Factorial analysis

S18478 - Anusha Rajaguru

April 2024

1 Introduction

Factor analysis is a method of modelling observed variables and their covariance structure in terms of a smaller number of underlying unobservable "factors".Also Exploratory Factor analysis(EFA) is a strong statistical method used in data analysing for large set of variables.Simply put EFA is a method of simplifying complex dataset.Considering mathematically,Factor model can be a series of multiple regression ,prediction each of the observable variables and common factors.This model represent to $X = LF+E$; Where X is the matrix of observed variables,L is the matrix loadings , F is the matrix of factors and E is the error term.Also to test the model accuracy and validation ,we used Confirmatory factor analysis(CFA) .In CFA claim a predestined hypothesis about number of factors and the pattern of loading Lawley and Maxwell, 1962.

In this report,I Selected a suitable dataset and Applying, analyzing the data using both Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA) using R statistical software.

2 Methodology

Considering the data set,Various organization are ready to provide aid to the countries.to select the countries that need the aid,Thus, it is necessary to classify countries using socio-economic and health factors that determine the overall development of the country.Therefore this data set consider the Death of children under 5 years of age per 1000 live births,Exports of goods and services per capita,Total health spending per capita, Imports of goods and services per capita, Net income per person like variables.

we consider Exploratory factor analysis and Confirmatory factor analysis use as statistical methods for analysis the dataset.

3 Results and discussion

Exploratory factor analysis reveled that the first two components explained by 95% ratio of variation.Also,factor loading are the correlation between the factors and the variables.factor 1 is strongly correlated with life expect,income and GDP.Also negatively correlated with child mort and total fertility.factor 2 is primarily related to exports and imports.therefor first and second factor is primarily measure of these variables.

4 Conclusion and recommendation

Exploratory factor analysis successfully used to analyze the dataset.In conclusion,the processes highlight the various options open and thus requires us to be sensible in choosing the methods depending on the data and the problem statement. Considering limitations of data set,the analysis was limited to the data set.Also in factor analysis can be used in only numerical variables.

References

Lawley, D. N., & Maxwell, A. E. (1962). Factor analysis as a statistical method. *Journal of the Royal Statistical Society. Series D (The Statistician)*, *12*(3), 209–229.

Factor analysis

S18478

2024-04-07

load the packages

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(dplyr)
library(ggplot2)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```r
library(visdat)
```

```
## Warning: package 'visdat' was built under R version 4.3.3
```

```r
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.3.3
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.3.3

##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.2

## corrplot 0.92 loaded
```

```r
 ##library(lavaan)
```

load the data set

```r
country_data <- read_csv(file = "../Multivariate project/Data/Country-data.csv")
```

```
## Rows: 167 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): country
## dbl (9): child_mort, exports, health, imports, income, inflation, life_expec...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(country_data)
```

```
## # A tibble: 6 x 10
##   country           child_mort exports health imports income inflation life_expec
##   <chr>                  <dbl>   <dbl>  <dbl>   <dbl>  <dbl>     <dbl>      <dbl>
## 1 Afghanistan             90.2    10     7.58    44.9   1610      9.44       56.2
## 2 Albania                 16.6    28     6.55    48.6   9930      4.49       76.3
## 3 Algeria                 27.3    38.4   4.17    31.4  12900     16.1        76.5
## 4 Angola                 119      62.3   2.85    42.9   5900     22.4        60.1
## 5 Antigua and Bar~        10.3    45.5   6.03    58.9  19100      1.44       76.8
## 6 Argentina               14.5    18.9   8.1     16    18700     20.9        75.8
## # i 2 more variables: total_fer <dbl>, gdpp <dbl>
```

```r
count_missing <- sum(is.na(country_data))
count_missing
```

```
## [1] 0
```

Remove empty column and rows

```r
country_data <- country_data %>%
  remove_empty(c("cols","rows"))
```

dimension of data set

