PCA

Ruiqi (Rickey) Huang

2/7/2021

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Read and summarize the data

```r
TimesRanking <- read.csv("TimesCountryAvg.csv")
#dim(TimesRanking)
head(TimesRanking)
```

```
##       country scoreAvg scoreRankAvg universityCount studentAvg
## 1   Singapore 77.40000     33.00000               2   27978.50
## 2   Hong Kong 62.41667     97.33333               6   14243.17
## 3 Netherlands 61.49231     86.69231              13   19599.15
## 4 Switzerland 57.39091    131.36364              11   11755.91
## 5     Belgium 54.31250    138.37500               8   23646.50
## 6  Luxembourg 53.70000    133.00000               1    4654.00
##   studentToStaffAvg intlStudentAvg femalePercentageAvg malePercentageAvg
## 1          16.90000           0.28                0.50              0.51
## 2          17.46667           0.37                0.38              0.28
## 3          18.51538           0.22                0.48              0.45
## 4          14.67273           0.33                0.45              0.46
## 5          33.07500           0.19                0.53              0.47
## 6          18.70000           0.49                0.51              0.49
##   TeachingAvg researchAvg citationAvg industryIncomeAvg intlOutlookAvg  Health
## 1    67.20000    80.40000    80.90000          67.65000       95.30000 Healthy
## 2    48.76667    53.93333    76.40000          54.98333       97.56667 Healthy
## 3    42.70769    55.58462    80.23077          72.95385       81.39231 Healthy
## 4    42.25455    43.57273    76.82727          60.94545       94.22727 Healthy
## 5    37.48750    48.18750    71.07500          72.67500       72.81250 Healthy
## 6    37.90000    36.50000    75.80000          45.20000       99.70000 Healthy
```

Create a unified scale for all possible variables

Create a covariance matrix S for our data

After calculating all the covariances, we could find the *total variance* by sum all variance we have (we calculate the $\sum_{j=1}^{k} S_{jj}$, and this is actually the sum of eigenvalues of S)

```r
S <- cov(TimesRanking[, 6:13])
S
```

```
##                     studentToStaffAvg intlStudentAvg femalePercentageAvg
## studentToStaffAvg         68.11084773  -0.0152830778       -0.0516302818
## intlStudentAvg            -0.01528308   0.0193196727        0.0007082537
## femalePercentageAvg       -0.05163028   0.0007082537        0.0095852246
```

```
## malePercentageAvg          0.16755099  -0.0009486980       -0.0037544792
## TeachingAvg               -5.48268733   0.5452874991       -0.1227469943
## researchAvg                9.50910222   0.8852364502       -0.1374788522
## citationAvg               17.34237117   1.3265014942        0.1538665164
## industryIncomeAvg          6.08180400   0.3482762559       -0.1773172500
##                       malePercentageAvg  TeachingAvg  researchAvg citationAvg
## studentToStaffAvg          0.167550993  -5.48268733   9.50910222  17.3423712
## intlStudentAvg            -0.000948698   0.54528750   0.88523645   1.3265015
## femalePercentageAvg       -0.003754479  -0.12274699  -0.13747885   0.1538665
## malePercentageAvg          0.007468359  -0.02435635  -0.05819818  -0.2182919
## TeachingAvg               -0.024356346  76.86102028 103.10372539 108.8117705
## researchAvg               -0.058198182 103.10372539 167.87015797 196.8685889
## citationAvg               -0.218291929 108.81177045 196.86858886 514.7632007
## industryIncomeAvg          0.034649212  54.61114640  87.60290104  73.5667509
##                       industryIncomeAvg
## studentToStaffAvg          6.08180400
## intlStudentAvg             0.34827626
## femalePercentageAvg       -0.17731725
## malePercentageAvg          0.03464921
## TeachingAvg               54.61114640
## researchAvg               87.60290104
## citationAvg               73.56675089
## industryIncomeAvg         87.26294500
```

```
sum(diag(S))
```

```
## [1] 914.9045
```

