Stats 519: Midterm

Due on March 12, 2009

Dr. Stephen Lee 1:30

Monte J. Shaffer

Problem 1

Let $A = \begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix}$

Program 1 Useful Functions

```
> vectorLength = function (vector) {sqrt(sum(vector^2));}
> dotProduct   = function (a,b)    {t(as.matrix(a))%*%as.matrix(b);}
> calcAngle    = function (a,b)    {180*acos(dotProduct(a,b)/(vectorLength(a)*vectorLength(b)))/pi;}
> doAnalysis   = function (a,b)
                         {
                         myAnswer = numeric(5);
                              myAnswer[1]=round(vectorLength(a),digits=3);
                              myAnswer[2]=round(vectorLength(b),digits=3);
                              myAnswer[3]=round(calcAngle(a,b),digits=1);       # degrees
                              myAnswer[4]=round(calcAngle(a,b)/180,digits=2);   # radians
                              myAnswer[5]=dotProduct(a,b);
                         myAnswer;
                         }
```

a) Find the eigenvalues by solving a quadratic equation, and thus find the corresponding eigenvectors of A. The equation to be solved is $(A - \lambda I)U = 0$. In the 2x2 case, this can be solved by setting the determinant $|A - \lambda I| = 0$ and solving. This has been shown to equal specifically $\lambda = \frac{1}{2}\left(\mathrm{tr}(A) \pm \sqrt{\mathrm{tr}^2(A) - 4\mathrm{det}(A)}\right) = \frac{1}{2}(4 \pm 2) = 3, 1$ which can also validate the solution.

$$\begin{bmatrix} 1 & 0 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} 1-\lambda_1 & 0 \\ 1 & 3-\lambda_2 \end{bmatrix}$$

From this, we solve the determinant $(1-\lambda_1)(3-\lambda_1) = 0$; $\lambda_1 = 1$ and $\lambda_2 = 3$. To find the eigenvectors we solve:

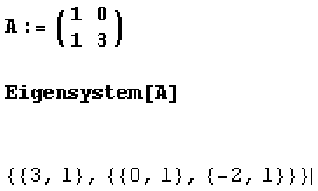
$$\begin{bmatrix} 1-\lambda_1 & 0 \\ 1 & 3-\lambda_2 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0$$

Using R or Mathematic I get the solutions described

b) Find the angle between the two eigenvectors. The angle is about 114 degrees.

```
> A=matrix(c(1,0,1,3),nrow=2,ncol=2,byrow=TRUE);
> eigen.A=eigen(A);
$values
[1] 3 1
$vectors
     [,1]       [,2]
[1,]    0  0.8944272
[2,]    1 -0.4472136
# u1 = eigen.A$vectors[1,];  ## FIRST EIGENVECTOR
# u2 = eigen.A$vectors[2,];  ## SECOND EIGENVECTOR
> finalAnswer = doAnalysis(eigen.A$vectors[1,],eigen.A$vectors[2,]);
> finalAnswer;
          ||u1|| ||u2|| theta pi radians  u1.u2
u1 and u2  0.894 1.095    114.1    0.63   -0.4
```

Problem 1 continued on next page. . .



$A := \begin{pmatrix}1 & 0\\1 & 3\end{pmatrix}$

```
Eigensystem[A]

{{3, 1}, {{0, 1}, {-2, 1}}}
```

Figure 1: Mathematica calculation of eigenvalues and eigenvectors: difference between two answers is mereley a rigid rotations; that is, 180 degrees

```
> finalAnswerMathematica;
              ||u1|| ||u2|| theta pi radians  u1.u2
     u1 and u2  1.000  2.236 63.400  0.350  1.000
```

Problem 2

Let $Y = XB$, where X is a nxp standardized data matrix, and B is a orthogonal matrix containing the standardized eigenvectors corresponding to the eigenvalues of the sample covariance matrix of X. Let D be a diagonal matrix with the corresponding eigenvalues along the diagonal.

