

# EXPLORATORY FACTOR ANALYSIS IN R

JAN DE LEEUW

ABSTRACT. Meet the abstract. This is the abstract.

## 1. INTRODUCTION

Suppose  $R$  is a positive definite correlation matrix of order  $n$ . In Exploratory Factor Analysis (EFA) we want to find a diagonal matrix  $\Phi$  such that  $R - \Phi$  has rank  $r < n$ . Thus  $R - \Phi$  must have  $n - r$  eigenvalues equal to zero or, equivalently,  $R^{-\frac{1}{2}}\Phi R^{-\frac{1}{2}}$  must have  $n - r$  eigenvalues equal to one.

In a more restrictive formulation we want to find a non-negative diagonal matrix  $\Phi$  such that  $R - \Phi$  is positive semi-definite of rank  $r < n$ . This is equivalent to requiring that  $R^{-\frac{1}{2}}\Phi R^{-\frac{1}{2}}$  is positive semi-definite and has its  $n - r$  largest eigenvalues equal to one.

In Swain [1975] a family of techniques for exploratory factor analysis is proposed that is asymptotically equivalent to the multinormal maximum likelihood method.

Minimize

$$(1) \quad F(\phi) = \sum_{k=r+1}^n f(\lambda_k(\sum_{i=1}^n \phi_i u_i u_i'))$$

The  $\lambda_k$  are the eigenvalues of its matrix argument, ordered from large to small. The  $u_i$  are the columns of the matrix  $R^{-\frac{1}{2}}$ .

---

*Date:* Wednesday 25<sup>th</sup> January, 2012 — 23h 25min — Typeset in LUCIDA BRIGHT.

2000 *Mathematics Subject Classification.* 00A00.

*Key words and phrases.* Binomials, Normals, L<sup>A</sup>T<sub>E</sub>X.

