Principal Components and Factor Analysis: An Example

36-350, Data Mining

1 October 2008

1 Data: The United States *circa* 1977

The `state.x77` data set is available by default in R; it's a compilation of data about the US states put together from the 1977 *Statistical Abstract of the United States*, with the actual measurements mostly made a few years before.[1] It's a little long in the tooth now but handy for teaching purposes.

The variables are:

Population	in thouusands
Income	dollars per capita
Illiterarcy	Percent of the population
Life Exp	Years of life expectancy at birth
Murder	Number of murders and non-negligent manslaughters per 100,000 people
HS Grad	Percent of adults who were high-school graduates
Frost	Mean number of days per year with low temperatures below freezing
Area	In square miles

You can do help(state.x77) for a little more detail. There is also a `state.center` data set, giving the longitude and latitude of the geographic center of each state (except for Alaska and Hawaii, which are artificially put somewhere off the west coast) — see Figures 1 and 2.

[1] The *Statistical Abstract* is "the best book published in America" (P. Krugman), an immensely valuable compilation of data about a huge range of aspects of American life, put out every year by the Census Bureau. It's available for free online at `http://www.census.gov/compendia/statab/`.

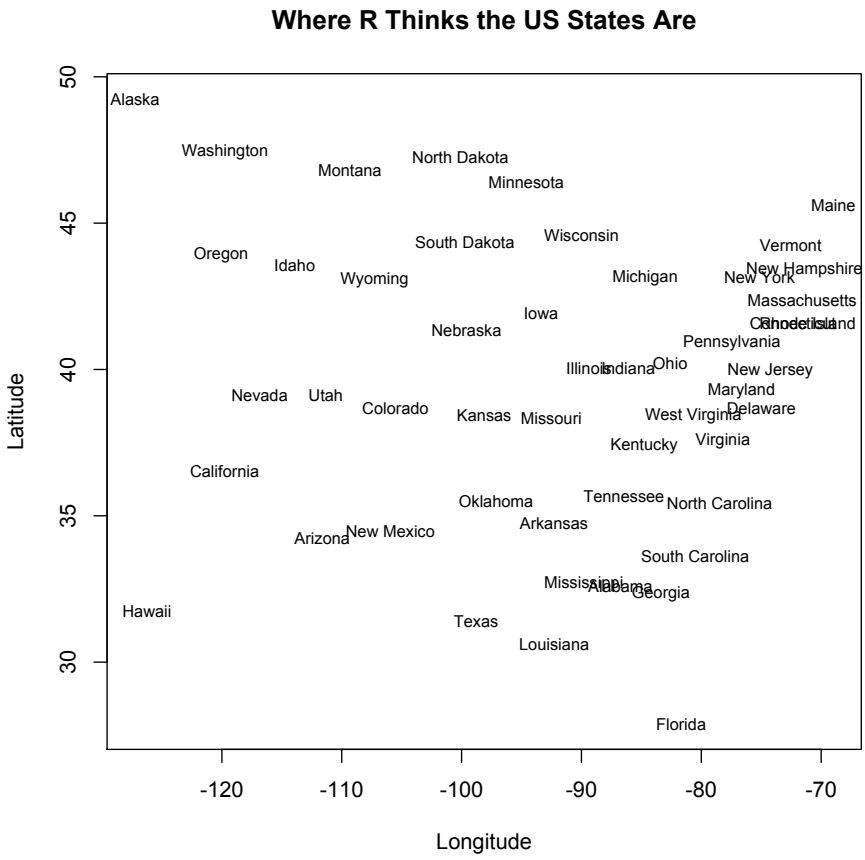

Figure 1: Geographic centers of the US states, per R's built-in `state.center` data set. N.B. Alaska and Hawaii are artificially brought close to the lower 48.

```
plot.new() # Start up a new plot
# How big is the plotting window? Set ranges from the data we'll be graphing
plot.window(xlim=range(state.center$x),ylim=range(state.center$y))
# Put the name of each state at its center. Shrink text 25% so there's less
# overlap between the names.
text(state.center,state.name,cex=0.75)
# Add the horizontal axis at the bottom and the vertical at the left.
axis(1)
axis(2)
# Draw a box.
box()
# Add the titles
title(main="Where R Thinks the US States Are",xlab="Longitude",ylab="Latitude")
```

Figure 2: R code for drawing Figure 1.

2 Principal Components

The command `prcomp` is the preferred command for principal component analysis in `R`. It performs a singular value decomposition directly on the data matrix; this is slightly more accurate than doing an eigenvalue decomposition on the covariance matrix but conceptually equivalent. A naive approach here is unrewarding.

```
> prcomp(state.x77)
Standard deviations:
[1] 8.532765e+04 4.465107e+03 5.591454e+02 4.640117e+01
[5] 6.043099e+00 2.461524e+00 6.580129e-01 2.899911e-01
```

. . . followed by more output specifying the principal components. The standard deviations here are the square roots of the eigenvalues. (EXERCISE: why are the *square roots* standard deviations?) To see this another way, use the `plot` command on the output of `prcomp`:

```
plot(prcomp(state.x77),type="l")
```

The output is Figure 3.

The thing to notice is that the first eigenvalue is immensely larger than the others. Naively, one might think this means that there's just one component here and our job is done. In reality, this is because we've chosen units where one variable is immensely larger than the others, so it varies much more.

```
> apply(state.x77,2,sd)
  Population       Income   Illiteracy     Life Exp
4.464491e+03 6.144699e+02 6.095331e-01 1.342394e+00
      Murder      HS Grad        Frost         Area
3.691540e+00 8.076998e+00 5.198085e+01 8.532730e+04
```

In other words, typical variations in area from state to state are orders of magnitude larger than anything else. And in fact if we look at the principal components, the first one is giving us area, and everything else is noise. We can see this by doing a `biplot` of the data and the original features together (Figure 4).

```
biplot(prcomp(state.x77),cex=(0.75,1))
# Plot original data points and feature vectors against principal
# components; shrink data-point names by 25% for contrast/legibility
```

Things are a lot better if we standardize the variables to have variance 1. `prcomp` will do this for us. (See Figure 5.)

```
biplot(prcomp(state.x77,scale.=TRUE),cex=c(0.5,0.75))
```

Looking at the plot, several things are striking.

