ST405 Mini Project

S/18/841

2024-04-05

Introduction

The National Health and Nutrition Examination Survey (NHANES) conducted by the Centers for Disease Control and Prevention (CDC) gathers extensive health and nutritional information from a diverse U.S. population. In this study, we narrow our focus to predicting respondents' age by extracting a subset of features from the NHANES dataset. These features include physiological measurements, lifestyle choices, and biochemical markers, hypothesized to correlate strongly with age.

Methodology

Dataset Description: The dataset used in this study is derived from NHANES 2013-2014, comprising various health and nutritional data from the U.S. population. Key variables include "SEQN" (Respondent Sequence Number), "age_group" (Respondent's Age Group), "RIDAGEYR" (Respondent's Age), "RIAGENDR" (Respondent's Gender), "PAQ605" (Engagement in Physical Activities), "BMXBMI" (Body Mass Index), "LBXGLU" (Blood Glucose after Fasting), "DIQ010" (Diabetes Status), "LBXGLT" (Oral Glucose Tolerance Test), and "LBXIN" (Blood Insulin Levels).

Statistical Methods Employed: This analysis employs Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA). EFA uncovers underlying patterns and correlations among variables, while CFA validates the factor structure and assesses model fit. Descriptive statistics such as means, standard deviations, and correlations are computed to summarize the dataset and examine relationships between variables.

```r
#Libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     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 errors
```

```r
library(ggplot2)
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 4.3.3
```

```
## corrplot 0.92 loaded
```

```r
library(ggcorrplot)
```

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

```r
library(GPArotation)
```

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

```
##
## Attaching package: 'GPArotation'
##
## The following objects are masked from 'package:psych':
##
##     equamax, varimin
```

```r
library(nFactors)
```

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

```
## Loading required package: lattice
##
## Attaching package: 'nFactors'
##
## The following object is masked from 'package:lattice':
##
##     parallel
```

```r
library(factoextra)
```

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

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

```r
library(psych)
library(lavaan)
```

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

## This is lavaan 0.6-17
## lavaan is FREE software! Please report any bugs.
##
## Attaching package: 'lavaan'
##
## The following object is masked from 'package:psych':
##
##     cor2cov
```

```r
###Data Loading and Inspection
nhanes_data <- read.csv("../Data/NHANES_age_prediction.csv")
```

```r
# View the structure of the dataset
str(nhanes_data)
```

```
## 'data.frame':    2278 obs. of  10 variables:
##  $ SEQN     : num  73564 73568 73576 73577 73580 ...
##  $ age_group: chr  "Adult" "Adult" "Adult" "Adult" ...
##  $ RIDAGEYR : num  61 26 16 32 38 50 14 57 75 43 ...
##  $ RIAGENDR : num  2 2 1 1 2 1 1 2 1 1 ...
##  $ PAQ605   : num  2 2 2 2 1 2 2 2 2 1 ...
##  $ BMXBMI   : num  35.7 20.3 23.2 28.9 35.9 23.6 38.7 38.3 38.9 28.9 ...
##  $ LBXGLU   : num  110 89 89 104 103 110 94 107 89 90 ...
##  $ DIQ010   : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ LBXGLT   : num  150 80 68 84 81 100 202 164 113 95 ...
##  $ LBXIN    : num  14.91 3.85 6.14 16.15 10.92 ...
```

```r
# View summary statistics of the dataset
summary(nhanes_data)
```

```
##       SEQN        age_group            RIDAGEYR       RIAGENDR
##  Min.   :73564   Length:2278        Min.   :12.0   Min.   :1.000
##  1st Qu.:76172   Class :character   1st Qu.:24.0   1st Qu.:1.000
##  Median :78749   Mode  :character   Median :41.0   Median :2.000
##  Mean   :78692                      Mean   :41.8   Mean   :1.511
##  3rd Qu.:81214                      3rd Qu.:58.0   3rd Qu.:2.000
##  Max.   :83727                      Max.   :80.0   Max.   :2.000
##      PAQ605          BMXBMI          LBXGLU           DIQ010          LBXGLT
##  Min.   :1.000   Min.   :14.50   Min.   : 63.00   Min.   :1.000   Min.   : 40
##  1st Qu.:2.000   1st Qu.:22.80   1st Qu.: 91.00   1st Qu.:2.000   1st Qu.: 87
##  Median :2.000   Median :26.80   Median : 97.00   Median :2.000   Median :105
##  Mean   :1.823   Mean   :27.96   Mean   : 99.55   Mean   :2.016   Mean   :115
##  3rd Qu.:2.000   3rd Qu.:31.20   3rd Qu.:104.00   3rd Qu.:2.000   3rd Qu.:130
##  Max.   :7.000   Max.   :70.10   Max.   :405.00   Max.   :3.000   Max.   :604
##      LBXIN
##  Min.   :  0.14
##  1st Qu.:  5.86
##  Median :  9.04
##  Mean   : 11.83
##  3rd Qu.: 14.44
##  Max.   :102.29
```

```r
# View the first few rows of the dataset
head(nhanes_data)
```

```
##    SEQN age_group RIDAGEYR RIAGENDR PAQ605 BMXBMI LBXGLU DIQ010 LBXGLT LBXIN
## 1 73564     Adult       61        2      2   35.7    110      2    150 14.91
## 2 73568     Adult       26        2      2   20.3     89      2     80  3.85
## 3 73576     Adult       16        1      2   23.2     89      2     68  6.14
## 4 73577     Adult       32        1      2   28.9    104      2     84 16.15
## 5 73580     Adult       38        2      1   35.9    103      2     81 10.92
## 6 73581     Adult       50        1      2   23.6    110      2    100  6.08
```