In (1)  $f$  is any twice differentiable function with

$$f(1) = 0,$$

$$f'(1) = 0,$$

$$f''(1) = 1,$$

and

$$f'(x) < 0 \text{ if } x < 1,$$

$$f'(x) > 0 \text{ if } x > 1.$$

Or, equivalently,

$$f(x) = \frac{1}{2}(x - 1)^2 + o((x - 1)^2).$$

## APPENDIX A. CODE

```

1 swain<-function(s,p,fgh,tra=ident,ph=lsFac(s,p),eps=1e-6,
2   itmax=30,verbose=TRUE)
3 {
4 n<-nrow(s); es<-eigen(s); lb<-es$values; kk<-es$vectors
5 id<-1:(n-p); nn<-1:n; itel<-1
6 ss<-kk%*%((1/sqrt(lb))*t(kk))
7 f<-fgh$f; g<-fgh$g; h<-fgh$h
8 ft<-tra$f; fi<-tra$i; gt<-tra$g; ht<-tra$h; th<-fi(ph)
9 repeat{
10   vv<-eigen(ss%*%(ph*ss))
11   vl<-vv$values; vk<-vv$vectors; vd<-vl[id]
12   gh<-gt(th); gk<-ht(th)
13   ff<-sum(f(vd)); uu<-ss%*%vk
14   gg<-drop((uu[,id]^2)%*%g(vd))*gh
15   gm<-max(abs(gg))
16   hh<-matrix(0,n,n)
17   for (i in id) {
18     vli<-vl[i]; ui<-uu[,i]; uw<-outer(ui,ui)
19     for (j in nn) {
20       vlj<-vl[j]; uj<-uu[,j]
21       bij<-ifelse(i==j,h(vli),2*g(vli)/(vli-vlj))
22       hh<-hh+bij*uw*outer(uj,uj)
23     }
24   }
25   hh<-(hh*outer(gh,gh))+diag(gh*gk)
26   dr<-solve(hh,gg); ch<-max(abs(dr)); dc<-sum(gg*dr)
27   hv<-(eigen(hh,only.values=TRUE)$values)[n]
28   if (verbose)
29     cat("itel",formatC(itel,format="d",width=4),
30         " function",formatC(ff,format="f",digits=8,
31           width=12),
32         " maxgrad",formatC(gm,format="f",digits=8,
33           width=12),
34         " mingrad",formatC(dc,format="f",digits=8,
35           width=12),
36         " residual",formatC(dr,format="f",digits=8,
37           width=12),
38         " error",formatC(ch,format="f",digits=8,
39           width=12),
40         " condition",formatC(dc,format="f",digits=8,
41           width=12),
42         " singular",formatC(dc,format="f",digits=8,
43           width=12),
44         " rank",formatC(dc,format="f",digits=8,
45           width=12),
46         " nullspace",formatC(dc,format="f",digits=8,
47           width=12),
48         " eigenvalues",formatC(dc,format="f",digits=8,
49           width=12),
50         " eigenvalues",formatC(dc,format="f",digits=8,
51           width=12),
52         " eigenvalues",formatC(dc,format="f",digits=8,
53           width=12),
54         " eigenvalues",formatC(dc,format="f",digits=8,
55           width=12),
56         " eigenvalues",formatC(dc,format="f",digits=8,
57           width=12),
58         " eigenvalues",formatC(dc,format="f",digits=8,
59           width=12),
60         " eigenvalues",formatC(dc,format="f",digits=8,
61           width=12),
62         " eigenvalues",formatC(dc,format="f",digits=8,
63           width=12),
64         " eigenvalues",formatC(dc,format="f",digits=8,
65           width=12),
66         " eigenvalues",formatC(dc,format="f",digits=8,
67           width=12),
68         " eigenvalues",formatC(dc,format="f",digits=8,
69           width=12),
70         " eigenvalues",formatC(dc,format="f",digits=8,
71           width=12),
72         " eigenvalues",formatC(dc,format="f",digits=8,
73           width=12),
74         " eigenvalues",formatC(dc,format="f",digits=8,
75           width=12),
76         " eigenvalues",formatC(dc,format="f",digits=8,
77           width=12),
78         " eigenvalues",formatC(dc,format="f",digits=8,
79           width=12),
80         " eigenvalues",formatC(dc,format="f",digits=8,
81           width=12),
82         " eigenvalues",formatC(dc,format="f",digits=8,
83           width=12),
84         " eigenvalues",formatC(dc,format="f",digits=8,
85           width=12),
86         " eigenvalues",formatC(dc,format="f",digits=8,
87           width=12),
88         " eigenvalues",formatC(dc,format="f",digits=8,
89           width=12),
90         " eigenvalues",formatC(dc,format="f",digits=8,
91           width=12),
92         " eigenvalues",formatC(dc,format="f",digits=8,
93           width=12),
94         " eigenvalues",formatC(dc,format="f",digits=8,
95           width=12),
96         " eigenvalues",formatC(dc,format="f",digits=8,
97           width=12),
98         " eigenvalues",formatC(dc,format="f",digits=8,
99           width=12),
100        " eigenvalues",formatC(dc,format="f",digits=8,
101          width=12),
102        " eigenvalues",formatC(dc,format="f",digits=8,
103          width=12),
104        " eigenvalues",formatC(dc,format="f",digits=8,
105          width=12),
106        " eigenvalues",formatC(dc,format="f",digits=8,
107          width=12),
108        " eigenvalues",formatC(dc,format="f",digits=8,
109          width=12),
110        " eigenvalues",formatC(dc,format="f",digits=8,
111          width=12),
112        " eigenvalues",formatC(dc,format="f",digits=8,
113          width=12),
114        " eigenvalues",formatC(dc,format="f",digits=8,
115          width=12),
116        " eigenvalues",formatC(dc,format="f",digits=8,
117          width=12),
118        " eigenvalues",formatC(dc,format="f",digits=8,
119          width=12),
120        " eigenvalues",formatC(dc,format="f",digits=8,
121          width=12),
122        " eigenvalues",formatC(dc,format="f",digits=8,
123          width=12),
124        " eigenvalues",formatC(dc,format="f",digits=8,
125          width=12),
126        " eigenvalues",formatC(dc,format="f",digits=8,
127          width=12),
128        " eigenvalues",formatC(dc,format="f",digits=8,
129          width=12),
130        " eigenvalues",formatC(dc,format="f",digits=8,
131          width=12),
132        " eigenvalues",formatC(dc,format="f",digits=8,
133          width=12),
134        " eigenvalues",formatC(dc,format="f",digits=8,
135          width=12),
136        " eigenvalues",formatC(dc,format="f",digits=8,
137          width=12),
138        " eigenvalues",formatC(dc,format="f",digits=8,
139          width=12),
140        " eigenvalues",formatC(dc,format="f",digits=8,
141          width=12),
142        " eigenvalues",formatC(dc,format="f",digits=8,
143          width=12),
144        " eigenvalues",formatC(dc,format="f",digits=8,
145          width=12),
146        " eigenvalues",formatC(dc,format="f",digits=8,
147          width=12),
148        " eigenvalues",formatC(dc,format="f",digits=8,
149          width=12),
150        " eigenvalues",formatC(dc,format="f",digits=8,
151          width=12),
152        " eigenvalues",formatC(dc,format="f",digits=8,
153          width=12),
154        " eigenvalues",formatC(dc,format="f",digits=8,
155          width=12),
