

Developing R software for simultaneous estimation of Q- and R-mode Factor Analyses using spatial and non spatial data

George Owusu
Department of Geography and Resource Development, University of Ghana, P O. Box LG 59, Accra,Ghana
* E-mail of the corresponding author: *owusugeorge@ug.edu.gh*

Abstract

Simultaneous use of R- and Q-mode Factor Analysis is a powerful similarity measurement among and between variables and objects of a continuous data, but its availability is lacking in R statistical software environment. I have developed a new R package called qrfactor that can perform Factor Analysis on spatial and non spatial data. The package contains one function called qrfactor() that can perform various versions of Factor Analyses such as PCA, R-mode Factor Analysis, Q-mode Factor Analysis, Simultaneous R- and Q-mode Factor Analysis, Principal Coordinate Analysis, as wells as Multidimensional Scaling (MDS) and cluster analysis. The qrfactor() function returns values such as eigenvalues, eigenvectors, loadings, scores, and indices. Unlike other R package factor analysis functions, plot.qrfactor() offers several annotated biplots for all possible combinations of eigenvectors, loadings, and scores as well as the possibility of plotting about 60 maps in gray and full colour scales. The empirical and Eckhart–Young theorem evaluations show that 'qrfactor' package is mathematically correct in estimation of simultaneous R-and Q-mode Factor Analysis. The results are also in agreement with the results of other classical statistical functions and packages. Using one function to estimate various dimensions of factor analyses reduces the learning curve in R environment.

Keywords: GIS, qrfactor, loadings, Multi-dimensional, R package, Factor scores, Cluster Analysis, Eckhart–Young, maps

1. Introduction

Many scientists normally work with continuous data that is characterized by multiple column variables and row observations. Developing a common space to analyse and display inter-variables, inter-objects, and variable-object relationships of a ratio data is important (de Mooy et al., 1988). Simultaneous computation of R- and Q-mode Factor Analysis readily offers solutions to these similarity measurements and classification (Engel et al., 1988; Walden et al., 1992). R-mode Factor Analysis has been extensively used to detect inter variables similarity of continuous data (Hair et al., 1998; Khan & Tewari, 2011). A combined R-mode and Q-mode has also been used to detect inter variables and inter objects relationships of a data, as well as relationships between variables and objects of a continuous data that few multivariate techniques can match (Davis, 2002).

R base (R Development, 2011) package mostly offers R-mode factor analysis techniques such as princomp() and prcomp() for PCA; and factanal() for maximum likelihood factor analysis. The R contributed packages also offer R-mode factor analysis techniques such as principal() in **psych** package (Revelle, 2011) for PCA; PCA() of FactoMineR package for PCA and Factor Analysis (Sébastien Lê et al., 2008). Apart from these functions mostly performing R-mode Factor Analysis, none of them performs simultaneous R- and Q-mode Factor Analysis. The PCA() function from FactoMineR package separately performs and plots R- and Q-mode Factor analyses; it is therefore not possible to compare variables to objects as well as observing the contributions of the variables for each dimension or coordinate of the reduced data. The biplot.princomp() function in R base package uses two different scales to display observations and variables; making comparison between observation and variables

impossible. The proposed plot.qrfactor() function in qrfactor() package (Owusu, 2011) offers similarity plots for all combinations variables and objects of factor analysis values such as eigenvectors, loadings, scores, coordinates, and dimensions.

In spatial data analysis, R base comes with many functions; for example, 'spatial' package (Ripley, 2011) is used for reading, visualizing, and analysing geographical data (Bivand, 2011). There are also several R contributed packages that supplement the R base packages for handling and analysing spatial data. A data becomes spatial when its coordinates are organized in a special way so that it is distinguished from other numbers such as time series data (Bivand et al., 2008). A joint effort from Bivand et al (2008) led to the writing of 'sp' package (Pebesma & Bivand, 2011) to specify the structure and definition of organisation of spatial data. The 'sp' package provides functions – such as spplot(), bubble() – for plotting of points, lines, polygons, and grids data. There have been several packages including rgdal, gstat, and maptools that depend on sp package for spatial analysis. Recently the 'raster' package (Hijmans & Etten, 2011) has brought a major 'extension of spatial data classes to virtualize access to large rasters, permitting large objects to be analyzed, and extending the analytical tools available for both raster and vector data' (Bivand, 2011).

This paper introduces 'qrfactor', an R package that can simultaneously measure similarities among and between i) objects ii) variables and iii) object - variable relationships of a continuous spatial and non spatial data. Scores and loadings of the data matrix are some of the return values of the package. Several map plots and scatter plots are offered by the package.

2. Methodology

2.1 Data Scaling

Davis (2002:566) discussed four main methods that researchers have been using to develop a scaled data (W) from original data (X) in order to compute Q- and R-mode FA (Factor Analysis) (Kulkami, 2012). The simplest method is to separately compute R-mode FA using the minor product of a scaled X data matrix (W'W) and Q-mode FA from the major product WW' and plot them together on the same axes. This method does not give correct similarity between objects and variables because different scaling techniques have been used to compute R-mode and Q-mode FAs from the original X data. This method is never implemented in 'qrfactor' package.

The second scaling method is to standardize the original data by centring each value and dividing it by column standard deviations. This method, applied in PCA() function (in FactoMineR package), princomp() and prcomp() in the base R, does not perform standardization for Q-mode FA because each element in the data matrix has also not been divided by square root of the sum of squares of rows, as required by Q-mode FA (Davis, 2002:566). This method is implemented in 'qrfactor' package by setting 'scale' parameter in the qrfactor function to 'normal': qrfactor(data, scale='normal').

Third, some researchers (Davis, 2002:566) avoid the scaling controversies by using the original X data matrix and computing minor (X'X) and major (XX') products for R- and Q-mode FA respectively. This 'raw-data' methodology, though mathematically correct, is very sensitive to measurement units of the variables, avoiding variance-covariance of the variables. In practice few researchers use it but because this technique is mathematically correct, there is an option in 'qrfactor' package to use it by setting the 'scale' parameter to 'data': qrfactor(data, scale='data').

Finally, Davis (2002:567) outlined two main scaling methods that R- and Q-mode FA can rely on for their

correct simultaneous estimations. The first method scales the original data (X) into a scaled data (W) by centring the elements of a data matrix X and divides it by the square root of total number of observations (n) as:

$$w_{ij} = \frac{x_{ij} - \overline{X_j}}{\sqrt{n}} \qquad (1)$$

where w_{ij} is the scaled element from X data matrix; x_{ij} is an element in X matrix data, and $\overline{X}$ is the column mean of X data matrix.

After using equation (1) to scale the original data X, the minor product matrix W'W contains variances and covariances of the variables and at the same time, the major product WW' being equivalent to the principal coordinates of a matrix for Q-mode FA (Davis, 2002:567). The Variables and observations can therefore be compared on the same dimensions. There is an option in using equation (1) to scale data in 'qrfactor' package by setting 'scale' input parameter to 'n': qrfactor(data, scale='n').

The second standardization method also adopted from Davis (2002:568) is the division of a centring element of X data by the product of column standard deviation and square root of total number of observations (n) as:

$$w_{ij} = \frac{x_{ij} - \overline{X_j}}{sj\sqrt{n}} \qquad (2)$$

where w_{ij} is the scaled element from X data matrix, x_{ij} is an element in X matrix data, sj is column standard deviations of X. Like equation (1) the minor product matrix W'W contains variances and covariances of the variables in a standardized form and at the same time the major product WW' being equivalent to the principal coordinates of a matrix for Q-mode FA (Davis, 2002:568). There is an option in using equation (2) in 'qrfactor' package by setting 'scale' input parameter to 'sd': qrfactor(data, scale='sd'). This is also the default data scaling method in the package because loadings range from 0 to 1.