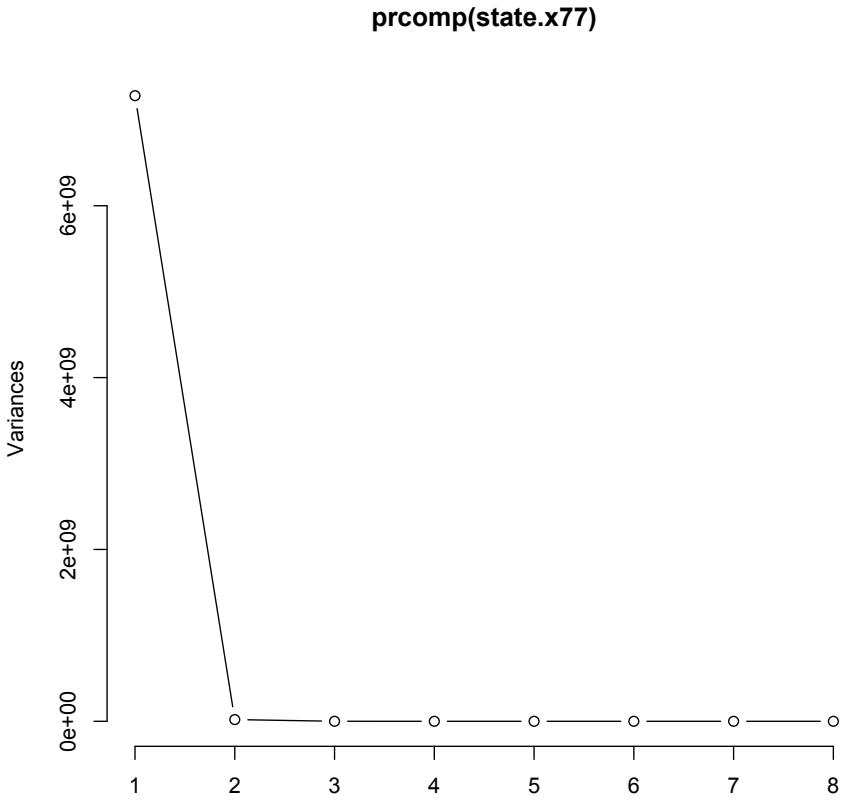

Figure 3: Scree plot for PCA of the unscaled `state.x77` data, created using.

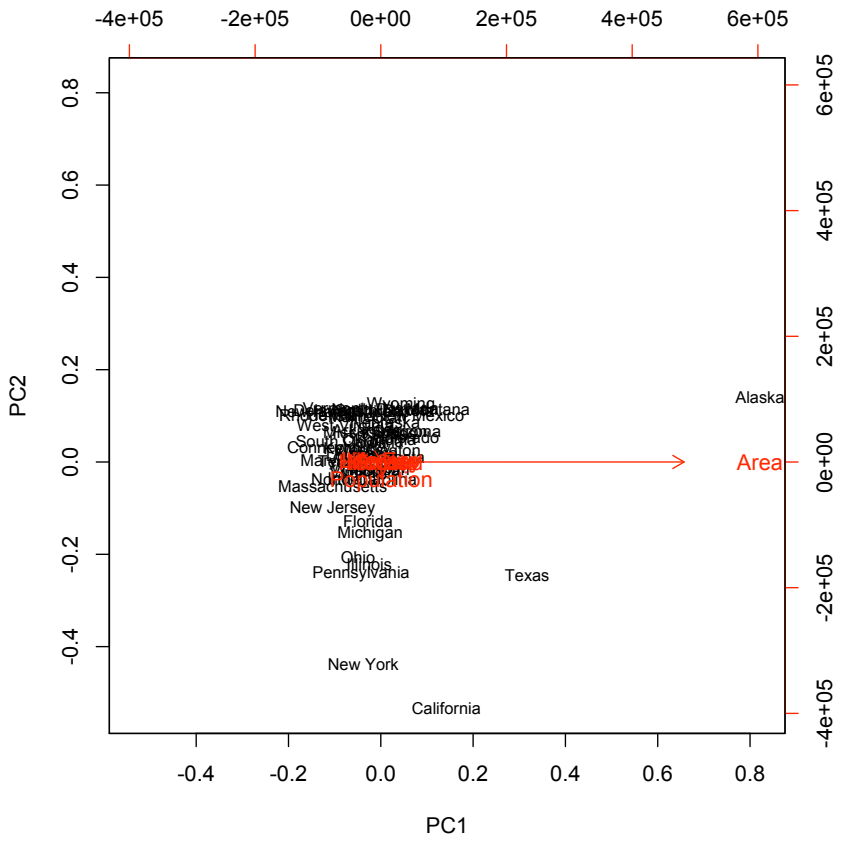

Figure 4: PCA plot of the state.x77 data set without scaling. Note how the first principal component is basically just area; in the units chosen, this is orders of magnitude more variable than anything else.

