Midterm.R

Surya

2019-03-29

```
## Surya Aenuganti Ushasri

protein <- read.csv("Protein_Consumption.csv",header=TRUE)
head(protein)
```

```
##       ï..Country Red.Meat White.Meat Egg Milk Fish Cereals Starchy.Foods
## 1        Albania       10          1   1    9    0      42             1
## 2        Austria        9         14   4   20    2      28             4
## 3        Belgium       14          9   4   18    5      27             6
## 4       Bulgaria        8          6   2    8    1      57             1
## 5 Czechoslovakia       10         11   3   13    2      34             5
## 6        Denmark       11         11   4   25   10      22             5
##   Pulses.Nuts.and.Oilseeds Fruits.and.Vegetables Total
## 1                        6                     2    72
## 2                        1                     4    86
## 3                        2                     4    89
## 4                        4                     4    91
## 5                        1                     4    83
## 6                        1                     2    91
```

```
attach(protein)
#View(protein)

# 1. Use principal components analysis to investigate the relationships
#between the countries on the basis of these variables

#The data set does not contain any categorical values and it also does not contain any missing values

#Removing the last column because it's just sum of all columns across each country

cor(protein[c(-1,-11)])
```

```
##                             Red.Meat  White.Meat         Egg       Milk
## Red.Meat                  1.00000000  0.18850977  0.57532001  0.5440251
## White.Meat                0.18850977  1.00000000  0.60095535  0.2974816
## Egg                       0.57532001  0.60095535  1.00000000  0.6130310
## Milk                      0.54402512  0.29748163  0.61303102  1.0000000
## Fish                      0.06491072 -0.19719960  0.04780844  0.1624624
## Cereals                  -0.50970337 -0.43941908 -0.70131040 -0.5924925
## Starchy.Foods             0.15383673  0.33456770  0.41266333  0.2144917
## Pulses.Nuts.and.Oilseeds -0.40988882 -0.67214885 -0.59519381 -0.6238357
## Fruits.and.Vegetables    -0.06393465 -0.07329308 -0.16392249 -0.3997753
##                                 Fish     Cereals Starchy.Foods
## Red.Meat                  0.06491072 -0.50970337     0.1538367
## White.Meat               -0.19719960 -0.43941908     0.3345677
## Egg                       0.04780844 -0.70131040     0.4126633
## Milk                      0.16246239 -0.59249246     0.2144917
## Fish                      1.00000000 -0.51714759     0.4386841
```

```
## Cereals                    -0.51714759  1.00000000    -0.5781345
## Starchy.Foods               0.43868411 -0.57813449     1.0000000
## Pulses.Nuts.and.Oilseeds -0.12226043  0.63605948    -0.4951880
## Fruits.and.Vegetables      0.22948842  0.04229293     0.0683567
##                          Pulses.Nuts.and.Oilseeds Fruits.and.Vegetables
## Red.Meat                               -0.4098888           -0.06393465
## White.Meat                             -0.6721488           -0.07329308
## Egg                                    -0.5951938           -0.16392249
## Milk                                   -0.6238357           -0.39977527
## Fish                                   -0.1222604            0.22948842
## Cereals                                 0.6360595            0.04229293
## Starchy.Foods                          -0.4951880            0.06835670
## Pulses.Nuts.and.Oilseeds                1.0000000            0.35133227
## Fruits.and.Vegetables                   0.3513323            1.00000000
```

```
pca <- prcomp(protein[,c(-1,-11)],scale=TRUE)
pca
```

```
## Standard deviations (1, .., p=9):
## [1] 2.0237432 1.2747169 1.0417887 0.9513238 0.6532516 0.5890163 0.5191570
## [8] 0.3667732 0.3339091
##
## Rotation (n x k) = (9 x 9):
##                                 PC1         PC2         PC3         PC4
## Red.Meat                 -0.3106693 -0.06957085 -0.35546338 -0.59650142
## White.Meat               -0.3159279 -0.21457197  0.62841986 -0.03961214
## Egg                      -0.4205930 -0.09986721  0.08050675 -0.25525634
## Milk                     -0.3788776 -0.16867961 -0.40414435  0.03223542
## Fish                     -0.1341071  0.65161517 -0.29971395  0.23487897
## Cereals                   0.4298291 -0.25366332  0.06815673  0.02030764
## Starchy.Foods            -0.2959618  0.38888491  0.28085511  0.30524504
## Pulses.Nuts.and.Oilseeds  0.4218085  0.12932932 -0.14030066 -0.25125596
## Fruits.and.Vegetables     0.1223681  0.50377330  0.34041535 -0.60376932
##                                  PC5         PC6         PC7          PC8
## Red.Meat                  0.39658595 -0.37671581  0.22797808 -0.049688240
## White.Meat               -0.31059983 -0.08129384  0.14601621 -0.028186225
## Egg                       0.06707700  0.66453033  0.03595386 -0.467400341
## Milk                     -0.31800256  0.01779923 -0.71798985  0.102202763
## Fish                     -0.30432982 -0.04476482  0.23683595 -0.440552318
## Cereals                   0.18501820 -0.19398782 -0.34306417 -0.720660760
## Starchy.Foods             0.67317396  0.02444741 -0.32554187  0.082975933
## Pulses.Nuts.and.Oilseeds  0.09378094  0.58676016 -0.03105426  0.217739473
## Fruits.and.Vegetables    -0.22763119 -0.15823653 -0.35941199  0.009714519
##                                 PC9
## Red.Meat                 -0.2506754
## White.Meat               -0.5766036
## Egg                       0.2750188
## Milk                     -0.1903416
## Fish                     -0.2600351
## Cereals                  -0.1921878
## Starchy.Foods            -0.1499922
## Pulses.Nuts.and.Oilseeds -0.5666397
## Fruits.and.Vegetables     0.2114057
```

```r
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.0237 1.2747 1.0418 0.9513 0.65325 0.58902 0.51916
## Proportion of Variance 0.4551 0.1805 0.1206 0.1006 0.04742 0.03855 0.02995
## Cumulative Proportion  0.4551 0.6356 0.7562 0.8568 0.90417 0.94272 0.97266
##                            PC8     PC9
## Standard deviation     0.36677 0.33391
## Proportion of Variance 0.01495 0.01239
## Cumulative Proportion  0.98761 1.00000
```