156        " eigenvalues",formatC(dc,format="f",digits=8,
157          width=12),
158        " eigenvalues",formatC(dc,format="f",digits=8,
159          width=12),
159      )
160    )
161  )
162  )
163  )
164  )
165  )
166  )
167  )
168  )
169  )
170  )
171  )
172  )
173  )
174  )
175  )
176  )
177  )
178  )
179  )
180  )
181  )
182  )
183  )
184  )
185  )
186  )
187  )
188  )
189  )
190  )
191  )
192  )
193  )
194  )
195  )
196  )
197  )
198  )
199  )
200  )
201  )
202  )
203  )
204  )
205  )
206  )
207  )
208  )
209  )
210  )
211  )
212  )
213  )
214  )
215  )
216  )
217  )
218  )
219  )
220  )
221  )
222  )
223  )
224  )
225  )
226  )
227  )
228  )
229  )
230  )
231  )
232  )
233  )
234  )
235  )
236  )
237  )
238  )
239  )
240  )
241  )
242  )
243  )
244  )
245  )
246  )
247  )
248  )
249  )
250  )
251  )
252  )
253  )
254  )
255  )
256  )
257  )
258  )
259  )
260  )
261  )
262  )
263  )
264  )
265  )
266  )
267  )
268  )
269  )
270  )
271  )
272  )
273  )
274  )
275  )
276  )
277  )
278  )
279  )
280  )
281  )
282  )
283  )
284  )
285  )
286  )
287  )
288  )
289  )
290  )
291  )
292  )
293  )
294  )
295  )
296  )
297  )
298  )
299  )
300  )
301  )
302  )
303  )
304  )
305  )
306  )
307  )
308  )
309  )
310  )
311  )
312  )
313  )
314  )
315  )
316  )
317  )
318  )
319  )
320  )
321  )
322  )
323  )
324  )
325  )
326  )
327  )
328  )
329  )
330  )
331  )
332  )
333  )
334  )
335  )
336  )
337  )
338  )
339  )
340  )
341  )
342  )
343  )
344  )
345  )
346  )
347  )
348  )
349  )
350  )
351  )
352  )
353  )
354  )
355  )
356  )
357  )
358  )
359  )
360  )
361  )
362  )
363  )
364  )
365  )
366  )
367  )
368  )
369  )
370  )
371  )
372  )
373  )
374  )
375  )
376  )
377  )
378  )
379  )
380  )
381  )
382  )
383  )
384  )
385  )
386  )
387  )
388  )
389  )
390  )
391  )
392  )
393  )
394  )
395  )
396  )
397  )
398  )
399  )
400  )
401  )
402  )
403  )
404  )
405  )
406  )
407  )
408  )
409  )
410  )
411  )
412  )
413  )
414  )
415  )
416  )
417  )
418  )
419  )
420  )
421  )
422  )
423  )
424  )
425  )
426  )
427  )
428  )
429  )
430  )
431  )
432  )
433  )
434  )
435  )
436  )
437  )
438  )
439  )
440  )
441  )
442  )
443  )
444  )
445  )
446  )
447  )
448  )
449  )
450  )
451  )
452  )
453  )
454  )
455  )
456  )
457  )
458  )
459  )
460  )
461  )
462  )
463  )
464  )
465  )
466  )
467  )
468  )
469  )
470  )
471  )
472  )
473  )
474  )
475  )
476  )
477  )
478  )
479  )
480  )
481  )
482  )
483  )
484  )
485  )
486  )
487  )
488  )
489  )
490  )
491  )
492  )
493  )
494  )
495  )
496  )
497  )
498  )
499  )
500  )
501  )
502  )
503  )
504  )
505  )
506  )
507  )
508  )
509  )
509  )
510  )
511  )
512  )
513  )
514  )
515  )
516  )
517  )
518  )
519  )
519  )
520  )
521  )
522  )
523  )
524  )
525  )
526  )
527  )
528  )
529  )
529  )
530  )
531  )
532  )
533  )
534  )
535  )
536  )
537  )
538  )
539  )
539  )
540  )
541  )
542  )
543  )
544  )
545  )
546  )
547  )
548  )
549  )
549  )
550  )
551  )
552  )
553  )
554  )
555  )
556  )
557  )
558  )
559  )
559  )
560  )
561  )
562  )
563  )
564  )
565  )
566  )
567  )
568  )
569  )
569  )
570  )
571  )
572  )
573  )
574  )
575  )
576  )
577  )
578  )
579  )
579  )
580  )
581  )
582  )
583  )
584  )
585  )
586  )
587  )
588  )
589  )
589  )
590  )
591  )
592  )
593  )
594  )
595  )
596  )
597  )
598  )
599  )
599  )
600  )
601  )
602  )
603  )
604  )
605  )
606  )
607  )
608  )
609  )
609  )
610  )
611  )
612  )
613  )
614  )
615  )
616  )
617  )
618  )
619  )
619  )
620  )
621  )
622  )
623  )
624  )
625  )
626  )
627  )
628  )
629  )
629  )
630  )
631  )
632  )
633  )
634  )
635  )
636  )
637  )
638  )
639  )
639  )
640  )
641  )
642  )
643  )
644  )
645  )
646  )
647  )
648  )
649  )
649  )
650  )
651  )
652  )
653  )
654  )
655  )
656  )
657  )
658  )
659  )
659  )
660  )
661  )
662  )
663  )
664  )
665  )
666  )
667  )
668  )
669  )
669  )
670  )
671  )
672  )
673  )
674  )
675  )
676  )
677  )
678  )
679  )
679  )
680  )
681  )
682  )
683  )
684  )
685  )
686  )
687  )
688  )
689  )
689  )
690  )
691  )
692  )
693  )
694  )
695  )
696  )
697  )
698  )
699  )
699  )
700  )
701  )
702  )
703  )
704  )
705  )
706  )
707  )
708  )
709  )
709  )
710  )
711  )
712  )
713  )
714  )
715  )
716  )
717  )
718  )
719  )
719  )
720  )
721  )
722  )
723  )
724  )
725  )
726  )
727  )
728  )
729  )
729  )
730  )
731  )
732  )
733  )
734  )
735  )
736  )
737  )
738  )
739  )
739  )
740  )
741  )
742  )
743  )
744  )
745  )
746  )
747  )
748  )
749  )
749  )
750  )
751  )
752  )
753  )
754  )
755  )
756  )
757  )
758  )
759  )
759  )
760  )
761  )
762  )
763  )
764  )
765  )
766  )
767  )
768  )
769  )
769  )
770  )
771  )
772  )
773  )
774  )
775  )
776  )
777  )
778  )
779  )
779  )
780  )
781  )
782  )
783  )
784  )
785  )
786  )
787  )
788  )
789  )
789  )
790  )
791  )
792  )
793  )
794  )
795  )
796  )
797  )
798  )
799  )
799  )
800  )
801  )
802  )
803  )
804  )
805  )
806  )
807  )
808  )
809  )
809  )
810  )
811  )
812  )
813  )
814  )
815  )
816  )
817  )
818  )
819  )
819  )
820  )
821  )
822  )
823  )
824  )
825  )
826  )
827  )
828  )
829  )
829  )
830  )
831  )
832  )
833  )
834  )
835  )
836  )
837  )
838  )
839  )
839  )
840  )
841  )
842  )
843  )
844  )
845  )
846  )
847  )
848  )
849  )
849  )
850  )
851  )
852  )
853  )
854  )
855  )
856  )
857  )
858  )
859  )
859  )
860  )
861  )
862  )
863  )
864  )
865  )
866  )
867  )
868  )
869  )
869  )
870  )
871  )
872  )
873  )
874  )
875  )
876  )
877  )
878  )
879  )
879  )
880  )
881  )
882  )
883  )
884  )
885  )
886  )
887  )
888  )
889  )
889  )
890  )
891  )
892  )
893  )
894  )
895  )
896  )
897  )
898  )
899  )
899  )
900  )
901  )
902  )
903  )
904  )
905  )
906  )
907  )
908  )
909  )
909  )
910  )
911  )
912  )
913  )
914  )
915  )
916  )
917  )
918  )
919  )
919  )
920  )
921  )
922  )
923  )
924  )
925  )
926  )
927  )
928  )
929  )
929  )
930  )
931  )
932  )
933  )
934  )
935  )
936  )
937  )
938  )
939  )
939  )
940  )
941  )
942  )
943  )
944  )
945  )
946  )
947  )
948  )
949  )
949  )
950  )
951  )
952  )
953  )
954  )
955  )
956  )
957  )
958  )
959  )
959  )
960  )
961  )
962  )
963  )
964  )
965  )
966  )
967  )
968  )
969  )
969  )
970  )
971  )
972  )
973  )
974  )
975  )
976  )
977  )
978  )
979  )
979  )
980  )
981  )
982  )
983  )
984  )
985  )
986  )
987  )
988  )
989  )
989  )
990  )
991  )
992  )
993  )
994  )
995  )
996  )
997  )
998  )
999  )
999  )

