Stats 519: HW 5 - EFA (CH 5)

Due on March 27, 2009

Dr. Stephen Lee 1:30

Monte J. Shaffer

Problem 0

Can the PC analysis result generalize well outside the original sample data? To answer this question, let us consider bootstrapping the original data to assess the validity of the PC analysis as follows: Resample 50 observations with replacement and form the linear combination of the bootstrapped data using the 1st PC direction of the original data set (i.e., scale(gsp.share)). Compute the variance of this linear combination. Compare this variance with the variance of the 1st PC from the boostrapped data sample. Repeat this process 1000 times to produce Figure 4.18 (pg 120) Lattin et al. and comment on the validation of the PCA result on the data set gsp.share.

Program 1 PCA for 0

```
setwd("C:/latex/statsMultiVariate/datasets");
     gsp.share = read.table("GSP_SHARE.txt");
myColumns = c("AGRICULTURE","MINING","CONSTRUC","MFR_DUR","MFR_NON","TRANSPORT","COMMUN","UTILITIES","WH
myStates = gsp.share[,1];

Xs= scale(gsp.share[,2:14]);  rownames(Xs) = myStates;  colnames(Xs) = myColumns;
Xs.PCA=princomp(Xs);          rownames(Xs.PCA$loadings)=myColumns;  rownames(Xs.PCA$scores)=myStates;
     summary(Xs.PCA);

# PCA Variable Factor Map
library(FactoMineR);
result = PCA(Xs); # graphs generated automatically

biplot(Xs.PCA);

## nFactors
n=dim(Xs)[1];p=dim(Xs)[2];
library(nFactors);
nResults=nScree(eig = as.numeric(Xs.PCA$sdev),aparallel = parallel(subject=n,var=p)$eigen$qevpea);
plotnScree(nResults, main="Component Retention Analysis");

# r_1 is x*u_1/x*u_1*         # r_2 is xu_1*/xu_1

nsim = 1000;
Xorig = as.matrix(Xs);
u_1 = as.matrix(Xs.PCA$loadings[,1]);
r_1 = r_2 = c();
for(i in 1:nsim)
     {
     Xboot = Xs[sample(1:50,50,replace=T),];
     Xboot.PCA = princomp(Xboot);
     u_1_star = as.matrix(Xboot.PCA$loadings[,1]);

     r_1[i] = var(Xboot%*%u_1)          /    var(Xboot%*%u_1_star);
     r_2[i] = var(Xorig%*%u_1_star)     /    var(Xorig%*%u_1);
     }
hist(r_1,breaks=33,main=nsim);
hist(r_2,breaks=33,main=nsim);
```

Proof. Proof. Two ratios can be used to try and explain what is captured in this exercise:

$$r_1 = \frac{var(X^*_{boot} \cdot u_1)}{var(X^*_{boot} \cdot u^*_1)} \text{ and } r_2 = \frac{var(X_S \cdot u^*_1)}{var(X_S \cdot u_1)}$$

where $u_1 = as.matrix(X_S.PCA\$loadings[,1])$ and $u^*_1 = as.matrix(X_{boot}.PCA\$loadings[,1])$, the first column of eigen values. FIGURE 1 shows summary of the PCA of the original scales (X_S) data.