2.2 Matrix Algebra and Eckhart-Young Theorem

The 'qrfactor' package uses a matrix algebra and Eckhart–Young theorem, as described in Davis (2002:568), to compute simultaneous Q- and R-mode FA. Once the original data matrix (X) is scaled (W) simultaneous Q- and R-mode FA was estimated using Matrix Algebra and Eckhart–Young Theorem. The theorem expressed interrelationships between a data matrix X and the eigenvalues and eigenvectors of its two cross-products matrices including minor and major products (Davis, 2002:568). In summary, Eckhart-Young Theorem states that for any real matrix (X) two orthogonal matrices, V and U, can be derived from which the product is a real diagonal matrix with no negative elements as:

$$X = V\Lambda U' \qquad (3)$$

where V is an n x r matrix whose columns is orthonormal, Λ is r x r square matrix containing r positive diagonal elements that are the singular values, U is an m x r matrix whose columns are orthonormal.

Consequently Davis (2002:568) derived a solution for Q- and R-mode FA based on Eckhart-Young Theorem, using scaled matrix W, through the following steps. First, in computing simultaneous Q and R-mode FA, one has to find the Minor Product (Mp) from the correlation matrix of the scaled matrix, W, computed from either equation (1) or equation (2) as:

$$Mp = W'W \quad (4)$$

where W' is the transpose of the scaled matrix W. Then eigenvalues and eigenvectors are derived from Mp (from equation (4)) for the computation of R-mode loadings (A^R) by multiplying each element of the eigenvector (U) by corresponding square root of eigenvalue or singular values (Λ) as:

$$A^R = U\Lambda \quad (5)$$

Q-mode loadings can be found as:

$$A^Q = WU \quad (6)$$

R-mode scores are computed as:

$$S^R = WA^R \quad (7)$$

Q-mode scores were computed as:

$$S^Q = W' A^Q \quad (8)$$

According to Eckhart–Young theorem we can have relationships between the variables and objects to measure similarity by reproducing the scaled matrix W as:

$$W = A^Q \Lambda^{-1/2} A^{R'} \quad (9)$$

If the computed Q- mode loadings, R-mode loadings, singular values products as in equation (9) do not produce the original input matrix then computation can be considered as wrong; it helps checking the correctness of a model. In Diagnostic evaluation, based on Eckhart–Young theorem 'qrfactor' predictions should be able to reproduce the scaled data matrix W (See Table 1) by using equation (9). In this evaluation I used qrfactor to measure similarity between and among Meuse river data set (Burrough & McDonnell, 1998) in the 'gstat' package (Pebesma, 2011). And because the data contains other variables, count and categorical, we only need to select the continuous data such as cadmium, copper, lead, and zinc as:

```
library(qrfactor)
library(gstat)
data(meuse)
variables= c("cadmium", "copper", "lead","zinc")
data=meuse[variables]
#Then we can create an object from 'qrfactor' class as:
mod1 <- qrfactor(data)
#Using 'mod1' predictions we can estimate equation (9) from R command as:
#Q- mode and transpose of R-mode loadings can be estimated as in equation (9)
AQ=mod1$q.loading
AR= t(mod1$r.loading)
# Singular value diagonal matrix
sv = diag(1/sqrt(mod1$eigen.value))
#Model scaled matrix W is
Wmod1=AQ%*%sv%*%AR
#print results
#Wmod1
```


#Print original matrix

x_standard<-scale(meuse[variables],center=TRUE,scale=TRUE)/sqrt(nrow(meuse[variables]))

2.3 Software Design and Development

2.3.1 R language

R is a strongly functional language and environment that can be used to statistically explore data and display graphics. It is a free command based software which brings users close to their data (R Development, 2011). R is strongly based on S language that was invented in mid 1970s by John Chambers with the aim of encouraging the user to "slide into programming, perhaps without noticing" (Chambers, 1998). In 1995, Ihaka and Gentleman (1996) released free and open source R software. R tree consists of R base system, Packages, repository, and a community organized on "Comprehensive R Archive Network" (CRAN). The conventional relationship between R and S languages has been maintained. It is always assumed, when developing a package, that one is working with S object oriented programming language, and once the building of package is complete then workers call it R (Leisch, 2009). R uses two main S language objects called S3 and S4 that define object oriented programming in R. S3 object classes and methods have been integrated into R from its beginning. The 'qrfactor' package was written in S3 object oriented programming method.

Table 1. Comparison between original scaled matrix (W) and *qrfactor scaled matrix (Wmod1) of the Meuse data in the gstat package.* This data is a subset (20 observations) of modelled and original data matrix. The original scaled matrix is computed from equation (2) while the modelled matrix is computed from equation (9) by using the loadings, scores, and eigenvalues from qrfactor function.

	Original Scaled Matrix				***qrfactor() modeled Scaled Matrix***			
Obs.	**cadmium**	**copper**	**lead**	**zinc**	***cadmium***	***copper***	***lead***	***zinc***
1	0.19	0.15	0.11	0.12	*0.19*	*0.15*	*0.11*	*0.12*
2	0.12	0.14	0.09	0.15	*0.12*	*0.14*	*0.09*	*0.15*
3	0.07	0.09	0.03	0.04	*0.07*	*0.09*	*0.03*	*0.04*
4	-0.01	0.14	-0.03	-0.05	*-0.01*	*0.14*	*-0.03*	*-0.05*
5	-0.01	0.03	-0.03	-0.04	*-0.01*	*0.03*	*-0.03*	*-0.04*
6	-0.01	0.07	-0.01	-0.04	*-0.01*	*0.07*	*-0.01*	*-0.04*
7	0.00	-0.03	-0.02	-0.03	*0.00*	*-0.03*	*-0.02*	*-0.03*
8	-0.01	-0.04	0.00	-0.01	*-0.01*	*-0.04*	*0.00*	*-0.01*
9	-0.02	-0.01	-0.01	-0.03	*-0.02*	*-0.01*	*-0.01*	*-0.03*
10	-0.04	-0.06	-0.05	-0.06	*-0.04*	*-0.06*	*-0.05*	*-0.06*
11	-0.04	-0.05	-0.05	-0.06	*-0.04*	*-0.05*	*-0.05*	*-0.06*
12	-0.03	-0.05	-0.04	-0.05	*-0.03*	*-0.05*	*-0.04*	*-0.05*
13	0.18	0.18	0.09	0.14	*0.18*	*0.18*	*0.09*	*0.14*
14	-0.02	-0.03	0.02	0.01	*-0.02*	*-0.03*	*0.02*	*0.01*
15	-0.03	-0.05	-0.02	-0.03	*-0.03*	*-0.05*	*-0.02*	*-0.03*
16	0.14	0.15	0.06	0.12	*0.14*	*0.15*	*0.06*	*0.12*
17	0.09	0.11	-0.01	0.03	*0.09*	*0.11*	*-0.01*	*0.03*
18	0.09	0.10	0.00	0.05	*0.09*	*0.10*	*0.00*	*0.05*
19	0.12	0.10	0.04	0.06	*0.12*	*0.10*	*0.04*	*0.06*
20	0.22	0.19	0.09	0.13	*0.22*	*0.19*	*0.09*	*0.13*

2.3.2 R packages

Leisch (2009) described an R Package as an extension of the R base system with code, data and documentation in a standardized format residing in a library: a directory containing all R installed packages. There are three main types of R package: 1. The base packages that are part of the R source tree, and maintained by R Core Team. 2. Recommended packages, part of every R installation, but not necessarily maintained by R Core Team. 3. Contributed packages that include the rest of packages from other contributors. A package normally goes through series of checks, tests and scrutiny before it is accepted on CRAN (Carslaw & Ropkins, 2011).