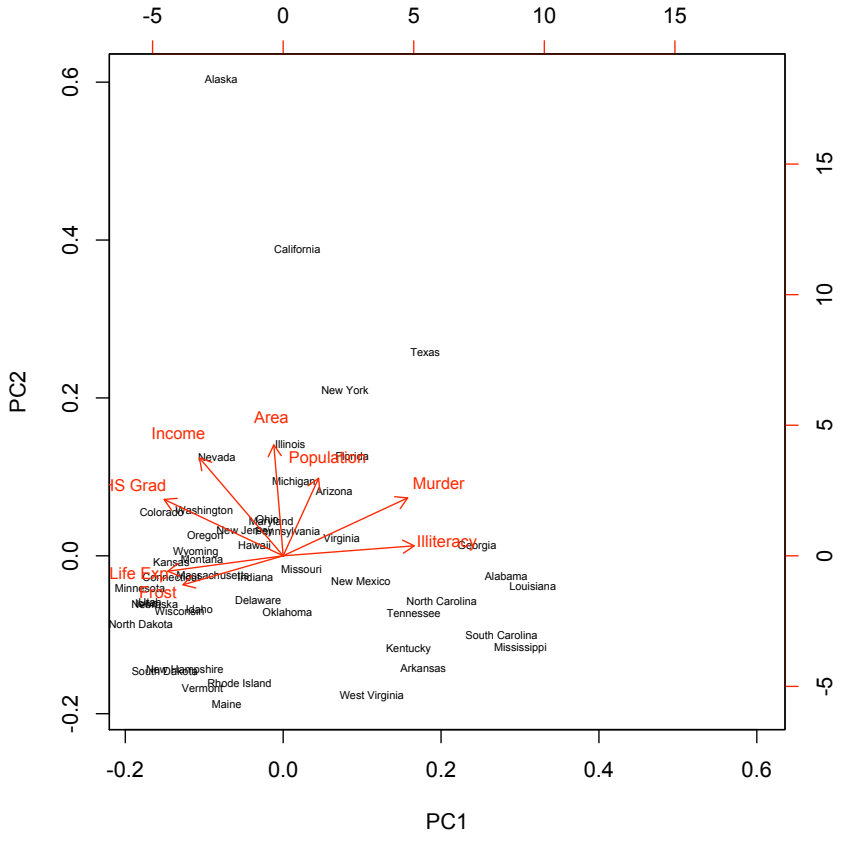

Figure 5: PCA plot of the state.x77, rescaling features to have variance 1. The first principal component distinguishes between cold, educated, long-lived states with low violence from warm, ill-educated, shorted-lived, murderous states, which also tend to be poorer. The second principal component most distinguishes big, populous, rich states from small, poor ones, which have some tendency to be less murderou and a very weak tendency to be warmer.

1. The first principal component, the one which summarizes the most variance, is very nearly the same as the illiteracy rate (on the positive dimension) or life expectancy (going the other way), and almost the same as the number of frost days per year. The murder rate projects very strongly on to the positive direction of the first principal component. Put a bit crudely, PC1 distinguishes between cold states with educated, harmless, long-lived populations, and warm, ill-educated, short-lived, violent states.

2. The second PC correlates extremely closely with area — states really do vary hugely in their area, even when you standardize. It also strongly correlates with population and income, and more weakly with secondary education and murder. More weakly, it negatively correlates with life expectancy and frost. The second PC distinguishes big rich educated states from small poor ignorant states, which tend to be a bit warmer, and less murderous.

3. The two leading components are *not* independent — high values of PC2 imply a very narrow range of PC1 values.

4. Alaska was something of an outlier; Hawaii, on the other hand, was as close to being typically American as anywhere. (Make of this what you will.)

Looking at the scree plot, there is no obvious point at which it levels off or otherwise breaks; you could argue for one principal component or five.

The fact that PC1 is so close to the "frost" variable suggests that part of what's being picked up here is the difference between the South and the rest of the country.[2] There are two ways we could try to follow up on that. One would be to include a *categorical* variable for the region. Unfortunately, PCA only works with numerical, metrical values. We could try to finesse *that* by assigning numbers to the categories, and people do that sometimes, but the results become totally hostage to which numbers we use — i.e., under our control, and not in a good way. There *are* models which will let us combine both variable types, but it will take us a while to develop them properly.

The other approach is to throw in latitude and longitude as variables. We'd need both, because, e.g., New Mexico isn't part of the South, while Kentucky is. Let's make that data.

```
> state.x77.centers = cbind(state.x77,state.center$x,state.center$y)
> colnames(state.x77.centers) = c(colnames(state.x77),"Longitude","Latitude")
```

The first line creates a new array by adding on the longitudes as latitudes as columns. The second names the columns, copying over the column names from `state.x77`. Let's run this through PCA (Figure 7).

```
> biplot(prcomp(state.x77.centers,scale.=TRUE),cex=c(0.5,0.8))
```

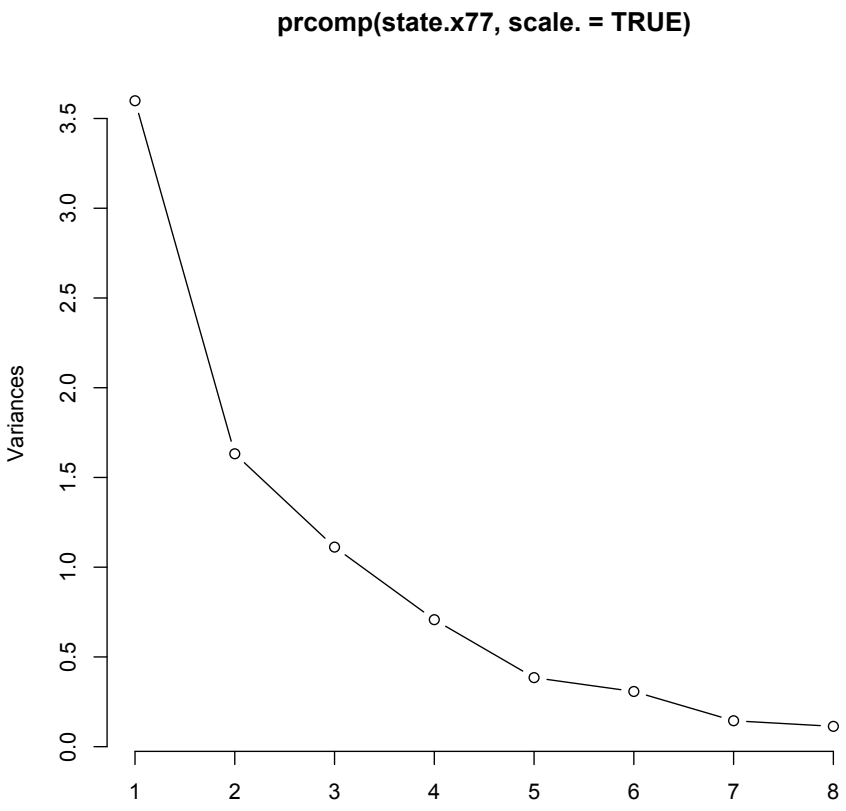

Figure 6: Screen plot for the `state.x77` data, with features scaled to variance 1. Notice the absence of a really sharp break. *Arguably* there is a break at 5.

