4.4 Principal Component Analysis

With the kind permission of Paul Francis, this exercise (and indeed much of the description of PCA in Section 4.5) is taken from a beautiful little paper by Paul and Bev Wills (Francis, P.J. and Wills, B.J., 1999, in ASP Conf. Ser. 162: Quasars and Cosmology, p363) describing and illustrating exactly how PCA works.

The answer to the PCA analysis of the data is given in this paper as follows:

Table 3. Results of Eigenanalysis – The Principal Components[a]

	PC1	PC2	PC3	PC4	PC5
Eigenvalue	6.4505	2.8157	1.5879	0.6257	0.5698
Proportion	0.496	0.217	0.122	0.048	0.044
Cumulative	0.496	0.713	0.835	0.883	0.927
Variable	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**
log L_{1216}	0.053	0.535	−0.123	−0.029	−0.405
α_x	0.295	−0.198	0.079	0.485	−0.155
FWHM Hβ	−0.330	0.077	−0.357	−0.082	−0.141
FeII/Hβ	0.341	−0.140	0.003	−0.487	−0.212
log EW [O III]	−0.310	0.016	0.255	0.394	−0.095
log FWHM C III]	−0.198	0.077	−0.623	0.054	0.402
log EW Lyα	−0.177	−0.502	−0.006	−0.143	0.033
log EW C IV	−0.336	−0.262	0.048	−0.050	−0.303
C IV/Lyα	−0.342	0.062	0.025	−0.074	−0.584
log EW C III]	−0.262	−0.413	−0.124	−0.176	−0.008
Si III]/C III]	0.342	−0.149	−0.018	−0.311	−0.116
N V/Lyα	0.231	−0.050	−0.573	0.107	−0.288
λ1400/Lyα	0.223	−0.351	−0.225	0.441	−0.216

Figure 1: The results of the Francis-Wills analysis. [a] denotes that 18 of the 22 QSO spectra were used, with 4 cases containing missing values. See below.

Francis and Wills describe this answer:

"We perform a PCA on the small sample of 22 QSOs discussed by Wills et al. (1998, in Structure and Kinematics of Quasar Broad Line Regions, ed Gaskell, M. et al., ASP Conf. Ser. 162, and 1998, in Quasars as Standard Candles for Cosmology, ed. Ferland, G., ASP Conf. Ser., in press), using a subset of the available measured properties shown in (the data table). Unavoidably, there are missing data, so the number of objects available depends on the variables chosen for the PCA....

...(The data table) presents, for each variable, the number of data points, the mean and the standard deviation. Notice the completely different units for different measured parameters....In order to weight the variables more or less equally, after subtracting the mean values, we normalize by the variance. The choice of weights is a difficult issue, and depends on the user's knowledge of the data, and preferences, as well as the use to which the results will be put. The results of performing a PCA on these normalized variables are shown (above, Figure 1). Columns (2)-(6) show the first 5 out of a total of 13 principal components. The first row gives the variances (eigenvalues) of the data along the direction of the corresponding principal component. The sums of all the

variances add up to the sums of the variances of the input variables, in this case, 13. By convention, the principal components are given in order of their contribution to the total variance. This is given as 'Proportion' in the second line, and the 'Cumulative' proportion on the third line. Thus, among the parameters we have chosen to use, the first principal component contributes 50% of the spectrum-to-spectrum variance, the second 22%, the third, 12%. The first two principal components together contribute 71% of the variance, the first 3, 84%, and the first 4, nearly 90%.

The columns of numbers for each principal component represent the weights assigned to each input variable. Thus $PC1 = 0.053x_1 + 0.295x_2 - 0.330x_3 \ldots$, where x_1, x_2, x_3 are the values of the normalized variables corresponding to $\log L_{1216}$, α_x, FWHM Hβ, etc. By convention these weights are chosen so that the sum of their squares = 1. This arbitrarily fixes the scale of the new variable. The sign of the new variable is therefore arbitrary....