Many aspects of basic statistics can be performed using R base system or with recommended packages. Most advance statistics such as multivariate techniques are however more popularly performed using some contributed packages (Table 2).

Table 2. Available Multivariate Statistics in R packages.

Statistics	**Packages**[1]	**Main functions**	**Source**
Generalized Linear Models	R Base	glm()	R Core (2011)
Discriminant Function Analysis	MASS	lda()	Venables & Ripley (2002)
		qda()	
Principal Components Analysis	psych	principal( )	Revelle (2011)
Factor Analysis	psych	factor.pa( )	Revelle (2011)
Correspondence Analysis	ca	ca()	Greenacre & Nenadic(2010)
Multidimensional Scaling	MASS	isoMDS()	Venables & Ripley (2002)
		dist()	
Cluster Analysis	fpc	kmeans( )	Hennig (2010)
	pvclust	pvclust()	Suzuki & Shimodaira (2011)
	Mclust	Mclust()	Fraley & Raftery (2011)
Structural Equation Modeling	sem	sem()	Fox & Byrnes (2011)
Simultaneous Q-and R-mode FA	qrfactor	qrfactor()	Owusu (2011)
Canonical	CCA	cc()	Gonz âlez & D éjean (2011)

1:These packages are available from R website: cran.r-project.org/web/packages/

2.3.3 qrfactor package

The ‘qrfactor’ package simultaneously estimates Q and R mode factor analysis loadings and scores. The package uses Eckhart-Young Theorem described under section 2.2 to estimate Q-and R-mode FA. The package consists of classes, functions or methods, arguments and returning values from the functions. The package also offers summary statistics of the model object, printing of returning values, and annotated plots of first two axes. It is designed with the principles of object oriented programming (see section 2.3.1). A contributed package like ‘qrfactor’ must be installed once before it is used on R as:

> install.packages(‘qrfactor’)

There are five functions in the ‘qrfactor’ package (Table 3). They include the main function qrfactor(); rq(), an internal function; and extensions of R base functions such as qrfactor.print(), qrfactor.summary() and qrfactor.plot(). The main function ‘qrfactor()’ (Table 3) accepts input data and a modelled scale parameter (Table 4). It is therefore passed as “qrfactor(data ,scale)‘’. Once an object is passed to the main function the ‘print()’ and ‘summary ()’ functions accept them as input parameter.

Table 3. Descriptions of the main functions in qrfactor package

Function	Description
qrfactor()	Returns the model object
rq()	An internal function for internal use only
print()	Print all the objects and values associated with the returned object
summary()	Gives a brief summary of the returned objects including correlations matrix, eigeinvalues, and many more.
plot()	When issued with the returned object it plots annotated first two axes of the loadings.

Table 4. Basic input data and arguments for qrfactor package

Argument	Descriptions
source	Data: a numeric design matrix for the model. All records must be numeric; it also accepts continuous data. Avoid using categorical variables and characters.
scale	An optional standardisation method that you want to use. Set it to "data" if you do not desire data transformation or scaling; set it to "sd" if you want the data to be standardised after centring it; set it to "n" if you want to divide the centred data by square root of the number of observations. The default is "sd".
object	an object of class "qrfactor", i.e., a fitted model.
x	an object of class "qrfactor", i.e., a fitted model.
...	any other R parameters can be added

The returning values of the package are basically the list elements of objects of the class ‘qrfactor’. Once an object is declared from qrfactor class; the list in Table 5 can be assessed with the object.

2.3.4 Graphical capabilities of qrfactor

The qrfactor package offers many combinations of plots and maps; one can plot many dimensions for eigenvectors, loadings and scores as:

plot(qrfactor(data),factors=c(1,2),type="loading”, plot="r",...)

The ‘factors’ parameter lists the number of factors one wants to plot; the default is ‘factors=c(1,2)’ for the first and the second dimensions. Setting ‘factors=c(1,3)’ will plot the first and third dimensions. The ‘type’ parameter indicates the type of plot one wants to do; it takes “map”, “cluster”, "scores", "loadings", “pca” or “eigenvectors”, ‘coord’. The ‘plot’ parameter iindicates the type of plots one desires: it takes "all" for all the 3 plots; "q" for q plot; "r" for r plot; and ‘qr’ for both q and r plots.

Table 5. The returning values associated with qrfactor object

Value	Descriptions
data	Original data for the model.
x.standard	It is the scaled matrix of the original data
correlation	The correlation matrix for the data
eigen.value	Eigen value of correlation matrix of the data
eigen.vector	Eigen vector of correlation matrix of the data
diagonal.matrix	Diagonal matrix of eigen vector
r.loading	R-mode loadings
q.loading	Q-mode loadings
loadings	combined loadings of R and Q on the same axis
q.scores	computed Q-mode scores
scores	combined R-mode and Q-mode scores on the same axis
variance	Percentage explained by eigen values
cumvariance	Cumulative Percentage explained by eigen values
rownames	row names of the loadings
variables	variables names of the loadings, of the original data
gisdata	spatial data input

3.0 Usage and illustrations

3.1 Examples and Evaluations

The qrfactor() is applied to fresh water resources distribution data in Africa. The data is available at The World's Water (Christian-Smith et al., 2011). It consists of 51 African countries and 8 variables that include Domestic water Use, Agricultural Water Use, Industrial Water Use, 2010 Population, Total Renewable Water Resources, Total water withdrawal, per capita water withdrawal, and per capita water resources. The data has been stored in 2 versions in qrfactor() external folder as 'Africanfreshwater.shp' and 'Africanfreshwater.csv'.

3.11 Modelling spatial and non-spatial data with qrfactor()

There are 4 ways qrfactor() reads data: 1) using external data files such as 'csv' or "txt" 2) using shapefiles, 3) using R dataframe and 4) using both "csv" and shape files, where the former is joined to the latter. I illustrate all the 4 ways as follows:

```
library(qrfactor)
source<- system.file("external", package = "qrfactor") #list the working folder
layer="Africanfreshwater" #indicate the shape file you want to work with; remember ".shp" is omitted
csv= system.file("external", "Africanfreshwater.csv", package = "qrfactor") #list the csv file
#list the variables you are working with
var=c( "Domestic", "Industry", "Agricultur", "Population", "Resources", "withdrawal","perCapitaW",
"perCapitaR")
mod1=qrfactor(csv,var=var) #model only CSV file
mod2=qrfactor(source,layer,var=var) #model shape file only
# model to match both csv and shape file with a common field: "COUNTRY" to produce a merged shapefile
mod3a=qrfactor(source,layer,var=var,m="COUNTRY",f=csv)
#Model a spataial data from qrfactor
Mod3b=qrfactor(mod3$gisdata[var])
#read shape into R dataframe and model the data
```


gisdata <- na.omit(readOGR(source, layer))
mod4=qrfactor(gisdata[var])
summary(mod4) #prints eigein values, (cumulative) percentage of variance explained etc
mod4$loadings #print combined loadings of R- and Q mode FA
mod4$r.loading #print R mode loadings
mod4$q.loading #print Q mode loadings
Please note that all the above models produce the same results. We can also add other parameters such as log transformation of input data, number of factors to enhance cluster analyses.