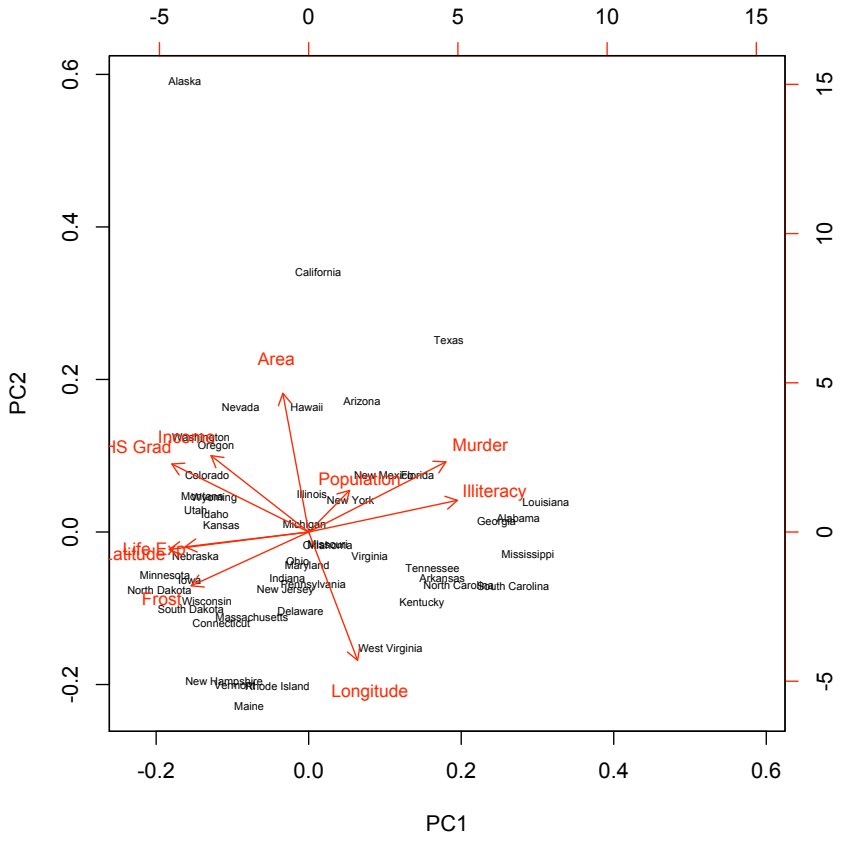

Figure 7: PCA plot when latitude and longitude are added as variables.

The first principal component is still nearly (but not quite) the life-expectancy-and-frost/illiteracy axis. But it's also nearly the latitude/illiterarcy axis. (The correlation between frost days and latitude is, not surprisingly, stronger than the correlation between latitude and life expectancy, but the latter have more similar projections on to the first two components!) Murder rate still strongly projects on to this axis, with the opposite sign from latitude, as one might expect. Longitude is measured in negative degrees east of the prime meridian, i.e., larger longitude is more easterly, and we see that there is a negative relationship between longitude and area — eastern states tend to be smaller than western states. This is now the second principal component.

One last tweak before setting PCA down. We might suspect that some of the features here aren't related to population or area on their own, but rather with population density — people per square mile. Population density is a nonlinear transformation of two of the features, so it can't really be faked with PCA. (EXERCISE: Why not?) However, nothing stops us adding it on our own (Figure 8).

```
> states.density = cbind(state.x77,
                         state.x77[,"Population"]/state.x77[,"Area"],
                         state.center$x,state.center$y)
> colnames(states.density) = c(colnames(state.x77),"Density","Longitude",
                               "Lattitude")
> biplot(prcomp(states.density,scale.=TRUE),cex=c(0.5,0.75))
```

The first component hasn't changed very much, but now the second component is very nearly density! In fact, because the first component is very nearly latitude and the second longitude, is this *very* much like a rotated version of the state map we started with. (Give the PCA plot a quarter-turn clockwise.) It's not quite the same — for example, Hawaii is closer to the center than California or Texas! — but it's actually a pretty good match.

Notice that in all our principal component plots, the "frost" and "murder" features have pointed in more or less the opposite directions. When we add in latitude and longitude, murder is very strongly negatively related to latitude, i.e., more southernly states tend to have a higher murder rate. This is interesting, but it would be more interesting if we could say something about why.

Presumably murders do not cause an absence of frost[3], but maybe hot weather makes people lose their tempers? If so, we might expect murder rates to rise across the country with climate change. On the other hand, we can randomize the frost values and see what happens.

```
> states.density.randfrost = states.density
> states.density.randfrost[,"Frost"] = sample(states.density[,"Frost"])
> biplot(prcomp(states.density.randfrost,scale.=TRUE),cex=c(0.5,0.8))
```

[2]It may especially suggest this if you were brought up just below the Mason-Dixon line, while being taught that the Civil War was "treason in defense of slavery."

[3]But see Frazer (1922).