$Y = XB$ **is analogous to** $Z = XU$

***a*)** Prove that the correlation of X is equivalent to the covariance matrix of X when X is standardized.

Proof. $S = \frac{1}{n-1} X_d' X_d$ where X_d is the differenced matrix $(x_i - \bar{x})$ defines the covariance of X.
$R = \frac{1}{n-1} X_s' X_s$ where X_s is the scaled or standardized matrix $(\frac{x_i - \bar{x}}{s_i})$ defines the correlation of X.
By property $R = D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$ the two forms can be demonstrated to be equal. When the data is standardized (mean=1, standard deviation=1), the diagonal elements that determine $D^{-\frac{1}{2}} = diag(\frac{1}{\sqrt{s_{ii}}}$ are all one, so $D^{-\frac{1}{2}}$ becomes the identity; therefore, $R = S$. □

***b*)** Find the mean of the columns of Y and show that it is always true.

Proof. $Y = XB$ is analogous to $Z = XU$ where Z represents the orthogonal projects of the eigenvectors U. X contains standardized data (mean=0, standard deviation = 1), so an orthogonal projection will likewise have the same mean. Remember, an orthogonal project is a linear combination of the x_i which all have mean=0; from expectation: $E[aX_1 + bX_2] = aE[X_1] + bE[X_2] = a(0) + b(0) = 0$. □

***c*)** Find the variance of the i^{th} columns of Y and show that it is always true.

Proof. $var(Z[i,]) = \lambda_i = var(PC_i)$ □

***d*)** Find the correlation of the first two columns of Y and show that it is always true.

Proof. $cor(Z[1,], Z[2,]) = 0$. Orthogonality implies no correlation between adjacent projections. The first projection is perpendicular to the second projection. $U'U = I$. □

***e*)** Find the maximum value of $var(a_1 \cdot X_1 + \ldots a_p \cdot X_p)$, in terms of the eigenvalues, for any constants $a_1, \ldots, a_p$, where X_i is the variable corresponds to the i^{th} column of X such that $a_1^2 + \ldots + a_p^2 = 1$.

Proof. $var(a_1 \cdot X_1 + \ldots a_p \cdot X_p) = a_1^2 \cdot var(X_1) + \ldots a_p^2 \cdot var(X_p)$. $Z = XU$ and $var(z_i) = \lambda_i$. $var(X_i) = 1$ because standardized, so maximum is 1. □

***f*)** Find the maximum value of $var(a_1 \cdot X_1 + \ldots a_p \cdot X_p)$, in terms of the eigenvalues, for any constants $a_1, \ldots, a_p$, where X_i is the variable corresponds to the i^{th} column of X such that $a_1^2 + \ldots + a_p^2 = 100$.

Proof. $var(a_1 \cdot X_1 + \ldots a_p \cdot X_p) = a_1^2 \cdot var(X_1) + \ldots a_p^2 \cdot var(X_p)$. $Z = XU$ and $var(z_i) = \lambda_i$. $var(X_i) = 1$ because standardized, so maximum is 100. □

***g*)** Express X in terms of Y and B.

Proof. $X = Z_s D^{\frac{1}{2}} U'$; D is found by determining the $cov(X)$ as seen in previous part above. $Z_s = XUD^{-\frac{1}{2}}$ is standardized so the variance is 1. □

***h*)** Express the correlation of X in terms of D and B.

Proof. $R = FF'$ where $F = UD^{\frac{1}{2}}$ □

Problem 3

Consider the following data set protein.txt which is available at
http://www.webpages.uidaho.edu/ stevel/519/Data/protein.txt

```
X=read.table('clipboard');
Pr = X[-1,-1]; # Pretty
P=as.numeric(as.matrix(Pr));
P=matrix(P,nrow=25,ncol=9);
rownames(Pr)=X[,1][-1];
     # colnames(P)=X[1,][-1]; #doesn't work
colnames(Pr)=c("RedMeat","WhiteMeat","Eggs","Milk","Fish","Cereals","Starch","Nuts","Fr&Veg");
```

a) Construct starplot for the data set protein.txt, which describes the protein intake in various countries in Europe. Do you recognize groups of countries with similar protein intake? It looks like Spain just eats more than everyone else! There appears to be other patterns, but not fully developed or understood. The variables are color coded based on the values of the second column (White Meat commonalities). You could argue that the Scandinavian countries have some dietary similarities: compare Norway, Sweden, and Finland. Let's do some data reduction to help identify underlying factors of dietary protein intake.

```
stars(Pr,key.loc=c(15,1),col.stars=2+round(P[26:50]));
```