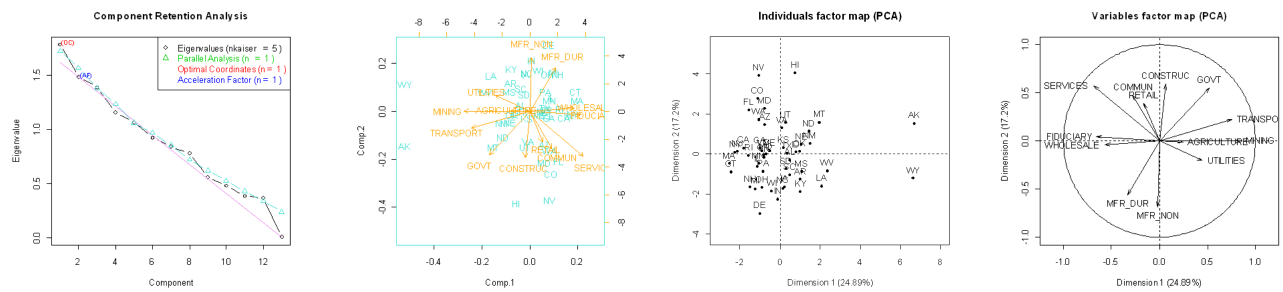

Figure 1: Basic PCA Explanation: Scree, BiPlot, Factor Maps using `princomp`

Results are reported for different bootstrap simulations (from 100 to 100,000).

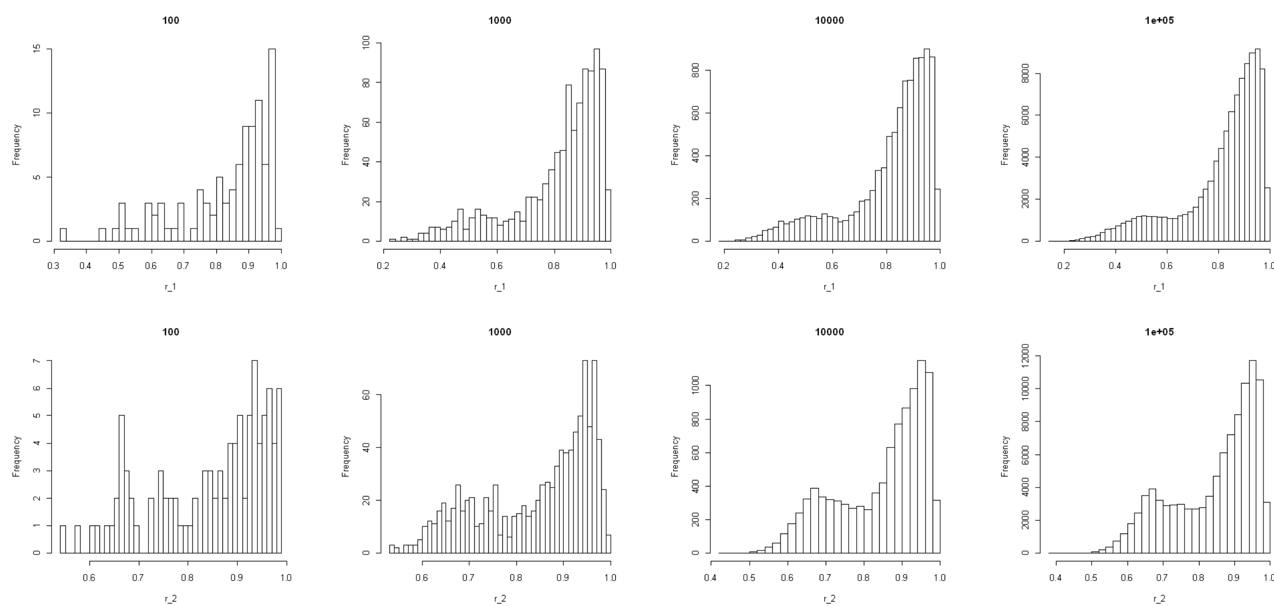

Figure 2: Comparing r_1 and r_2 for different number of bootstraps

From pg. 119–120, bootstrapping will work if we assume the data is representative of the true distribution. Is the variation explained by the first PC due to chance? To answer this we use one of the ratios above. If the ratio is close to 1 – we can conclude that the variation captured is systematic to the sample (and boot), thereby concluding generalizability. As seen above, the mass of each ratio is close to one, so if the assumptions underlying bootstrapping hold, we can establish generalizability. As noted in the text, the values for Alaska and Wyoming, if not included in the bootstrap draw, change the variance of PC1.