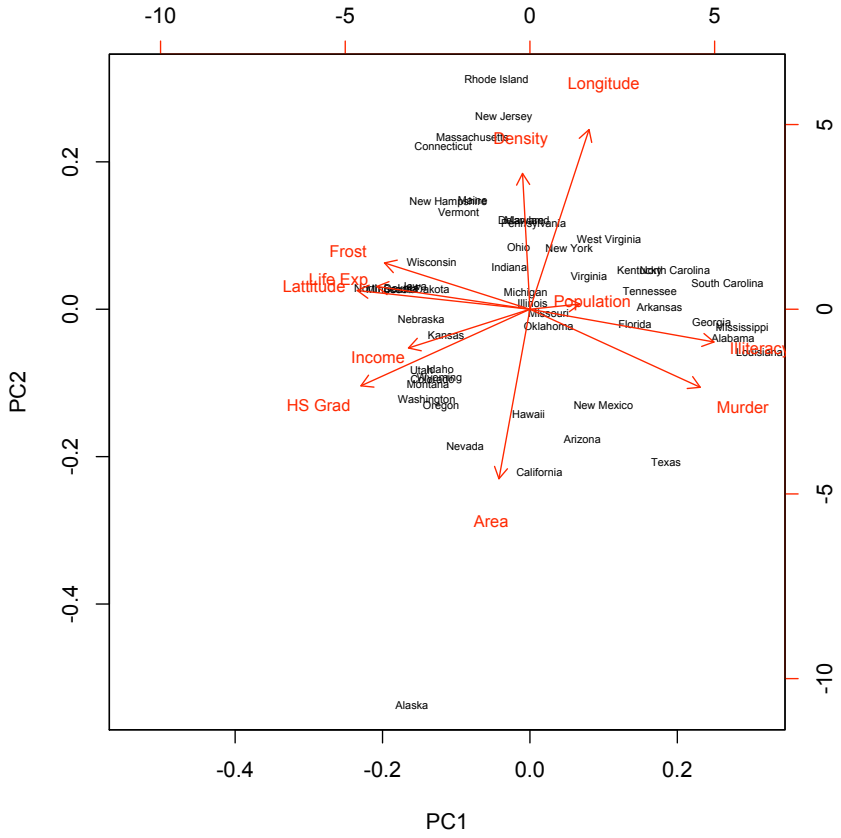

Figure 8: PCA plot including the additional variable of density (inhabitants per square mile). Compare this to the map. — Note that the placement of points here looks fairly random, i.e., the two components are closer to being independent (though they're still not).

This makes a copy of the `states.density` array. Then it randomly permutes the "Frost" column, and repeats the PCA. The R function for sampling from a vector is `sample`, but it defaults to sampling without replacement, and making the sample size as large as the original, which gives a random permutation of the vector. (See `help(sample)`.) We could also randomize by, say, putting in Gaussian values with the same mean and variance, but this is just as easy and does not commit us to distributional assumptions.

What the biplot shows is that randomizing frost doesn't actually have much effect on the first two components, and in particular the homicide rate remains very close to PC1.

Maybe currently-hot places have some other characteristic, not so obvious from the data set, which actually affects homicide?[4] These are the kinds of questions which may be suggested by a method like PCA, but which it can't actually answer.

[4]For an interesting and largely-persuasive conjecture about *why* murder rates are higher in the south, see Nisbett and Cohen (1996).

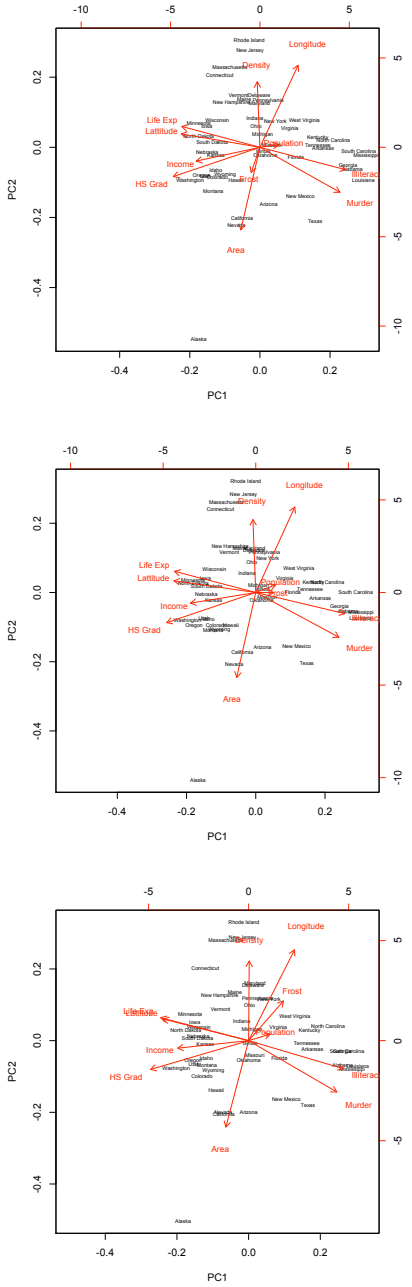

Figure 9: PCA plots with the `Frost` variable randomly permuted, for three different permutations. Note that it typically has a substantial projection on to PC1 and PC2; it's bimodal, so it's easy for fluctuations to make it look reasonably correlated with other variables.

3 Factor Analysis

Now let's do some factor analyses. The R command for maximum likelihood factor estimation is `factanal`.

We'd like to make plots similar to the ones we've made for PCA. Unfortunately the output of `factanal` is not structured in the way that `biplot` likes. Fortunately it's not too hard to bridge them (code in Fiure 10).

We'd also like to make scree plots (Figure 11).

Now let's let rip. We'll start with a single factor.

```
> factanal(states.density,factors=1)

Call:
factanal(x = states.density, factors = 1)

Uniquenesses:
Population     Income Illiteracy   Life Exp     Murder
     0.959      0.768      0.186      0.545      0.358
   HS Grad      Frost       Area    Density  Longitude
     0.505      0.512      0.999      0.998      0.976
 Lattitude
     0.347

Loadings:
           Factor1
Population -0.202
Income      0.481
Illiteracy -0.902
Life Exp    0.674
Murder     -0.801
HS Grad     0.704
Frost       0.699
Area
Density
Longitude  -0.153
Lattitude   0.808

               Factor1
SS loadings      3.847
Proportion Var   0.350

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 216.17 on 44 degrees of freedom.
The p-value is 1.42e-24
```

Let's break this down. We're looking for a single factor which summarizes as much of the correlations among the variables as possible. What we find is

```
# Make a biplot from the output of factanal
# Presumes: fa.fit is a fitted factor model of the
#           type returned by factanal
#           fa.fit contains a scores object
#           fa.fit has at least two factors!
# Inputs: fitted factor analysis model, additional
#         parameters to pass to biplot()
# Side-effects: Makes biplot
# Outputs: None
biplot.factanal <- function (fa.fit,...)
{
# Get the first two columns of scores, i.e.,
# scores on first two factors
x = fa.fit$scores[,1:2]
# Get the loadings on the first two factors
y = fa.fit$loadings[,1:2]
biplot(x,y,...)
}
```