```r
dim(country_data)
```

```
## [1] 167  10
```

columns name

```r
str(country_data)
```

```
## tibble [167 x 10] (S3: tbl_df/tbl/data.frame)
##  $ country   : chr [1:167] "Afghanistan" "Albania" "Algeria" "Angola" ...
##  $ child_mort: num [1:167] 90.2 16.6 27.3 119 10.3 14.5 18.1 4.8 4.3 39.2 ...
##  $ exports   : num [1:167] 10 28 38.4 62.3 45.5 18.9 20.8 19.8 51.3 54.3 ...
##  $ health    : num [1:167] 7.58 6.55 4.17 2.85 6.03 8.1 4.4 8.73 11 5.88 ...
##  $ imports   : num [1:167] 44.9 48.6 31.4 42.9 58.9 16 45.3 20.9 47.8 20.7 ...
##  $ income    : num [1:167] 1610 9930 12900 5900 19100 18700 6700 41400 43200 16000 ...
##  $ inflation : num [1:167] 9.44 4.49 16.1 22.4 1.44 20.9 7.77 1.16 0.873 13.8 ...
##  $ life_expec: num [1:167] 56.2 76.3 76.5 60.1 76.8 75.8 73.3 82 80.5 69.1 ...
##  $ total_fer : num [1:167] 5.82 1.65 2.89 6.16 2.13 2.37 1.69 1.93 1.44 1.92 ...
##  $ gdpp      : num [1:167] 553 4090 4460 3530 12200 10300 3220 51900 46900 5840 ...
```

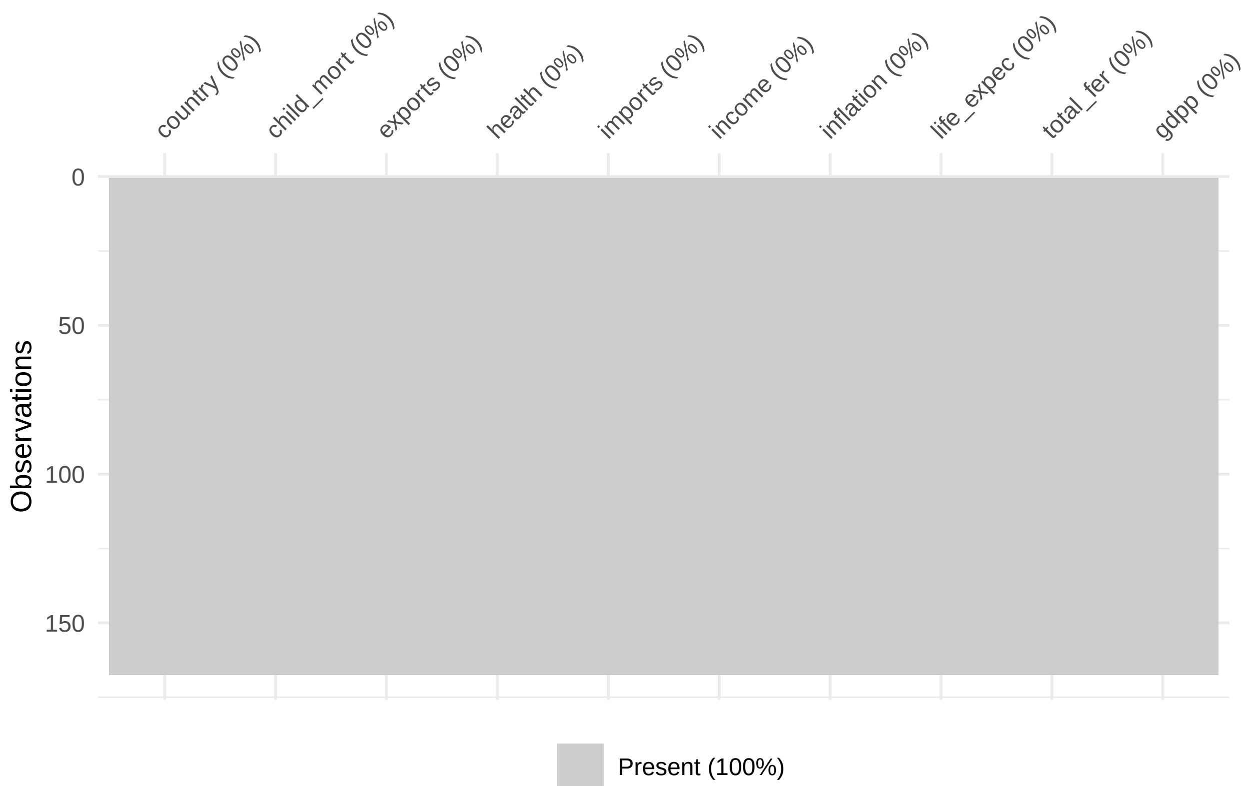
summary of data set