□

Problem 1

The MBA_CAR dataset may be found on the website. This has ratings on 16 attributes of 10 different car models 1 through 10, from MBA students. There are some missing values, so use read.table with the na.strings='.' option, and na.omit to remove the incomplete observations. [See problem 5.7, pg 169 of Lattin)] Save the car variable separately, and remove it from your data matrix. Conduct a principal components analysis on this data. How many components would you keep? Can you interpret the loadings on those components meaningfully? Does rotation of the loadings for the components you are keeping improve interpretability?

Program 2 PCA for 0

```
setwd("C:/latex/statsMultiVariate/datasets");
mba.cars = na.omit(read.table("MBA_CAR_ATTRIB.txt",na.strings='.'));
     myCars = c("BMW 328i","Ford Explorer","Infiniti J30","Jeep Grand Cherokee","Lexus ES300","Chrysler
     myAttributes = c("Exciting","Dependable","Luxurious","Outdoorsy","Powerful","Stylish","Comfortable

cars = mba.cars[,3:18];       colnames(cars) = myAttributes;

Cs= scale(cars);          colnames(Cs) = myAttributes;
Cs.PCA=princomp(Cs);      rownames(Cs.PCA$loadings)=myAttributes;
     summary(Cs.PCA);

plot(Cs.PCA$scores[,1],Cs.PCA$scores[,2]);
     library(rgl);
plot3d(Cs.PCA$scores[,1:3], type="s", radius=.1, col=rainbow(10));

# PCA Variable Factor Map
library(FactoMineR)
result = PCA(Cs) # graphs generated automatically

## nFactors
n=dim(Cs)[1];p=dim(Cs)[2];
library(nFactors);
     nResults = nScree(eig = as.numeric(Cs.PCA$sdev),aparallel = parallel(subject = n, var = p)$eigen$qe
plotuScree(as.numeric(Cs.PCA$sdev));  ## basic scree
plotnScree(nResults, main="Component Retention Analysis");

biplot(Cs.PCA);

Cs.PCA$loadings;
Cs.rotatedLoadingsVarimax = varimax(Cs.PCA$loadings); Cs.rotatedLoadingsVarimax$rot[,1:3];
Cs.rotatedLoadingsPromax = promax(Cs.PCA$loadings); Cs.rotatedLoadingsPromax$rot[,1:3];

Cs.rotated = Cs.rotatedLoadingsVarimax$rot[,1:3];
     rownames(Cs.rotated)=myAttributes; colnames(Cs.rotated)=c("Comp.1","Comp.2","Comp.3");
```

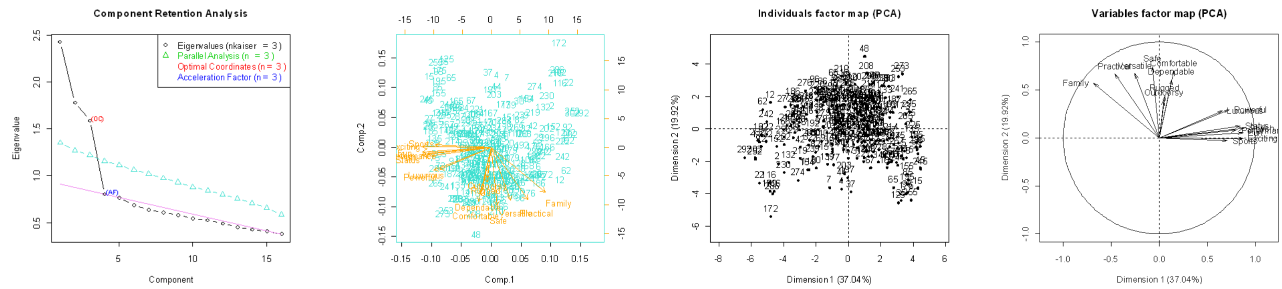

Figure 3: Basic PCA Explanation: Scree, BiPlot, Factor Maps using `princomp`

I would keep 3 PCs; since PCA only gives one solution, a rotation of the loadings should be rigid (in this case in 3-d space), and the rotation would help to find meaning in the 3 PCs; I could use varimax or promax rotation, which in this case are equivalent. Based on the rotations of the loadings, I would interpret the following components: PC1 is correlated with Exciting and Practical and Sporty. PC2 is correlated with Dependable, Stylish, Comfortable, Safe. PC3 is correlated Luxurious, Versatile, Status Symbol.

