Principal components

Principal components is a general analysis technique that has some application within regression, but has a much wider use as well.

Technical Stuff

We have yet to define the term "covariance", but do so now. Remember when we pointed out that if adding two independent random variables X and Y, then
Var(X + Y) = Var(X) + Var(Y).

What happens if X and Y are not independent? Then
Var(X + Y) = Var(X) + Var(Y) + something left-over.

This something left over is called the covariance. It's definition can be found be examining the definition of the variance:
$Var(X) = E(X - m)^2$ where m is the mean E(X).

So $Var(X + Y) = E((X+Y) - (mx + my))^2 = E(X - mx)^2 + E(Y - my)^2 + E(X-mx)(Y-my)$
(This is just a bit of algebra and some applications of using the Expectation as a linear operator.)

This last term, E(X-mx)(Y-my) == Cov(X,Y) is the covariance. It is positive if, when X is bigger than its mean, Y tends to be bigger then its mean, or if both are negative, etc. It measures the relation between X and Y. It's the numerator in the correlation:
Cov(X,Y)/SD(X)SD(Y)

Often, when examining several variables simultaneously, it's productive to look at the covariance matrix. This is the matrix that has the variances on the diagonal, and the covariances on the off-diagonals. It is a symmetric matrix, since Cov(X,Y) = Cov(Y,X). Here's a covariance matrix for two-variables:

Cov(X,X)	Cov(X,Y)
Cov(Y,X)	Cov(Y,Y)

Here's the covariance matrix for the first four variables of the ozone data set:

```
> cov(cbind(ozone, temp, inversionht, pressure))
                  ozone         temp inversionht    pressure
ozone          67.60740     82.98688   -9264.183    77.97842
temp           82.98688    160.97822  -13410.142   165.61216
inversionht -9264.18278 -13410.14220 3344832.428 -2716.72462
pressure       77.97842    165.61216   -2716.725  1029.90152
```

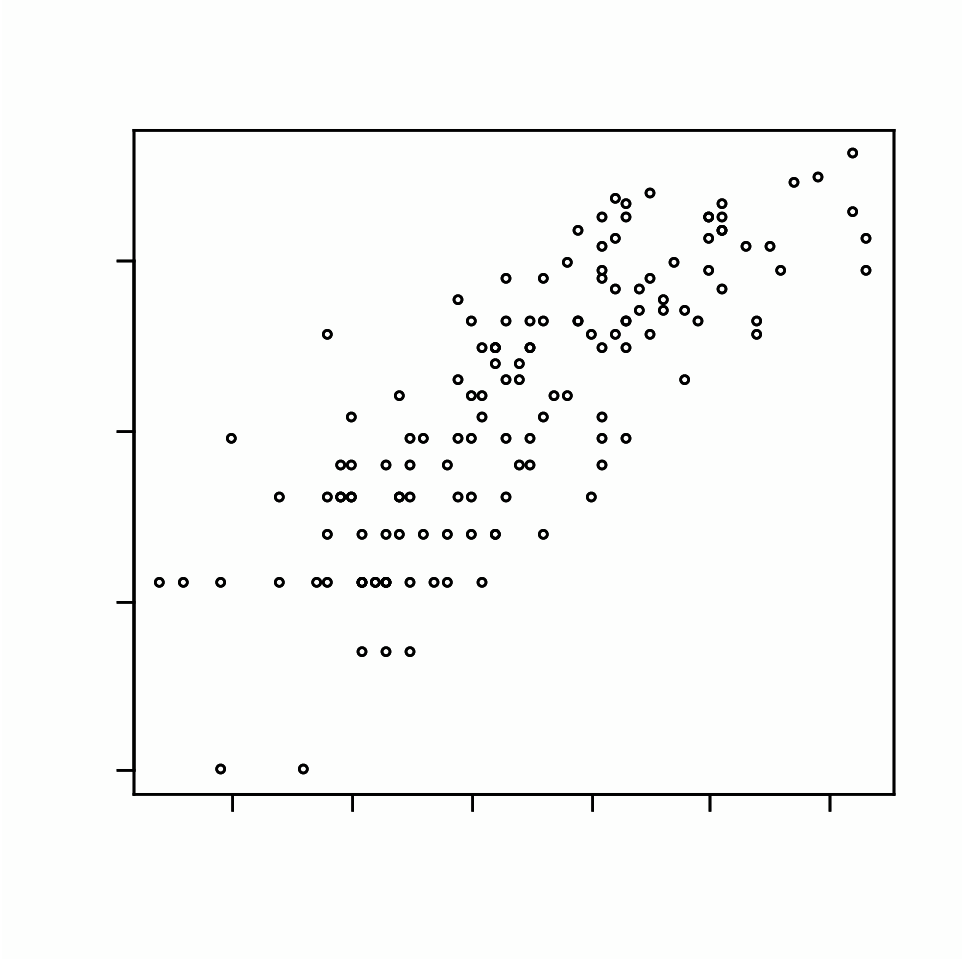
It's difficult to interpret the magnitudes of the covariances, although the signs have some interesting meanings. Often, a correlation matrix is easier to interpret:

```
> cor(cbind(ozone, temp, inversionht, pressure))
                ozone       temp inversionht    pressure
ozone       1.0000000  0.7954791 -0.61605974  0.29551496
temp        0.7954791  1.0000000 -0.57791324  0.40673431
inversionht -0.6160597 -0.5779132  1.00000000 -0.04628717
pressure    0.2955150  0.4067343 -0.04628717  1.00000000
```