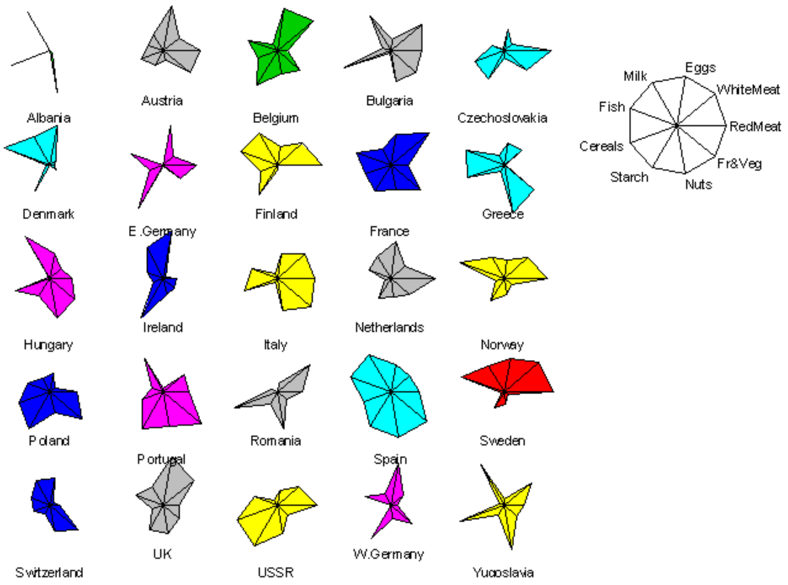


Figure 2: Star Plots of European Countries

b) Perform PCA to the scaled data set protein.txt.

```
Ps = scale(P);
P.princomp=princomp(Ps);
     rownames(P.princomp$loadings)=c("RedMeat","WhiteMeat","Eggs","Milk","Fish","Cereals","St
     rownames(P.princomp$scores)=X[,1][-1];
```

```
          summary(P.princomp);
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4    Comp.5
Standard deviation     1.9611680 1.2528366 1.0405781 0.9573283 0.6672967
Proportion of Variance 0.4451597 0.1816666 0.1253244 0.1060738 0.0515376
Cumulative Proportion  0.4451597 0.6268263 0.7521507 0.8582245 0.9097621
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
          plot(P.princomp$scores[,1],P.princomp$scores[,2]);
     text(P.princomp$scores[,1],P.princomp$scores[,2],cex=0.7,lwd=2, labels=X[,1][-1]);
     library(rgl);
     plot3d(P.princomp$scores[,1:3], type="s", radius=.1, col=rainbow(10));
     text3d(P.princomp$scores[,1:3],text=X[,1][-1]);
```