```r
#Reading from the summary of pca table we can see that upto pc5 about 90% of variance is captured

plot(pca)
#from the above plot we see that pca1 accounts for maximum variance in the data
pca$x
```

```
##              PC1         PC2          PC3         PC4        PC5
##  [1,]  3.4062175 -1.43187183 -1.596648133 -0.08434257  0.4124395
##  [2,] -1.3961709 -1.07844406  1.234558817 -0.02919248 -0.7564630
##  [3,] -1.6271911  0.27394175 -0.009163712 -0.41608341  0.9108462
##  [4,]  3.0996115 -1.50333675  0.082356700 -0.30660707 -0.2970873
##  [5,] -0.4277883 -0.57418064  1.159335459  0.21991003  0.3701307
##  [6,] -2.4422594  0.28305004 -0.676942687  1.02016258 -0.6562849
##  [7,] -1.4249913  0.60782538  1.746831101  0.87710306  0.6028516
##  [8,] -1.7006498 -0.58298031 -1.972677332  1.58071748 -0.2011453
##  [9,] -1.4354297  0.89590251 -0.161539920 -1.95053301  0.3099538
## [10,]  2.3291742  0.86546599 -1.227337046 -1.75741320 -0.6575195
## [11,]  1.4302687 -0.95052166  1.782611863  0.26555332 -0.1057918
## [12,] -2.5809791 -0.82037615 -0.161750192 -0.51252848  0.8610870
## [13,]  1.5501576  0.16192833 -0.053056104 -1.33599650 -0.7676190
## [14,] -1.7115591 -0.78012960  0.766301047 -0.25865817 -0.9164207
## [15,] -0.9571511  1.10929163 -1.319851198  1.21615923 -0.4173226
## [16,] -0.1285106  0.63184836  1.522555810 -0.03104612 -0.1228267
## [17,]  1.8854364  4.23632323  0.235407502  0.64127627 -0.3296311
## [18,]  2.6361730 -1.10164486  0.169166371  0.60431439  0.1965040
## [19,]  1.4042842  2.43957843  0.249276728 -0.24228673  0.6238140
## [20,] -1.9196053 -0.08881654 -1.085799797  0.90373795 -0.7886161
## [21,] -0.8862644 -0.79798276 -0.228906351 -1.06865159 -0.7103254
## [22,] -1.9396765 -0.32877834 -1.274231236 -1.19215725  1.2311866
## [23,]  0.8607657 -0.15774231 -0.215679913  1.04275420  1.2112175
## [24,] -1.8007758 -0.34409820  0.872728311 -0.26262846 -0.1813817
## [25,]  3.7769132 -0.96425165  0.162453908  1.07643653  0.1784042
##                 PC6         PC7         PC8         PC9
##  [1,] -0.2667144820  0.94892837  0.84693053  0.15478609
##  [2,]  0.0237975418  0.05758584 -0.05177819  0.11624278
##  [3,] -0.1269263837  0.22683921 -0.22319293 -0.09689498
##  [4,] -0.5842119100  0.39976618 -0.90940273  0.25018422
##  [5,] -0.7261570266  0.29971869 -0.06798719  0.25074519
##  [6,]  0.0627184045  0.48030200 -0.56925372 -0.50886295
##  [7,]  0.2138448106  0.53117349 -0.18580431  0.29526903
##  [8,] -0.2058406000 -0.97347796  0.28022893  0.12113082
##  [9,] -1.4755527601 -0.03008584 -0.06846045 -0.51649154
```

```
## [10,]  1.0097312103 -0.57538334 -0.34740216 -0.45103458
## [11,]  0.8657732666  0.11900810  0.19668872 -0.44150330
## [12,]  0.6415595029 -0.43471746  0.03742272 -0.05217871
## [13,]  0.0312818001 -0.14708797 -0.12872601  0.85624862
## [14,]  0.3040553671  0.06091030  0.35043459 -0.28870555
## [15,]  0.0038561601  0.04796743 -0.05700862  0.18258443
## [16,] -0.3479854540 -1.31643147 -0.01492251  0.31505313
## [17,] -0.5280805539  0.53140483  0.20289705 -0.20295441
## [18,]  0.1708000230 -0.04058813 -0.17580879 -0.13304725
## [19,]  1.0132276525 -0.14851022  0.27557451  0.36210459
## [20,]  0.2848678709  0.41870881 -0.19737555  0.30259740
## [21,] -0.6895174928 -0.21158255  0.59042991  0.03956071
## [22,]  0.6339274501  0.43367349 -0.24441516  0.13761916
## [23,] -0.5814776989 -0.72141844 -0.05214970 -0.11645720
## [24,]  0.2726945750  0.39030488  0.53225955 -0.13919641
## [25,]  0.0003287263 -0.34700826 -0.01917849 -0.43679928
```

```
pca.cty <- cbind(data.frame(protein[,1]),pca$x)
pca.cty
```

```
##      protein...1.        PC1         PC2          PC3         PC4
## 1         Albania  3.4062175 -1.43187183 -1.596648133 -0.08434257
## 2         Austria -1.3961709 -1.07844406  1.234558817 -0.02919248
## 3         Belgium -1.6271911  0.27394175 -0.009163712 -0.41608341
## 4        Bulgaria  3.0996115 -1.50333675  0.082356700 -0.30660707
## 5  Czechoslovakia -0.4277883 -0.57418064  1.159335459  0.21991003
## 6         Denmark -2.4422594  0.28305004 -0.676942687  1.02016258
## 7    East Germany -1.4249913  0.60782538  1.746831101  0.87710306
## 8         Finland -1.7006498 -0.58298031 -1.972677332  1.58071748
## 9          France -1.4354297  0.89590251 -0.161539920 -1.95053301
## 10         Greece  2.3291742  0.86546599 -1.227337046 -1.75741320
## 11        Hungary  1.4302687 -0.95052166  1.782611863  0.26555332
## 12        Ireland -2.5809791 -0.82037615 -0.161750192 -0.51252848
## 13          Italy  1.5501576  0.16192833 -0.053056104 -1.33599650
## 14    Netherlands -1.7115591 -0.78012960  0.766301047 -0.25865817
## 15         Norway -0.9571511  1.10929163 -1.319851198  1.21615923
## 16         Poland -0.1285106  0.63184836  1.522555810 -0.03104612
## 17       Portugal  1.8854364  4.23632323  0.235407502  0.64127627
## 18        Romania  2.6361730 -1.10164486  0.169166371  0.60431439
## 19          Spain  1.4042842  2.43957843  0.249276728 -0.24228673
## 20         Sweden -1.9196053 -0.08881654 -1.085799797  0.90373795
## 21    Switzerland -0.8862644 -0.79798276 -0.228906351 -1.06865159
## 22 United Kingdom -1.9396765 -0.32877834 -1.274231236 -1.19215725
## 23           USSR  0.8607657 -0.15774231 -0.215679913  1.04275420
## 24   West Germany -1.8007758 -0.34409820  0.872728311 -0.26262846
## 25     Yugoslavia  3.7769132 -0.96425165  0.162453908  1.07643653
##           PC5           PC6         PC7         PC8         PC9
## 1   0.4124395 -0.2667144820  0.94892837  0.84693053  0.15478609
## 2  -0.7564630  0.0237975418  0.05758584 -0.05177819  0.11624278
## 3   0.9108462 -0.1269263837  0.22683921 -0.22319293 -0.09689498
## 4  -0.2970873 -0.5842119100  0.39976618 -0.90940273  0.25018422
## 5   0.3701307 -0.7261570266  0.29971869 -0.06798719  0.25074519
## 6  -0.6562849  0.0627184045  0.48030200 -0.56925372 -0.50886295
## 7   0.6028516  0.2138448106  0.53117349 -0.18580431  0.29526903
## 8  -0.2011453 -0.2058406000 -0.97347796  0.28022893  0.12113082
```