```r
summary(country_data)
```

```
##    country            child_mort        exports            health
##  Length:167         Min.   :  2.60   Min.   :  0.109   Min.   : 1.810
##  Class :character   1st Qu.:  8.25   1st Qu.: 23.800   1st Qu.: 4.920
##  Mode  :character   Median : 19.30   Median : 35.000   Median : 6.320
##                     Mean   : 38.27   Mean   : 41.109   Mean   : 6.816
##                     3rd Qu.: 62.10   3rd Qu.: 51.350   3rd Qu.: 8.600
##                     Max.   :208.00   Max.   :200.000   Max.   :17.900
##     imports             income         inflation         life_expec
##  Min.   :  0.0659   Min.   :   609   Min.   : -4.210   Min.   :32.10
##  1st Qu.: 30.2000   1st Qu.:  3355   1st Qu.:  1.810   1st Qu.:65.30
##  Median : 43.3000   Median :  9960   Median :  5.390   Median :73.10
##  Mean   : 46.8902   Mean   : 17145   Mean   :  7.782   Mean   :70.56
##  3rd Qu.: 58.7500   3rd Qu.: 22800   3rd Qu.: 10.750   3rd Qu.:76.80
##  Max.   :174.0000   Max.   :125000   Max.   :104.000   Max.   :82.80
##    total_fer          gdpp
##  Min.   :1.150   Min.   :   231
##  1st Qu.:1.795   1st Qu.:  1330
##  Median :2.410   Median :  4660
##  Mean   :2.948   Mean   : 12964
##  3rd Qu.:3.880   3rd Qu.: 14050
##  Max.   :7.490   Max.   :105000
```

factor analysis missing values