```r
#Dimensions of the dataset
dim(nhanes_data)
```

```
## [1] 2278   10
```

```r
# Check for missing values
colSums(is.na(nhanes_data))
```

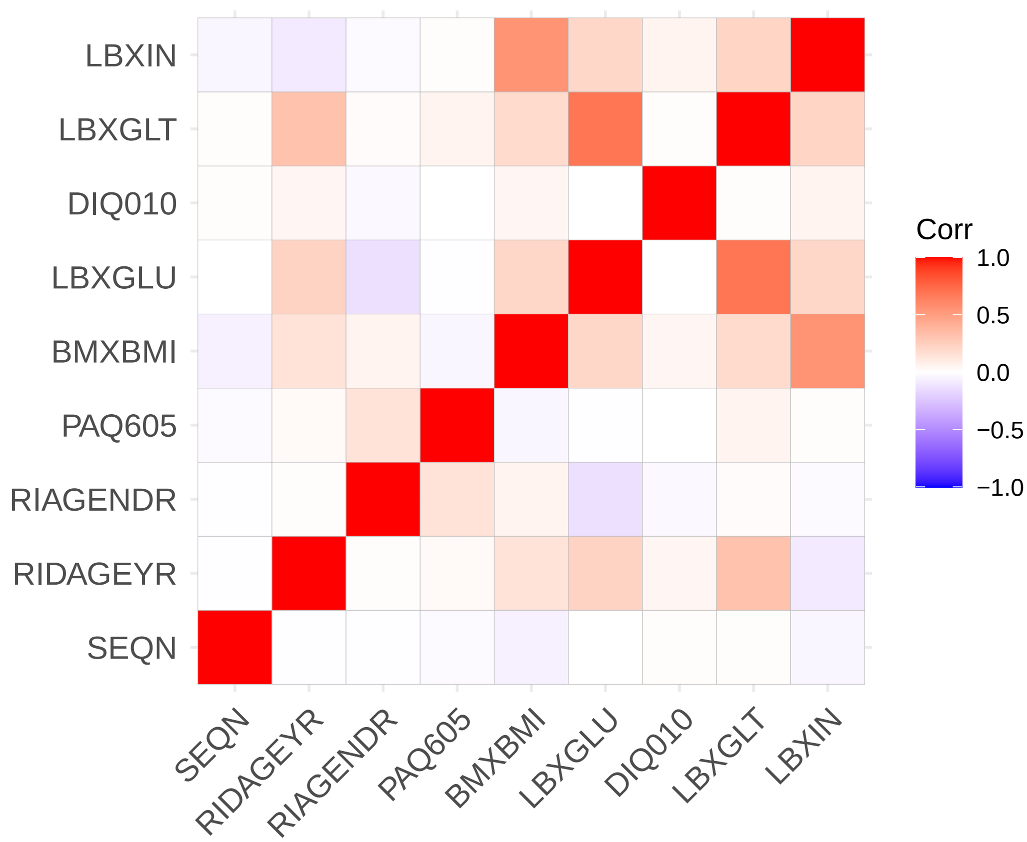
```
##      SEQN age_group  RIDAGEYR  RIAGENDR    PAQ605    BMXBMI    LBXGLU    DIQ010
##         0         0         0         0         0         0         0         0
##    LBXGLT     LBXIN
##         0         0
```

```r
###Data Preprocessing
# Select only numeric variables from the NHANES dataset
numerical_nhanes_data <- nhanes_data[, sapply(nhanes_data, is.numeric)]

# Scale the numeric variables
normalized_data <- scale(numerical_nhanes_data)

# View the first few rows of the normalized data
head(normalized_data)
```

```
##           SEQN   RIDAGEYR   RIAGENDR     PAQ605     BMXBMI     LBXGLU
## [1,] -1.755294  0.9527694  0.9772131  0.4445743  1.0684040  0.5839564
## [2,] -1.753924 -0.7836767  0.9772131  0.4445743 -1.0560381 -0.5898946
## [3,] -1.751186 -1.2798041 -1.0228690  0.4445743 -0.6559808 -0.5898946
## [4,] -1.750844 -0.4860002 -1.0228690  0.4445743  0.1303387  0.2485704
## [5,] -1.749817 -0.1883237  0.9772131 -2.0622083  1.0959942  0.1926727
## [6,] -1.749474  0.4070292 -1.0228690  0.4445743 -0.6008005  0.5839564
##           DIQ010     LBXGLT       LBXIN
## [1,] -0.08753347  0.7441596  0.31641794
## [2,] -0.08753347 -0.7432641 -0.82158128
## [3,] -0.08753347 -0.9982510 -0.58595576
## [4,] -0.08753347 -0.6582685  0.44400555
## [5,] -0.08753347 -0.7220152 -0.09412608
## [6,] -0.08753347 -0.3182859 -0.59212936
```

```r
#Compute the correlation matrix
cor_matrix<-cor(normalized_data)
```

```r
#Visualize the correlation matrix
ggcorrplot(cor_matrix)
```




```r
# Compute eigenvalues
eigen_values<- eigen(cor_matrix)
eigen_values
```

```
## eigen() decomposition
## $values
## [1] 2.1418878 1.3348263 1.1767796 1.0160941 0.9879332 0.9034897 0.7669540
## [8] 0.3898257 0.2822095
##
## $vectors
##              [,1]        [,2]        [,3]        [,4]         [,5]        [,6]
##  [1,]  0.03646881  0.15073249  0.13750447 -0.38628977  0.881753293 -0.11384612
##  [2,] -0.28593232  0.41656848 -0.13248116 -0.20694686 -0.264312936 -0.48300427
##  [3,]  0.03376956 -0.12259966 -0.70518475 -0.06876054  0.115613452 -0.44014485
##  [4,] -0.01553227  0.06523268 -0.65792079 -0.09663964  0.070342547  0.62229942
##  [5,] -0.41684427 -0.50287947 -0.03407413 -0.04021793  0.001373012 -0.26987367
##  [6,] -0.53783408  0.28434144  0.11179158  0.12922155  0.064669696  0.19742268
##  [7,] -0.05282271 -0.08903474  0.11254312 -0.87865877 -0.325365980  0.18019687
```