The 1's on the diagonal reflect the fact that variables are perfectly correlated with themselves. At a glance we can see that pressure has the lowest correlation with ozone, and that there are strong correlations within the predictors. (Word of warning: correlations measure only linear relationships.)

What is a principal component?

For simplicity sake, we'll consider only two variables: log(ozone) and temp:



The SD of temp (horizontal axis) is 12.7, and the sd of log of ozone is .80. These SDs measure

the spread of each variable across the horizontal and vertical axes, respectively. What if we wanted to rotate the coordinate system so that the SDs were maximized? This would be equivalent to re-drawing the axes to coincide with the axes of the ellipse formed by the cloud of data-points. (It's an approximate ellipse in this case, because the data are not quite normal.)

The "first principal component" finds the direction which has the maximum variation. This direction turns out to be determined by the eigen-vector of the covariance matrix associated with the largest eigen value. The first eigenvalue turns out to be the variance along this direction. The second principal component is constrained to be statistically indepedent of the fist and to maximize the variation. This turns out to be the second eigenvector. There are as many principal components as there are variables.

There are three ways to find these in R. The first is to apply the "eigen" function to the covariance matrix:

```
> eigen(cov(cbind(temp,lozone)))
$values
[1] 161.3894110   0.2172475

$vectors
            lozone        temp
temp   -0.99872355  0.05051013
lozone -0.05051013 -0.99872355
```

The second is to apply the "prcomp" command to a matrix containing the data, or a data frame or table:

```
> out <- prcomp(cbind(lozone, temp))
> names(out)
[1] "sdev"     "rotation" "x"
> out$rotation
               PC1         PC2
lozone -0.05051013  0.99872355
temp   -0.99872355 -0.05051013
> out$sdev
[1] 12.7039132  0.4660981
```

Note that these are the square-roots of the eigenvalues above.

The third is to apply the "princomp" command -- which does the same as prcomp but using a slightly different numerical procedure (one that can be elss stable.)

What do we do with this information? First, think about how we would go about "converting" the data from the old coordinate system to the new. The new axes are given by the column

vectors labeled PC1 (new horizontal axis) and PC2 above. So to find where a point belongs on the new horizontal axis, we multiple the old coordinates by PC1. (This is equivalent to projecting the old point onto the new axis.) Thus, the PC1 vector gives us a recipe: the new point is -.05 parts lozone and -.99872 temp.