Figure 10: Function for making biplots from the output of `factanal`. If I were a real programmer, I'd add in tests for the presumptions, and tell R to run this function when `biplot` is called with an input of class `factanal`. But I'm not.

```
# Make scree (eigenvalue-magnitude) plots from
# the output of factanal()
# Input: fitted model of class factanal,
#        x-axis label (default "factor"),
#        y-axis label (default "eigenvalue")
#        graphical parameters to pass to plot()
# Side-effects: Plots eigenvalues vs. factor number
# Output: None
screeplot.factanal <- function(fa.fit,xlab="factor",ylab="eigenvalue",...) {
# sum-of-squares function for repeated application
sosq <- function(v) {sum(v^2)}
# Get the matrix of loadings
my.loadings <- as.matrix(fa.fit$loadings)
# Eigenvalues can be recovered as sum of
# squares of each column
evalues <- apply(my.loadings,2,sosq)
plot(evalues,xlab=xlab,ylab=ylab,...)
}
```

Figure 11: Scree-plot function for objects of class `factanal`.

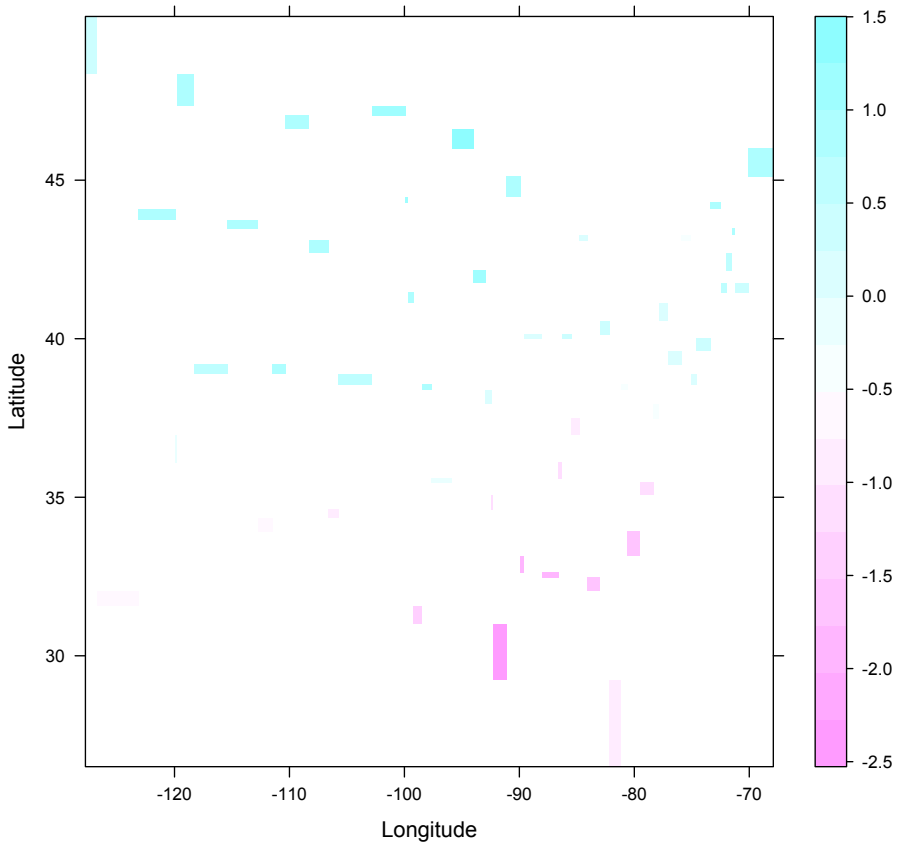

Figure 12: Plot of the factor scores for a one-factor analysis by geographic location.

something with a strong positive relationship with latitude; specifically, a one standard deviation increase in the factor increases latitude by 0.808. There is also a positive relationship with high school graduation rates, income, and frost, though those are all weaker. It has a strong negative relationship with illiteracy (-0.902) and homicide (-0.801), and a weaker negative one with population and longitude. (Geographically, there is less of this factor in the southeast than elsewhere — Figure 12.)

The size of the eigenvalue for this factor is 3.847. This is the amount of variance in the (standarized) features which is summarized by this factor. The total variance is just equal to the number of features (EXERCISE: why?), so we're capturing $3.847/11 = 0.350$ of the variance.

Finally, the last part of the output does a simple goodness-of-fit test, which

```
library(lattice) # For more graphics commands!
state.g = cbind(state.center$x,
                state.center$y,
                factanal(states.density,factors=1,scores="regression")$scores)
colnames(state.g) = c("Longitude","Latitude","G")
levelplot(state.g[,"G"]~state.g[,"Longitude"]*state.g[,"Latitude"],
          xlab="Longitude",ylab="Latitude")
# See help(levelplot) for more
```

is only reliable if we make the assumption that everything is Gaussian and the parameters are estimated by maximum likelihood. The factor model is then a *restriction* of the general multivariate Gaussian model, for which the maximum likelihood estimate is just the empirical covariance matrix. What we know from general statistical theory is that twice the log likelihood ratio between an unrestricted model and a restricted model,

$$\mathcal{R} = 2L(\widehat{\theta}_{\text{unrestricted}}) - 2L(\widehat{\theta}_{\text{restricted}}) \tag{1}$$

has a χ^2 distribution if the restricted model is right and they are both estimated by maximum likelihood. The number of degrees of freedom is the number of extra parameters which are free in the unrestricted model. The actual calculations here are tedious, so it's better to have the computer do them. Basically it amounts to comparing the correlation matrix we'd expect under the fitted model and the one we actually observe.

The upshot in this case is that the fit is *horrible*. So much, then for the one factor model. How about two?

The plotting command is

```
biplot.factanal(factanal(states.density,factors=2,scores="regression"),
                cex=c(0.5,0.8))
```

which delivers Figure 13.