```
## [8,] -0.54741455  0.31135426 -0.07198483  0.05664455  0.070340646  0.09564474
## [9,] -0.38736744 -0.58809690  0.04347161  0.01790093  0.139069676  0.13905225
##             [,7]        [,8]        [,9]
## [1,]  0.12997570  0.02008895 -0.00868437
## [2,]  0.52436617 -0.27986519 -0.16702184
## [3,] -0.50091859 -0.03013175 -0.15130786
## [4,]  0.39005911  0.09378322 -0.01258607
## [5,]  0.28949716  0.61239269  0.19724977
## [6,] -0.26706440  0.29822754 -0.63029195
## [7,] -0.25398540  0.04132088 -0.00504460
## [8,] -0.28540781 -0.16189105  0.68792686
## [9,]  0.06310858 -0.64783339 -0.19882551
```

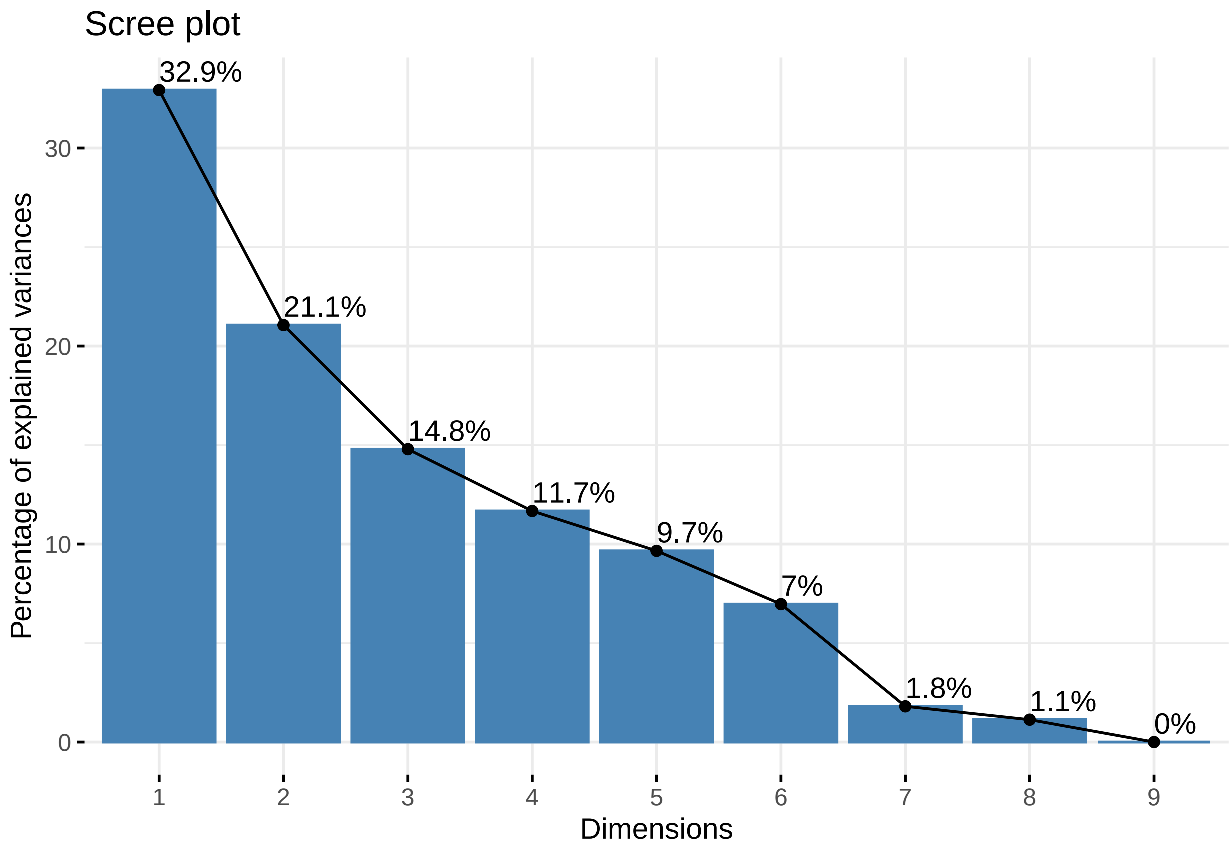
```
# Principal Component Analysis (PCA)
PCA <- princomp(cor_matrix)
PCA
```