```r
vis_miss(country_data)
```



Get the numerical data

```r
country_data_numerical <- country_data[,2 :10]
head(country_data_numerical)
```

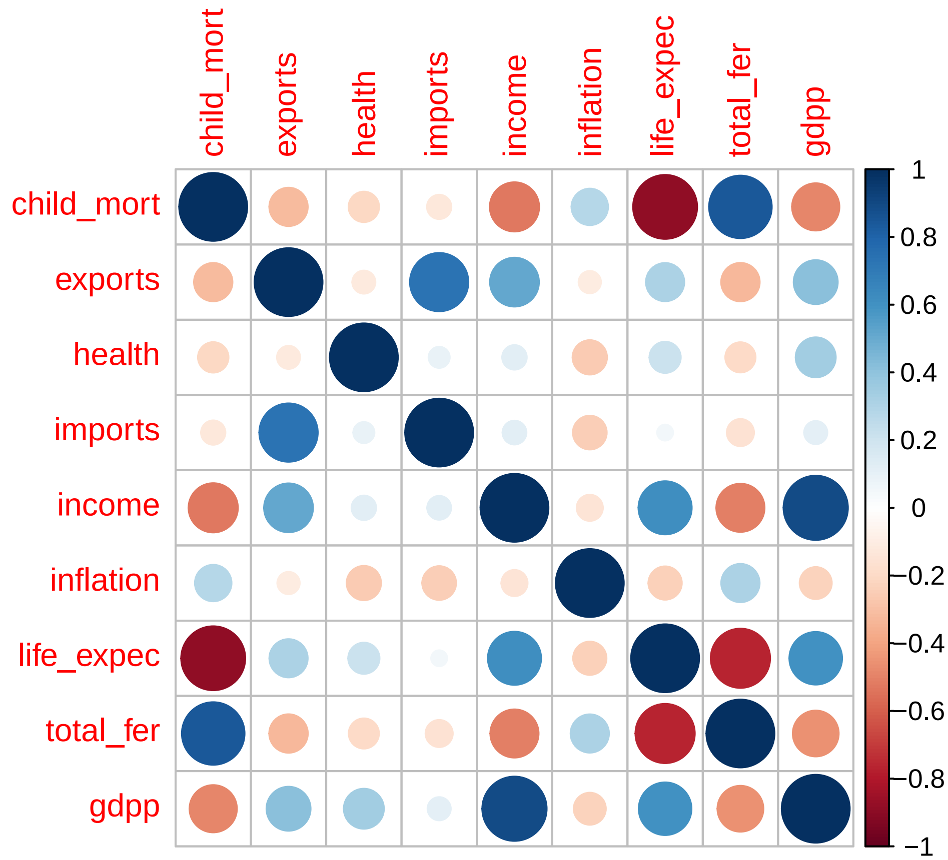
```
## # A tibble: 6 x 9
##   child_mort exports health imports income inflation life_expec total_fer  gdpp
##        <dbl>   <dbl>  <dbl>   <dbl>  <dbl>     <dbl>      <dbl>     <dbl> <dbl>
## 1       90.2    10     7.58    44.9   1610      9.44       56.2      5.82   553
## 2       16.6    28     6.55    48.6   9930      4.49       76.3      1.65  4090
## 3       27.3    38.4   4.17    31.4  12900     16.1        76.5      2.89  4460
## 4      119      62.3   2.85    42.9   5900     22.4        60.1      6.16  3530
## 5       10.3    45.5   6.03    58.9  19100      1.44       76.8      2.13 12200
## 6       14.5    18.9   8.1     16    18700     20.9        75.8      2.37 10300
```

normalized data

```r
data_normalized <- scale(country_data_numerical)
```

correlation of dataset

```r
corr_matrix_country <- cor(data_normalized)
corrplot(corr_matrix_country)
```



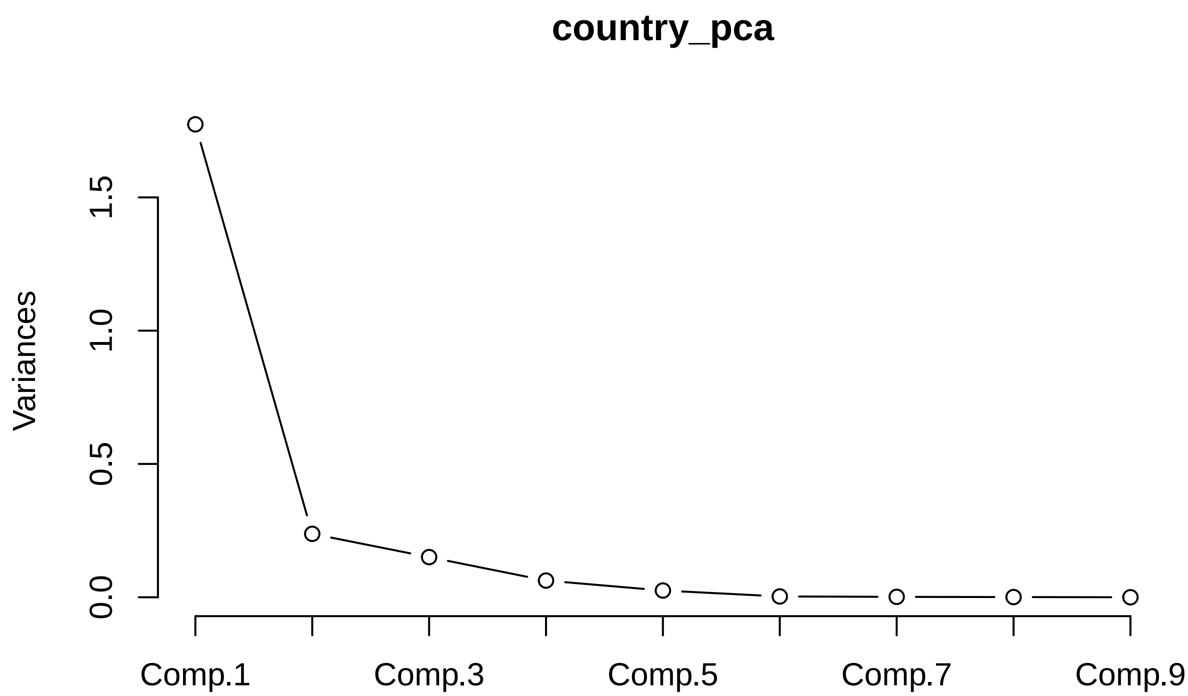
Principle components