```



```

63         g=function(x) (x-1),
64         h=function(x) 1)
65
66 james<-list(f=function(x) 0.5*log(x)^2,
67                 g=function(x) log(x)/x,
68                 h=function(x) (1-log(x))/(x^2))
69
70 # parameter transformations
71
72 logit<-list(f=function(x) 1/(1+exp(-x)),
73                 i=function(y) log(y/(1-y)),
74                 g=function(x) {p<-1/(1+exp(-x)); p*(1-p)},
75                 h=function(x) {p<-1/(1+exp(-x)); p*(1-p)*(1-2*p
76                               )})
77
77 ident<-list(f=function(x) x,
78                 i=function(y) y,
79                 g=function(x) rep(1,length(x)),
80                 h=function(x) rep(0,length(x)))
81
82 square<-list(f=function(x) x^2,
83                  i=function(y) sqrt(y),
84                  g=function(x) 2*x,
85                  h=function(x) rep(2,length(x)))
86
87 expo<-list(f=function(x) exp(x),
88                 i=function(y) log(y),
89                 g=function(x) exp(x),
90                 h=function(x) exp(x))
91
92
93
94 swainNLM<-function(s,p,fgh,info=0,verbose=1) {
95 n<-nrow(s); es<-eigen(s); lb<-es$values; kk<-es$vectors
96 id<-1:(n-p); nn<-1:n; itel<-1
97 ss<-kk%*%((1/sqrt(lb)) * t(kk))