```
## Call:
## princomp(x = cor_matrix)
##
## Standard deviations:
##       Comp.1       Comp.2       Comp.3       Comp.4       Comp.5       Comp.6
## 5.562252e-01 4.448188e-01 3.728026e-01 3.311339e-01 3.012191e-01 2.558409e-01
##       Comp.7       Comp.8       Comp.9
## 1.302323e-01 1.030893e-01 3.155841e-09
##
##  9  variables and  9 observations.
```

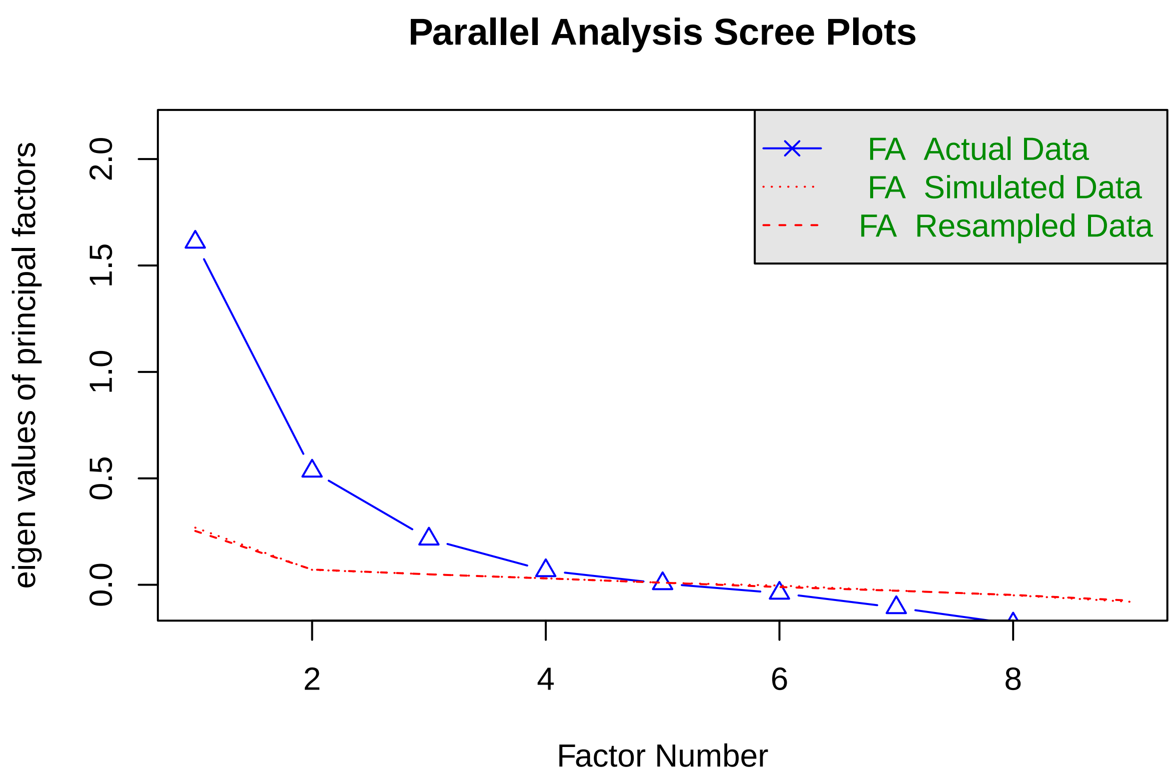
```
summary(PCA)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
## Standard deviation     0.5562252 0.4448188 0.3728026 0.3311339 0.30121906
## Proportion of Variance 0.3292547 0.2105702 0.1479069 0.1166911 0.09655962
## Cumulative Proportion  0.3292547 0.5398248 0.6877318 0.8044229 0.90098254
##                            Comp.6     Comp.7     Comp.8       Comp.9
## Standard deviation     0.25584094 0.13023230 0.10308935 3.155841e-09
## Proportion of Variance 0.06965795 0.01804962 0.01130989 1.059890e-17
## Cumulative Proportion  0.97064049 0.98869011 1.00000000 1.000000e+00
```

```
# Visualize the eigenvalues
fviz_eig(PCA,addlabels=TRUE)
```




```r
# Perform parallel analysis for factor extraction
fa.parallel(normalized_data, fm = "pa", fa = "fa")
```




```
## Parallel analysis suggests that the number of factors =  4  and the number of components =  NA
```

```r
# Compute covariance matrix
covariance_matrix <- cov(normalized_data)
covariance_matrix
```

```
##                  SEQN     RIDAGEYR     RIAGENDR       PAQ605      BMXBMI
## SEQN      1.000000000 -0.008805540 -0.012962358 -0.019700576 -0.06134269
## RIDAGEYR -0.008805540  1.000000000  0.006397727  0.025973321  0.14716313
## RIAGENDR -0.012962358  0.006397727  1.000000000  0.151076048  0.06387261
## PAQ605   -0.019700576  0.025973321  0.151076048  1.000000000 -0.04293533
## BMXBMI   -0.061342686  0.147163129  0.063872606 -0.042935333  1.00000000
## LBXGLU   -0.004146668  0.229623521 -0.132342297 -0.007848679  0.20833006
## DIQ010    0.014102187  0.049970042 -0.032768864 -0.002599301  0.04713345
## LBXGLT    0.006035900  0.318044363  0.017405743  0.060412732  0.19337459
## LBXIN    -0.040028480 -0.091879158 -0.016660137  0.010011012  0.55271719
##                LBXGLU       DIQ010      LBXGLT       LBXIN
## SEQN     -0.004146668  0.014102187 0.006035900 -0.04002848
## RIDAGEYR  0.229623521  0.049970042 0.318044363 -0.09187916
## RIAGENDR -0.132342297 -0.032768864 0.017405743 -0.01666014
## PAQ605   -0.007848679 -0.002599301 0.060412732  0.01001101
## BMXBMI    0.208330061  0.047133447 0.193374586  0.55271719
## LBXGLU    1.000000000 -0.004427431 0.685579317  0.21191124
## DIQ010   -0.004427431  1.000000000 0.009795877  0.05898570
## LBXGLT    0.685579317  0.009795877 1.000000000  0.21727159
## LBXIN     0.211911242  0.058985703 0.217271586  1.00000000
```

```r
# Kaiser-Meyer-Olkin (KMO) Test
KMO(r=nhanes_data[,-c(2)])
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = nhanes_data[, -c(2)])
## Overall MSA =  0.53
## MSA for each item =
##      SEQN RIDAGEYR RIAGENDR   PAQ605   BMXBMI   LBXGLU   DIQ010   LBXGLT
##      0.63     0.50     0.32     0.45     0.52     0.57     0.52     0.55
##    LBXIN
##      0.50
```