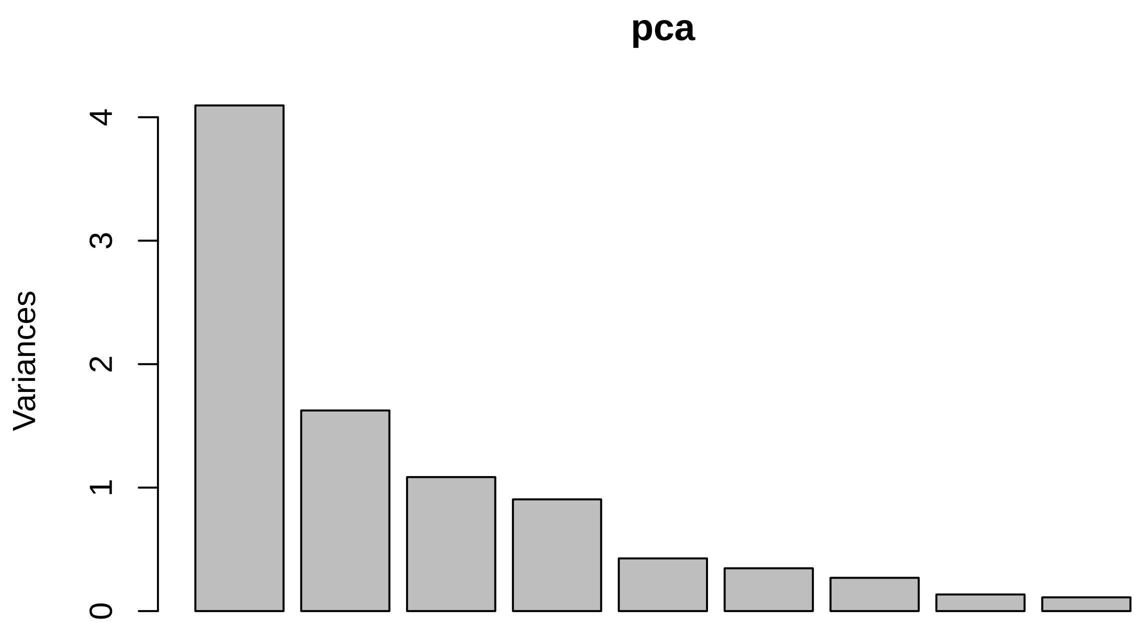
```
## 9   0.3099538 -1.4755527601 -0.03008584 -0.06846045 -0.51649154
## 10 -0.6575195  1.0097312103 -0.57538334 -0.34740216 -0.45103458
## 11 -0.1057918  0.8657732666  0.11900810  0.19668872 -0.44150330
## 12  0.8610870  0.6415595029 -0.43471746  0.03742272 -0.05217871
## 13 -0.7676190  0.0312818001 -0.14708797 -0.12872601  0.85624862
## 14 -0.9164207  0.3040553671  0.06091030  0.35043459 -0.28870555
## 15 -0.4173226  0.0038561601  0.04796743 -0.05700862  0.18258443
## 16 -0.1228267 -0.3479854540 -1.31643147 -0.01492251  0.31505313
## 17 -0.3296311 -0.5280805539  0.53140483  0.20289705 -0.20295441
## 18  0.1965040  0.1708000230 -0.04058813 -0.17580879 -0.13304725
## 19  0.6238140  1.0132276525 -0.14851022  0.27557451  0.36210459
## 20 -0.7886161  0.2848678709  0.41870881 -0.19737555  0.30259740
## 21 -0.7103254 -0.6895174928 -0.21158255  0.59042991  0.03956071
## 22  1.2311866  0.6339274501  0.43367349 -0.24441516  0.13761916
## 23  1.2112175 -0.5814776989 -0.72141844 -0.05214970 -0.11645720
## 24 -0.1813817  0.2726945750  0.39030488  0.53225955 -0.13919641
## 25  0.1784042  0.0003287263 -0.34700826 -0.01917849 -0.43679928
```

```
library(devtools)
install_github("vqv/ggbiplot")
```