```r
country_pca <- princomp(corr_matrix_country)
summary(country_pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     1.3318939 0.4877206 0.38807506 0.25012129 0.15875869
## Proportion of Variance 0.7865816 0.1054743 0.06677839 0.02773996 0.01117582
## Cumulative Proportion  0.7865816 0.8920558 0.95883424 0.98657420 0.99775002
##                             Comp.6       Comp.7       Comp.8       Comp.9
## Standard deviation     0.054823748 0.0377897948 0.0253093633 9.983147e-09
## Proportion of Variance 0.001332729 0.0006332184 0.0002840318 4.419157e-17
## Cumulative Proportion  0.999082750 0.9997159682 1.0000000000 1.000000e+00
```

visualizaion plot- screeplot

```r
plot(country_pca, type = "l")
```



1)correlation

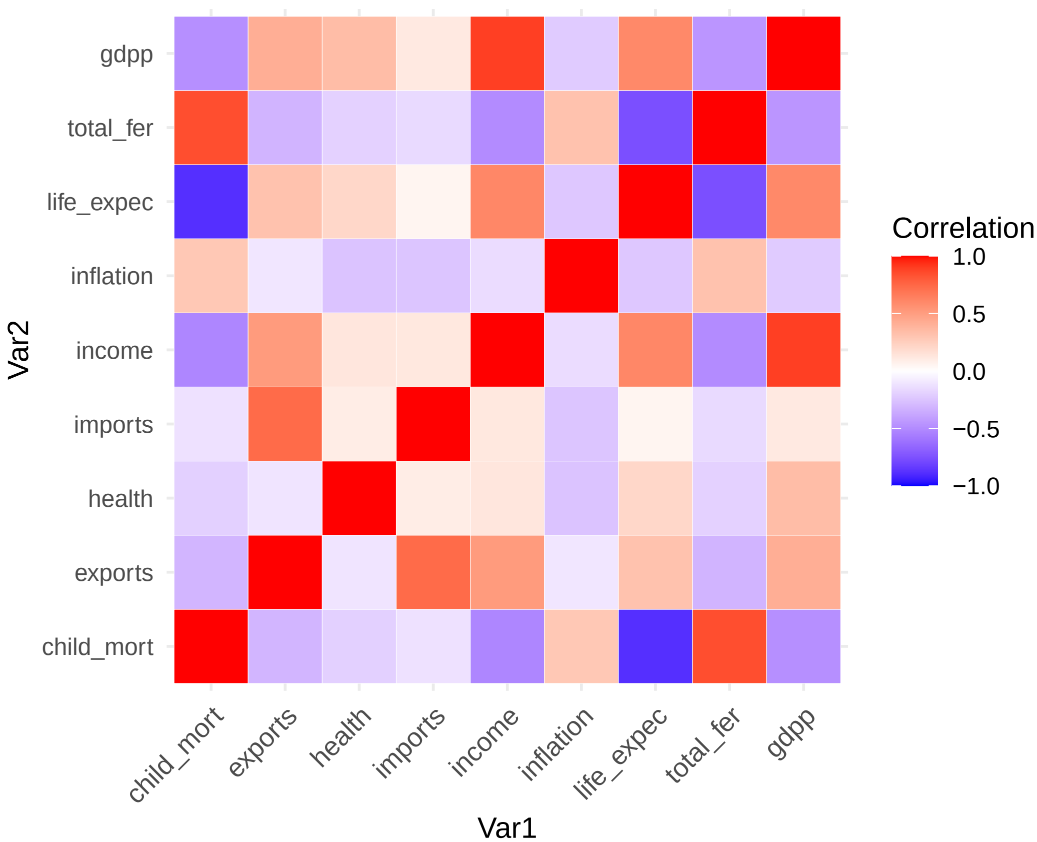
```
correlation_matrix <- cor(country_data[, -1])
print(correlation_matrix)   # View correlation matrix
```