This isn't too far from the principal components plot, but it's also not the same. How are we doing on model fitting?

```
               Factor1 Factor2
SS loadings      3.886   1.962
Proportion Var   0.353   0.178
Cumulative Var   0.353   0.532

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 144.43 on 34 degrees of freedom.
The p-value is 1.45e-15
```

Again, horrible, though we're now accounting for over half the variance.

What about three?

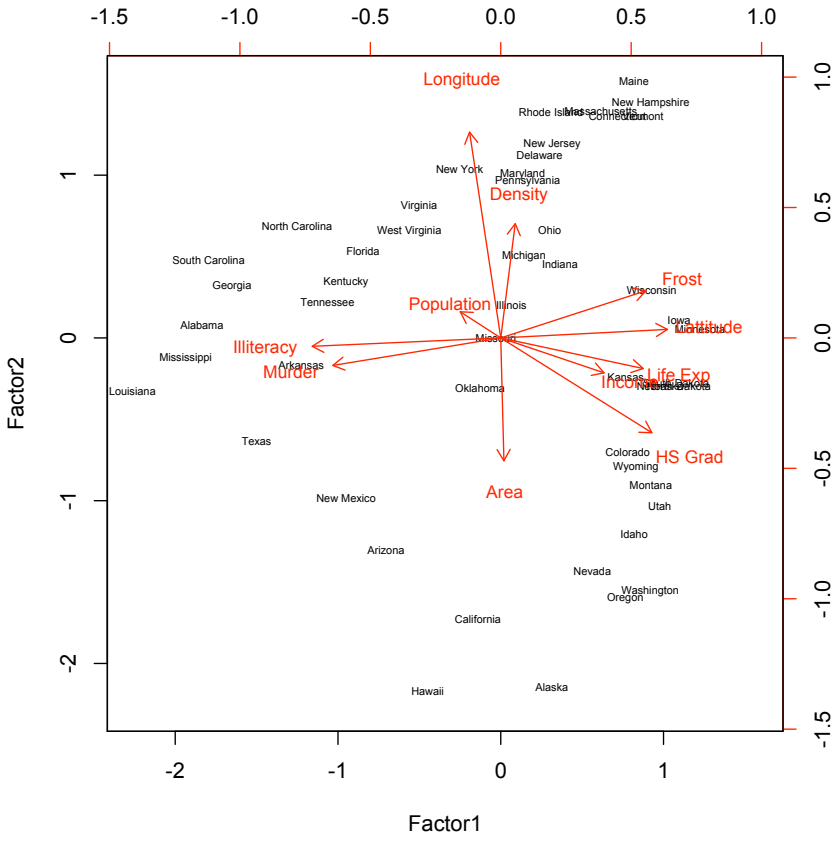

Figure 13: Biplot of the two-factor model.

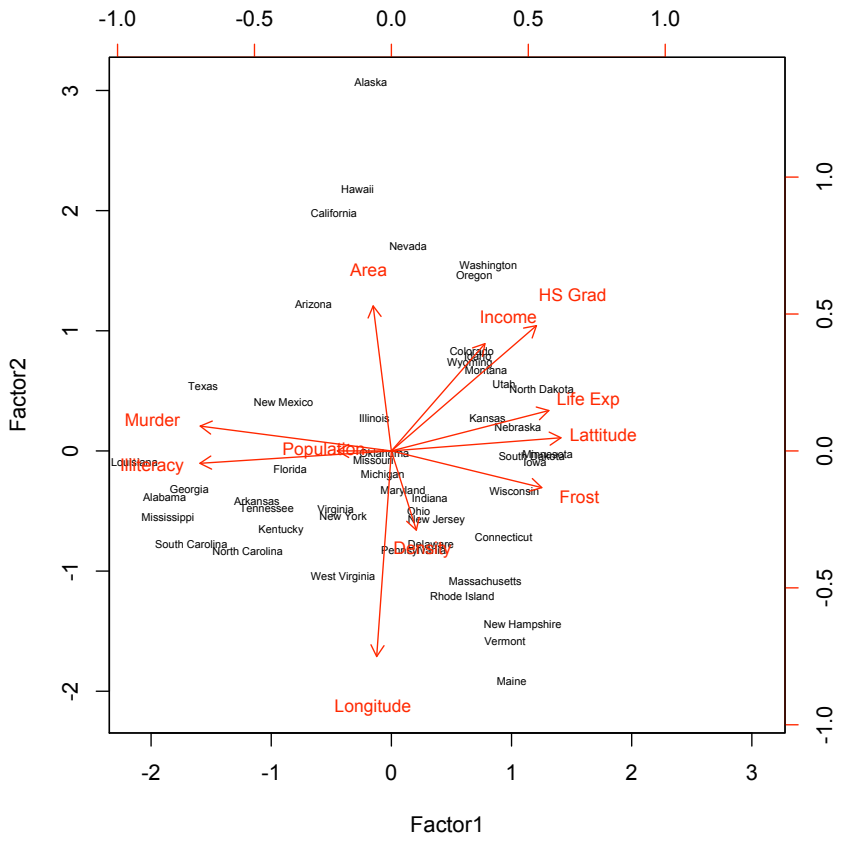

Figure 14: Biplot for the three-factor model, showing scores for the two largest factors.

```
               Factor1 Factor2 Factor3
SS loadings      3.819   2.098   1.265
Proportion Var   0.347   0.191   0.115
Cumulative Var   0.347   0.538   0.653

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 99.02 on 25 degrees of freedom.
The p-value is 9.17e-11
```

Notice that adding more factors is causing the loadings on to the first two factors, and hence the arrows in the biplot, to move around. This doesn't happen with PCA.