The first principal component is elongated with variance about 6.5 times that of any individual measurements, and accounts for about half the total variance. This is therefore likely to be highly significant. If all measured, normalized quantities contributed equally to PC1, they would all have weight 0.277 ($1/\sqrt{13}$ for 13 variables), but each variable contributes more or less than this. One way to test the significance of the contribution of any one measured variable, is to perform the PCA without that variable, then check the significance of the correlation between that variable and the scores of the new principal component. This procedure shows that all measured variables except L_{1216}, log FWHM CIII], and log EW Lyα, correlate with PC1, but correlations involving NV/Lyα and λ1400/Lyα are not very strong. PC2, accounting for 22% of the variance in this dataset, appears to link the EW Lyα, EW CIV, and EW CIII] with L_{1216}, so EW CIV and EW CIII] appear to contribute to both PC1 and PC2, but EW Lyα contributes predominantly to PC2. Is PC2 a significant component? A similar correlation test shows that individually the EWs do anti-correlate with L_{1216}, but this result depends on the lowest EWs for the highest luminosity QSO PG1226+023 and the highest EWs for the low luminosity QSO PG1202+281. However L_{1216} correlates significantly (Pearson's ordinary correlation coefficient = -0.77) with PC2 formed when L_{1216} is excluded. Thus there is a significant overall correlation between EW and L_{1216}, although a larger sample is clearly needed to investigate the individual EW correlations. Another test may be to check correlations between observed measurements for those measurements that contribute to only one significant principal component - for example, CIV/Lyα vs. FeII/Hβ...

As a rule of thumb, any principal component with variance greater than 1, should be considered seriously. It is also worth investigating any principal component with variance rather greater than that of the remaining principal components. In our example, this could mean the first three principal components."

We consider the example step-by-step to make it easy to trace your own solution through.

(1) The first step is to take the original data and put it into normalized or weighted

form so that the effect of different scales and different units is effectively removed. In doing so, note the tiny mistake in Francis & Wills; the mean is subtracted from each of the 13 variables and they are then normalized by the *standard deviation*, not the variance, as dimensional analysis shows.

Here again is the data table:

qso data 1	2	3	4	5	6	7	8	9	10	11	12	13
1	45.66	1.51	3.684	0.23	1.18	3.520	2.08	1.78	0.450	1.240	0.306	0.179	0.143
2	45.83	1.57	3.496	0.25	1.26	3.432	2.19	1.78	0.400	1.240	0.164	0.189	0.093
3	44.99	0.88	3.660	0.20	1.23	3.654	2.27	1.85	0.420	1.480	0.222	0.175	0.092
4	45.41	1.89	3.236	0.54	0.78	3.403	1.90	1.51	0.330	1.140	0.385	0.228	0.134
5	46.00	1.73	3.465	0.47	1.00	3.446	2.14	1.71	0.340	1.200	0.440	0.254	0.126
6	44.77	1.22	3.703	0.29	1.56	3.434	2.72	2.41	0.690	1.870	0.164	0.154	0.098
7	46.03	1.36	3.715	0.20	1.00	3.514	2.12	1.95	0.540	1.200	0.037	0.121	0.056
8	46.74	0.94	3.547	0.57	0.70	3.477	1.64	1.44	0.450	1.000	0.280	0.174	0.018
9	45.55	1.51	3.468	0.28	1.28	3.406	2.01	1.68	0.410	1.150	0.303	0.131	0.064
10	45.42	1.69	3.446	0.59	0.90	3.351	2.19	1.85	0.410	1.300	0.291	0.135	0.097
11	45.34	1.52	3.556	0.46	1.00	3.548	2.14	1.80	0.410	1.290	0.357	0.203	0.116
12	45.74	1.93	3.281	1.23	0.30	3.229	1.91	1.59	0.390	1.090	0.568	0.227	0.161
13	45.08	1.74	3.418	1.25	0.30	3.434	2.32	1.78	0.290	1.400	0.688	0.210	0.142
14	45.54	1.41	3.405	0.36	1.76	3.300	2.03	1.82	0.490	1.210	0.265	0.126	0.117
15	45.23	2.08	3.161	1.19	1.00	3.192	2.14	1.54	0.210	1.050	0.747	0.141	0.092
16	45.92	1.91	3.394	1.45	0.30	3.479	1.99	1.34	0.210	1.060	0.809	0.335	0.164
17	46.04	1.21	3.833	0.16	1.76	3.546	2.02	2.05	0.750	1.280	0.228	0.182	0.050
18	45.48	1.94	3.652	0.32	0.95	3.631	2.14	1.80	0.390	1.360	0.197	0.217	0.118
mean	45.56	1.57	3.498	0.56	0.99	3.446	2.11	1.78	0.421	1.279	0.362	0.212	0.108
std dev	0.45	0.33	0.175	0.41	0.43	0.119	0.21	0.24	0.134	0.193	0.208	0.052	0.038

