Stata example

Principle component analysis (PCA): Construct economic activity indicators for India

Step 1: loading the time series dataset with high-frequency indicators (data should be in a wide format)

Date variable	Indicator1	Indicator2	Indicator3	Indicator4	…
2020m1					…
2020m2					…
2020m3					…
2020m4					…
…	…	…	…	…	…

.clear all

.set more off

.use *HF_IND_pcaanalysis_wide*

.local indicators *electricityg_gwh ewaybilltot_lcu expnonoil_usd_mln impnonoil_usd_mln ip_2010usd_sa pmmanu_indx_nsa petroleumcons_tonneth portcargo_tonnemln railfreight_tonnemn crudesteelprod_tonneth ipiconstr_2011 carregistr_u_sa pmserv_indx_nsa aircargotraff_tonneth airpassngtraff_p*

.describe `indicators'

variable name	storage type	display format	value label	variable label
electricityg_~h	double	%10.0g		Electricity generation, GWh
ewaybilltot_lcu	double	%10.0g		E-way bill total
expnonoil_usd~n	double	%10.0g		Exports non oil, USD mln
impnonoil_usd~n	double	%10.0g		Imports non oil, USD mln
ip_2010usd_sa	double	%10.0g		Industrial Production, constant 2010 USD, sa
pmmanu_indx_nsa	double	%10.0g		Manufacturing PMI, nsa
petroleumcons~h	double	%10.0g		Petroleum consumption
portcargo_ton~n	double	%10.0g		Port cargo traffic, tonne mln
railfreight_t~n	double	%10.0g		Railway Traffic (Container Service),Tonne mln
crudesteelpro~h	double	%10.0g		Crude steel production, tonne thousand
ipiconstr_2011	double	%10.0g		Industrial Production Index: Infrastructure & Construction Goods, 2011-12=100
carregistr_u_sa	double	%10.0g		Car registrations
pmserv_indx_nsa	double	%10.0g		Services PMI, nsa
aircargotraff~h	double	%10.0g		Air cargo traffic, tonne th
airpassngtraf~p	double	%10.0g		Air passenger traffic, person

.correlation `indicators'

	electr~h	ew~t_lcu	expnon~n	impnon~n	ip_201~a	pmma~nsa	petrol~h	portca~n	railfr~n	crudes~h
electricit~h	1.0000									
ewaybillto~u	0.9147	1.0000								
expnonoil_~n	0.9655	0.9689	1.0000							
impnonoil_~n	0.7758	0.8749	0.8576	1.0000						
ip_2010usd~a	0.9293	0.9970	0.9834	0.8885	1.0000					
pmmanu_i~nsa	0.8448	0.9690	0.9107	0.8394	0.9563	1.0000				
petroleumc~h	0.9830	0.9475	0.9864	0.8663	0.9632	0.8971	1.0000			
portcargo_~n	0.5457	0.7588	0.6849	0.8267	0.7513	0.8590	0.6785	1.0000		
railfreigh~n	0.6671	0.8486	0.7680	0.9592	0.8427	0.8691	0.7750	0.9090	1.0000	
crudesteel~h	0.9291	0.9869	0.9814	0.8937	0.9918	0.9690	0.9723	0.7997	0.8537	1.0000
ipicons~2011	0.9573	0.9818	0.9862	0.8747	0.9875	0.9565	0.9865	0.7552	0.8234	0.9958
carregistr~a	0.5607	0.8240	0.7062	0.8611	0.8086	0.8022	0.6545	0.7803	0.9114	0.7735
pmserv_i~nsa	0.7905	0.9565	0.8861	0.8825	0.9447	0.9886	0.8655	0.9068	0.9216	0.9554
aircargotr~h	0.8163	0.9692	0.9058	0.9228	0.9613	0.9777	0.8888	0.8837	0.9426	0.9643
airpassngt~p	0.6402	0.8600	0.7712	0.9448	0.8530	0.8689	0.7508	0.9008	0.9860	0.8508

	ipi~2011	carreg~a	pmse~nsa	aircar~h	airpas~p
ipicons~2011	1.0000				
carregistr~a	0.7290	1.0000			
pmserv_i~nsa	0.9316	0.8691	1.0000		
aircargotr~h	0.9431	0.8948	0.9934	1.0000	
airpassngt~p	0.8113	0.9608	0.9297	0.9491	1.0000

Note: the more correlated of the data, the better to apply PCA

Step 2: Generate weights (loadings) by PCA

Firstly, we need to determine the number of components we want to retain. One of the commonly used criteria is the eigenvalue-one criterion, also known as the Kaiser criterion (Kaiser, 1960). With this approach, we retain and interpret any component with an eigenvalue greater than 1, which means any retained component should account for at least as much variation as any of the original indicator. Stata option *mineigen* allows us to retain components with eigenvalues larger than 1. Yet, in practice, we may want to use the scree plot of the eigenvalues to examine if there is a "break" in the plot with the remaining components explaining considerably less variation.