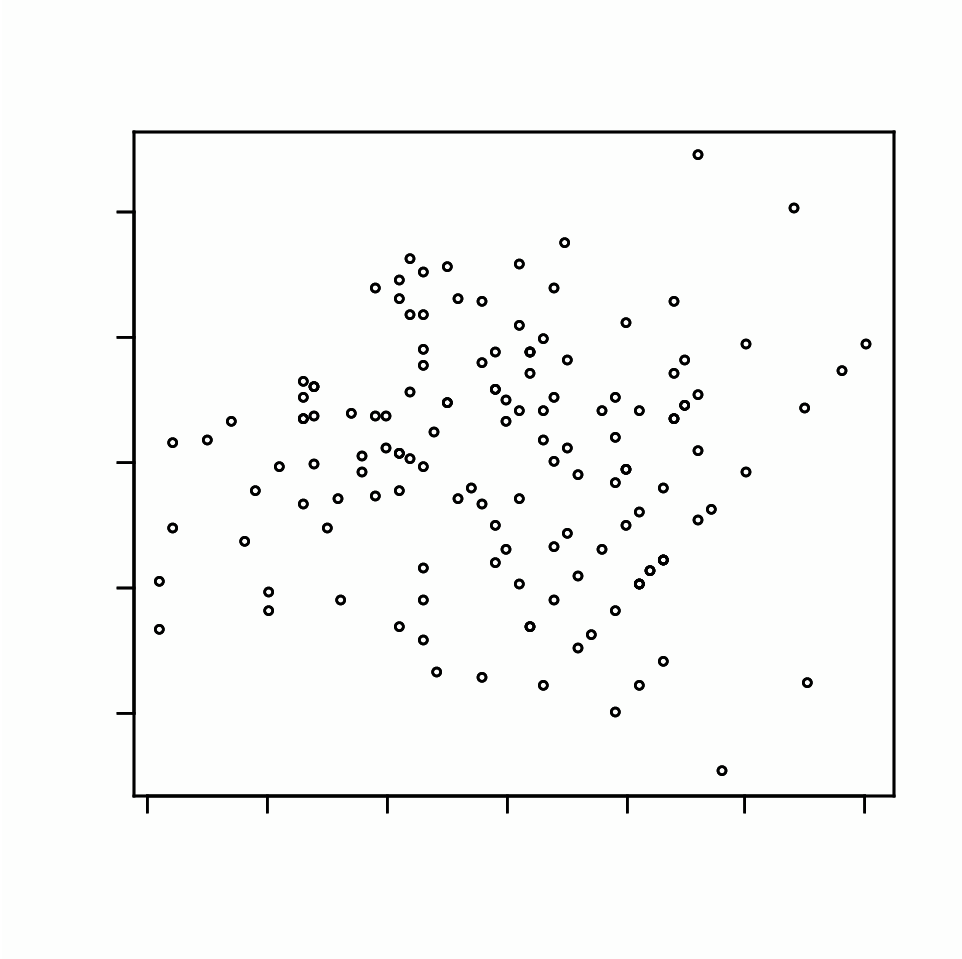
The values of the PCs (the eigenvectors) are sometimes called the "loadings".

This tells us that, in the direction of maximal variation, temperature matters more than ozone. So we might interpret temperature to be the factor that accounts for the most variation.

To carry this further: the sum of the eigenvalues can be used as a measure of "total" variation. The eigenvalue associated with a particular principal component measures the percent of the total variation "explained" by that principal covariance. Thus, the total variation in temperature and ozone is the sum of the eigenvalues: 161.6067. 99.8% of this variation, (161.389441/1161.9067) is carried by the first principal component.

One way of interpreting this is to say that once we've calculated the first principle component, there's little to be gained by translating the second. Thus, we can throw away the second, and we now have a one-dimensional data set.

The translated data are stored under the "x" name in the output. Here's a plot:



Of course, it looks just like that last, but rotated.

What do we do with it?

PCs are useful in regression for mitigating the problem of multi-colinearity -- predictors that are correlated with each other.

Here's the idea:

Rotate the predictors to principal components
The principal components are statisticall independent
If most of the total variation is explained by only a few principal components, then there's no need to keep the rest.

This falls under the more general heading of "dimension reduction". We take a p-dimensional data-set and reduce its dimensions while, hopefully, retaining most of the important information. The main thing that's lost is interpretability. But sometimes, the PC results in a linear combination of predictors that makes sense. If that happens, you call the new variable an "index" and viola, you have a useful model.

Other applications:

Sometimes the PC analysis is the endpoint. It can be a useful way for exploring the relations among variables, particularly if you're not sure which should be the predictor. Biologists sometimes find this useful. For example, a biologist is interested in studying the biological diversity of a certain type of squirrel. So in a particular ecological "niche", a few squirrels are captured, and various characteristics: height, weight, waist, tail length, etc. are measured. A PC analysis then determines that the first PC "loads heavily" on variables that determine the overall size of the squirrel, and the next PC seems to determine general "shape" characteristics. (the phrase "loads heavily" means that those variables get high values on a particular PC.) The other PCs explain very little of the variation, and are discarded. The biologist now has two indices: shape and size, with which to do her analysis.