3.12 The Plot Capabilities of qrfactor()
The qrfactor() offers 4 main types of plotting capabilities; they include the default plots, cluster based plots, map based plots and diagnostic plots (Figure 1):
par(mfrow=c(1,2))
plot(mod4) #default plot
#customising default plot as shown on Figure 1
plot(mod4,rowname="COUNTRY",cex=c("means"),legend="topleft",values=c("cluster"),pch=23)
plot(mod4,cex=c("Industry"),type="cluster") # series of cluster plots
plot(mod4,type="map" ,cex=c("Resources")) # series of map plots
plot(mod4,type="diagnose") # diagnostic plots for normality and outlier identification.

Figure 1 has been plotted by adding more labels to the defaults plots and the labels include the row means of the data and cluster values, and then also a legend. Figure 1 helps interpretation of the loadings of R and Q-mode Factor Analysis. It can be seen on Figure 1 that Domestic water Use (Domestic), Industrial Water Use (Industry), Agricultural Water Use (Agricultur), and Per Capita Fresh Water Withdrawal (perCapitaW) in Africa are highly linearly related on Factor loading 1 while Per capita Annual Water Resources (perCapitaR) and Total Annual Water Resources (Resources) highly linearly related on Factor loading 2. While Factor Loading 1 can be named Water Use, Factor loading 2 can be named as Resources Availability. It is also evident from Figure 1 that there is an inverse relationship between the rate of Water Use and Water Availability because the latter are having positive Factor 1 loadings while the water use is having negative Factor loadings. The more water resources a country may have, in Africa, the less it uses water, according to Figure 1. It is also evident from Figure 1 that the closer the distance between the countries the more similar they are. We can therefore use qrfactor() to identify various groups of water usage and water resources in Africa.

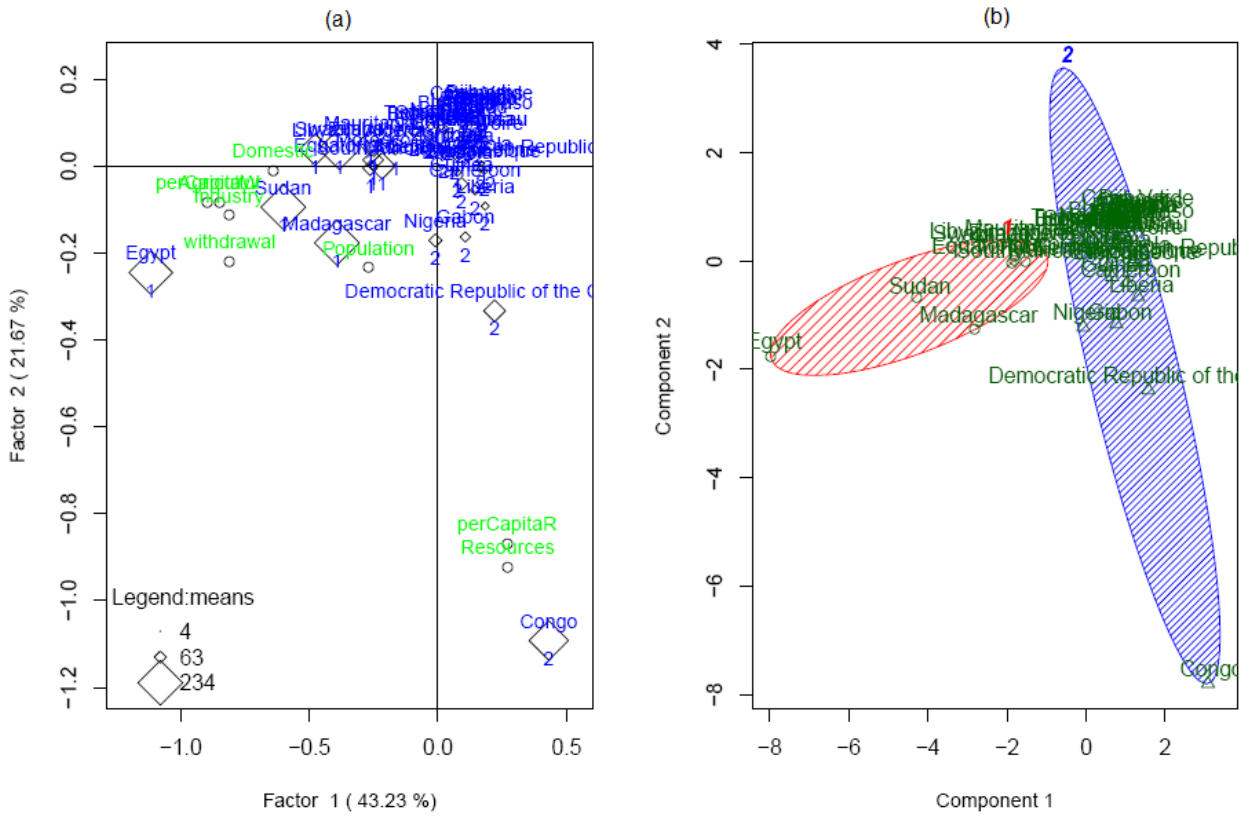

Figure 1: The (a) qrfactor plot of the Factor loadings and (b) clusplot of kmeans cluster plot of African Fresh Water data. The qrfactor plot symbols of the countries have been scaled with the values of the row means of the data; and the legend the plot is at the bottom left. The plot was generated by setting plot.qrfactor() "type" parameter to "cluster": plot(mod4,cex=c("means "),type="cluster")

When we label the countries with kmeans clusters (qrfactor() by default classifies observations into 2; this can be changed with "nfactors" parameter) qrfactor() perfectly matches kmeans classification. Most of the North and Southern African Countries are classified as group 1 while sub-Saharan countries as group 2. However, qrfactor Q- and R mode Factor Analysis goes beyond cluster analysis by identifying the importance of the variables to each observation or cluster. The closer an observation (country) to a variable the more it is related to the variables. The qrfactor() package is able to identify on Figure 1 that the countries closer to 'Water Use' variables are dominated by high rate of water use while the distant countries show lower rate of water usage. The same interpretation can be done on Factor Loading 2 with countries such as DR Congo and Congo having higher means on water resources variables.

3.13 Plotting qrfactor() Maps

Maps in qrfactor() is displayed by a command:

plot(mod4,type="map")

The qrfactor() package displays maps from 1) Input Shape File Data, 2) Factor Scores, 3) Factor Indices, 4) Mean Indices, 5) Factor Ranks, 6) Mean Rank, and 6) clusters. The package produces about 60 paged maps with all qrafctor() maps in gray scale and full color with considerable number of them labeled with row names. One can also change the label of a map, and all the plots in qrfactor(), with another name; for instance, using a

unique field called “code” in the spatial data as:

plot(mod4,type="map",rowname="code")

There are three main types of input data maps: original matrix data map, normalized matrix data map and individual variables maps all in gray (Figure 2) and full color versions. The scaled map facilitates spatial interpretation of input data across all the variables on the matrix plot.