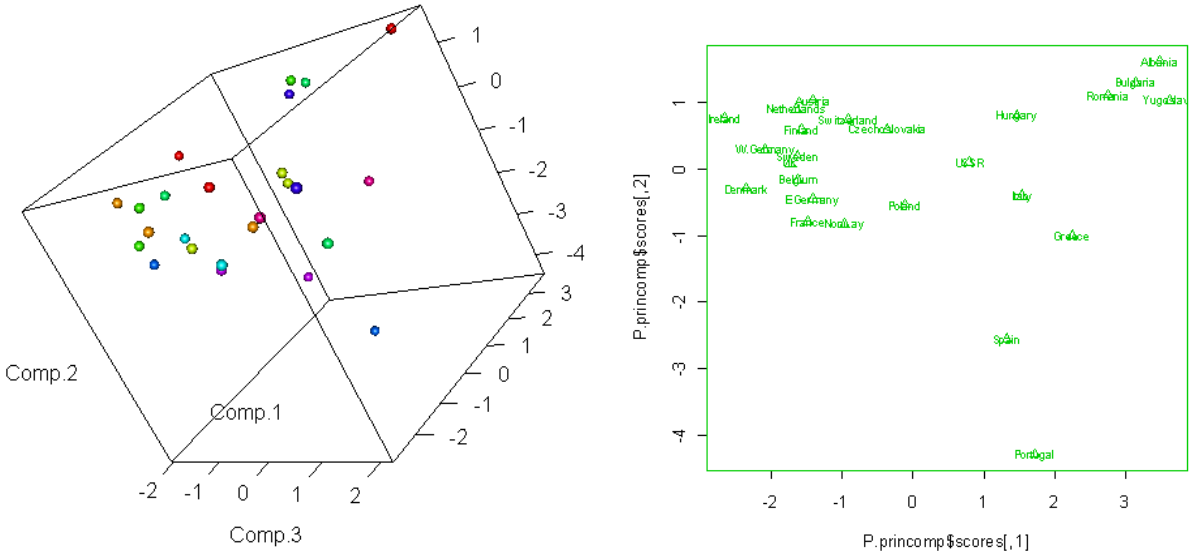


Figure 3: RGL and scatterplot

c) Give a possible interpretation for the first four principal components. Using the general rule of thumb $(cor(X, Z) = F$ where $f_{ij} > |0.4|$, I would interpret the first PC to represent staple proteins: Eggs, Cereal, Nuts. PC2 to represent Fish/Fruits and Vegetables (healthy proteins requiring accessibility). PC3 to represent WhiteMeat (and crossloads with Fruits and Vegetables). PC4 to represent Red Meat (again crossloading with Fruits and Vegetables).

```
> P.princomp$loadings
Loadings:
          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
RedMeat   -0.303         0.298  0.646 -0.322  0.460
WhiteMeat -0.311  0.237 -0.624         0.300  0.121
Eggs      -0.427        -0.182  0.313        -0.361
Milk      -0.378  0.185  0.386         0.200 -0.618
Fish      -0.136 -0.647  0.321 -0.216  0.290  0.137
```

```
Cereals    0.438  0.233               -0.238
Starch    -0.297 -0.353 -0.243 -0.337 -0.736 -0.148
Nuts       0.420 -0.143         0.330 -0.151 -0.447
Fr&Veg     0.110 -0.536 -0.408  0.462  0.234 -0.119
```

d) Generate a biplot of the first two principal components. Interpret the plot.

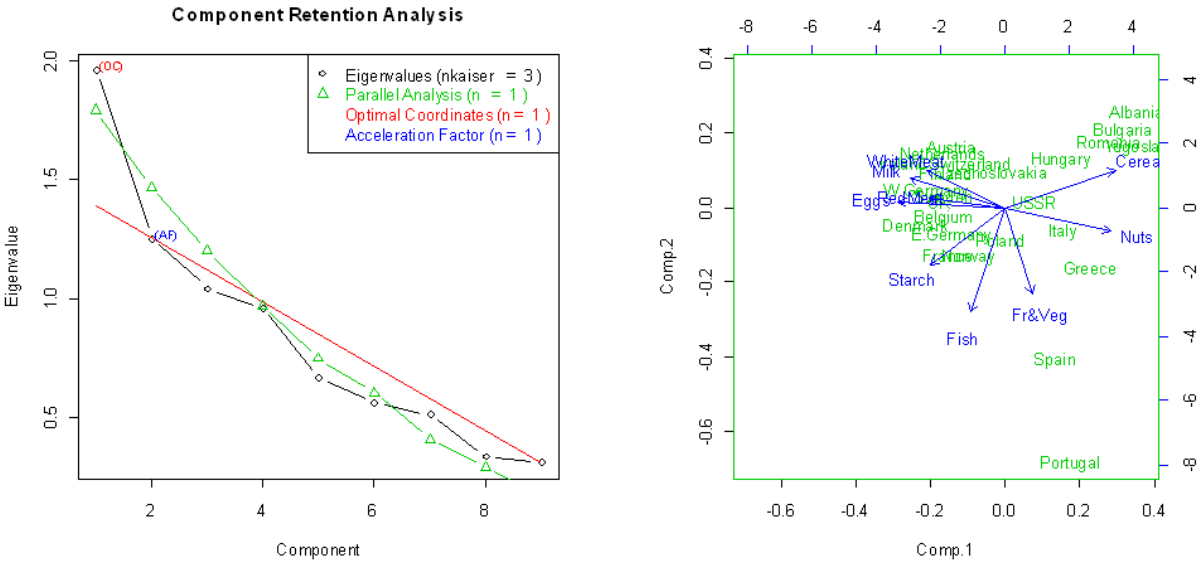


Figure 4: Scree Analysis and Biplot of PCA

e) Suppose we want to achieve some dimension reduction. Using the Horn's procedure, how many principal components would you use to describe these data? We can generate random samples or bootstrap from the data (sample from the data) and run comparable analyses. There appears to be a double elbow at 2 and 4, so the boostrapping would help smooth the data. I also use nFactors, a library that includes this Horn procedure (called "parallel" in the legend). Based on the crossing point (green onto original black), we would conclude that there is only one factor worth keeping - a general factor of protein intake habits in Europe: lots of cereals, nuts, and vegetables, not alot of eggs, milk or meat. Very different from Americans or Argentines!

```
> ## nFactors
> n=25;p=9;
> library(nFactors);
     > nResults = nScree(eig = as.numeric(P.princomp$sdev),aparallel = parallel(subject = n, var =
> plotuScree(as.numeric(P.princomp$sdev));
> plotnScree(nResults, main="Component Retention Analysis");
```