find eigenvalues and eigenvectors of S

```
s.eigen <- eigen(S)
s.eigen
```

```
## eigen() decomposition
## $values
## [1] 6.631527e+02 1.460963e+02 6.949147e+01 2.904171e+01 7.092600e+00
## [6] 1.436054e-02 1.104666e-02 4.314181e-03
##
## $vectors
##               [,1]          [,2]          [,3]          [,4]         [,5]
## [1,]  3.118182e-02  0.0361542981  0.9784538847  0.1447369922 -0.139241131
## [2,]  2.569285e-03 -0.0014031869 -0.0010704334  0.0036249392  0.003512949
## [3,]  7.346979e-06  0.0020199662 -0.0009010043  0.0004800651  0.003268692
## [4,] -3.071561e-04 -0.0005673364  0.0025546129 -0.0002710978 -0.004419459
## [5,]  2.498929e-01 -0.4130435127 -0.1602562882  0.3983335700 -0.763268512
## [6,]  4.242159e-01 -0.5319615965  0.0372443694  0.3872507995  0.621023794
## [7,]  8.472824e-01  0.5106801893 -0.0351748762 -0.1289049789 -0.058847984
## [8,]  1.967943e-01 -0.5332017971  0.1196370304 -0.8085718587 -0.094111059
##               [,6]          [,7]          [,8]
## [1,] -0.0016289660  0.0024238021  1.771678e-03
## [2,] -0.9623460710 -0.2691096581  3.787921e-02
## [3,] -0.2380591905  0.7675888099 -5.950794e-01
## [4,]  0.1310447413 -0.5816882223 -8.027699e-01
## [5,] -0.0009573243  0.0040913786  6.329005e-04
## [6,]  0.0057824675 -0.0009069602 -1.616399e-03
```

```
## [7,]  0.0010322690 -0.0006046305  1.770973e-04
## [8,] -0.0026579719  0.0015557640 -8.779132e-05
```

The eigenvectors represent the principal components of S. The eigenvalues of S are used to find the proportion of the total variance explained by the components.

```
for (s in s.eigen$values) {
  print(s / sum(s.eigen$values))
}
```

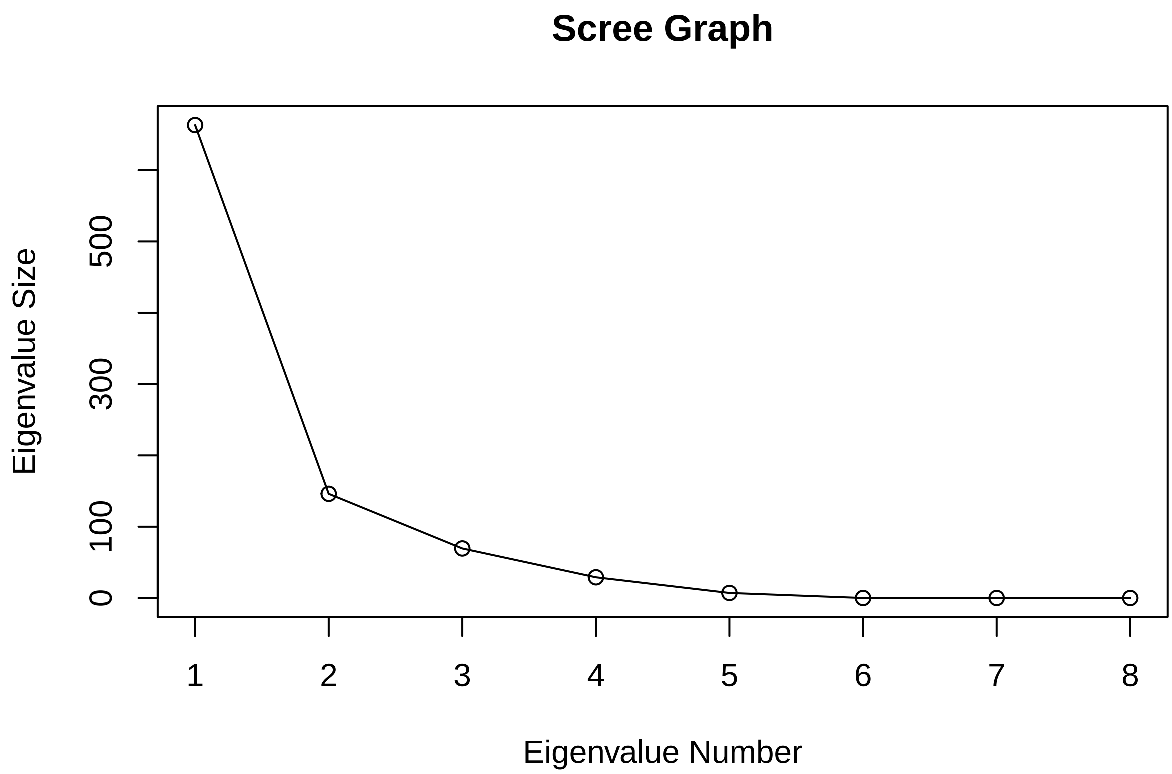
```
## [1] 0.7248327
## [1] 0.1596848
## [1] 0.07595489
## [1] 0.03174288
## [1] 0.007752284
## [1] 1.569622e-05
## [1] 1.207412e-05
## [1] 4.715444e-06
```