Loadings:				Rotated Loadings:		
	Comp.1	Comp.2	Comp.3	Comp.1	Comp.2	Comp.3
Exciting	-0.359		0.131	0.359	-0.030	-0.019
Dependable		-0.341	-0.268	0.031	0.417	-0.221
Luxurious	-0.270	-0.159	-0.291	-0.097	0.224	0.506
Outdoorsy		-0.221	0.506	0.100	-0.293	0.280
Powerful	-0.299	-0.166	0.143	-0.034	-0.027	-0.114
Stylish	-0.364			0.140	-0.387	-0.030
Comfortable		-0.391	-0.235	0.143	0.530	0.149
Rugged		-0.248	0.489	0.346	0.051	-0.028
Fun	-0.359			-0.318	0.046	-0.277
Safe		-0.417	-0.224	-0.236	-0.444	-0.047
Performance	-0.329		-0.146	0.169	-0.074	-0.255
Family	0.281	-0.320		0.121	-0.149	-0.053
Versatile	0.103	-0.376	0.240	-0.091	0.018	0.444
Sports	-0.292		0.310	-0.416	0.039	-0.027
Status	-0.347		-0.139	-0.224	0.163	-0.447
Practical	0.190	-0.373		0.508	0.006	-0.190

Problem 2

Now examine this same data with an exploratory factor analysis. How many factors should you use? How do you interpret them?

There are two techniques I could use to determine the number of factors to use. From the PCA analysis, I would conclude 3 factors. This is because the form of PCA ($R = FF'$ to $R \approx F_c F_c'$ where I keep c principal components) is similar to the model specification of EFA ($R = \Lambda_c \Lambda_c' + \Theta$). If the common factors are large and Θ as the specific factors are small (or close to zero), then the PCA and EFA are close to the same. Using the statistics from `factanal` we get conflicting results, since this is a χ^2 test of model fit.

Factors	d.f.	χ^2	p-value	Conclusion
1	104	1634.44	2.84e-273	Reject H_0: 1 solution is sufficient
2	89	994.86	5.19e-153	Reject H_0: 2 solutions are sufficient
3	75	228.56	1.98e-17	Reject H_0: 3 solutions are sufficient
4	62	144.33	1.63e-08	Reject H_0: 4 solutions are sufficient
5	50	90.91	0.00036	Reject H_0: 5 solutions are sufficient
6	39	51.97	0.0799	Fail to Reject H_0: 6 solutions are sufficient

Figure 4: Basic EFA Model Fit Using χ^2

The model fit requires 6 factors, but the PCA tests suggest 3. Model fit will be assumed going forward, but this demonstrates that maybe the data should not be reduced; commonalities and uniqueness might explain part of the problem:

```
> Cs.EFA.none = factanal(Cs,factors=6,rotation='none');
> Cs.EFA.varimax = factanal(Cs,factors=6,rotation='varimax');
```

```
> round(1-Cs.EFA$uniquenesses,digits=2);
   Exciting  Dependable   Luxurious   Outdoorsy    Powerful     Stylish Comfortable      Rugged
       0.79        0.44        0.72        0.77        0.60        0.86        0.60        0.83
     Family   Versatile      Sports      Status   Practical
       0.80        0.60        0.69        0.76        0.70
```