```
##             child_mort    exports      health     imports     income  inflation
## child_mort   1.0000000 -0.3180932 -0.20040206 -0.12721092 -0.5243150  0.2882762
## exports     -0.3180932  1.0000000 -0.11440840  0.73738083  0.5167836 -0.1072944
## health      -0.2004021 -0.1144084  1.00000000  0.09571668  0.1295786 -0.2553758
## imports     -0.1272109  0.7373808  0.09571668  1.00000000  0.1224062 -0.2469943
## income      -0.5243150  0.5167836  0.12957861  0.12240625  1.0000000 -0.1477560
## inflation    0.2882762 -0.1072944 -0.25537579 -0.24699428 -0.1477560  1.0000000
## life_expec  -0.8866761  0.3163126  0.21069212  0.05439053  0.6119625 -0.2397050
## total_fer    0.8484781 -0.3200106 -0.19667399 -0.15904843 -0.5018401  0.3169211
## gdpp        -0.4830322  0.4187248  0.34596553  0.11549817  0.8955714 -0.2216311
##              life_expec  total_fer       gdpp
## child_mort  -0.88667610  0.8484781 -0.4830322
## exports      0.31631260 -0.3200106  0.4187248
## health       0.21069212 -0.1966740  0.3459655
## imports      0.05439053 -0.1590484  0.1154982
## income       0.61196247 -0.5018401  0.8955714
## inflation   -0.23970496  0.3169211 -0.2216311
## life_expec   1.00000000 -0.7608747  0.6000891
## total_fer   -0.76087469  1.0000000 -0.4549103
## gdpp         0.60008913 -0.4549103  1.0000000
```

```r
correlation_long <- melt(correlation_matrix)
```

correlation graph

```r
ggplot(correlation_long, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1), space =
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 10, hjust = 1)) +
  coord_fixed()
```



2) factor analysis examine factor structure

```r
factor_country <- fa(country_data_numerical,nfactors = 2,scores = "none")
```

```
## Loading required namespace: GPArotation

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : I
## am sorry, to do these rotations requires the GPArotation package to be
## installed

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs = np.obs, :
## The estimated weights for the factor scores are probably incorrect.  Try a
## different factor score estimation method.
```

```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected.  Examine the results carefully
```

```r
factor_country
```

```
## Factor Analysis using method =  minres
## Call: fa(r = country_data_numerical, nfactors = 2, scores = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##              MR1   MR2    h2     u2 com
## child_mort -0.84  0.27 0.783  0.217 1.2
## exports     0.61  0.82 1.041 -0.041 1.8
## health      0.24 -0.17 0.086  0.914 1.8
## imports     0.30  0.61 0.466  0.534 1.5
## income      0.76  0.04 0.575  0.425 1.0
## inflation  -0.32  0.01 0.101  0.899 1.0
## life_expec  0.87 -0.30 0.853  0.147 1.2
## total_fer  -0.79  0.20 0.658  0.342 1.1
## gdpp        0.73 -0.02 0.540  0.460 1.0
##
##                        MR1  MR2
## SS loadings           3.82 1.28
## Proportion Var        0.42 0.14
## Cumulative Var        0.42 0.57
## Proportion Explained  0.75 0.25
## Cumulative Proportion 0.75 1.00
##
## Mean item complexity =  1.3
## Test of the hypothesis that 2 factors are sufficient.
##
## df null model =  36  with the objective function =  7.21 with Chi Square =  1169.74
## df of  the model are 19  and the objective function was  2.44
##
## The root mean square of the residuals (RMSR) is  0.1
## The df corrected root mean square of the residuals is  0.14
##
## The harmonic n.obs is  167 with the empirical chi square  124.87  with prob <  1.4e-17
## The total n.obs was  167  with Likelihood Chi Square =  392.01  with prob <  2e-71
##
## Tucker Lewis Index of factoring reliability =  0.371
## RMSEA index =  0.343  and the 90 % confidence intervals are  0.315 0.374
## BIC =  294.77
## Fit based upon off diagonal values = 0.94
```