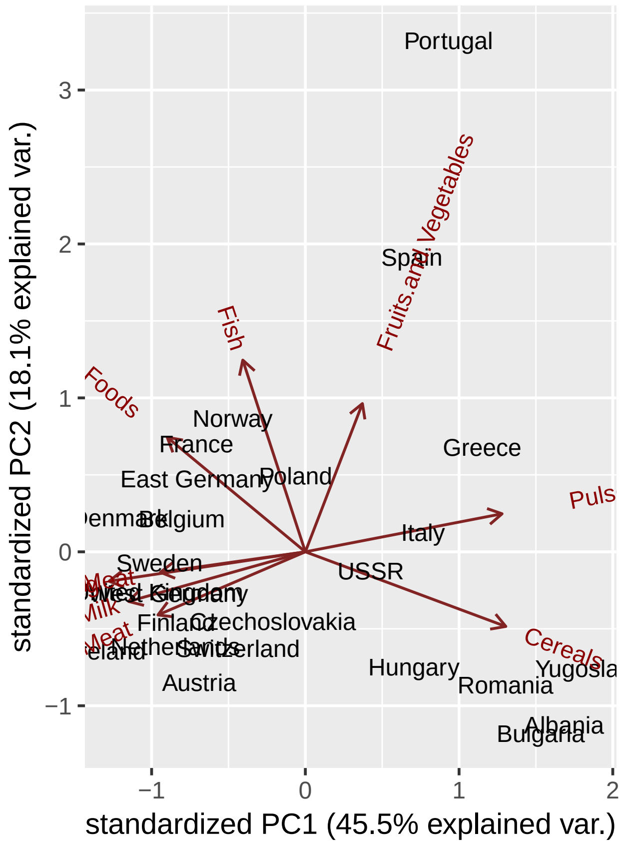
```
## Skipping install of 'ggbiplot' from a github remote, the SHA1 (7325e880) has not changed since last i
##   Use `force = TRUE` to force installation
```

```
library(ggbiplot)
```

```
## Loading required package: ggplot2

## Loading required package: plyr

## Loading required package: scales

## Loading required package: grid
```




```r
ggbiplot(pca,ellipse=TRUE,labels=protein[,1])
```




```r
#When we plot the countries and their 2 principal components leading to 54% of variance
#It Shows the concentraion of countries according to the protein consumption from various sources




# 2. Carry out cluster analysis to study relation between countries on their diet
#I will be using agglomerative clustering approach because there are less than 50 points and it's easy
#Creating a distant matrix using euclidean distance
row.names(protein) <- protein[,1]
dist.mat <- dist(protein[c(-1,-11)], method="euclidean")
dist.mat
```