Since EFA may have multiple solutions, rotations should be used consistently to find meaning in the data reduction. F1 is Exciting, Luxurious, Power, Stylish, Fun, Performance, NOT family, Sporty, Status Symbol. F2 is Dependable, Luxurious, Comfortable, Safe. F3 is Outdoorsy, Rugged. F4 is Family, Versatile, Practical. F5 is Performance (cross-loaded with F1). F6 is Fun (also cross-loaded with F1). The common factor of F1 is an overall bias, in my opinion, to the fact that the ten models were mostly high-end with a few family cars. Also, two obvious SUVs are a factor. In the future, I would suggest doing either analysis on a class of cars: e.g., all SUVs, all sedans, etc. In marketing, this is how perceptual maps are generally created with attributes.

```
Varimax Rotated
Loadings:
            Factor1 Factor2 Factor3 Factor4 Factor5 Factor6
Exciting     0.855           0.142  -0.164
Dependable   0.104   0.640           0.132
Luxurious    0.579   0.530  -0.209  -0.169          -0.179
Outdoorsy                    0.858   0.157
Powerful     0.662   0.175   0.307           0.183
Stylish      0.889   0.135          -0.103          -0.182
Comfortable  0.149   0.739           0.156
Rugged                       0.894   0.148
Fun          0.903                  -0.123           0.390
Safe                 0.762           0.198   0.102
Performance  0.724   0.203  -0.172  -0.137   0.617
Family      -0.555   0.321   0.143   0.604
Versatile            0.193   0.432   0.606
Sports       0.699  -0.235   0.357  -0.105
Status       0.788   0.258  -0.120  -0.172   0.107  -0.145
Practical   -0.284   0.322   0.115   0.709
```

Problem 3

Rerun your final EFA model with the option scores= 'regression'. Construct a scatterplot or pairs plot of the scores, setting color or pch to the car ID variable. Use legend, text, or identify to label the cars. What do your plots tell you about the similarities and differences between the 10 car models?

I am going to do "paired plots" in addition to the pairs plots (of the varimax rotation with option scores='regression') so that I can interpret the color schema.

```
> Cs.EFA.varimax.regression = factanal(Cs,factors=6,rotation='varimax',scores='regression');
```

```
> legendV = c(1:10);
> plot(legendV,legendV,pch=myScores[,1],col=myScores[,1],xlab="",ylab="",main="LEGEND");
> text(legendV,legendV,pch=myScores[,1],col=myScores[,1],xlab="",ylab="",main="",labels=myCars);

> myScores=cbind(mba.cars[,2],Cs.EFA.varimax.regression$scores);
> pairs(myScores[,2:7],pch=myScores[,1],col=myScores[,1],cex=.1);
> library(MASS);
> parcoord(myScores[,2:7],pch=myScores[,1],col=myScores[,1]);

> myMeans = read.table('clipboard');
> colnames(myMeans)=c("CarID","n","Factor1","Factor2","Factor3","Factor4","Factor5","Factor6");
## clustering activity
> parcoord(myMeans[,3:8],pch=myScores[,1],col=myScores[,1]);
```

```
## paired plots
for(j in 2:7)
     {
     for(i in 2:7)
          {
          plot(myScores[,i],myScores[,j],pch=myScores[,1],col=myScores[,1],xlab=i-1,ylab=j-1);
          }
     }
```

