S&DS 563 / F&ES 758b - Multivariate Statistics Homework #6
Option B Factor Analysis

Lanxin Jiang (lj345), Grace Sun (ys544), Chenglin Lu (cl939)

2018-04-19

```r
#Import Dataset
Sys.setenv(JAVA_HOME="C:\\Program Files\\Java\\jre-9.0.4\\")
library(rJava)
library(RWeka)
CKD <-
read.arff("C:/Users/lanxin/Documents/GitHub/Chronic_Kidney_Disease/Chronic_Kidney_Disease/chronic_kidney_disease_full.arff")
```

1. Look through indicators (questions). Think about which indicators might be related through latent factors. (nothing to turn in here)

```r
    CKD.numeric <- CKD[,c(1:2,10:18)]
    # Remove Missing Observations
    CKD.numeric <- CKD.numeric[complete.cases(CKD.numeric),]
```

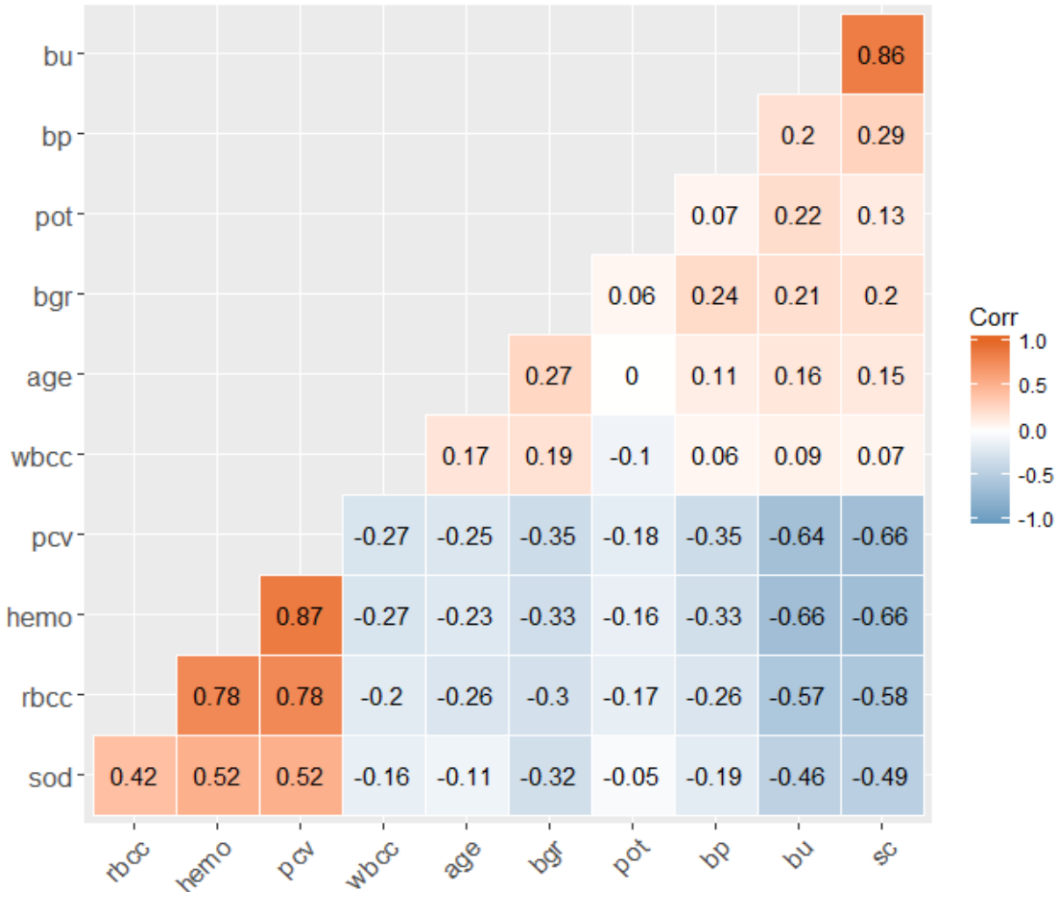
The dataset obtained from the UCI Machine Learning Repository describes the chronic kidney disease status and blood measurement of patients from Apollo Hospitals in India. There are 400 observations in this dataset. The following variables might be related through latent factors.

age	-	age
bp	-	blood pressure
bgr	-	blood glucose random
bu	-	blood urea
sc	-	serum creatinine
sod	-	sodium
pot	-	potassium
hemo	-	hemoglobin
pcv	-	packed cell volume
wbcc	-	white blood cell count
rbcc	-	red blood cell count

After removal of missing values, our dataset contains 215 observattions.