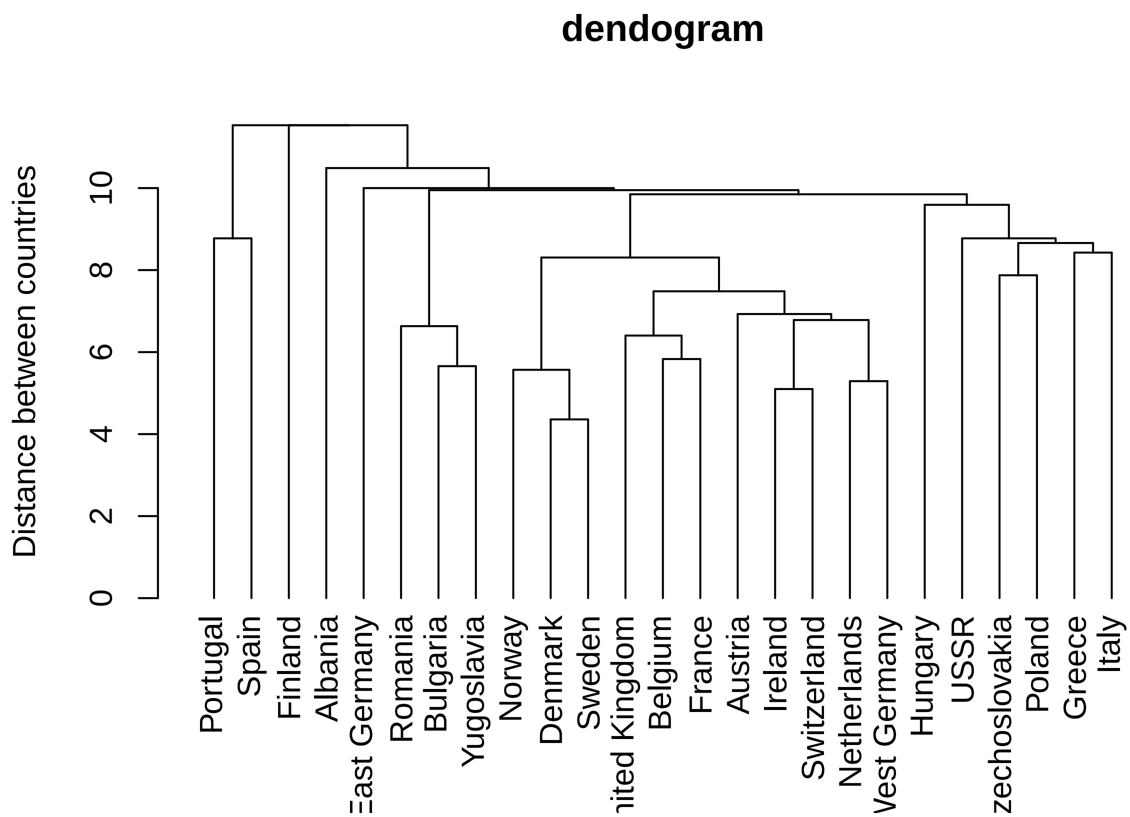
```
##                  Albania   Austria   Belgium  Bulgaria Czechoslovakia
## Austria        23.194827
## Belgium        21.563859  8.306624
## Bulgaria       16.278821 32.756679 33.075671
## Czechoslovakia 15.264338  9.848858 10.295630 24.698178
## Denmark        30.116441 11.958261 10.862780 40.767634      18.920888
## East Germany   22.737634 10.630146 10.000000 33.674916      10.198039
## Finland        31.288976 17.578396 17.378147 41.255303      23.790755
## France         23.643181 11.180340  5.830952 34.000000      13.266499
## Greece         12.449900 20.024984 18.439089 19.748418      15.362291
## Hungary        13.000000 16.881943 19.026298 18.708287       9.591663
## Ireland        27.784888 10.000000  9.219544 38.794329      17.117243
## Italy          10.488088 14.899664 13.820275 21.283797       8.660254
## Netherlands    28.495614  6.928203  9.949874 39.230090      16.031220
```

```
## Norway         26.664583 13.527749 10.488088 38.548671      18.000000
## Poland         17.464249 10.049876 12.083046 24.939928       7.874008
## Portugal       22.891046 22.891046 19.313208 33.600595      18.947295
## Romania        10.816654 25.709920 26.267851  8.246211      18.000000
## Spain          16.881943 17.635192 14.142136 29.495762      13.190906
## Sweden         29.933259 12.884099 11.532563 41.725292      19.773720
## Switzerland    24.657656  7.483315  7.681146 35.623026      14.177447
## United Kingdom 24.494897 12.489996  6.403124 37.080992      15.842980
## USSR           11.401754 18.973666 18.466185 16.822604      12.688578
## West Germany   28.774989  9.486833  9.746794 40.632499      16.401219
## Yugoslavia     15.968719 32.046841 32.741411  5.656854      24.494897
##                  Denmark East Germany   Finland    France    Greece
## Austria
## Belgium
## Bulgaria
## Czechoslovakia
## Denmark
## East Germany   15.748016
## Finland        12.328828    24.454039
## France         12.409674    14.491377 18.055470
## Greece         24.779023    22.538855 24.698178 18.920888
## Hungary        26.608269    16.911535 30.133038 21.725561 15.231546
## Ireland         9.110434    16.703293 11.618950 10.246951 23.259407
## Italy          21.886069    15.842980 24.186773 15.652476  8.426150
## Netherlands     8.306624    13.228757 15.459625 12.041595 24.596748
## Norway          6.928203    15.165751 12.328828 13.114877 21.633308
## Poland         18.055470    14.352700 20.149442 14.000000 12.489996
## Portugal       24.020824    15.198684 31.511903 22.022716 22.472205
## Romania        33.704599    26.795522 34.234486 27.928480 13.784049
## Spain          21.118712    11.916375 26.645825 17.606817 16.733201
## Sweden          4.358899    16.217275 11.532563 13.747727 25.317978
## Switzerland    10.049876    15.066519 13.076697  8.306624 20.174241
## United Kingdom 11.090537    15.132746 15.459625  7.681146 21.189620
## USSR           25.436195    21.470911 25.159491 19.974984  8.774964
## West Germany   10.148892    