```r
factor_analysis <- factanal(country_data_numerical,factors = 2,scores = "none")
factor_analysis
```

```
##
## Call:
## factanal(x = country_data_numerical, factors = 2, scores = "none")
##
## Uniquenesses:
## child_mort    exports     health    imports     income  inflation life_expec
##      0.101      0.005      0.904      0.431      0.510      0.912      0.133
```

```
##  total_fer       gdpp
##      0.265      0.594
##
## Loadings:
##            Factor1 Factor2
## child_mort -0.945
## exports     0.260   0.963
## health      0.250  -0.183
## imports             0.753
## income      0.590   0.376
## inflation  -0.294
## life_expec  0.928
## total_fer  -0.851  -0.103
## gdpp        0.572   0.281
##
##                Factor1 Factor2
## SS loadings      3.372   1.773
## Proportion Var   0.375   0.197
## Cumulative Var   0.375   0.572
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 362.21 on 19 degrees of freedom.
## The p-value is 3.04e-65
```

uniqueness

```r
factor_analysis$uniquenesses
```

```
## child_mort    exports     health    imports     income  inflation life_expec
##  0.1013083  0.0050000  0.9040554  0.4308088  0.5096096  0.9121534  0.1332916
##  total_fer       gdpp
##  0.2652859  0.5944138
```

community - square loadings for variables

```r
apply(factor_analysis$loadings^2, 1, sum)
```

```
## child_mort    exports     health    imports     income  inflation life_expec
## 0.89869197 0.99501613 0.09599866 0.56920060 0.49037948 0.08786639 0.86670979
##  total_fer       gdpp
## 0.73471933 0.40558382
```

1- uniqueness specific variences

```r
1 - apply(factor_analysis$loadings^2, 1, sum)
```

```
##  child_mort     exports      health     imports      income   inflation
## 0.101308031 0.004983868 0.904001340 0.430799396 0.509620517 0.912133613
##  life_expec   total_fer        gdpp
## 0.133290206 0.265280668 0.594416182
```

residual matrix

```r
Loading_factor <- factor_analysis$loadings
specific_variance<- diag(factor_analysis$uniquenesses)
S <- factor_analysis$correlation
Sigma_matrix <- Loading_factor %*% t(Loading_factor) + specific_variance

Loading_factor
```

```
##
## Loadings:
##            Factor1 Factor2
## child_mort -0.945
## exports     0.260   0.963
## health      0.250  -0.183
## imports             0.753
## income      0.590   0.376
## inflation  -0.294
## life_expec  0.928
## total_fer  -0.851  -0.103
## gdpp        0.572   0.281
##
##                Factor1 Factor2
## SS loadings      3.372   1.773
## Proportion Var   0.375   0.197
## Cumulative Var   0.375   0.572
```

```r
specific_variance
```

```
##            [,1]  [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
##  [1,] 0.1013083 0.000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##  [2,] 0.0000000 0.005 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##  [3,] 0.0000000 0.000 0.9040554 0.0000000 0.0000000 0.0000000 0.0000000
##  [4,] 0.0000000 0.000 0.0000000 0.4308088 0.0000000 0.0000000 0.0000000
##  [5,] 0.0000000 0.000 0.0000000 0.0000000 0.5096096 0.0000000 0.0000000
##  [6,] 0.0000000 0.000 0.0000000 0.0000000 0.0000000 0.9121534 0.0000000
##  [7,] 0.0000000 0.000 0.0000000 0.0000000 0.0000000 0.0000000 0.1332916
##  [8,] 0.0000000 0.000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##  [9,] 0.0000000 0.000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000
##            [,8]      [,9]
##  [1,] 0.0000000 0.0000000
##  [2,] 0.0000000 0.0000000
##  [3,] 0.0000000 0.0000000
##  [4,] 0.0000000 0.0000000
##  [5,] 0.0000000 0.0000000
##  [6,] 0.0000000 0.0000000
##  [7,] 0.0000000 0.0000000
##  [8,] 0.2652859 0.0000000
##  [9,] 0.0000000 0.5944138
```