```r
cor_matrix <- cor_matrix[,KMO(cor_matrix)$MSAi>0.5]
round(KMO(cor_matrix)$MSA,2)
```

```
## [1] 0.54
```

```r
# Perform Bartlett's test of sphericity
cortest.bartlett(normalized_data)
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 3084.423
##
## $p.value
## [1] 0
##
## $df
## [1] 36
```

```r
# Compute the determinant of the covariance matrix
det(covariance_matrix)
```

```
## [1] 0.2574618
```

```r
# Compute proportion of variance explained by each principal component
Pop_var_exp <- eigen_values$values/sum(eigen_values$values)*100
round(Pop_var_exp,3)
```

```
## [1] 23.799 14.831 13.075 11.290 10.977 10.039  8.522  4.331  3.136
```

```r
# Total variance explained by all principal components
sum(eigen_values$values)
```

```
## [1] 9
```

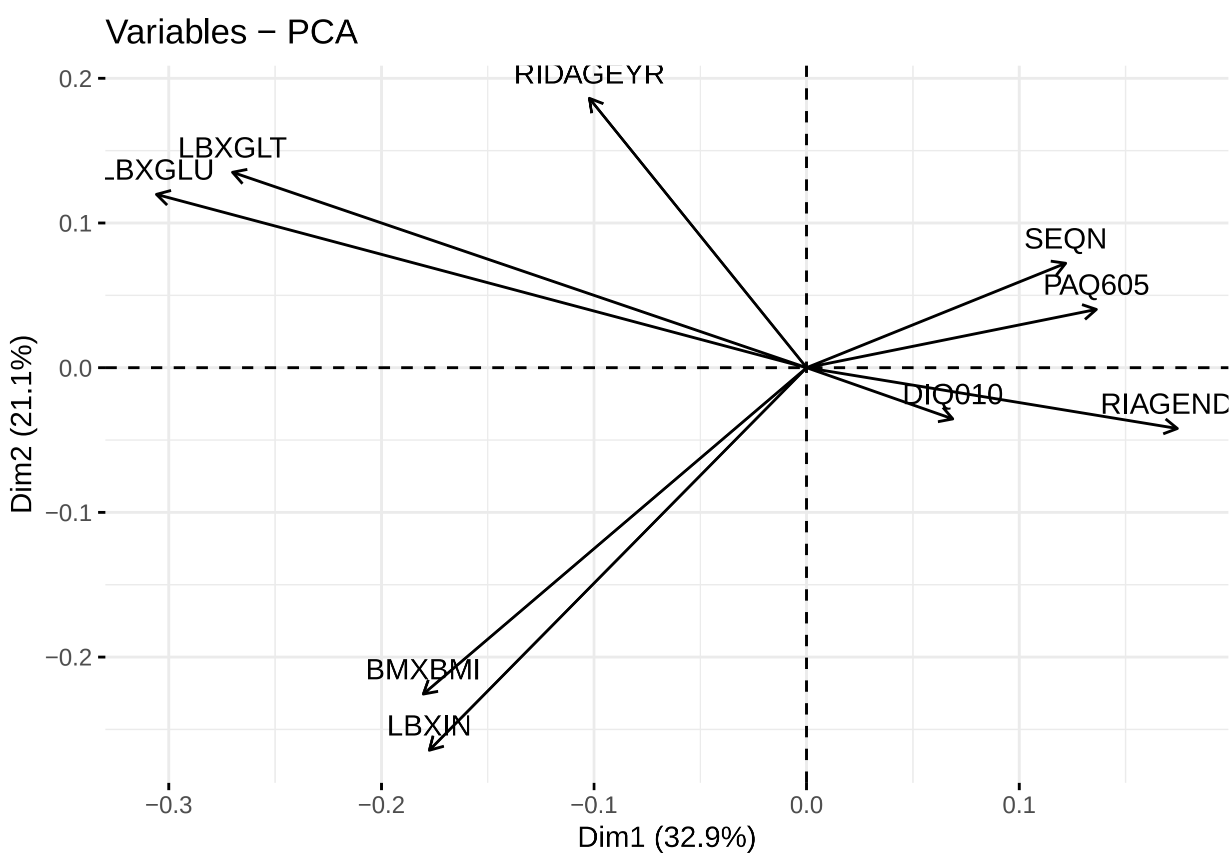
```r
# Sum of proportion of variance explained
sum(Pop_var_exp)
```