10.908712 18.947295 12.609520 26.362853
## Yugoslavia     39.824616    33.075671 39.673669 34.205263 18.708287
##                  Hungary   Ireland     Italy Netherlands    Norway
## Austria
## Belgium
## Bulgaria
## Czechoslovakia
## Denmark
## East Germany
## Finland
## France
## Greece
## Hungary
## Ireland        24.879711
## Italy          10.630146 20.099751
## Netherlands    23.259407  7.211103 20.149442
## Norway         25.179357 11.357817 18.841444   11.618950
## Poland         11.661904 16.217275  8.660254   15.842980 16.733201
## Portugal       21.886069 27.276363 18.275667   24.979992 20.124612
## Romania        11.916375 31.764760 14.525839   32.046841 31.272992
```

```
## Spain          16.431677 21.748563 11.532563   20.518285 16.792856
## Sweden         27.404379  8.944272 22.000000    8.602325  5.567764
## Switzerland    21.794495  5.099020 16.492423    6.782330 11.090537
## United Kingdom 24.145393  7.483315 17.720045   11.224972 10.535654
## USSR           12.206556 23.280893  9.591663   24.738634 23.000000
## West Germany   24.062419  9.797959 20.976177    5.291503 12.288206
## Yugoslavia     17.720045 38.039453 20.760539   38.379682 37.255872
##                   Poland  Portugal   Romania     Spain    Sweden
## Austria
## Belgium
## Bulgaria
## Czechoslovakia
## Denmark
## East Germany
## Finland
## France
## Greece
## Hungary
## Ireland
## Italy
## Netherlands
## Norway
## Poland
## Portugal       21.189620
## Romania        17.776389 27.802878
## Spain          15.165751  8.774964 22.759613
## Sweden         19.000000 24.124676 34.452866 20.663978
## Switzerland    13.228757 24.939928 28.722813 19.313208  9.899495
## United Kingdom 16.941074 23.151674 30.380915 17.804494 10.392305
## USSR           10.723805 24.372115  9.949874 18.411953 26.229754
## West Germany   18.027756 22.847319 33.481338 18.947295  9.695360
## Yugoslavia     23.748684 32.787193  6.633250 28.425341 40.632499
##                Switzerland United Kingdom      USSR West Germany
## Austria
## Belgium
## Bulgaria
## Czechoslovakia
## Denmark
## East Germany
## Finland
## France
## Greece
## Hungary
## Ireland
## Italy
## Netherlands
## Norway
## Poland
## Portugal
## Romania
## Spain
## Sweden
## Switzerland
## United Kingdom    7.874008
```

```
## USSR              20.736441      22.181073
## West Germany       9.695360      10.862780 26.532998
## Yugoslavia        35.028560      36.783148 15.524175    39.962482
```

```
#Invoking hclust using single linkage
dist.nn <- hclust(dist.mat, method = "single")
plot(as.dendrogram(dist.nn),ylab="Distance between countries",main="dendogram")
```




```
# From the plot we can see various clusters
#Note: Portugal and spain are clustered in 1 and scandinavian nations are clustered in 1.


# 3. Identify the important factors underlying the observed variables and examine the relationships bet
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:scales':
##
##     alpha, rescale

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