The first two principal components account for 88.3% of the total variance.

(*TODO: Here we need to know what level of total variance we want our model to explain*)

A scree graph of the eigenvalues can be plotted to visualize the proportion of variance explained by each subsequential eigenvalue.

```
plot(s.eigen$values, xlab = 'Eigenvalue Number', ylab = 'Eigenvalue Size', main = 'Scree Graph')
lines(s.eigen$values)
```




```
s.eigen$vectors
```

```
##                [,1]         [,2]         [,3]         [,4]         [,5]
## [1,]  3.118182e-02  0.0361542981  0.9784538847  0.1447369922 -0.139241131
```

```
## [2,]  2.569285e-03 -0.0014031869 -0.0010704334  0.0036249392  0.003512949
## [3,]  7.346979e-06  0.0020199662 -0.0009010043  0.0004800651  0.003268692
## [4,] -3.071561e-04 -0.0005673364  0.0025546129 -0.0002710978 -0.004419459
## [5,]  2.498929e-01 -0.4130435127 -0.1602562882  0.3983335700 -0.763268512
## [6,]  4.242159e-01 -0.5319615965  0.0372443694  0.3872507995  0.621023794
## [7,]  8.472824e-01  0.5106801893 -0.0351748762 -0.1289049789 -0.058847984
## [8,]  1.967943e-01 -0.5332017971  0.1196370304 -0.8085718587 -0.094111059
##               [,6]          [,7]          [,8]
## [1,] -0.0016289660  0.0024238021  1.771678e-03
## [2,] -0.9623460710 -0.2691096581  3.787921e-02
## [3,] -0.2380591905  0.7675888099 -5.950794e-01
## [4,]  0.1310447413 -0.5816882223 -8.027699e-01
## [5,] -0.0009573243  0.0040913786  6.329005e-04
## [6,]  0.0057824675 -0.0009069602 -1.616399e-03
## [7,]  0.0010322690 -0.0006046305  1.770973e-04
## [8,] -0.0026579719  0.0015557640 -8.779132e-05
```

use the eigenvectors to find out the results

The elements of the eigenvectors of S are the 'coefficients' or 'loadings' of the principal components.

here if we decide we are going to use the first two components as the principal components we can write out the formula for these two principal compnents from the eigenvectors result above:

1^{st} principal component

$y_1 = 0.03118182 * StudentToStaffAvg + 0.0361542981 * intlStudentAvg + 0.9784538847 * femalePersentageAvg + 0.144736992 * malepercentageAvg - 0.139241131 * teachingAvg - 0.0016289660 * reseachAvg + 0.0024238021 * citation + 0.001771678 * industryIncomeAvg$

$y_1 = 0.002569285 * StudentToStaffAvg - 0.0014031869 * intlStudentAvg - 0.0010704334 * femalePersentageAvg + 0.0036249392 * malepercentageAvg + 0.003512949 * teachingAvg - 0.9623460710 * reseachAvg - 0.2691096581 * citation + 0.03787921e * industryIncomeAvg$

Analyze the principal components

From the result above the y_1 combines variables StudentToStaffAvg, intlStudent, femalePercentageAvg, and teaching, while the component y_2 measures the change in the variables like reserchAvg, citation, and industryIncomAvg.

In this case, we might name our component y_1 as Demographical factor and name the second component y_2 as Educational Outcome factor.

Another method to do PCA

```r
TimesRanking.pca <- prcomp(TimesRanking[,6:13], scale = TRUE)
TimesRanking.pca
```

```
## Standard deviations (1, .., p=8):
## [1] 1.8106116 1.2609273 1.0072916 0.8621236 0.7349729 0.7033148 0.5290381
## [8] 0.2431800
##
## Rotation (n x k) = (8 x 8):
##                              PC1          PC2          PC3        PC4
## studentToStaffAvg     0.03244624 -0.331575039  0.854989353 -0.1982215
## intlStudentAvg        0.33736531  0.165088148  0.107063311  0.7652856
## femalePercentageAvg -0.06882505  0.619131198  0.366583867 -0.1008543
```

```
## malePercentageAvg   -0.02596116 -0.657538438  0.033164499  0.3417735
## TeachingAvg          0.50310224 -0.009263632 -0.189516015 -0.1006269
## researchAvg          0.53202162 -0.011920320  0.008629425 -0.1267727
## citationAvg          0.40458227  0.146466566  0.264329973  0.1443715
## industryIncomeAvg    0.42411712 -0.159509547 -0.127223882 -0.4483584
##                             PC5         PC6          PC7          PC8
## studentToStaffAvg    0.18622349 -0.22615495 -0.148390466  0.104219537
## intlStudentAvg       0.48113483 -0.12461464  0.116845642  0.033668027
## femalePercentageAvg  0.10168293  0.67410837 -0.051213194  0.000215117
## malePercentageAvg   -0.13605819  0.65521585  0.004229929 -0.034966748
## TeachingAvg         -0.01439532  0.10417536 -0.601434558  0.572692743
## researchAvg         -0.03918907  0.02646607 -0.280413874 -0.787271834
## citationAvg         -0.74390885 -0.07452615  0.377367147  0.150952856
## industryIncomeAvg    0.38708584  0.18024460  0.615569045  0.127363066
```

```
summary(TimesRanking.pca)
```