```

```

98  nlm(f=fSwain,p=lsFac(s,p),ss,id,fgf,info,print.level=
      verbose,hessian=TRUE)
99 }
100
101 fSwain <- function(ph, ss, id, fgh, info) {
102   f<-fgf$f; g<-fgf$g; h<-fgf$h; n<-length(ph); nn<-1:n
103   vv<-eigen(ss%*%(ph*ss))
104   v1<-vv$values; vk<-vv$vectors; vd<-v1[id]
105   ff<-sum(f(vd)); uu<-ss%*%vk
106   gg<-drop((uu[,id]^2)%*%g(vd))
107   hh<-matrix(0,n,n)
108   for (i in id) {
109     vli<-v1[i]; ui<-uu[,i]; uw<-outer(ui,ui)
110     for (j in nn) {
111       vlj<-v1[j]; uj<-uu[,j]
112       bij<-ifelse(i==j,h(vli),2*g(vli)/(vli-vlj))
113       hh<-hh+bij*uw*outer(uj,uj)
114     }
115   }
116   if (info > 0)
117     attr(ff, "gradient") <- gg
118   if (info > 1)
119     attr(ff, "hessian") <- hh
120   return(ff)
121 }
122
123 swain2<-matrix(
124   c(1.000,0.624,0.626,0.271,0.400,0.340,0.319,0.496,
125     0.624,1.000,0.573,0.285,0.263,0.185,0.340,0.396,
126     0.626,0.573,1.000,0.120,0.301,0.296,0.249,0.380,
127     0.271,0.285,0.120,1.000,0.157,0.239,0.270,0.253,
128     0.400,0.263,0.301,0.157,1.000,0.524,0.582,0.560,
129     0.340,0.185,0.296,0.239,0.524,1.000,0.563,0.553,
130     0.319,0.340,0.249,0.270,0.582,0.563,1.000,0.651,
131     0.496,0.396,0.380,0.253,0.560,0.553,0.651,1.000),
132   ,8,8)

```

```
132  
133 harman23<-Harman23.cor$cov  
134  
135 harman74<-Harman74.cor$cov
```

## REFERENCES

- A.J. Swain. A Class of Factor Analysis Estimation Procedures with Common Asymptotic Sampling Properties. *Psychometrika*, 40:315-335, 1975.

DEPARTMENT OF STATISTICS, UNIVERSITY OF CALIFORNIA, LOS ANGELES, CA 90095-1554

*E-mail address*, Jan de Leeuw: `deleeuw@stat.ucla.edu`

*URL*, Jan de Leeuw: <http://gifi.stat.ucla.edu>