Part of the normalization process is to compute the means and standard deviations of each of the 13 variables. The column variables computed here differ slightly from those of Francis and Wills, presumably because they used data from all 22 QSOs. Here we have rejected entirely the 4 QSOs with incomplete data; the differences are not significant, as we shall see. The table of normalized data $(x(i,j) - \overline{x_j})/\sigma_j$ is then as follows:

qso data 1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.14	-0.14	1.014	-0.80	0.39	0.636	-0.13	0.09	0.215	-0.069	-0.252	-0.170	1.005
2	0.52	0.04	-0.061	-0.75	0.57	-0.103	0.39	0.09	-0.157	-0.069	-0.934	0.022	-0.300
3	-1.35	-2.03	0.877	-0.88	0.50	1.760	0.77	0.38	-0.008	1.175	-0.655	-0.246	-0.326
4	-0.42	0.99	-1.548	-0.04	-0.55	-0.346	-0.99	-1.06	-0.678	-0.588	0.128	0.771	0.770
5	0.89	0.51	-0.238	-0.21	-0.03	0.015	0.15	-0.21	-0.604	-0.276	0.392	1.270	0.561
6	-1.84	-1.01	1.123	-0.66	1.27	-0.086	2.90	2.77	2.001	3.197	-0.934	-0.649	-0.170
7	0.96	-0.59	1.192	-0.88	-0.03	0.585	0.06	0.81	0.885	-0.276	-1.544	-1.283	-1.266
8	2.54	-1.85	0.231	0.03	-0.73	0.275	-2.22	-1.36	0.215	-1.313	-0.377	-0.265	-2.258
9	-0.11	-0.14	-0.221	-0.68	0.62	-0.321	-0.47	-0.34	-0.083	-0.536	-0.266	-1.091	-1.057
10	-0.40	0.40	-0.347	0.08	-0.27	-0.782	0.39	0.38	-0.083	0.242	-0.324	-1.014	-0.196
11	-0.57	-0.11	0.282	-0.24	-0.03	0.870	0.15	0.17	-0.083	0.190	-0.007	0.291	0.300
12	0.31	1.11	-1.291	1.64	-1.66	-1.805	-0.94	-0.72	-0.232	-0.847	1.007	0.752	1.475
13	-1.15	0.54	-0.507	1.69	-1.66	-0.086	1.00	0.09	-0.976	0.760	1.584	0.425	0.979
14	-0.13	-0.44	-0.582	-0.48	1.74	-1.210	-0.37	0.26	0.513	-0.225	-0.449	-1.187	0.326
15	-0.82	1.56	-1.977	1.55	-0.03	-2.116	0.15	-0.94	-1.571	-1.054	1.867	-0.899	-0.326
16	0.72	1.05	-0.644	2.18	-1.66	0.292	-0.56	-1.79	-1.571	-1.002	2.165	2.824	1.553
17	0.98	-1.04	1.867	-0.97	1.74	0.854	-0.42	1.23	2.448	0.138	-0.626	-0.112	-1.423
18	-0.26	1.14	0.831	-0.58	-0.15	1.567	0.15	0.17	-0.232	0.553	-0.775	0.560	0.352

This process of 'data adjustment', weighting, normalizing, whatever, is critical to the outcome, in particular to whether we understand the significance of the results, and

whether the error/covariance matrix really does the job we expect of it. As emphasized by Paul and Bev, there are many ways of doing this: we can take logs of the data, we can weight by factors other than standard deviations based on prior knowledge, etc. So, dare we see what the present weighting system looks like? Figure 1 plots the run of the 18 points, one from each QSO, for each of the 13 data.

