be done by combining the various functions in this lab. (Hint: start by making a color plot, and then convert it to a contour plot.) Show the original predictors as arrows on the plot and make sure the aspect ratio is 1. Turn in the plot and keep a copy for the homework.