The next group of plots is factor scores maps, which facilitate the spatial distribution interpretation of factor scores. Due difficulty in interpretation of factor scores qrfactor() package develops indices from factor scores in order to aid interpretation. Each factor score is ranked and the ranks are divided by the maximum rank to produce factor index. The package displays two main factor indices: factor 1 index and factor 2 index. In our African freshwater data set factor 1 index can be described as Water Use index of Africa while Factor 2 as water resources Availability index of Africa (Figure 4). The package also produces the rank version (Figure 3) of the indices in order to know the relative importance of the factors.

The map version of figure 1 therefore throws more lights on spatial distribution of water use and water availability. Most of the forest zones are ranked high on water availability index but low on water use while the desert zones are showing the opposite. The position of Egypt on figure 1, found between factor 1 loading and factor 2, has been elucidated by Figures 3 and 4; apart from Egypt ranking the highest on water use, the Nile water is also causing it to rank relatively high on water availability. The group 1 in the Figure 5 is characterized by water availability indicators such as high agriculture water use, high industrial water use, high domestic water use, and high rate of water withdrawal but relatively low rate of annual water resources. Group 2 (Figure 5) on the other is characterized by high rate of renewable water resources and per capita water resources but relatively lower rate of water use indicators such as domestic water use, agriculture water and industrial water use.

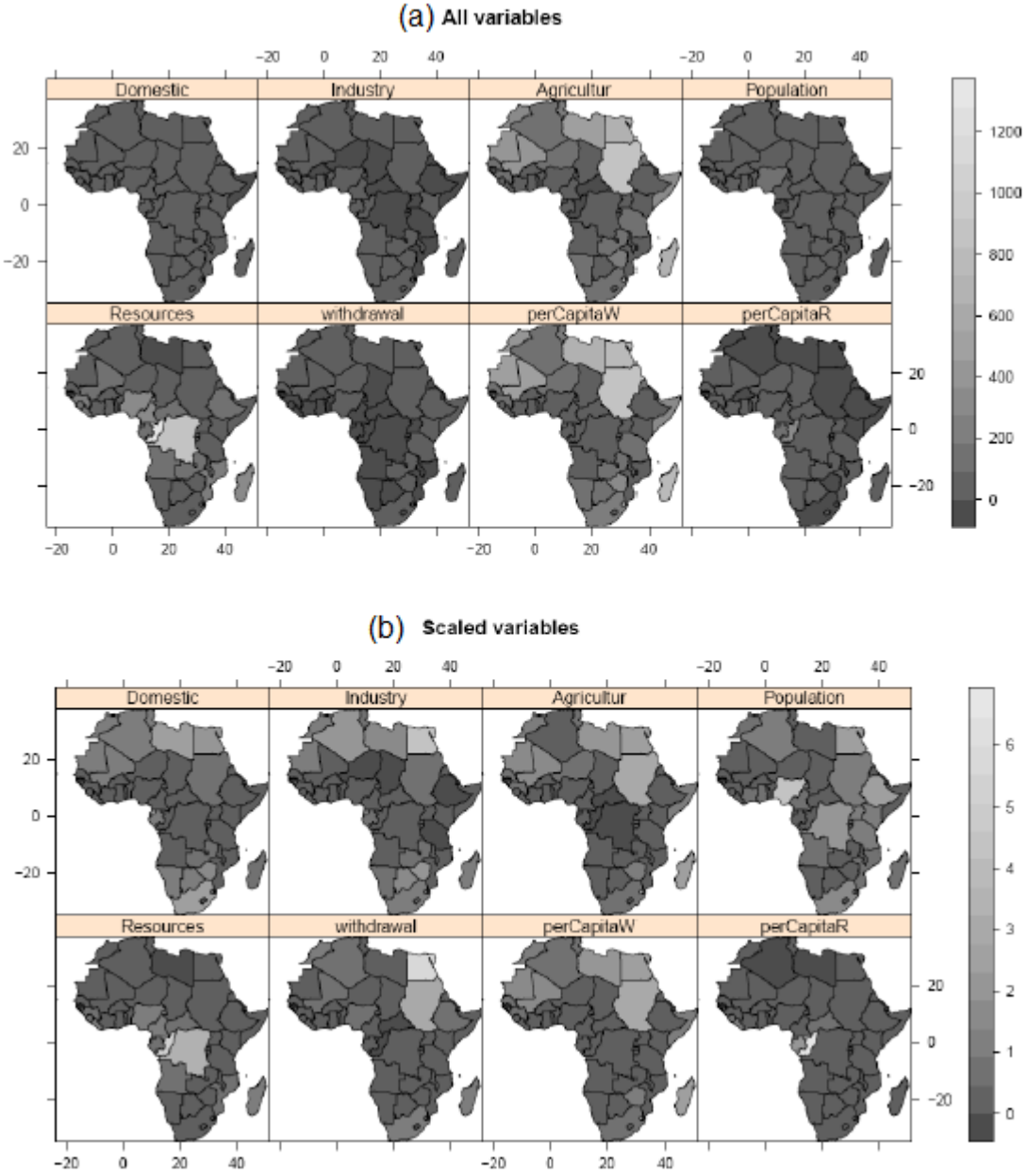

Figure 2: The qrfactor() plotting of spatial input of (a) original data and (b) Normalized data. The plot was generated by setting plot.qrfactor() "type" parameter to "map": plot(mod4,cex=c("means "),type="map")

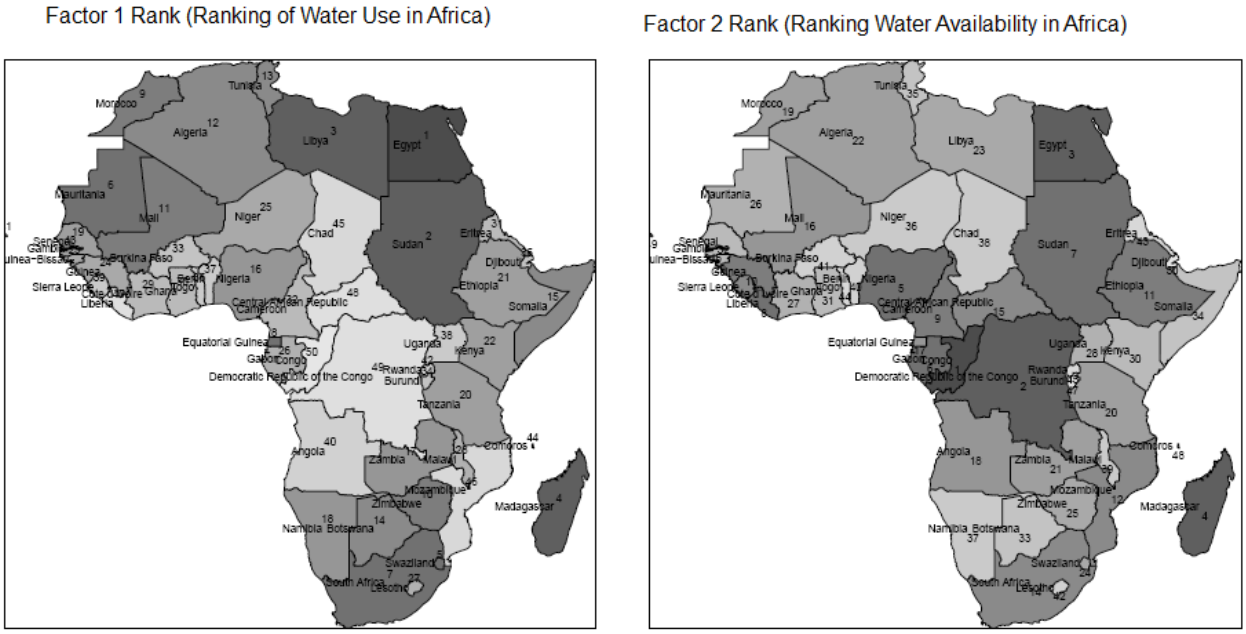

Figure 3: Ranking African Countries based on Factor Scores for the first two factors by qrfactor(). The plot was generated by setting plot.qrfactor() "type" parameter to "map": plot(mod4,cex=c("means "),type="map").