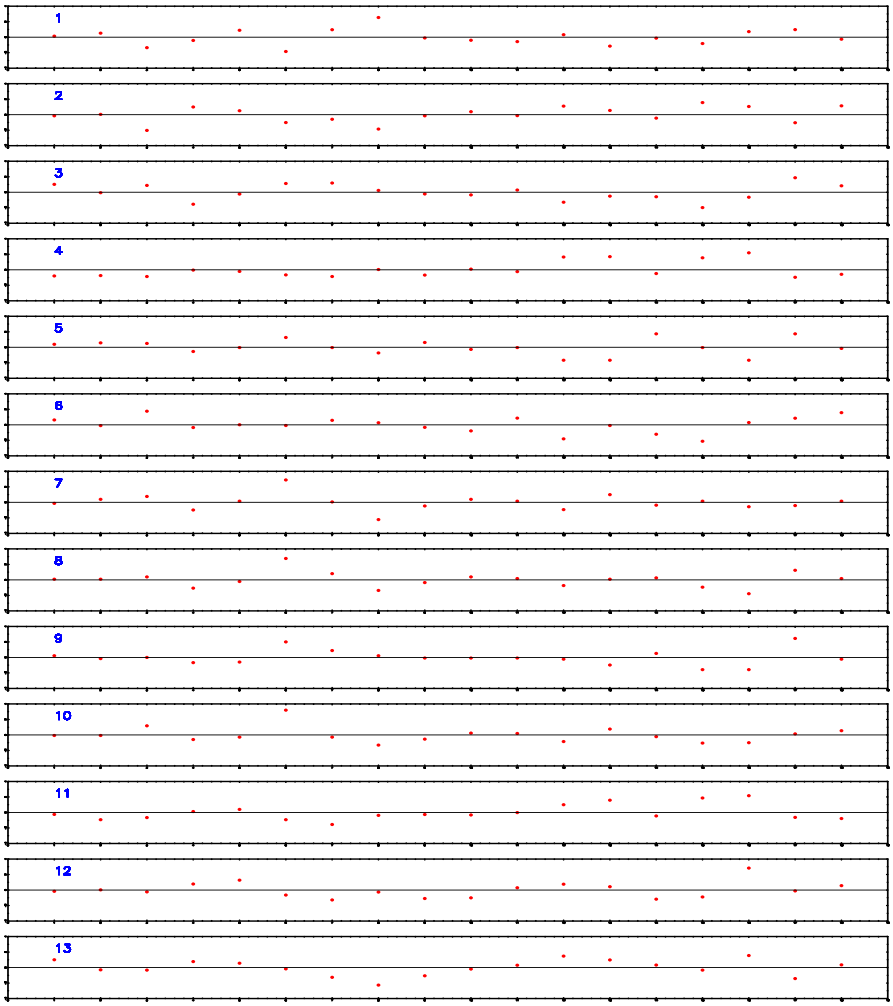

Figure 2: The run of each of the 13 data sets, data from 18 QSOs. QSO number is along the x-axis. Each plot is $\pm 4\sigma$.

Looks pretty good, does it not? All the points are there; there's only one deviation $> 3\sigma$ in 234 points, not far off expectation for Gaussian distributions, and the distributions look reasonable. We may be confident that the results will be understandable.

(2) Step 2 is to construct the covariance or error matrix. This is a 13×13 symmetric matrix:

$$\mathbf{C} = \begin{vmatrix} < x_1^2 > & < x_1 x_2 > & \\ < x_1 x_2 > & < x_2^2 > & \\ & & \ddots \end{vmatrix} \tag{1}$$

(2)

The following few lines of Fortran set it up:

```
c normalized data in xn(18,13); set up the upper-right off-diagonal elements
         do 100 j=1,13
            do 100 k=j+1,13
               do 110 i=1,18
                  e1(j,k)=e1(j,k)+xn(i,j)*xn(i,k)
 110           continue
               e1(j,k)=e1(j,k)/18.
 100     continue

c set up the diagonal elements (which should all be 1.0 - just checking!)
         do 130 j=1,13
            do 128 i=1,18
            e1(j,j)=e1(j,j)+xn(i,j)*xn(i,j)
 128     continue
            e1(j,j)=e1(j,j)/18.
 130     continue

c reflect the off-diagonal matrix elements about the diagonal
         do 120 j=1,13
            do 120 k=j+1,13
               e1(k,j)=e1(j,k)
 120     continue
```