2. Compute the correlation matrix between all indicators (you may want to do this in batches). Comment on relationships you do/do not observe.

```
library(ggcorrplot)

library(dplyr)

CKD.numeric %>%
      cor(use="pairwise.complete.obs", method = "pearson") %>%
      ggcorrplot(hc.order = TRUE, type = "lower",  lab = TRUE,
        outline.col = "white",
        ggtheme = ggplot2::theme_gray,
        colors = c("#6D9EC1", "white", "#E46726"))
```



From the above correlation matrix plot, the more “orange” indicates that two variables more positively correlated, and the more “blue” indicates more negative correlation. Overall, the correlations are high among variables which indicates groupings of homogeneous sets. We could observe some negative correlation and some positive ones. Sc, bu, bp, pot, bgr, age, or wbcc is negatively associated with hemo, rbcc or sod, while hemo, rbcc or sod is positively related with rbcc, sod, or hemo. There might be more than one latent factor.

3. Compute KMO or other measure (i.e. just look at matrix produced above) to comment on suitability of data for factor analysis.

```
library(rela)
fact=paf(as.matrix(CKD.numeric))
#KMO Kaiser-Meyer-Olkin (KMO) measure of adequacy
summary(fact)
```

```
## $KMO
## [1] 0.84752
##
## $MSA
##          MSA
## age  0.84867
## bp   0.83081
## bgr  0.86249
## bu   0.78294
## sc   0.78360
## sod  0.94068
## pot  0.62611
## hemo 0.87072
## pcv  0.86616
## wbcc 0.80101
## rbcc 0.92007
##
## $Bartlett
## [1] 1178.9
##
## $Communalities
##      Initial Communalities Final Extraction
## age                0.12261          0.128799
## bp                 0.17465          0.129793
## bgr                0.21062          0.229236
## bu                 0.77144          0.835088
## sc                 0.77936          0.828293
## sod                0.34748          0.332024
## pot                0.10319          0.047724
## hemo               0.80392          0.835327
## pcv                0.80797          0.859192
## wbcc               0.14645          0.159265
## rbcc               0.65930          0.647896
##
## $Factor.Loadings
##          [,1]      [,2]
## age  -0.26999 -0.236442
## bp   -0.34824 -0.092315
## bgr  -0.38454 -0.285243
## bu   -0.80931  0.424396
## sc   -0.81833  0.398292
## sod   0.57534  0.031768
## pot  -0.18086  0.122533
```

```
## hemo  0.90254  0.144036
## pcv   0.91077  0.172320
## wbcc -0.23748 -0.320731
## rbcc  0.79409  0.131597
##
## $RMS
## [1] 0.038906
```

KMO measure is above 0.8, so factor analysis is meritoriously recommended for this dataset. There are 215 cases for 11 parameters. The sample size is fair for factor analysis. Therefore, the dataset is suitable for factor analysis.

4. Use Principle Components (or appropriate option in Factor Analysis) to decide on a number of latent factors. You can use Scree Plot, eigenvalue>1, or parallel analysis.