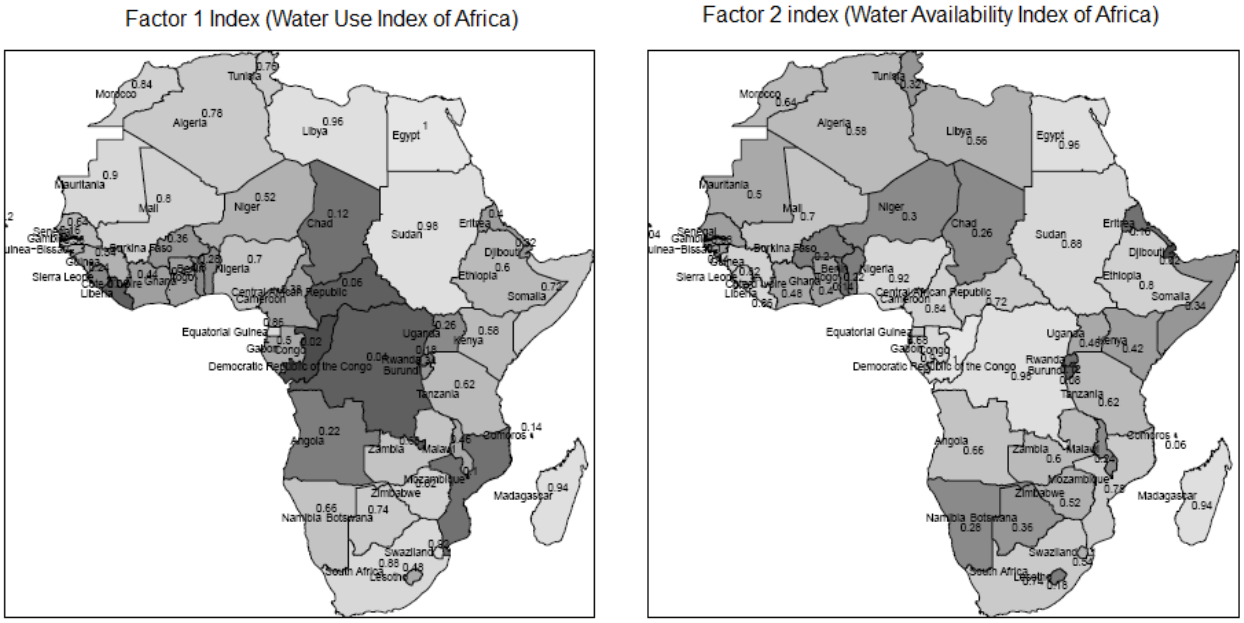

Figure 4: Indexing African water usage and resources with qrfactor(). The indices have been developed by ranking the factor score and dividing each by the maximum rank. The plot was generated by setting plot.qrfactor() "type" parameter to "cluster": plot(mod4,cex=c("means "),type="map").

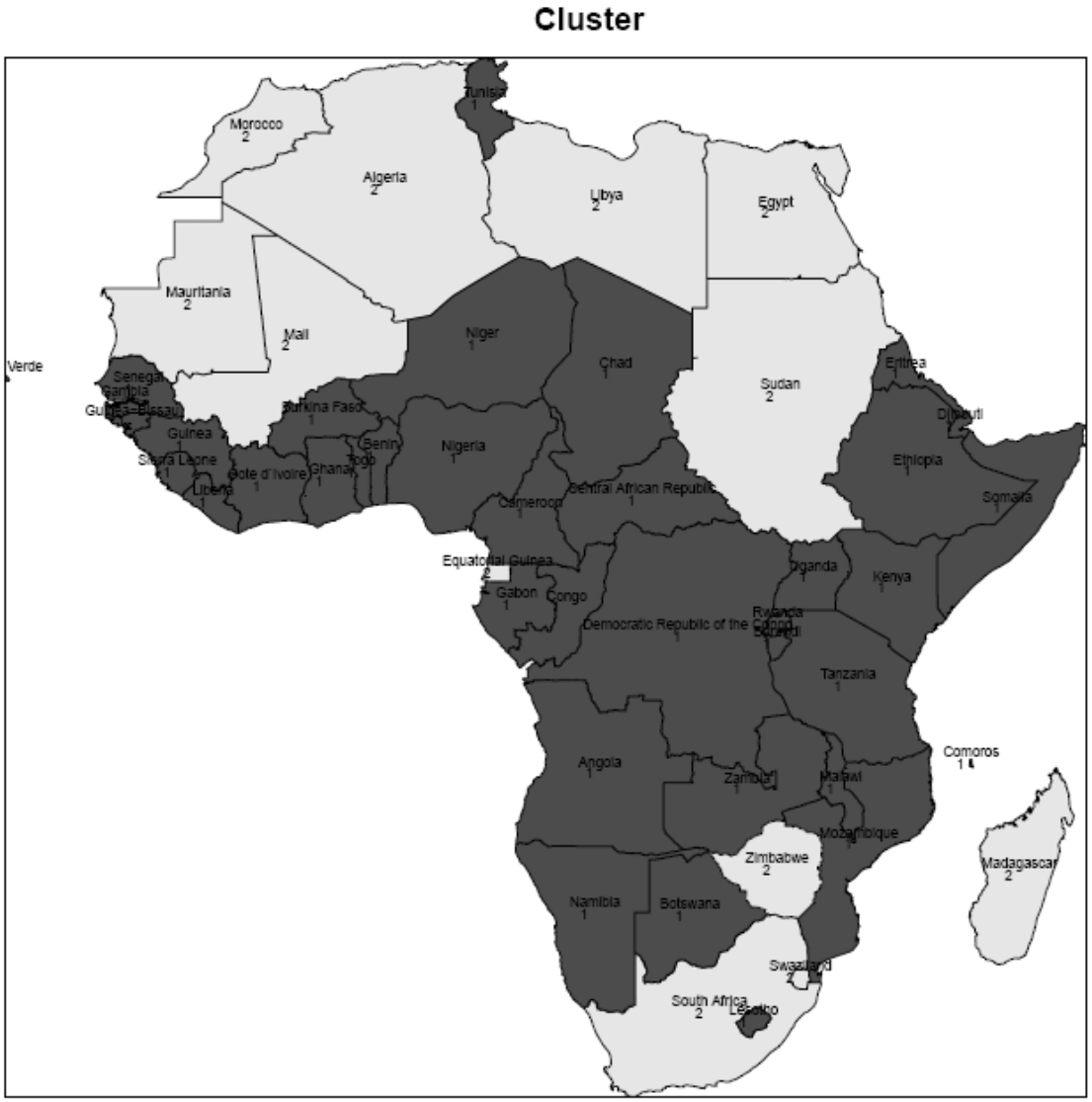

Figure 5: Clustering African Water Usage and Resources with qrfactor().The plot was generated by setting plot.qrfactor() "type" parameter to "map": plot(mod4,cex=c("means "),type="map").