.pca `indicators', mineigen (1)

. screeplot, yline(1)

Principal components/correlation

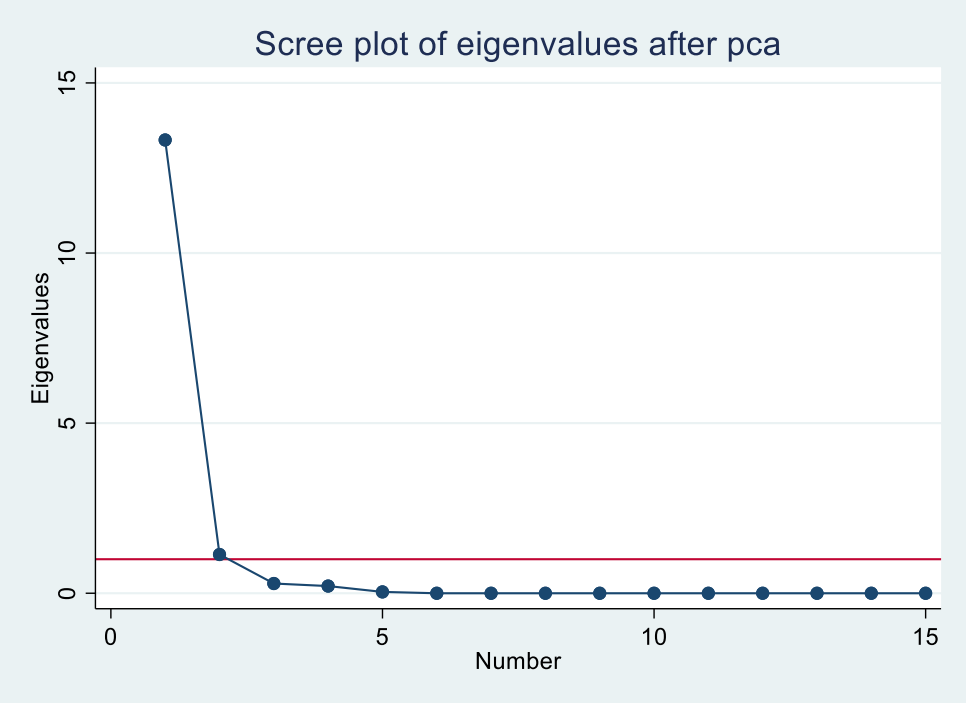
Number of obs	=	6
Number of comp.	=	2
Trace	=	15
Rho	=	0.9642

Rotation: (unrotated = principal)

Component	Eigenvalue	Difference	Proportion	Cumulative
Comp1	13.3241	12.185	0.8883	0.8883
Comp2	1.13909	.853385	0.0759	0.9642
Comp3	.285704	.0743605	0.0190	0.9833
Comp4	.211343	.171582	0.0141	0.9973
Comp5	.039761	.039761	0.0027	1.0000
Comp6	0	0	0.0000	1.0000
Comp7	0	0	0.0000	1.0000
Comp8	0	0	0.0000	1.0000
Comp9	0	0	0.0000	1.0000
Comp10	0	0	0.0000	1.0000
Comp11	0	0	0.0000	1.0000
Comp12	0	0	0.0000	1.0000
Comp13	0	0	0.0000	1.0000
Comp14	0	0	0.0000	1.0000
Comp15	0	.	0.0000	1.0000

Principal components (eigenvectors)

Variable	Comp1	Comp2	Unexplained
electricit~h	0.2396	-0.4430	.01123
ewaybillto~u	0.2691	-0.1213	.01844
expnonoil_~n	0.2596	-0.2786	.01333
impnonoil_~n	0.2571	0.1377	.098
ip_2010usd~a	0.2691	-0.1475	.01022
pmmanu_i~nsa	0.2661	-0.0189	.05618
petroleumc~h	0.2568	-0.3051	.01538
portcargo_~n	0.2331	0.3628	.1262
railfreigh~n	0.2532	0.3160	.03218
crudesteel~h	0.2702	-0.1420	.004006
ipicons~2011	0.2665	-0.2124	.002285
carregistr~a	0.2349	0.3759	.1036
pmserv_i~nsa	0.2681	0.1040	.03006
aircargotr~h	0.2718	0.0939	.00585
airpassngt~p	0.2532	0.3458	.009794



Note: The PCA reveals the first and second principle component explains 89%, 96% of the variation of the high-frequency indicators, respectively. In this case, as the second principle is barely above 1, we could only retain the first principle component to explain the most of variation in the data. In Stata, the option components (#) with the command pca could keep the maximum number of principle components you want to retain.