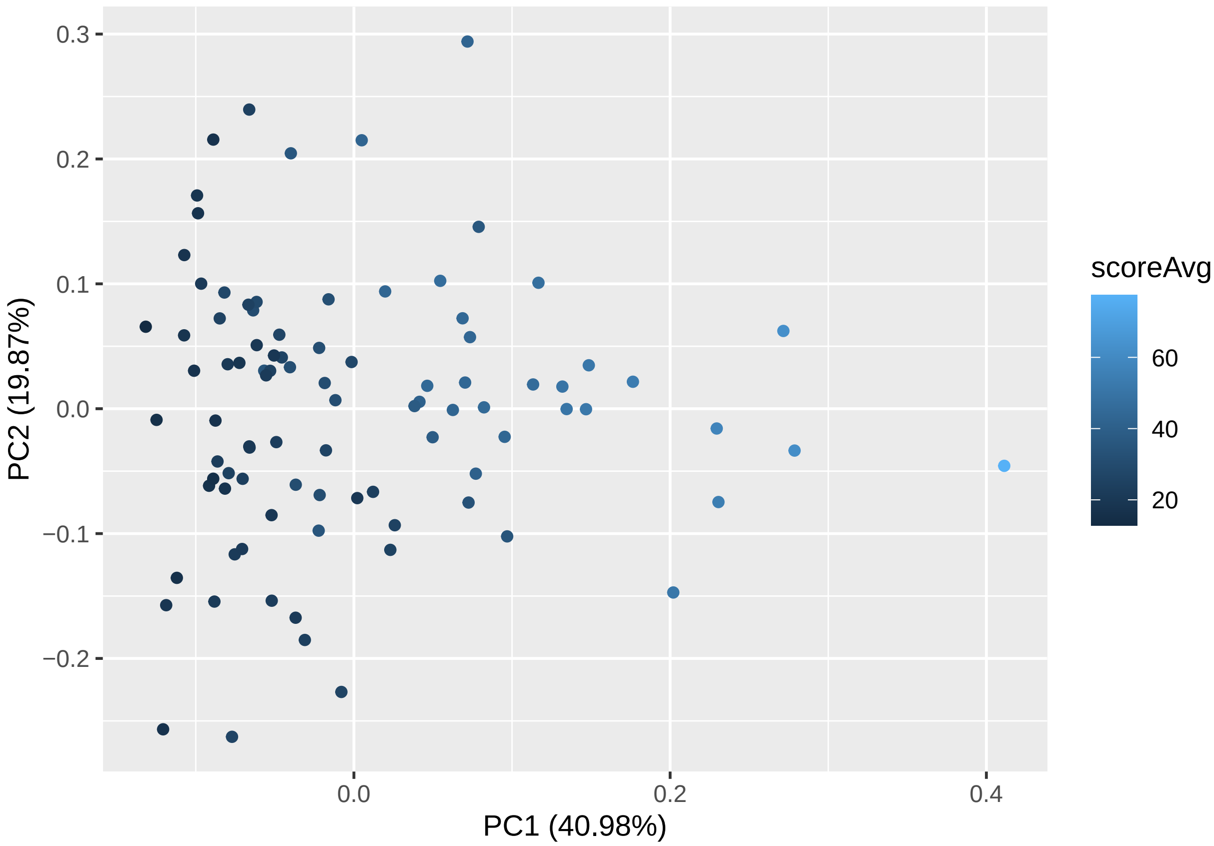
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.8106 1.2609 1.0073 0.86212 0.73497 0.70331 0.52904
## Proportion of Variance 0.4098 0.1987 0.1268 0.09291 0.06752 0.06183 0.03499
## Cumulative Proportion  0.4098 0.6085 0.7354 0.82827 0.89579 0.95762 0.99261