3.14 Diagnostic plots, Data Transformation, and some Statistics

The package also checks the normality and outlier's identification of the input data, thereby helping the modeller to delete or transform the input data. The univariate normality is assessed by drawing serious of histograms of each variable. The multivariate normality is assessed by drawing qqplot of all input data. In case the data is not normally distributed the data can be transformed by setting the "scale" parameter of qrfactor() to "log" or "sqrt" for log and square root linear transformation of the entire data respectively. Options are also made to transform only selected variables that are not normally distributed with the "transform" parameter. There are four different graphs that are produced for the assessment of the outliers with the package. One pair of graph identified outliers based on 97.5% quantile while the other is based on adjusted quantile. Outlier identification and variable transformation can be concurrently issued with one of the following commands:

#input spatial data

source<- system.file("external", package = "qrfactor") #list the working folder

layer="Africanfreshwater" #indicate the shape file you want to work with; remember ".shp" is omitted


```
#diagnose the original data
diagnoseMod=qrfactor(source,layer,var=var)
plot(diagnoseMod,rowname="COUNTRY",type="diagnose")
#diagnose the square root transformed data
sqrtMod=qrfactor(source,layer,var=var,scale="sqrt")
plot(sqrtMod,rowname="COUNTRY",type="diagnose",plot="map")
#diagnose the log transformed data
logMod=qrfactor(source,layer,var=var,scale="log")
plot(logMod,rowname="COUNTRY",type="diagnose")
#Transformation on only selected variables and print maps
transformMod=qrfactor(source,layer,var=var,scale="sqrt", transform=c("Industry","Resources","Agricultur"))
plot(transformMod,rowname="COUNTRY",type="diagnose", plot="map")
```

The qrfactor() is also statistically able to measure difference between the variables; and this is useful if one is modelling a temporal data. It prints ANOVA table and non-parametric independent 2-group Mann-Whitney U Test of the variables as:

```
plot(transformMod,rowname="COUNTRY",type="anova", plot="map")
```

The cluster plot of qrfactor also prints ANOVA table of the clusters as well as box plots of the clusters:plot(transformMod, plot="cluster")

3.2 Evaluation of the qrfactor package

Table 1 shows that qrfactor is mathematically correct. Figure 1(a) and 1(b) show that qrfactor and kmeans plots are the same and having exact observation position on their respective scatter plots. The Q-mode factor loadings are therefore correct. The r-mode results of the qrfactor as shown on Figure 1(a) was also compared to R base prcomp(), and PCA() in FactoMineR Package (Sébastien Lê et al., 2008). The code below produces the output on Table 6:

```
data=read.csv(system.file("external","Africanfreshwater.csv", package = "qrfactor"),header=TRUE)
rownames(data)= data$COUNTRY
var=c( "Domestic", "Industry", "Agricultur", "Population", "Resources", "withdrawal","perCapitaW",
"perCapitaR")
data=data[var]
sink("results.txt", append=FALSE, split=FALSE)
mod=qrfactor(data)
cat("PCA standard deviation for qrfactor\n")
sqrt(mod$eigen.value)
cat("PCA Loadings for qrfactor\n")
mod$pca
cat("qrfactor R mode Loadings\n")
mod$r.loading
cat("\nPCA results for R base prcomp()\n")
mod2pca=prcomp(data,center = TRUE, scale. = TRUE)
mod2pca
cat("\nLoadings from PCA() in FactoMineR Package\n")
```


library(FactoMineR)
pca=PCA(data)
pcavarcoord
sink()

We can see that qrfactor results are in complete agreement with both R base PCA function and FactoMineR PCA() function. It is clear from the Table 6 that qrfactor command does not only correctly simulate PCA but also R-mode loadings as in *FactoMineR*. The FactoMineR PCA () function automatically generates two plots for observations (Q-mode) and variables (R-mode) (Figure 6). The figure is quite similar to Figure 1(a) generated by plot(qrfactor(data)) except the inversion of the axes. The inversion of axes does not affect similarity measurement and the loading can be inverted back by multiplying it by -1 (Davis 2002:569). Though FactoMineR correctly measure similarity among variables on one hand and among observations on the other hand on different plots, it is unable to simultaneously measure relationship among observations and variables because of its use of different scales. The scale of FactoMineR PCA individual factor dimensions range from -4 to 10 (Figure 6): this makes it impossible to compare variables and observations. Therefore the qrfactor package is able to measure similarity between variables and observation; the scale of qrfactor axis ranges from -1 to +1(figure 1(a)).

4. Availability

The qrfactor package is freely available on CRAN at cran.r-project.org/web/packages/qrfactor/. Series of tests scripts are also offered for PCA, R-mode Factor Analysis, Q-mode Factor Analysis, Q-and R-mode Factor Analysis, principal coordinate analysis (PCO), Multi dimensional scaling (MDS) and African freshwater example as used in this text. It is licensed under GPL-2.

Table 6. Extracted results of evaluation of qrfactor using R base and FactoMineR packages. The qrfactor results were estimated from qrfactor() function; R base results were extracted from prcomp() and FactoMineR results from PCA(). All of the functions use a common fresh water data set in qrfactor package.

PCA standard deviation for qrfactor					
[1]	1.859577	1.316515	1.102615	0.987424	0.606468
PCA Loadings for qrfactor					
	PC1	PC2	PC3	PC4	PC5
Domestic	-0.34465	-0.00806	0.446523	0.503894	-0.43929
Industry	-0.43657	-0.08574	0.189117	0.436898	0.303595
Agricultur	-0.45496	-0.063	0.01863	-0.51593	-0.20809
Population	-0.14481	-0.17751	-0.75466	0.33894	-0.3919
Resources	0.145888	-0.70057	-0.05516	-0.04328	-0.19589
qrfactor R mode Loadings					
	Factor1	Factor2	Factor3	Factor4	Factor5
Domestic	-0.6409	-0.01061	0.492343	0.497557	-0.26642
Industry	-0.81184	-0.11288	0.208523	0.431404	0.184121
Agricultur	-0.84604	-0.08294	0.020541	-0.50944	-0.1262
Population	-0.26929	-0.23369	-0.8321	0.334677	-0.23768
Resources	0.27129	-0.92231	-0.06082	-0.04274	-0.1188

PCA results for R base prcomp()					
Standard deviations:					
[1]	1.859577	1.316515	1.102615	0.987424	0.606468
	PC1	PC2	PC3	PC4	PC5
Domestic	-0.34465	-0.00806	-0.44652	0.503894	0.43929
Industry	-0.43657	-0.08574	-0.18912	0.436898	-0.3036
Agricultur	-0.45496	-0.063	-0.01863	-0.51593	0.20809
Population	-0.14481	-0.17751	0.754664	0.33894	0.391904
Resources	0.145888	-0.70057	0.055158	-0.04328	0.19589
Loadings from PCA() in FactoMineR Package					
	Dim.1	Dim.2	Dim.3	Dim.4	Dim.5
Domestic	0.640899	0.010611	-0.49234	0.497557	0.266415
Industry	0.81184	0.112878	-0.20852	0.431404	-0.18412
Agricultur	0.84604	0.082936	-0.02054	-0.50944	0.1262
Population	0.269288	0.233695	0.832104	0.334677	0.237677
Resources	-0.27129	0.922308	0.060818	-0.04274	0.118801