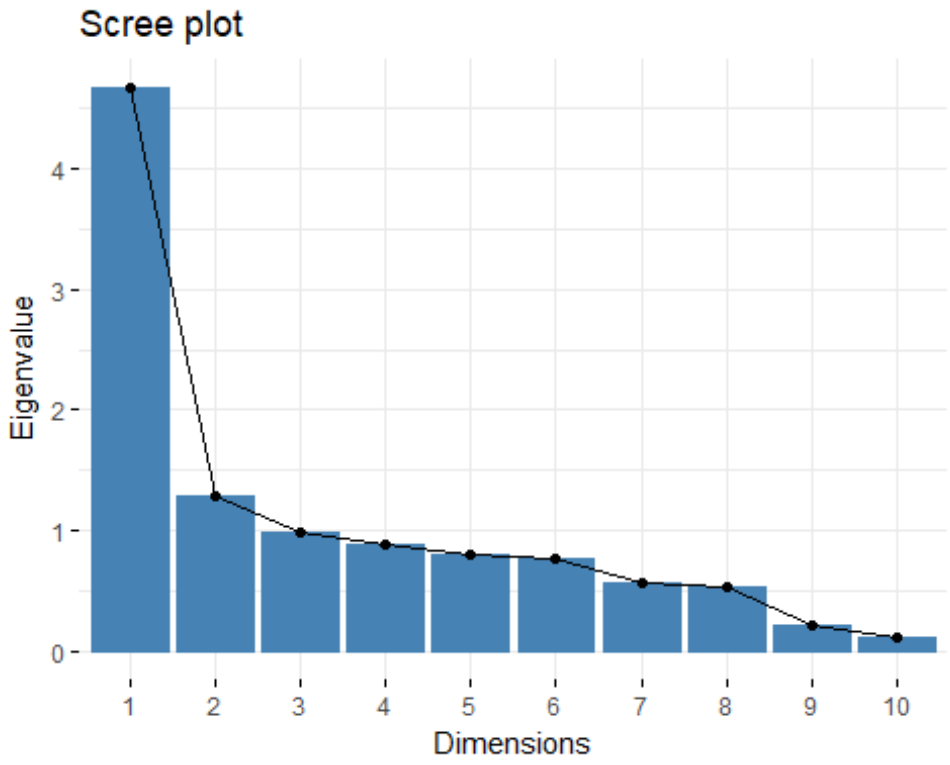
```
    library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
https://goo.gl/13EFCZ
```

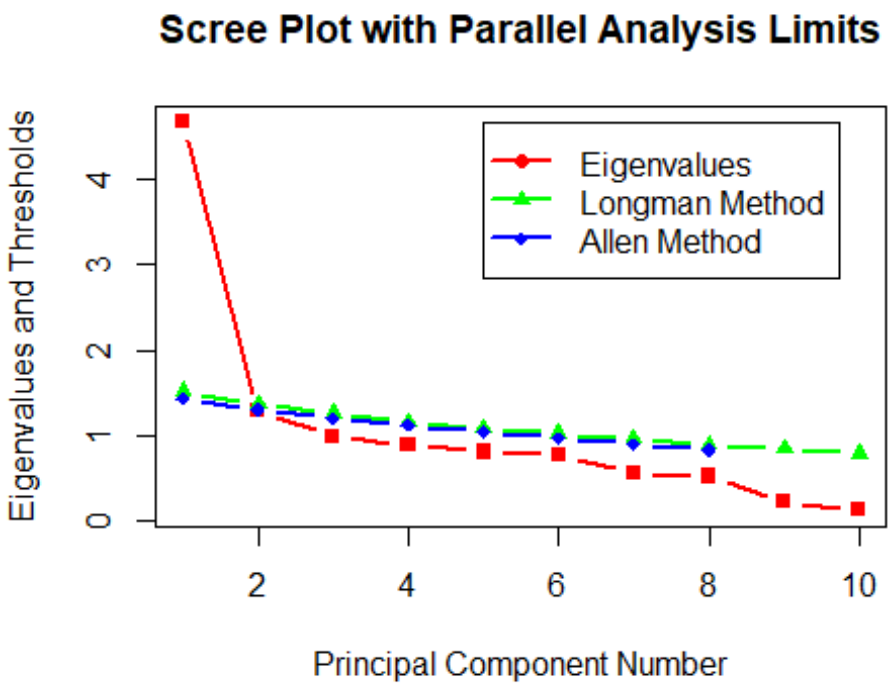
```
    pc1 <- princomp(CKD.numeric, cor=TRUE)
    pc1
```

```
## Call:
## princomp(x = CKD.numeric, cor = TRUE)
##
## Standard deviations:
##  Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9
## 2.16072 1.13517 0.99613 0.94493 0.90178 0.87723 0.75197 0.72896 0.47483
## Comp.10 Comp.11
## 0.36013 0.35041
##
##  11  variables and  215 observations.
```

```
    # Screeplot
    fviz_screeplot(pc1, choice="eigenvalue")
```




```r
source("http://www.reuningscherer.net/STAT660/R/parallel.r.txt")

# Parallel analysis plot
parallelplot(pc1)
```



The first two eigenvalues are larger than 1. The scree plot and the parallel analysis both show that the first two components should retain. Thus, there should be two latent variables.