```
## [1] 100
```

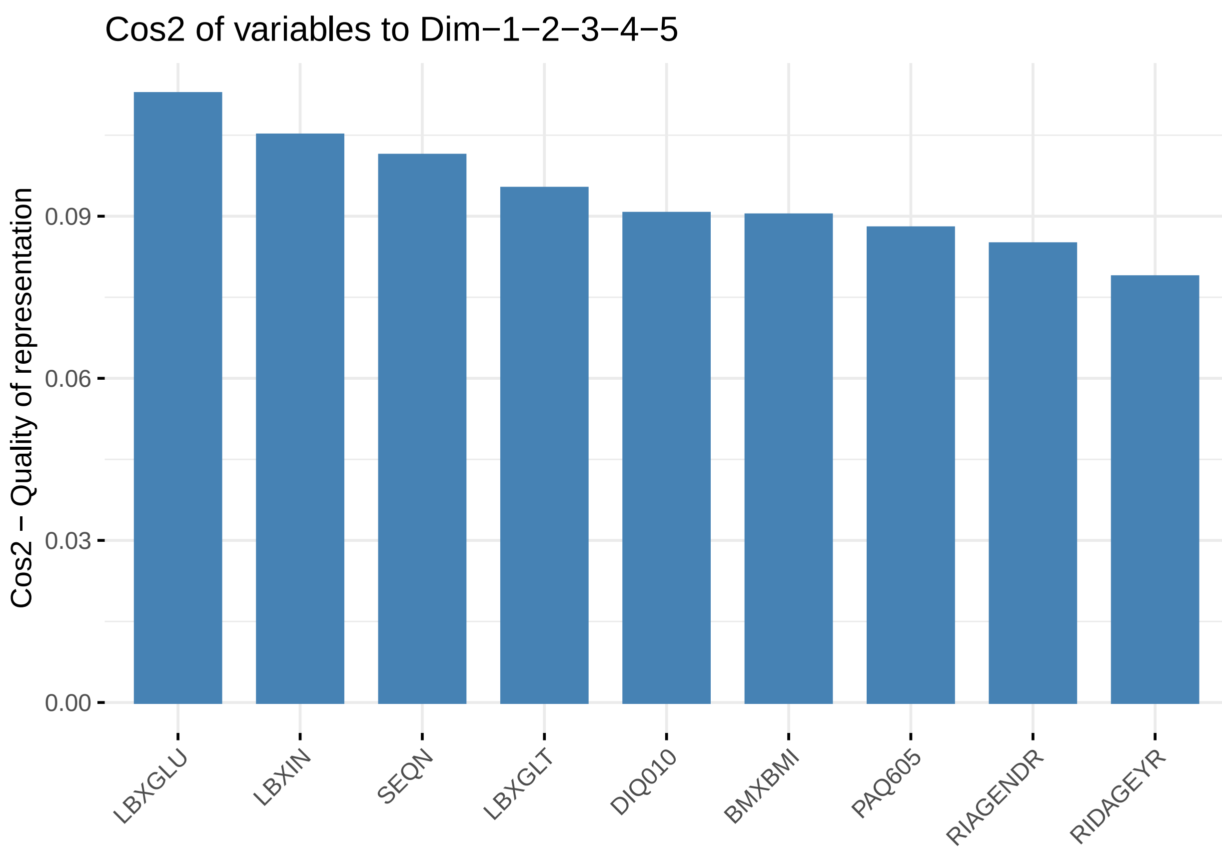
```r
# Factor Loadings using PCA
PCA$loadings[,1:5]
```

```
##              Comp.1      Comp.2       Comp.3      Comp.4      Comp.5
## SEQN      0.2190399  0.16226286  0.510407355  0.63629338  0.08430368
## RIDAGEYR -0.1837077  0.41854584 -0.067064946 -0.33372524  0.47956831
## RIAGENDR  0.3134556 -0.09449175 -0.509188553  0.04348126  0.42649093
## PAQ605    0.2449615  0.09052242 -0.474162993 -0.01117905 -0.63348046
## BMXBMI   -0.3241362 -0.50705491 -0.054899611 -0.02398356  0.26552462
## LBXGLU   -0.5497668  0.26924425 -0.019558318  0.10951391 -0.19625453
## DIQ010    0.1235834 -0.07958088  0.478670383 -0.66895238 -0.20065759
## LBXGLT   -0.4854211  0.30354996 -0.135388369  0.07378786 -0.09980103
## LBXIN    -0.3190583 -0.59438565  0.005590889  0.12740548 -0.14308124
```

```r
# Visualize the variables in PCA
fviz_pca_var(PCA, col.var = "black")
```



```r
# Visualize the squared cosines of variables in PCA
fviz_cos2(PCA, choice = "var", axes = 1:5)
```




```r
# Factor Analysis using factanal
numerical_nhanes_data.fa<-factanal(numerical_nhanes_data,factors = 3)
numerical_nhanes_data.fa
```

```
##
## Call:
## factanal(x = numerical_nhanes_data, factors = 3)
##
## Uniquenesses:
##      SEQN RIDAGEYR RIAGENDR   PAQ605   BMXBMI   LBXGLU   DIQ010   LBXGLT
##     0.998    0.850    0.005    0.974    0.678    0.425    0.995    0.151
##     LBXIN
##     0.005
##
## Loadings:
##          Factor1 Factor2 Factor3
## SEQN
## RIDAGEYR  0.366  -0.117
## RIAGENDR                  0.993
## PAQ605                    0.156
## BMXBMI    0.134   0.547
## LBXGLU    0.739   0.158
```

```
## DIQ010
## LBXGLT    0.903   0.155   0.101
## LBXIN             0.995
##
##                Factor1 Factor2 Factor3
## SS loadings      1.528   1.357   1.033
## Proportion Var   0.170   0.151   0.115
## Cumulative Var   0.170   0.321   0.435
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 160.19 on 12 degrees of freedom.
## The p-value is 4.79e-28
```

```r
# Compute squared loadings
apply(numerical_nhanes_data.fa$loadings^2,1,sum)
```

```
##        SEQN    RIDAGEYR    RIAGENDR      PAQ605      BMXBMI      LBXGLU
## 0.001958084 0.149571781 0.995000045 0.025756857 0.322403421 0.574617189
##      DIQ010      LBXGLT       LBXIN
## 0.004520238 0.849247707 0.995001183
```

```r
# Factor Analysis using fa function
nhanes_data_PC<- fa(covariance_matrix ,nfactors = 3,rotate = "varimax",n.obs
= 1000 ,covar = TRUE,fm = "pa")
```