Here is the resulting 13×13 covariance matrix:

```
 1.0000 -0.1530  0.1135 -0.0414 -0.1420  0.0627 -0.7656 -0.4387  0.0620 -0.6803 -0.0962  0.1764 -0.3794
-0.1530  1.0000 -0.6775  0.6117 -0.5009 -0.4853 -0.0647 -0.4348 -0.6603 -0.3460  0.6255  0.4159  0.6514
 0.1135 -0.6775  1.0000 -0.7000  0.5029  0.7748  0.2860  0.6694  0.7656  0.5151 -0.7008 -0.2118 -0.4287
-0.0414  0.6117 -0.7000  1.0000 -0.7829 -0.5204 -0.1602 -0.5852 -0.6826 -0.3701  0.9295  0.5139  0.5182
-0.1420 -0.5009  0.5029 -0.7829  1.0000  0.1549  0.3013  0.6476  0.6979  0.3944 -0.6505 -0.5894 -0.4519
 0.0627 -0.4853  0.7748 -0.5204  0.1549  1.0000  0.1207  0.2595  0.2923  0.3465 -0.4627  0.1881 -0.1898
-0.7656 -0.0647  0.2860 -0.1602  0.3013  0.1207  1.0000  0.7653  0.2489  0.8897 -0.1574 -0.1864  0.1630
-0.4387 -0.4348  0.6694 -0.5852  0.6476  0.2595  0.7653  1.0000  0.7925  0.8609 -0.6196 -0.4830 -0.2307
 0.0620 -0.6603  0.7656 -0.6826  0.6979  0.2923  0.2489  0.7925  1.0000  0.5117 -0.7328 -0.4608 -0.5046
-0.6803 -0.3460  0.5151 -0.3701  0.3944  0.3465  0.8897  0.8609  0.5117  1.0000 -0.3930 -0.2054  0.0287
-0.0962  0.6255 -0.7008  0.9295 -0.6505 -0.4627 -0.1574 -0.6196 -0.7328 -0.3930  1.0000  0.5622  0.5626
 0.1764  0.4159 -0.2118  0.5139 -0.5894  0.1881 -0.1864 -0.4830 -0.4608 -0.2054  0.5622  1.0000  0.6198
-0.3794  0.6514 -0.4287  0.5182 -0.4519 -0.1898  0.1630 -0.2307 -0.5046  0.0287  0.5626  0.6198  1.0000
```

(3) All we have to do now is solve 13 13th order equations in 13 unknowns to get the eigenvalues of this matrix! But this is 150-year old technology; for symmetric matrices, *Jacobi rotations* do the trick, each plane rotation or transformation designed to get rid of one off-diagonal matrix element. "The Jacobi method is absolutely foolproof for all real symmetric matrices" – *Numerical Recipes*. The *Numerical Recipes* routine (*jacobi*, how did you guess), when supplied with the covariance matrix returns the eigenvalues, the array of eigenvectors, and the number of rotations required, which turns out to be about $3 * 13^2 = 500$. The cpu time required is insignificant.

The routine *eigsrt* orders the eigenvalues (hardly necessary when there are only 13) and the eigenvectors (helpful). Here are the results from putting the covariance array into *jacobi* and the results from *jacobi* into *eigsrt*:

Rotations: 459
Eigenvalues: 6.451 2.820 1.589 0.624 0.565 0.343 0.261 0.172 0.122 0.023 0.019 0.010 0.002

Eigenvectors:

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
-0.055	0.534	0.126	-0.018	0.408	0.193	-0.128	-0.322	-0.418	0.075	0.250	0.280	-0.226
-0.294	-0.197	-0.082	0.490	0.151	0.511	-0.456	0.146	0.282	0.090	0.147	-0.018	-0.071
0.330	0.077	0.357	-0.081	0.149	0.133	-0.213	0.422	-0.296	0.111	0.150	-0.480	0.366
-0.342	-0.139	-0.006	-0.484	0.222	-0.001	-0.074	0.184	0.013	0.656	-0.297	0.146	0.015
0.310	0.016	-0.252	0.396	0.093	-0.619	-0.389	-0.017	-0.064	0.352	-0.019	0.105	0.018
0.198	0.075	0.624	0.044	-0.399	0.007	-0.183	0.234	0.132	0.064	-0.129	0.394	-0.351
0.177	-0.503	0.005	-0.138	-0.026	0.127	-0.312	-0.352	-0.396	-0.101	-0.283	-0.242	-0.391
0.336	-0.262	-0.051	-0.046	0.302	0.196	-0.049	0.046	-0.041	-0.276	-0.214	0.601	0.441
0.342	0.064	-0.031	-0.067	0.581	-0.034	0.180	0.215	0.411	-0.112	-0.128	-0.171	-0.479
0.261	-0.414	0.124	-0.177	0.012	0.016	0.146	-0.257	0.203	0.294	0.698	0.101	-0.016
-0.342	-0.149	0.015	-0.310	0.125	-0.399	-0.362	0.301	-0.056	-0.469	0.348	0.106	-0.113
-0.231	-0.053	0.571	0.112	0.288	-0.258	-0.088	-0.465	0.291	-0.083	-0.190	-0.159	0.279
-0.223	-0.351	0.225	0.441	0.207	-0.136	0.499	0.251	-0.424	0.054	0.019	0.087	-0.135