There is a trade-off between simplicity (retaining as few as possible components) and completeness (explaining the variation in the data as much as possible)!

The components could be rotated to simplify the structure of loadings matrix and facilitate the interpretation results. (Kaiser 1960, Abdi 2010) The Kaiser's varimax rotation is an orthogonal rotation with the purpose to maximize the squared loadings variance across indicators summed over all components. With the option *blanks(#)*, the loadings are shown blank for those indicators with loadings less than #. Thus, it reported the indicators for each component with the highest correlations with the common score.

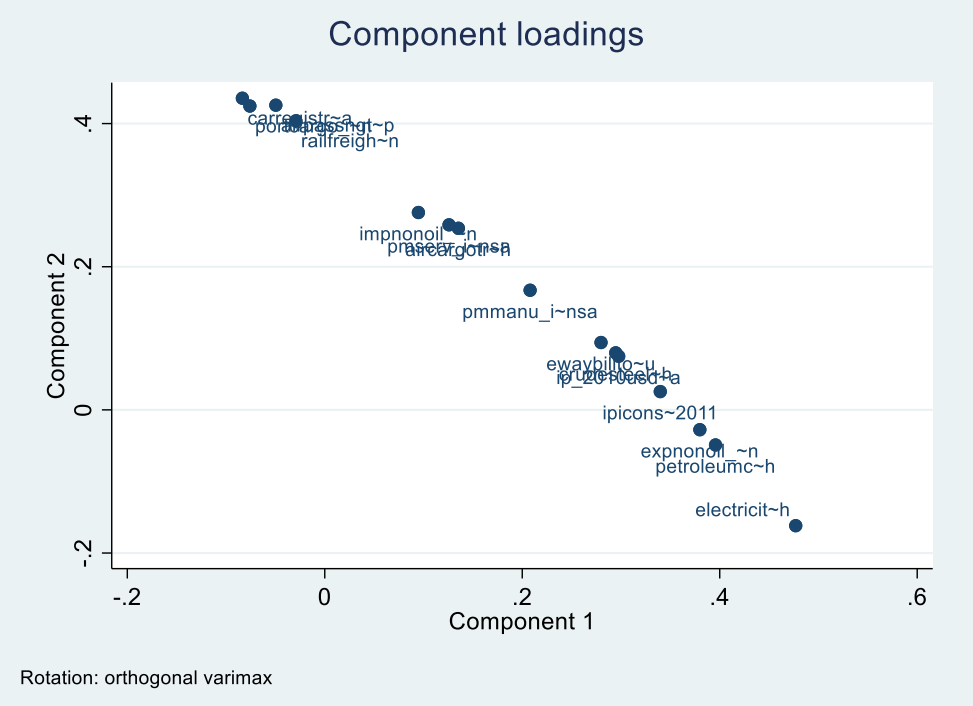
.rotate, varimax blanks(*.3*)

Rotated components (blanks are abs(loading)<.3)

Variable	Comp1	Comp2	Unexplained
electricit~h	0.4770		.01123
ewaybillto~u			.01844
expnonoil_~n	0.3799		.01333
impnonoil_~n			.098
ip_2010usd~a			.01022
pmmanu_i~nsa			.05618
petroleumc~h	0.3957		.01538
portcargo_~n		0.4245	.1262
railfreigh~n		0.4038	.03218
crudesteel~h			.004006
ipicons~2011	0.3398		.002285
carregistr~a		0.4354	.1036
pmserv_i~nsa			.03006
aircargotr~h			.00585
airpassngt~p		0.4257	.009794