```
## maximum iteration exceeded
```

```
## 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.
```

```r
nhanes_data_PC
```

```
## Factor Analysis using method =  pa
## Call: fa(r = covariance_matrix, nfactors = 3, n.obs = 1000, rotate = "varimax",
##     covar = TRUE, fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            PA1   PA2   PA3     h2     u2 com
## SEQN      0.00 -0.05 -0.03 0.0038  0.996 1.5
## RIDAGEYR  0.36 -0.04  0.06 0.1366  0.863 1.1
## RIAGENDR -0.05  0.04  0.68 0.4637  0.536 1.0
## PAQ605    0.04  0.00  0.22 0.0488  0.951 1.1
## BMXBMI    0.16  0.54  0.05 0.3159  0.684 1.2
## LBXGLU    0.74  0.17 -0.15 0.6036  0.396 1.2
## DIQ010    0.01  0.06 -0.03 0.0043  0.996 1.7
## LBXGLT    0.89  0.17  0.07 0.8233  0.177 1.1
## LBXIN     0.04  1.02 -0.07 1.0381 -0.038 1.0
##
##                        PA1  PA2  PA3
## SS loadings           1.50 1.39 0.55
## Proportion Var        0.17 0.15 0.06
## Cumulative Var        0.17 0.32 0.38
```

```
## Proportion Explained  0.44 0.40 0.16
## Cumulative Proportion 0.44 0.84 1.00
##
## Mean item complexity =  1.2
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  36  with the objective function =  1.36 with Chi Square =  1350.33
## df of  the model are 12  and the objective function was  0.08
##
## The root mean square of the residuals (RMSR) is  0.03
## The df corrected root mean square of the residuals is  0.05
##
## The harmonic n.obs is  1000 with the empirical chi square  57.25  with prob <  7.1e-08
## The total n.obs was  1000  with Likelihood Chi Square =  80.36  with prob <  3.5e-12
##
## Tucker Lewis Index of factoring reliability =  0.844
## RMSEA index =  0.075  and the 90 % confidence intervals are  0.06 0.092
## BIC =  -2.53
## Fit based upon off diagonal values = 0.98
```

```
# Rotate the factors
nhanes_data_PC_rotate <- fa(cor_matrix ,nfactors = 3,rotate =
"varimax",n.obs = 1000 ,cor = TRUE,fm = 'pa',max.iter = 1000)
```

```
## 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.
```

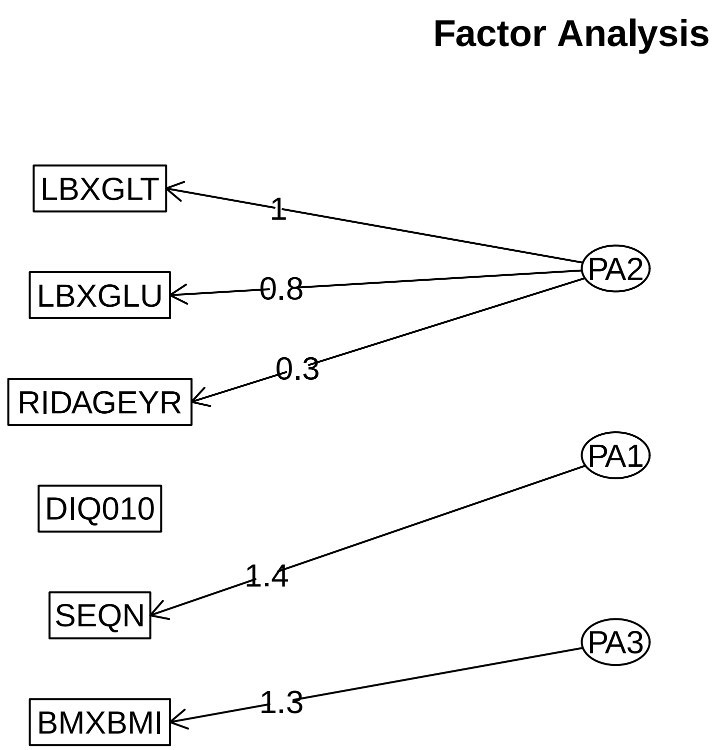
```
## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate, : An
## ultra-Heywood case was detected.  Examine the results carefully
```

```
nhanes_data_PC_rotate
```

```
## Factor Analysis using method =  pa
## Call: fa(r = cor_matrix, nfactors = 3, n.obs = 1000, rotate = "varimax",
##     max.iter = 1000, fm = "pa", cor = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##            PA2   PA1   PA3   h2     u2 com
## SEQN     -0.14  1.40 -0.15 2.01 -1.011 1.0
## RIDAGEYR  0.35 -0.12 -0.06 0.14  0.863 1.3
## BMXBMI    0.07 -0.13  1.29 1.67 -0.673 1.0
## LBXGLU    0.85 -0.08  0.09 0.73  0.271 1.0
## DIQ010   -0.30 -0.11 -0.11 0.11  0.889 1.6
## LBXGLT    1.04 -0.09  0.02 1.09 -0.091 1.0
##
##                        PA2  PA1  PA3
## SS loadings           2.03 2.03 1.70
## Proportion Var        0.34 0.34 0.28
## Cumulative Var        0.34 0.68 0.96
## Proportion Explained  0.35 0.35 0.29
## Cumulative Proportion 0.35 0.71 1.00
##
```

```
## Mean item complexity =  1.2
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  15  with the objective function =  2.25 with Chi Square =  11.61
## df of  the model are 0  and the objective function was  0.01
##
## The root mean square of the residuals (RMSR) is  0.01
## The df corrected root mean square of the residuals is  NA
##
## The harmonic n.obs is  9 with the empirical chi square  0.04  with prob <  NA
## The total n.obs was  9  with Likelihood Chi Square =  0.02  with prob <  NA
##
## Tucker Lewis Index of factoring reliability =  Inf
## Fit based upon off diagonal values = 1
```