##                            PC8
## Standard deviation     0.24318
## Proportion of Variance 0.00739
## Cumulative Proportion  1.00000
```

Plot the principal components

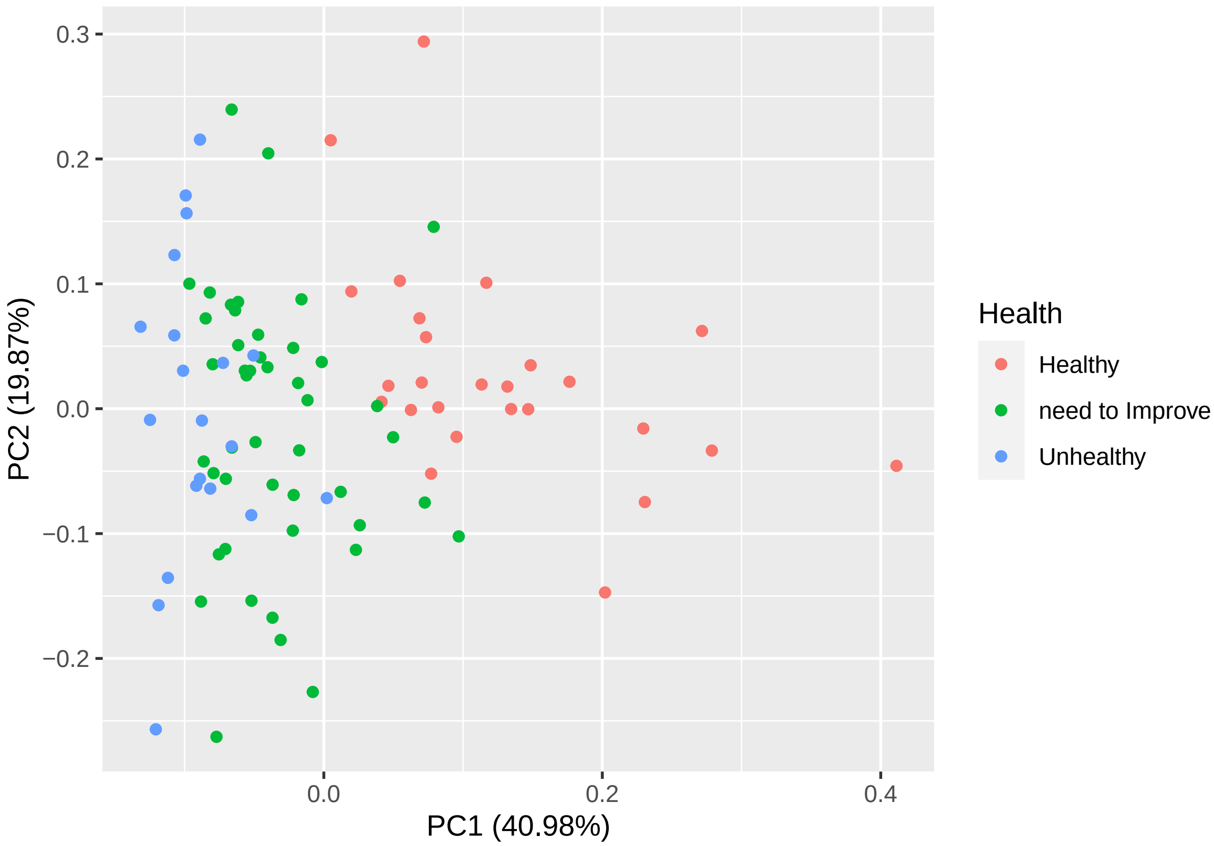
```
TR.PCA.plot <- autoplot(TimesRanking.pca, data = TimesRanking, colour = 'scoreAvg')
```