5. Perform a series of factor analyses using orthogonal models. First, try at least two extraction methods (choose from Principle Components, Principle Axis Factoring, Iterative Principle Components, Maximum Likelihood). Use some method for comparing extraction methods to choose a 'best' method (i.e. RMSR or # residuals greater than .05).

```
#extraction method:Maximum Likelihood
fact1=factanal(CKD.numeric,factors=2,rotation="none")
library(psych)

#extraction method:Principle Axis Factoring
fact2=fa(CKD.numeric,nfactors=2,rotate="none", fm="pa")
#extraction method:iterative PCA
fact3=fa(CKD.numeric,nfactors=2,rotate="none", SMC=FALSE, fm="pa")
#function to get RMSR, or proportion of residuals
comp_fa<-function(method,fact){
  #get reproduced correlation matrix
  repro=fact$loadings%*%t(fact$loadings)
  #residual correlation matrix
  resid=cor(CKD.numeric)-repro
  #get root-mean squared residuals
  len=length(resid[upper.tri(resid)])
  RMSR=sqrt(sum(resid[upper.tri(resid)]^2)/len)
  #get proportion of residuals greater than 0.05 in absolute   value
```

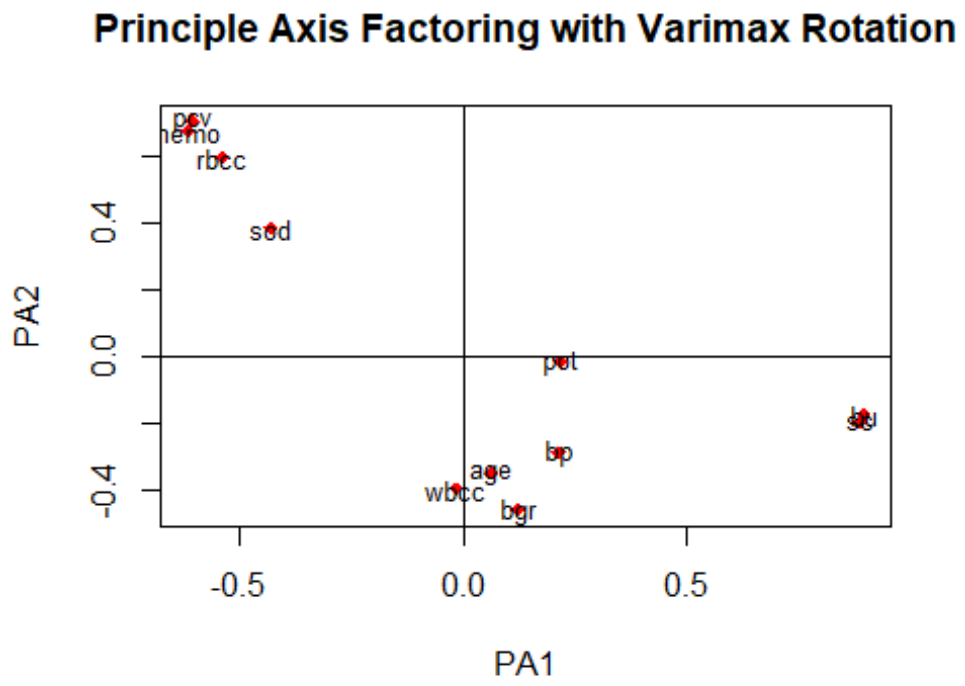
```
  prop=sum(rep(1,len)[abs(resid[upper.tri(resid)])>0.05])/len
  out<-paste("Method:", method, ", RMSR:" ,round(RMSR,3),", proportion of
residuals greater than 0.05 in absolute value", round(prop,3))
  print(out)
  return(out)
}

out1<-comp_fa("Maximum Likelihood",fact1)

## [1] "Method: Maximum Likelihood , RMSR: 0.044 , proportion of residuals
greater than 0.05 in absolute value 0.164"

out2<-comp_fa("Principle Axis Factoring",fact2)

## [1] "Method: Principle Axis Factoring , RMSR: 0.039 , proportion of
residuals greater than 0.05 in absolute value 0.164"

out3<-comp_fa("Iterative PCA",fact3)

## [1] "Method: Iterative PCA , RMSR: 0.039 , proportion of residuals greater
than 0.05 in absolute value 0.164"
```

The above three extration methods without any rotation have pretty similar results. By comparing RMSR, maximum likelihood method is a little bit worse than the other two methods, which might result from the fact that these variables don't follow the multivariate normal distribution. Thus, either principle axis factoring or iterative principle component is a better approach. We will use principle axis factoring for the next question.