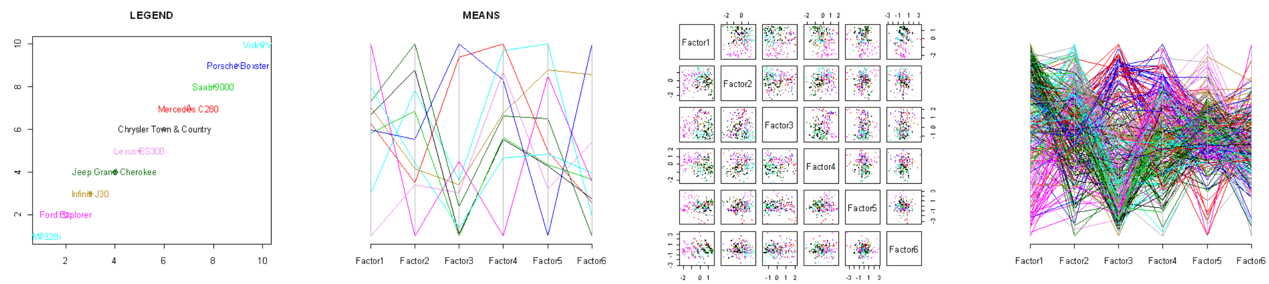

Figure 5: EFA Factors: Legend, Pairs, ParCoord, ParCoord with means

I also compared the means of the car models, which is a classification (clustering) technique based on the common factors. The means for each model can help us identify a given car, please see FIGURE 6 and MEANS of FIGURE 5.

Car ID	Car Name	N	Factor 1 Exciting, Luxurious, Power, Stylish, Fun, Performance, NOT family, Sporty, Status Symbol	Factor 2 Dependable, Luxurious, Comfortable, Safe	Factor 3 Outdoorsy, Rugged	Factor 4 Family, Versatile, Practical	Factor 5 Performance	Factor 6 Fun
1	BMW 328i	29	0.60	-0.22	-0.63	-0.36	-0.12	-0.03
2	Ford Explorer	29	-0.03	0.01	1.46	0.28	-0.70	0.47
3	Infiniti J30	28	-0.09	0.28	-0.73	-0.19	-0.20	-0.06
4	Jeep Grand Cherokee	30	0.07	-0.41	1.30	0.57	-0.11	-0.15
5	Lexus ES300	30	0.20	0.68	-0.70	-0.21	-0.21	-0.14
6	Chrysler Town and Country	31	-1.61	-0.43	-0.24	0.35	-0.37	0.09
7	Mercedes C280	27	0.40	0.93	-0.40	-0.01	0.13	-0.28
8	Saab 9000	30	0.30	-0.28	-0.14	0.01	0.47	0.35
9	Porsche Boxster	30	1.25	-0.93	0.12	-1.00	0.43	-0.07
10	Volvo V90	30	-0.98	0.48	-0.10	0.52	0.65	-0.20

Figure 6: Vehical Model Types and Factor Comparison

Problem 4

The PSYCH_TESTS dataset is a triangular correlation matrix for nine tests on 145 children, an expansion of the five-test data we looked at in class and in the text. Use read.tri to import this matrix and use it to answer question 5.1.

Analyze these data using factor analysis. How many factors are there? How would you interpret them? How do the results differ from the results based on five tests presented in the chapter?

```
setwd("C:/latex/statsMultiVariate/datasets");
read.tri = function(file)
     {
     x = scan(file);
     lx = length(x);
     d = (sqrt(8*lx+1)-1)/2;
     m = matrix(0, d, d);
     m[upper.tri(m, T)] = x;
     m = m + t(m) - diag(diag(m));
     return(m);
     }

R=read.tri("PSYCH_TESTS.txt");

psychQuestions = c("Visual perception","Cubes","Lozenges","Paragraph Comprehension","Sentence completio
psychQuestionShort = c("VP","Cub","Loz","PC","SC","WM","Add","CD","SCC");

     rownames(R)=psychQuestions;
     colnames(R)=psychQuestionShort;

     n=145;p=dim(R)[2];
R.e = eigen(R);
Lambda = R.e$values;
U = R.e$vectors;
## nFactors
library(nFactors);
nResults = nScree(eig = R.e$values,aparallel = parallel(subject = n, var = p)$eigen$qevpea);
plotuScree(R.e$values);
plotnScree(nResults, main="Component Retention Analysis");

psych.EFA.none = factanal(cov=R,factors=5,rotation='none');
psych.EFA.varimax = factanal(cov=R,factors=5,rotation='varimax');
```