```
## Warning: `select_()` is deprecated as of dplyr 0.7.0.
## Please use `select()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
TR.PCA.plot
```




```r
TR.PCA.plot2 <- autoplot(TimesRanking.pca, data = TimesRanking, colour = 'Health')
TR.PCA.plot2
```

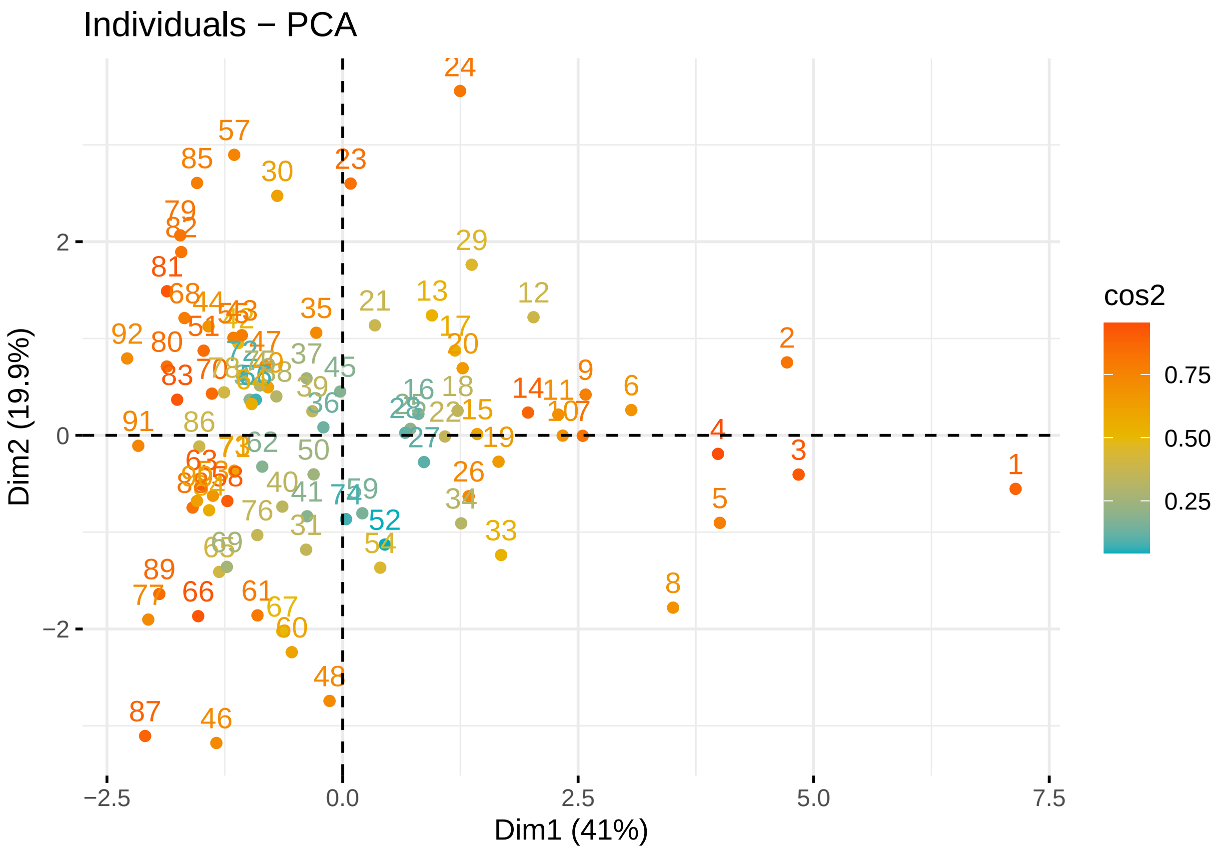


TODO: Here we still need to find some possible categorical or numerical response variable (output) to measure the level of health (may be some indirectly related variables), in order to chack whether our PC1 & 2 like above.

In the plots above, I used the scoreAvg (Times ranking score) to show the cluster in the first plot, which shows that a higher score means a combination of a high PC1 and a high PC2. For the seconf plot, I manually labelled countries with score from 40 to the max as healthy, countries with score from 20 to 40 as neet to improve, and countries from min to 20 as unhealthy. The plot shows that the countries with "unhealthy" education system have a low PC1 and a median PC2, and the countries "need improvement" in education have a median PC1 and a low or median PC2, while the countries with a "healthy" education sytem have a high PC1 and a median or high PC2. This way of evaluating Health variable may not be practical, some more reasonable response variable need to be chosen.

Visualizations

```
TR.Individuals <- fviz_pca_ind(TimesRanking.pca,
                              col.ind = "cos2", #color by the quality of representation
                              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), #set up the colors
                              reprel = TRUE #avoid test overlapping
                              )
TR.Individuals
```


Individuals − PCA
Dim1 (41%)
Dim2 (19.9%)
cos2
0.75
0.50
0.25
−2.5
0.0
2.5
5.0
7.5
2
0
−2