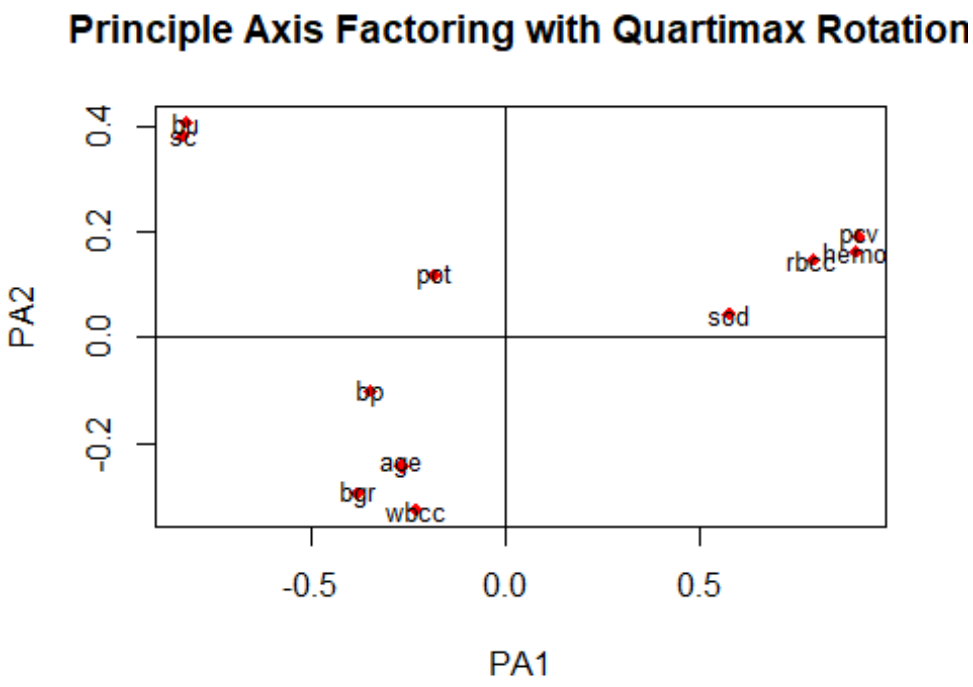
6. Once you've chosen an extraction method, try a varimax and/or a quartimax rotation. Pick one of these rotations and discuss the interpretation of the final factors. Make one or more loading plots as appropriate.

```
#extraction method:Principle Axis Factoring
fact21=fa(CKD.numeric,nfactors=2,rotate="varimax", fm="pa")
fact22=fa(CKD.numeric,nfactors=2,rotate="quartimax", fm="pa")

## Loading required namespace: GPArotation

out1<-comp_fa("Varimax",fact21)

## [1] "Method: Varimax , RMSR: 0.039 , proportion of residuals greater than
0.05 in absolute value 0.164"

out2<-comp_fa("Quartimax",fact22)

## [1] "Method: Quartimax , RMSR: 0.039 , proportion of residuals greater
than 0.05 in absolute value 0.164"

#get loading plot for first two factors
plot(fact21$loadings, pch=18, col='red',main="Principle Axis Factoring with
Varimax Rotation")
abline(h=0)
```

```
abline(v=0)
text(fact21$loadings, labels=names(CKD.numeric),cex=0.8)
```




```
plot(fact22$loadings, pch=18, col='red',main="Principle Axis Factoring with
Quartimax Rotation")
abline(h=0)
abline(v=0)
text(fact22$loadings, labels=names(CKD.numeric),cex=0.8)
```



The residuals for the two types of rotation are the same because rotation won't change the the fit between the observed and reproduced correlation matrices. The above two loading plots seem to be different. Varimax ratation enables each indicator to have a high loading on one and only one factor, while quartimax in addition let indicators have high loading on one factor. The loading plot with varimax rotation shows that bu, sc and pot load heavily on factor one; age, wbcc, and bgr load heavily on factor two. The rest load on both factors, and they load heavily on factor one when the rotation is quartimax. Thus, it is reasonable to conclude that these measurements of patients are a function of two latent factor.