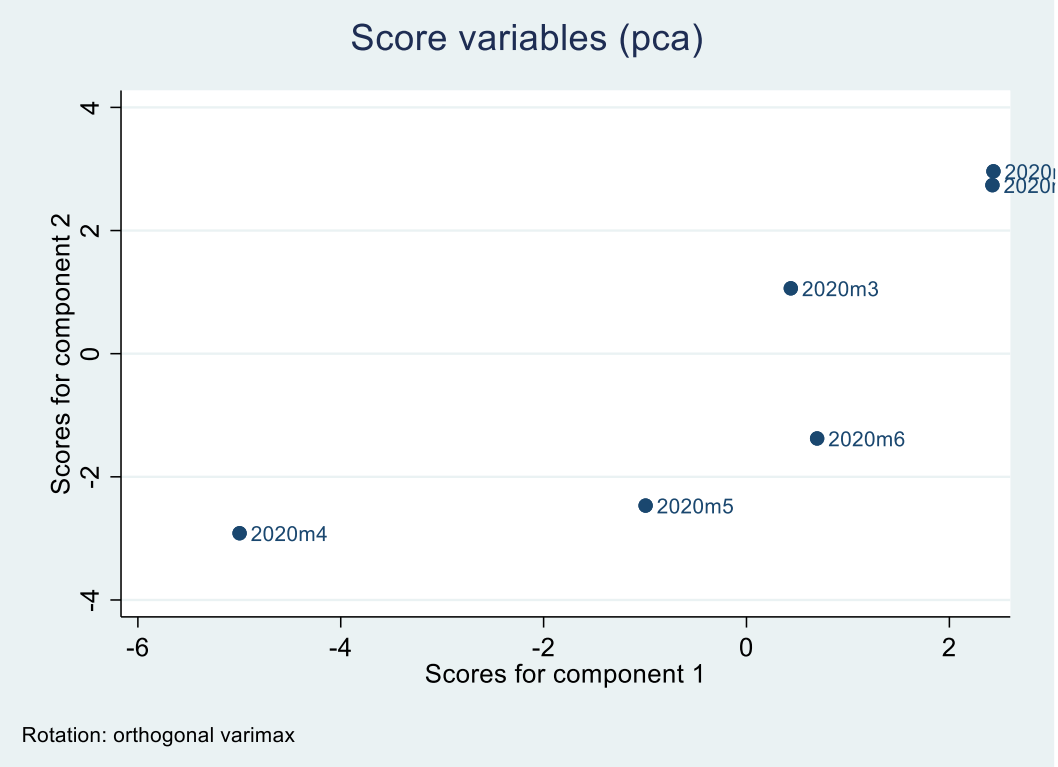
If we retain more than one principle component, we could scatter plot the loadings for the first two principle components and score variables. It tells which indicators have more loadings for each component and check the outlier of observations.

.loadingplot



. local id *date_m*

. scoreplot, mlabel(`id')



Note: In my dataset, the date variable is date_m. It tells 2020m4 is an outlier for the score variables, which consistent with the fact the adverse effect of COVID-19 or related lockdown measures weights on economic activities the most in April in India this year.

Then, we could generate a scoring coefficients matrix with the same elements of loadings. The dimension of matrix A is determined by the number of indicators and components.

. predict *pc1 pc2*, score

.return list

.mat A=r(scoef)

.mat list A

Scoring coefficients for orthogonal varimax rotation
sum of squares(column-loading) = 1

Variable	Comp1	Comp2
electricit~h	0.4770	-0.1618
ewaybillto~u	0.2798	0.0941
expnonoil_~n	0.3799	-0.0277
impnonoil_~n	0.0948	0.2757
ip_2010usd~a	0.2976	0.0749
pmmanu_i~nsa	0.2080	0.1671
petroleumc~h	0.3957	-0.0490
portcargo_~n	-0.0759	0.4245
railfreigh~n	-0.0293	0.4038
crudesteel~h	0.2947	0.0797
ipicons~2011	0.3398	0.0256
carregistr~a	-0.0834	0.4354
pmserv_i~nsa	0.1258	0.2585
aircargotr~h	0.1354	0.2537
airpassngt~p	-0.0495	0.4257

A[15,2]

	Comp1	Comp2
electricit~h	.47697559	-.1618025
ewaybillto~u	.2797758	.0940735
expnonoil_~n	.37985707	-.02770954
impnonoil_~n	.09482913	.27574661
ip_2010usd~a	.29763061	.07486519
pmmanu_i~nsa	.20795775	.16708388
petroleumc~h	.39574485	-.04903132
portcargo_~n	-.07586772	.42450024
railfreigh~n	-.02925846	.40382533
crudesteel~h	.29467064	.07970228
ipicons~2011	.33980974	.02555407
carregistr~a	-.08343226	.43538588
pmserv_i~nsa	.12584905	.25853554
aircargotr~h	.13536647	.25368597
airpassngt~p	-.04954814	.42567751

Step 3: Construct principle components (economic activity indicators)

To compute principle components, we use loadings (scoring coefficients) of indicators as weights.

```
.local a=rowsof (A)
.local b=colsof (A)
.forvalues j=1/`b'{
                gen component`j'=0
                }

                 foreach var of varlist `indicators' {
                        forvalues i = 1/`a' {
                        forvalues j = 1/`b' {
                        replace component`j'=component`j'+ `var'* A[`i',`j']
                        }
                        }
                 }
```

Last but not least, standardize the economic activity indicators with January 2020 equal to 100.

```
. foreach var of varlist component* {
        gen `var'_ind=(`var'/`var'[1])*100

        }
```