Since the data is not available, the goodness of fit χ^2 and the scores are not available. The degrees of freedom are a determining variable to determine model fit. [http://tolstoy.newcastle.edu.au/R/e2/help/07/05/16135.html] The `criteria` variable is assumed to be related to some fit measure (AIC, BIC, LL, etc.); I will choose 5, even though PCA would suggest 3.

```
factanal(factors = 5, covmat = R, rotation = "varimax")

Uniquenesses:
   VP   Cub   Loz    PC    SC    WM   Add    CD   SCC
0.523 0.595 0.509 0.005 0.392 0.011 0.005 0.464 0.367

Loadings:
                         Factor1 Factor2 Factor3 Factor4 Factor5
Visual perception         0.213           0.409   0.510
Cubes                                     0.622
Lozenges                  0.187           0.605   0.281   0.105
Paragraph Comprehension   0.959           0.218          -0.130
Sentence completion       0.720   0.121   0.113   0.207   0.142
Word meaning              0.769           0.193           0.590
Addition                  0.147   0.980           0.114
Counting dots                     0.546   0.137   0.467
Straight-curved capitals  0.190   0.315   0.230   0.667

               Factor1 Factor2 Factor3 Factor4 Factor5
SS loadings      2.175   1.381   1.089   1.083   0.401
Proportion Var   0.242   0.153   0.121   0.120   0.045
Cumulative Var   0.242   0.395   0.516   0.636   0.681

The degrees of freedom for the model is 1 and the fit was 0.0019
```

Factor 1 is Verbal (Paragraph Comprehension, Sentence completion, word Meaning); Factor 2 is Addition; Factor 3 is Visual (Visual perception, Cubes, Lozenges); Factor 4 is Spatial (Visual perception, Counting dots, Straight-curved capitals); Factor 5 is Language (Word Meaning). These 5 factors explain 68% of the variance in the data. PCA keeps each component consistent when one is added or removed, EFA is not like that. Since more factors are chosen than in the book examples, we can pull out more of the meaning from the data collected. There are still Math and Language components, but the abilities are further refined and explained using more data and more factors.

```
## FROM http://tolstoy.newcastle.edu.au/R/e2/help/07/05/16135.html


multifactanal = function(factors=1:3, ...)
     {
     names(factors) = factors;
     ret = lapply(factors, function(factors)
          {
          try(factanal(factors=factors, ...))
          });
     class(ret) = "multifactanal";
     ret;
     }
```

```
summary.multifactanal = function(object,...)
     {
     do.call("rbind", lapply(object, summary.factanal));
     }

print.multifactanal = function(x,...)
     {
     ret = summary.multifactanal(x);
     print(ret, ...);
     invisible(ret);
     }

summary.factanal =function(object, ...)
     {
     if (inherits(object, "try-error"))
          {
          c(n=NA, items=NA, factors=NA, total.df=NA, rest.df=NA, model.df=NA, LL=NA, AIC=NA, AICc=NA, BI
          }
          else
               {
                    n = object$n.obs;
                    p = length(object$uniquenesses);
                    m = object$factors;


                    model.df = (p*m) + (m*(m+1))/2 + p - m^2;
                    total.df = p*(p+1)/2;
                    rest.df = total.df - model.df; # = object$dof
                    LL = -as.vector(object$criteria["objective"]);
                    k = model.df;
                    aic = 2*k - 2*LL;
                    aicc = aic + (2*k*(k+1))/(n-k-1);
                    bic = k*log(n) - 2*LL;
                    c(n=n, items=p, factors=m, total.df=total.df, rest.df=rest.df, model.df=model.df, LL=
               }
     }


multifactanal(factors=1:5, covmat=R, n.obs=145);

    n items factors total.df rest.df model.df           LL      AIC      AICc       BIC
1 145     9       1       45      27       18 -1.234756121 38.46951  43.89808  92.05072
2 145     9       2       45      19       26 -0.440387442 52.88077  64.77908 130.27585
3 145     9       3       45      12       33 -0.069084915 66.13817  86.35439 164.37038
4 145     9       4       45       6       39 -0.018929229 78.03786 107.75214 194.13047
5 145     9       5       45       1       44 -0.001949608 88.00390 127.60390 218.98018
```