One simple check of this step: the eigenvalues must add up to the trace of the array, the sum of the diagonal elements, 13 of course.

We see that the eigenvalues are virtually identical to those in the results table given at the outset in Figure 1; and the eigenvectors likewise. These eigenvector columns are the weights assigned to the input variables by each eigenvalue; e.g. $PC1 = -0.55x_1 - 0.294x_2 + 0.330x_3 \ldots$; the sign of this new variable is arbitrary and note that for the current solution this sign differs in three out of the 5 PCAs presented in the initial table.

As Francis and Wills advise, one way to test the significance of the contribution of any one variable to the eigenvalue (total variance) is to remove it and perform the analysis again. There is perhaps an earlier step, namely what confidence to place in the eigenvalue itself being of significance. The eigenvalues are plotted in Figure 3 below.

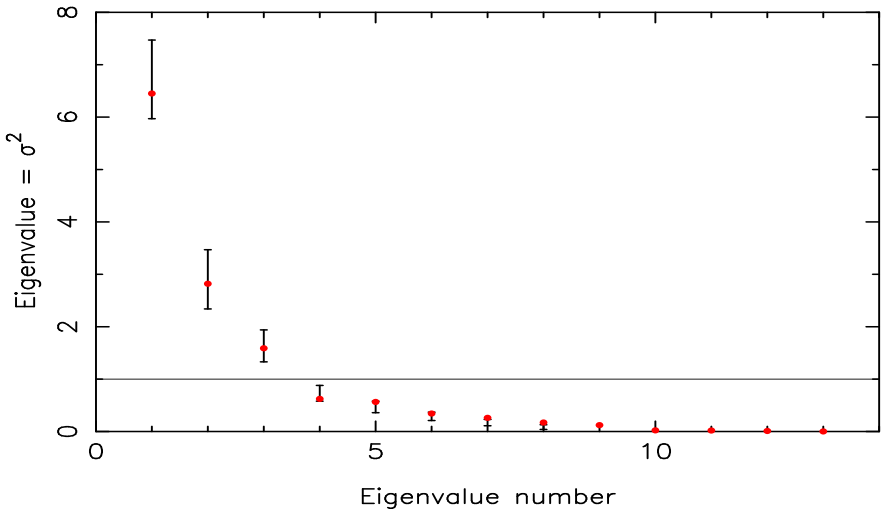

Figure 3: The set of 13 eigenvalues. The error bars are approximate $\pm 1\sigma$ as determined from a bootstrap analysis described below.

As Francis and Wills note, the first eigenvalue looks highly significant. It is then well

worth investigating the individual contributors to it. But how strongly are we to believe that there is significance in the PCAs described by eigenvectors 2 and 3?

Once the formalism to set up the steps (1), (2) and (3) above is complete, doing a bootstrap test is particularly easy. All we need to do is to select 18 of our QSOs *at random and with replacement*, standard bootstrap procedure (§6.6), rebuild our normalized data array, reform the covariance matrix, and solve for the eigenvalues again. We can do this as many times as we like, because the cpu time is not really an issue; and the range of results tells us our expected distributions for all the eigenvalues. Figure 4 shows the results of 10000 trials and these trials took a total of about 10 sec on a slow (800 MHz) laptop.