```
#Do an eigen value decomposition removing the last column
pc <- principal(protein[c(-1,-11)], nfactors=4, rotate="varimax")
pc
```

```
## Principal Components Analysis
## Call: principal(r = protein[c(-1, -11)], nfactors = 4, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##                            RC3   RC1   RC2   RC4   h2    u2 com
## Red.Meat                  0.08  0.92  0.01  0.02 0.86 0.138 1.0
## White.Meat                0.94  0.14 -0.08 -0.01 0.91 0.086 1.1
## Egg                       0.59  0.66  0.13 -0.09 0.81 0.193 2.1
## Milk                      0.20  0.68  0.21 -0.51 0.81 0.188 2.3
## Fish                     -0.21  0.10  0.92  0.09 0.91 0.089 1.2
## Cereals                  -0.42 -0.56 -0.61  0.07 0.87 0.133 2.8
## Starchy.Foods             0.52  0.01  0.71  0.03 0.77 0.226 1.8
## Pulses.Nuts.and.Oilseeds -0.69 -0.34 -0.28  0.41 0.83 0.166 2.6
## Fruits.and.Vegetables    -0.05 -0.04  0.14  0.95 0.93 0.071 1.1
##
##                        RC3  RC1  RC2  RC4
## SS loadings           2.25 2.21 1.89 1.36
## Proportion Var        0.25 0.25 0.21 0.15
## Cumulative Var        0.25 0.50 0.71 0.86
## Proportion Explained  0.29 0.29 0.25 0.18
## Cumulative Proportion 0.29 0.58 0.82 1.00
##
## Mean item complexity =  1.8
## Test of the hypothesis that 4 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.05
##  with the empirical chi square  4.94  with prob <  0.55
##
## Fit based upon off diagonal values = 0.98
```

```r
#From the summary we can see that upto 4 factors the variables explain about 86% of the variance
round(pc$values, 3)
```

```
## [1] 4.096 1.625 1.085 0.905 0.427 0.347 0.270 0.135 0.111
```

```r
pc$loadings
```

```
##
## Loadings:
##                          RC3    RC1    RC2    RC4   
## Red.Meat                         0.925              
## White.Meat                0.941  0.142              
## Egg                       0.594  0.655  0.128       
## Milk                      0.197  0.684  0.208 -0.513
## Fish                     -0.214         0.921       
## Cereals                  -0.418 -0.557 -0.614       
## Starchy.Foods             0.518         0.711       
## Pulses.Nuts.and.Oilseeds -0.688 -0.337 -0.277  0.413
## Fruits.and.Vegetables                   0.144  0.951
##
##                  RC3   RC1   RC2   RC4
## SS loadings    2.249 2.207 1.895 1.361
## Proportion Var 0.250 0.245 0.211 0.151
## Cumulative Var 0.250 0.495 0.706 0.857
```

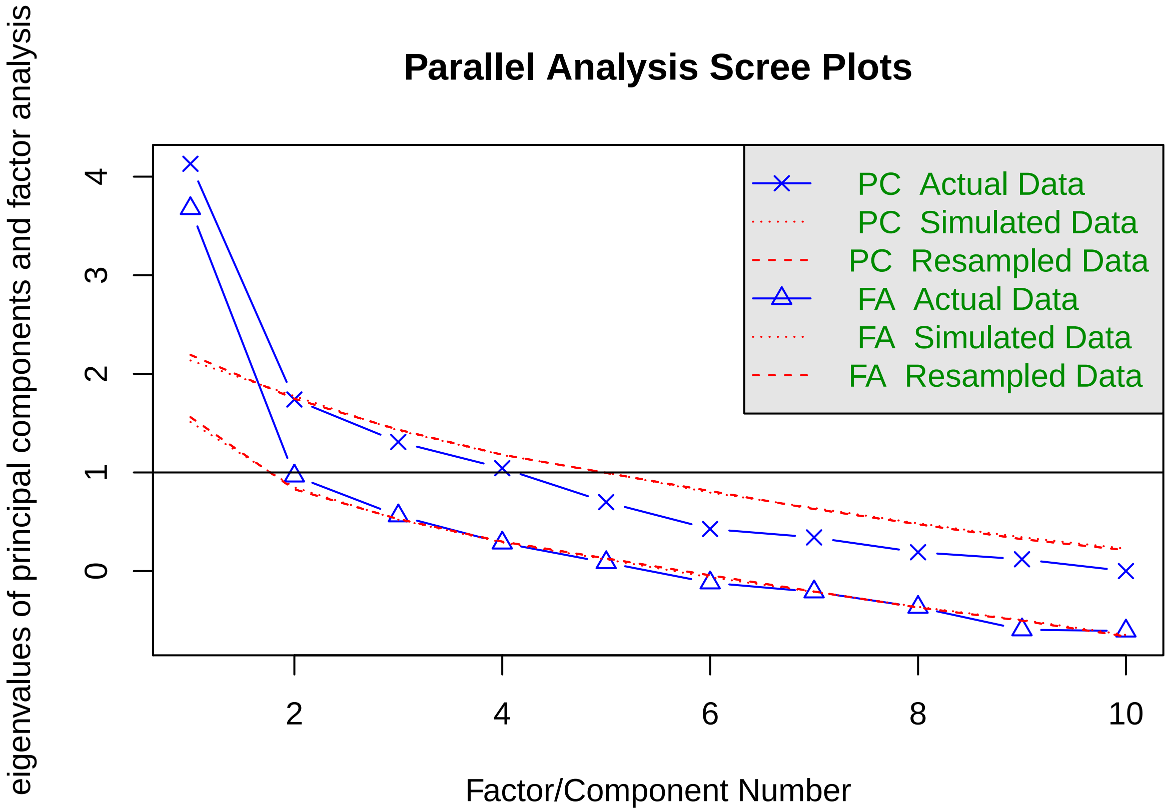
```r
# Communalities
pc$communality
```