Let's go all the way to the largest number of factors that we can fit (Figures 15 and 16).

```
> states.density.fa6 <- factanal(states.density,factors=6,scores="regression")
> biplot.factanal(states.density.fa6,cex=c(0.5,0.75))
> screeplot.factanal(states.density.fa6,type="b")
```

The fit is finally not bad:

```
               Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
SS loadings      2.411   1.920   1.603   1.501   1.074   0.986
Proportion Var   0.219   0.175   0.146   0.136   0.098   0.090
Cumulative Var   0.219   0.394   0.539   0.676   0.774   0.863

Test of the hypothesis that 6 factors are sufficient.
The chi square statistic is 5.11 on 4 degrees of freedom.
The p-value is 0.276
```

This said, it's not right to just take this p-value naively. We have done a whole bunch of hypothesis tests, and the more tests you do at a fixed size, the more likely you are to have a false positive. So we should really do some kind of control for the multiple tests. We will return to this issue later.

Notice also that this goodness-of-fit test is not checking lots of the background assumptions to the factor-analysis model: that the data have a Gaussian distribution, for example, or that the observations are IID samples. (Just how plausible is it that Oklahoma is *completely independent* of Texas, or Connecticut of New York?)

Notice also that we haven't even begun to address the issue of rotations, i.e., of applying some transformation to the factors which doesn't change their observable implications, but might make things like Figure 15 easier to understand. (This is controlled by the `rotation` argument to `factanal`.)

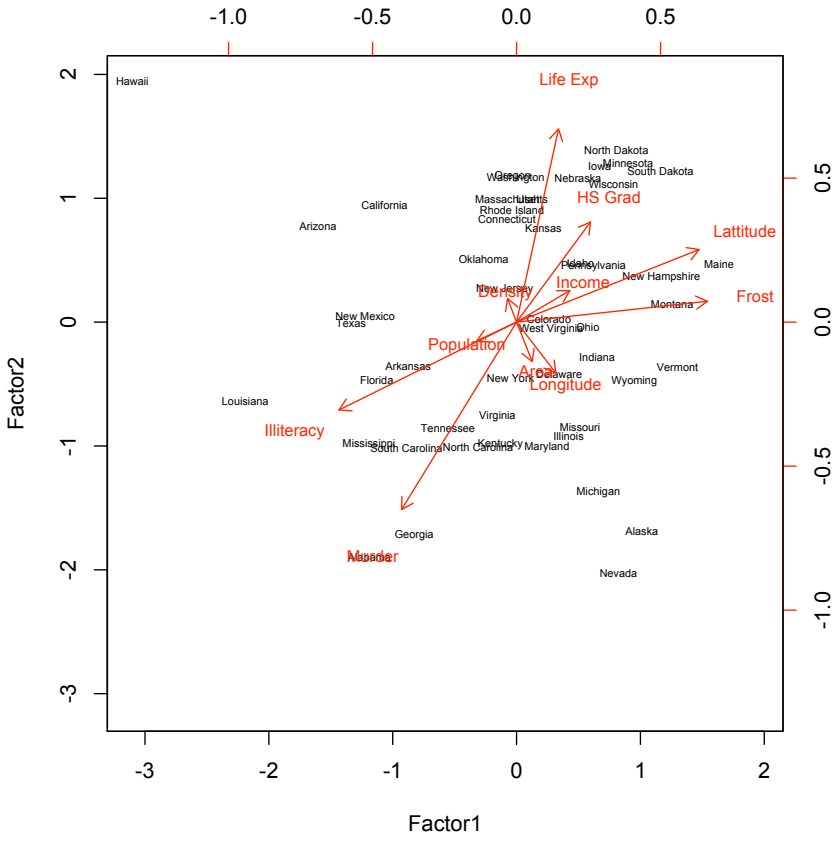

Figure 15: Biplot with six factors.

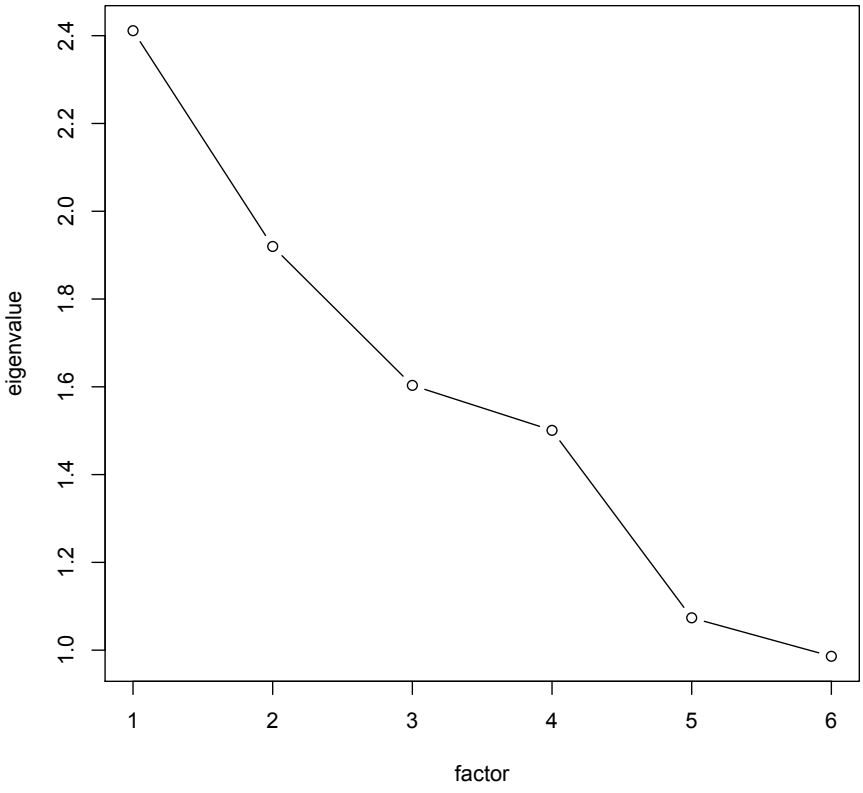

Figure 16: Scree plot for six factors.

References

Frazer, James George (1922). *The Golden Bough: A Study in Magic and Religion*. London: Macmillan, abridged edn. URL `http://www.gutenberg.org/etext/3623`.

Nisbett, Richard E. and Dov Cohen (1996). *Culture of Honor: The Psychology of Violence in the South*. Boulder, Colorado: Westview Press.