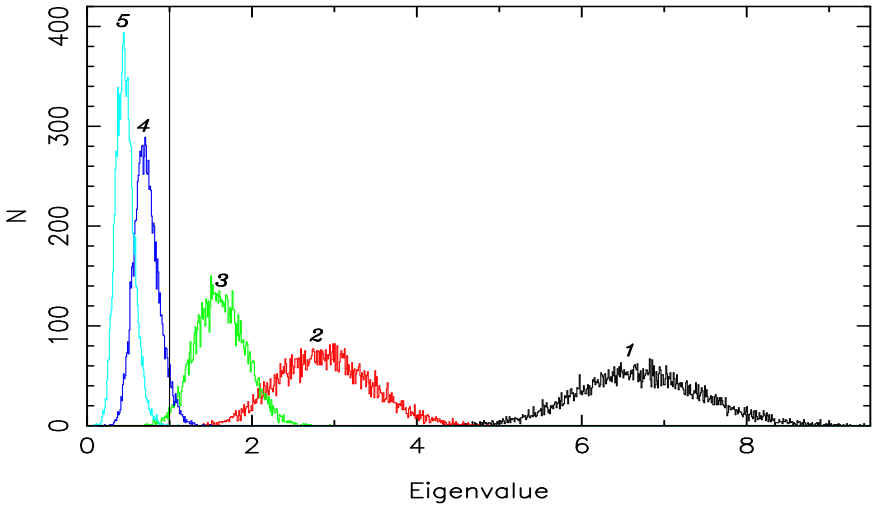

Figure 4: The results of 10000 bootstrap runs to determine the spread for each of the eigenvalues determined in the Francis and Wills example. Frequency runs north; the value of each eigenvalue runs east-west, with colour-coding used to discriminate between the 5 top eigenvalues plotted.

The error bars in Figure 3 come from these distributions. They show the clear significance of eigenvectors 1 and 2; and eigenvector 3 is almost certainly significant. We can forget about the rest; the chances of eigenvalue 4 being 1.0 are minuscule, and eigenvalue 5 and the rest never get anywhere near it.

It is worth, then, trying to understand the physical significance of PCAs 1, 2 and 3, as Francis and Wills discuss.

PCA represents the ultimate powerful way of searching for correlations in a stack of data. It is so simple to perform and no special numerical skills are required. There are a few caveats of course.

1. The distribution of points in the multi-dimension space must be essentially unimodal. Consider interpreting what PCA might mean in the simple 2D case in which there are two separate blobs of (x,y) points. PCA is certainly going to find the axis joining the blobs. It is not going to tell you why there are two blobs

or even that there are two blobs. What will you make of the variance in the other direction, the other PCA, which may have to reflect entirely different blob scatters somehow?

Thus the data need to be of quadratic form; they need to cluster continuously around the PCA, but they need not do this necessarily in a Gaussian manner. In fact the method is immensely forgiving in terms of distribution, provided the 'unimodal' condition is met - you will always get an answer, although interpretation may be difficult.

2. For this latter reason it is important to investigate at the outset what the form of the data scatter will be, with plots such as that of Figure 2. It may well be worth considering other methods of central location for zero-pointing, such as the median; and methods of normalizing other than a standard deviation computed from rms.

3. PCA software is available in widely used software packages - SPSS, SAS, Minitab. It is also available at Paul Francis's web site
http://msowww.anu.edu.au/ pfrancis/
If using this, please observe the acknowledgement requested by Paul.

But in the end, is it worth learning about another data interface? And will you understand what a PCA package has done for you? The PCA tools are simple; for standard PCA you only need a routine for solving a symmetric matrix. To understand the errors you need the bootstrap. That's it.

To advance your knowledge of PCA considerably, consider the following.

1. Set up a simple 2D correlation, pairs of (x, y) either invented, or (better) designed as a bivariate Gaussian correlation (§4.2). Use less than 100 pairs, but a reasonable degree of correlation ($\rho > 0.5$). Find the PCAs first geometrically, by rotating the axis to minimize/maximize variances, and then through determining the eigenvalues and eigenvectors of the covariance matrix. Understand completely the relation between the results from each approach.

2. Set up a similar experiment in 3D, with different correlation coefficients (§4.2). Derive the PCA; understand the relation of the eigenvalues and eigenvectors to your input parameters.

3. Consider robustness: try to fool the PCA by throwing in outliers, or even by superposing two blobs of points in 2D or 3D experiments to see how PCA performs and under what conditions it produces believable answers.

4. In all of these, use bootstrap tests to estimate errors on eigenvalues. Relate these errors to errors anticipated given the input data.