```
##                  Red.Meat                White.Meat                       Egg
##                 0.8623000                 0.9136165                 0.8067014
##                      Milk                      Fish                   Cereals
##                 0.8123492                 0.9110157                 0.8666324
##              Starchy.Foods Pulses.Nuts.and.Oilseeds    Fruits.and.Vegetables
##                 0.7744128                 0.8343631                 0.9293892
```

```
#We can see that fish,white meat and fruits&vegetables account for most common variance among the count

fa.parallel(protein[-1]) # See factor recommendation
```

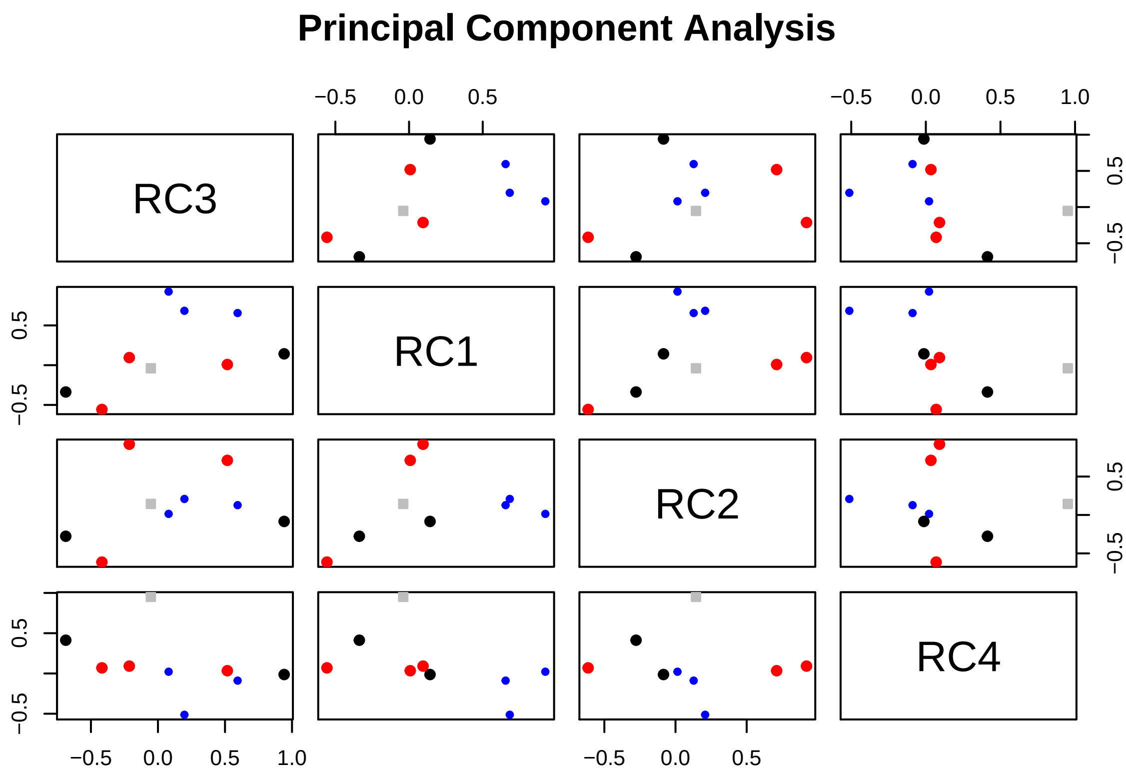
```
## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(R): Matrix was not positive definite, smoothing was
## done

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done

## 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 extraction method.

## In factor.scores, the correlation matrix is singular, an approximation is used

## Warning in cor.smooth(r): Matrix was not positive definite, smoothing was
## done

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =
## rotate, : A loading greater than abs(1) was detected. Examine the loadings
## carefully.

## 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 extraction method.

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

## 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 extraction method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =
## rotate, : A loading greater than abs(1) was detected. Examine the loadings
## carefully.

## 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 extraction method.

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

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =
## rotate, : A loading greater than abs(1) was detected. Examine the loadings
## carefully.
```

```
## 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 extraction method.

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

## 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 extraction method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =
## rotate, : A loading greater than abs(1) was detected. Examine the loadings
## carefully.

## 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 extraction method.

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

## 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 extraction method.
```




```
## Parallel analysis suggests that the number of factors =  1  and the number of components =  1
#From the above plot of "PC Actual Data" we can see that after 4 factors the eigen value crosses at 1 a
```

```r
fa.plot(pc) # See Correlations within Factors
```




```r
fa.diagram(pc) # Visualizing the relationship
```

Components Analysis