Correlation vs. Covariance

One can also do a PC analysis on the correlation matrix, rather than the covariance matrix. You will get slightly different results. A guiding rule is this: if the variables are measured on different scales or have very different variances, use correlation. (Correlations are unitless). Otherwise, use covariances.

Application to Ozone

One problem with the ozone data set was that there was a lot of colinearity among the predictors. Can we produce a useful model using PC?

Because the variables are measured in different units, it makes sense to use the correlation matrix rather than the covariance matrix.

The results are mixed, which isn't surprising. Here's the output:

```
> out2 <- prcomp(cbind(temp, inversionht, pressure, visibility,
height, humidty, temp2, windspeed), scale=T)
> summary(out2)
Importance of components:
                          PC1   PC2   PC3    PC4    PC5    PC6    PC7    PC8
Standard deviation     1.919 1.350 1.036 0.7973 0.6519 0.4141 0.3912 0.1832
Proportion of Variance 0.461 0.228 0.134 0.0795 0.0531 0.0214 0.0191 0.0042
Cumulative Proportion  0.461 0.688 0.823 0.9021 0.9552 0.9767 0.9958 1.0000
> names(out2)
[1] "sdev"     "rotation" "x"
> out2$rotation
                   PC1          PC2         PC3          PC4         PC5
temp        -0.4750898 -0.006356703 -0.23656449 -0.061962588  0.2651884
inversionht  0.4055684 -0.265532894 -0.18068873  0.199125002  0.6500196
pressure    -0.2096776 -0.590569963 -0.05010618 -0.440499507  0.2067317
visibility   0.2505009  0.009950117 -0.71428618 -0.518268335 -0.3004242
```

```
height      -0.3950753  0.288023619 -0.32341566  0.076227499  0.4643747
humidty     -0.3180481 -0.467044124  0.32385135 -0.181507254 -0.1092342
temp2       -0.4740096  0.260226425 -0.08993665  0.004649711 -0.1680065
windspeed   -0.1454797 -0.460189502 -0.42556722  0.674604220 -0.3440273
                    PC6         PC7          PC8
temp        -0.056048668 -0.67180322 -0.435501227
inversionht  0.258402087 -0.24814687  0.376607822
pressure    -0.535397537  0.22404041  0.176435088
visibility   0.260396995 -0.01660971 -0.002505468
height       0.207737901  0.61714059 -0.104346309
humidty      0.727282617  0.04064863 -0.019135473
temp2        0.008634943 -0.21544566  0.790423464
windspeed   -0.058823366  0.08776061 -0.036688417
```

From this we see that we need to keep at least 5 PCs to have 95% of the variation preserved. The first PC is interesting in that it seems to contrast inversionht and visibility with a roughly equal average of the others. So the first PC is biggest when the difference between visibility and inversionht with the other variables is biggest. Perhaps someone well aquainted with this phenomenon could explain this, but I can't.

We could, if we wished, keep the first 5 PCs, and redo the regression using them.

```
> newdata <- out2$x[,1:5]
> fit4 <-lm(ozone ~ newdata)
> summary.lm(fit4)

Call:
lm(formula = ozone ~ newdata)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5015  -3.0063  -0.1044   2.7530  13.4531

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.46099    0.38734  29.589   <2e-16 ***
newdataPC1  -3.55333    0.20252 -17.545   <2e-16 ***
newdataPC2   0.21450    0.28787   0.745   0.4575
newdataPC3   0.24829    0.37512   0.662   0.5092
newdataPC4   0.01843    0.48754   0.038   0.9699
newdataPC5   1.13138    0.59626   1.897   0.0599 .
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1
```

```
Residual standard error: 4.599 on 135 degrees of freedom
Multiple R-Squared: 0.6983,    Adjusted R-squared: 0.6871
F-statistic: 62.49 on 5 and 135 DF,  p-value:     0
```

The fit is somewhat less than successful. The residuals suggest the fit is quite poor.

What went wrong? Well, in some sense, nothing. This is a tool which sometimes works, sometimes doesn't. But one problem is that PC works best when the distributions are all normal, which is not the case here. Transformations followed by PC might do a better job.