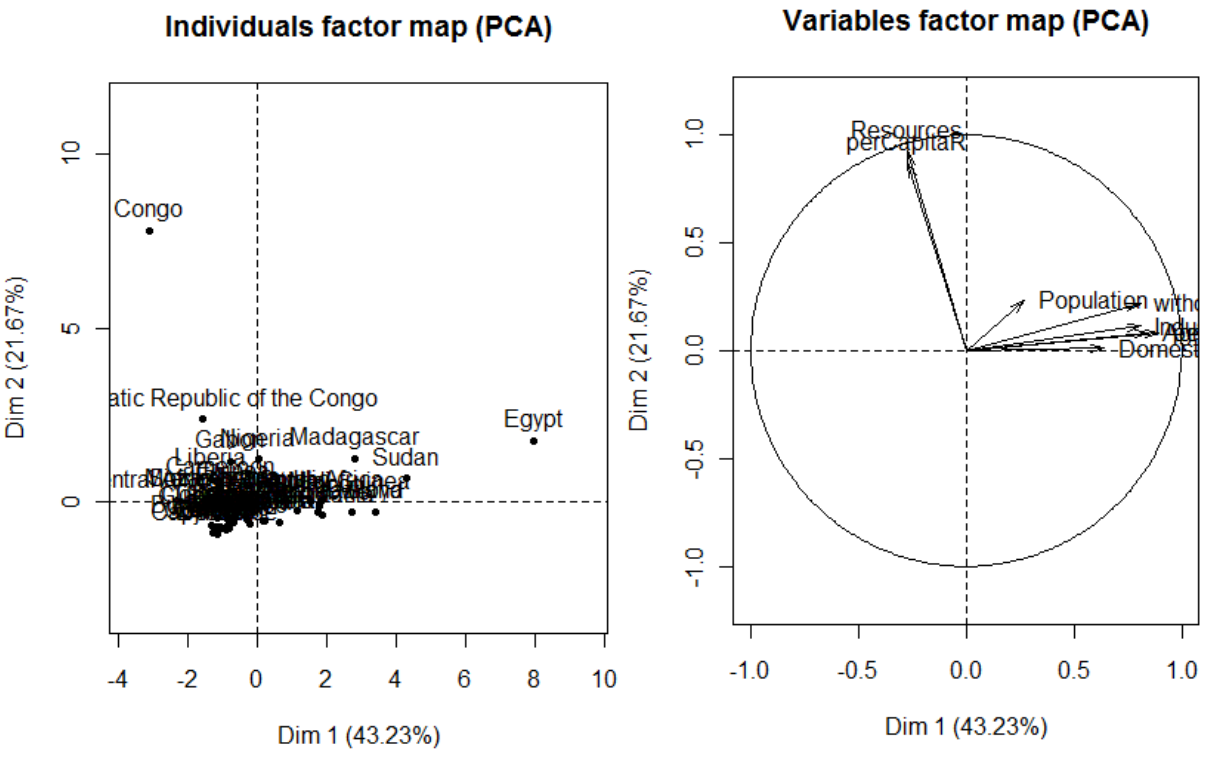

Figure 6: Representation of observations (left) and variables (right) on PC1-PC2 obtained using the FactoMineR PCA().

5. Conclusion

Simultaneous R- and Q-mode Factor Analysis has been simulated using Eckhart–Young mathematical theory and R computing environment with a new R package called ‘qrfactor’. The package, well tested, can easily be

installed on R. The main function returns values such as R-mode loadings, Q-mode loadings, and combined loadings of R- and Q-mode on the same axes. It also returns Q-mode scores, R-mode scores and combined R- and Q-mode scores on the same axis. Other standard R functions such as 'summary ()', 'print ()' and 'plot ()' have been extended and can be applied to the objects of 'qrfactor' class. The evaluation of the package results show that it is mathematically, computationally, and empirically correct. Therefore this package can be used to estimate Simultaneous R- and Q-mode Factor Analysis.

References

Bivand R. S., Pebesma E. J. and Gomez-Rubio V. (2008). *Applied Spatial Data Analysis with R.* : Springer.

Burrough P. A. and Mcdonnell R. A. (1998). *Principles of Geographical Information Systems*: Oxford University Press.

Carslaw D. C. and Ropkins K. (2011). openair - An R package for air quality data analysis. *Environmental Modelling & Software*.

Chambers J. M. (1998). *Programming with data: A guide to the S language*. Verlag, Berlin, Germany: Springer.

Christian-Smith J., Gleick P., H.; and Cooley H. (2011). World's Water, from http://www.worldwater.org/data.html

Davis J. C. (2002). *Statistics and Data Analysis in Geology*: Wiley.

De Mooy H., Van Hattum J. T. A. and Vriend S. P. (1988). A RQ-mode factor-analysis program for microcomputers. A Pascal program. *Computers & Geosciences, 14*(4), 449-465.

Engel M. H., Imbus S. W. and Zumberge J. E. (1988). Organic geochemical correlation of Oklahoma crude oils using R- and Q-mode factor analysis. *Organic Geochemistry, 12*(2), 157-170.

Hair J. F., Anderson R. E., Tatham R. L. and Black W. C. (1998). *Multivariate Data Analysis*: Prentice-Hall, Upper Saddle River, NJ.

Hijmans R. J. and Etten J. V. (2011). raster: Geographic analysis and modeling with raster data **R package version 1.9-55**. Retrieved 23 December 2011, from http://cran.r-project.org/web/packages/raster/index.html

Ihaka R. and Gentleman R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics, 5*, 299-314.

Khan Z. A. and Tewari R. C. (2011). R-mode factor analysis of lithologic variables from cyclically deposited Late Paleozoic Barakar sediments in Singrauli Gondwana sub-basin, Peninsular India. *Journal of Asian Earth Sciences, 40*(1), 144-149.

Kulkami K. (2012). Single Sampling Plan for Variables Under measurement Error for non-normal Distribution. *Mathematical Theory and Modeling 2*(9).

Leisch F. (2009). Creating R Packages: A Tutorial. *Compstat 2008-Proceedings in Computational Statistics.* .

Owusu G. (2011). qrfactor: Simultaneous simulation of Q and R mode factor analyses. R package version 1.2 Retrieved 5 January 2012, from **http://CRAN.R-project.org/package=qrfactor**

Pebesma E. and Bivand R. (2011). sp: classes and methods for spatial data-R package version 0.9-91 Retrieved 5 January 2012, from http://cran.r-project.org/web/packages/sp/index.html

R Development C. T. (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Revelle W. (2011). *psych: Procedures for Personality and Psychological Research*. Evanston: Northwestern University Available from http://personality-project.org/r/psych.manual.pdf.

Ripley B. (2011). spatial: Functions for Kriging and Point Pattern Analysis R package version 7.3-3.

S ébastien L ê, Julie Josse and Husson F. (2008). FactoMineR: An R Package for Multivariate Analysis. *Journal of Stattistical Software, 25*(1).

Walden J., Smith J. and Dackombe R. (1992). The use of simultaneous R- and Q-mode factor analysis as a tool for assisting interpretation of mineral magnetic data. *Mathematical Geology, 24*(3), 227-247.