```r
S
```

```
##            child_mort    exports      health     imports     income  inflation
## child_mort  1.0000000 -0.3180932 -0.20040206 -0.12721092 -0.5243150  0.2882762
```

```
## exports    -0.3180932  1.0000000 -0.11440840  0.73738083  0.5167836 -0.1072944
## health     -0.2004021 -0.1144084  1.00000000  0.09571668  0.1295786 -0.2553758
## imports    -0.1272109  0.7373808  0.09571668  1.00000000  0.1224062 -0.2469943
## income     -0.5243150  0.5167836  0.12957861  0.12240625  1.0000000 -0.1477560
## inflation   0.2882762 -0.1072944 -0.25537579 -0.24699428 -0.1477560  1.0000000
## life_expec -0.8866761  0.3163126  0.21069212  0.05439053  0.6119625 -0.2397050
## total_fer   0.8484781 -0.3200106 -0.19667399 -0.15904843 -0.5018401  0.3169211
## gdpp       -0.4830322  0.4187248  0.34596553  0.11549817  0.8955714 -0.2216311
##             life_expec  total_fer       gdpp
## child_mort -0.88667610  0.8484781 -0.4830322
## exports     0.31631260 -0.3200106  0.4187248
## health      0.21069212 -0.1966740  0.3459655
## imports     0.05439053 -0.1590484  0.1154982
## income      0.61196247 -0.5018401  0.8955714
## inflation  -0.23970496  0.3169211 -0.2216311
## life_expec  1.00000000 -0.7608747  0.6000891
## total_fer  -0.76087469  1.0000000 -0.4549103
## gdpp        0.60008913 -0.4549103  1.0000000
```

```r
Sigma_matrix  # varience-covariance matrix
```

```
##             child_mort    exports      health     imports      income
## child_mort  1.0000002 -0.3180581 -0.22225935 -0.09589550 -0.58637352
## exports    -0.3180581  1.0000161 -0.11162277  0.73626483  0.51593312
## health     -0.2222593 -0.1116228  1.00005404 -0.12772633  0.07851642
## imports    -0.0958955  0.7362648 -0.12772633  1.00000936  0.30802566
## income     -0.5863735  0.5159331  0.07851642  0.30802566  0.99998910
## inflation   0.2808394 -0.1090013 -0.06738367 -0.03761456 -0.18659871
## life_expec -0.8825478  0.3161174  0.21745310  0.09717565  0.57715848
## total_fer   0.8119045 -0.3201706 -0.19371693 -0.11276168 -0.54121984
## gdpp       -0.5613663  0.4188352  0.09136061  0.23512268  0.44322122
##               inflation  life_expec  total_fer        gdpp
## child_mort  0.28083944 -0.88254781  0.8119045 -0.56136629
## exports    -0.10900126  0.31611745 -0.3201706  0.41883517
## health     -0.06738367  0.21745310 -0.1937169  0.09136061
## imports    -0.03761456  0.09717565 -0.1127617  0.23512268
## income     -0.18659871  0.57715848 -0.5412198  0.44322122
## inflation   1.00001980 -0.27583550  0.2540760 -0.17782310
## life_expec -0.27583550  1.00000138 -0.7974597  0.55222318
## total_fer   0.25407597 -0.79745973  1.0000052 -0.51534473
## gdpp       -0.17782310  0.55222318 -0.5153447  0.99999761
```

model fit

```r
# Define the model
model <- '
    f1 =~ child_mort + exports + health  + imports
    f2 =~ income + inflaion + life_expec + total_fer + gdpp
    f1 ~~ f1
    f2 ~~ f2'

#fit_model <- cfa(model, data = country_data_numerical)
#summary(fit_model)
```