```r
# Plot the factor diagram
fa.diagram(nhanes_data_PC_rotate)
```




```r
###Confirmatory Factor Analysis (CFA):
# Define the CFA model
variables <-
normalized_data[,c("RIDAGEYR","RIAGENDR","BMXBMI","LBXGLU","LBXGLT","LBXIN")]
#define the CFA model
model <- '
Factor1 =~ BMXBMI+LBXIN
```

```r
Factor2 =~ LBXGLT+ LBXGLU+ RIDAGEYR
Factor3=~RIAGENDR
'
# Fit the CFA model
fit <- cfa(model, data = variables)
# Assess model fit
summary(fit, fit.measures = TRUE)
```

```
## lavaan 0.6.17 ended normally after 27 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        14
##
##   Number of observations                          2278
##
## Model Test User Model:
##
##   Test statistic                               318.591
##   Degrees of freedom                                 7
##   P-value (Chi-square)                           0.000
##
## Model Test Baseline Model:
##
##   Test statistic                              2980.033
##   Degrees of freedom                                15
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.895
##   Tucker-Lewis Index (TLI)                       0.775
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -18060.330
##   Loglikelihood unrestricted model (H1)     -17901.035
##
##   Akaike (AIC)                               36148.660
##   Bayesian (BIC)                             36228.894
##   Sample-size adjusted Bayesian (SABIC)      36184.414
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.140
##   90 Percent confidence interval - lower         0.127
##   90 Percent confidence interval - upper         0.153
##   P-value H_0: RMSEA <= 0.050                    0.000
##   P-value H_0: RMSEA >= 0.080                    1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.051
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Factor1 =~
##     BMXBMI            1.000
##     LBXIN             0.968    0.087   11.102    0.000
##   Factor2 =~
##     LBXGLT            1.000
##     LBXGLU            0.849    0.042   20.366    0.000
##     RIDAGEYR          0.376    0.027   13.755    0.000
##   Factor3 =~
##     RIAGENDR          1.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Factor1 ~~
##     Factor2           0.216    0.021   10.250    0.000
##     Factor3           0.026    0.019    1.397    0.162
##   Factor2 ~~
##     Factor3          -0.025    0.020   -1.210    0.226
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .BMXBMI            0.429    0.052    8.312    0.000
##    .LBXIN             0.465    0.049    9.503    0.000
##    .LBXGLT            0.191    0.037    5.156    0.000
##    .LBXGLU            0.417    0.029   14.271    0.000
##    .RIDAGEYR          0.885    0.027   32.849    0.000
##    .RIAGENDR          0.000
##     Factor1           0.571    0.057   10.057    0.000
##     Factor2           0.808    0.047   17.258    0.000
##     Factor3           1.000    0.030   33.749    0.000
```

```r
# Standardized estimates (factor loadings)
parameterEstimates(fit, standardized = TRUE, ci = TRUE)
```

```
##         lhs op      rhs   est    se      z pvalue ci.lower ci.upper std.lv
## 1   Factor1 =~   BMXBMI 1.000 0.000     NA     NA    1.000    1.000  0.755
## 2   Factor1 =~    LBXIN 0.968 0.087 11.102  0.000    0.797    1.139  0.731
## 3   Factor2 =~   LBXGLT 1.000 0.000     NA     NA    1.000    1.000  0.899
## 4   Factor2 =~   LBXGLU 0.849 0.042 20.366  0.000    0.767    0.931  0.763
## 5   Factor2 =~ RIDAGEYR 0.376 0.027 13.755  0.000    0.322    0.429  0.338
## 6   Factor3 =~ RIAGENDR 1.000 0.000     NA     NA    1.000    1.000  1.000
## 7    BMXBMI ~~   BMXBMI 0.429 0.052  8.312  0.000    0.328    0.530  0.429
## 8     LBXIN ~~    LBXIN 0.465 0.049  9.503  0.000    0.369    0.560  0.465
## 9    LBXGLT ~~   LBXGLT 0.191 0.037  5.156  0.000    0.119    0.264  0.191
## 10   LBXGLU ~~   LBXGLU 0.417 0.029 14.271  0.000    0.360    0.475  0.417
## 11 RIDAGEYR ~~ RIDAGEYR 0.885 0.027 32.849  0.000    0.833    0.938  0.885
```

```
## 12 RIAGENDR ~~ RIAGENDR  0.000 0.000     NA     NA    0.000    0.000  0.000
## 13  Factor1 ~~  Factor1  0.571 0.057 10.057  0.000    0.459    0.682  1.000
## 14  Factor2 ~~  Factor2  0.808 0.047 17.258  0.000    0.716    0.900  1.000
## 15  Factor3 ~~  Factor3  1.000 0.030 33.749  0.000    0.942    1.058  1.000
## 16  Factor1 ~~  Factor2  0.216 0.021 10.250  0.000    0.174    0.257  0.318
## 17  Factor1 ~~  Factor3  0.026 0.019  1.397  0.162   -0.011    0.063  0.035
## 18  Factor2 ~~  Factor3 -0.025 0.020 -1.210  0.226   -0.065    0.015 -0.027
##    std.all std.nox
## 1    0.756   0.756
## 2    0.732   0.732
## 3    0.899   0.899
## 4    0.763   0.763
## 5    0.338   0.338
## 6    1.000   1.000
## 7    0.429   0.429
## 8    0.465   0.465
## 9    0.192   0.192
## 10   0.417   0.417
## 11   0.886   0.886
## 12   0.000   0.000
## 13   1.000   1.000
## 14   1.000   1.000
## 15   1.000   1.000
## 16   0.318   0.318
## 17   0.035   0.035
## 18  -0.027  -0.027
```

```r
#remotes::install_github("rstudio/htmltools")

#install.packages("html",dependencies = TRUE)
```