Assignment 3

Erik Pak

2024-08-06

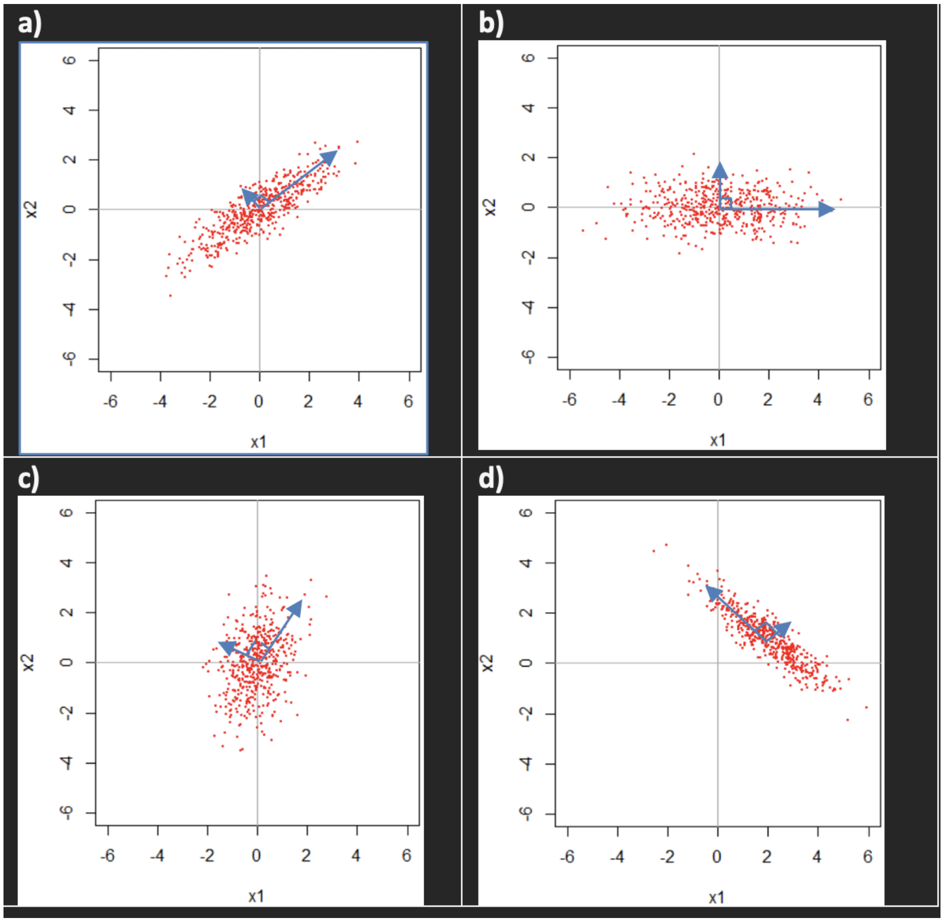
Load Libraries

```r
library(tidyverse)      # Data manipulation package
library(corrplot)       # correlation plot
library(psych)          # scree plot, pairs panels plot & corr.test
```

Problem 2 Eigenvector Diagram

approx. c(eigenvalue) = sqrt(a^2 + b^2)

a) sqrt(2^2 + 4^2) = 4.5
b) sqrt(4^2 + 0^2) = 4.0
c) sqrt(2^2 + 2^2) = 2.8
d) sqrt(2^2 + 3^2) = 3.6



Problem 3)

```r
# create matrices
M <- matrix(c(14, 2, 2, 11), ncol = 2)
N <- matrix(c(5,-1, 2,-2,10,-2,0,3,3), ncol = 3, byrow = T)
v <- matrix(c(-1,0,1), nrow = 3, byrow = T)

# print matrix M
cat("The 2x2 matrix:\n")
```

```
## The 2x2 matrix:
```

```r
print(M)
```

```
##      [,1] [,2]
## [1,]   14    2
## [2,]    2   11
```

```r
# a) Calculating Eigenvalues and eigenvectors of M
# same vectors but different direction
m <- eigen(M)
cat("Eigenvalues(s): \n", m$values, "\nEigenvector(s): \n")
```

```
## Eigenvalues(s):
##  15 10
## Eigenvector(s):
```

```r
print(round(m$vectors, digits = 1))
```

```
##      [,1] [,2]
## [1,] -0.9  0.4
## [2,] -0.4 -0.9
```

```r
# print matrix N
cat("The 3x3 matrix:\n")
```

```
## The 3x3 matrix:
```

```r
print(N)
```

```
##      [,1] [,2] [,3]
## [1,]    5   -1    2
## [2,]   -2   10   -2
## [3,]    0    3    3
```

```r
# b) Calculating Eigenvectors of N
n <- eigen(N)
cat("Eigenvector(s): \n")
```

```
## Eigenvector(s):
```

```r
print(round(n$vectors, digits = 1))
```

```
##      [,1] [,2] [,3]
## [1,]  0.0  0.6 -0.7
## [2,] -0.9  0.6  0.0
## [3,] -0.4  0.6  0.7
```

```r
# c) Eigenvalues of N
cat("Eigenvalues(s): \n", n$values)
```

```
## Eigenvalues(s):
##  9 6 3
```

Problem 4) Principal Component Analysis

Begin with the "census2.csv" datafile, which contains census data on various tracts in a district.

The fields in the data are:

1. Total Population (thousands)
2. Professional degree (percent)
3. Employed age over 16 (percent)
4. Government employed (percent)
5. Median home value (dollars)

```r
# set working directory
setwd("~/Downloads/Data")

# load dataset
census <- read.csv("Census2.csv")

# display
head(census)
```

```
##   Population Professional Employed Government MedianHomeVal
## 1       2.67         5.71    69.02       30.3        148000
## 2       2.25         4.37    72.98       43.3        144000
## 3       3.12        10.27    64.94       32.0        211000
## 4       5.14         7.44    71.29       24.5        185000
## 5       5.54         9.25    74.94       31.0        223000
## 6       5.04         4.84    53.61       48.2        160000
```

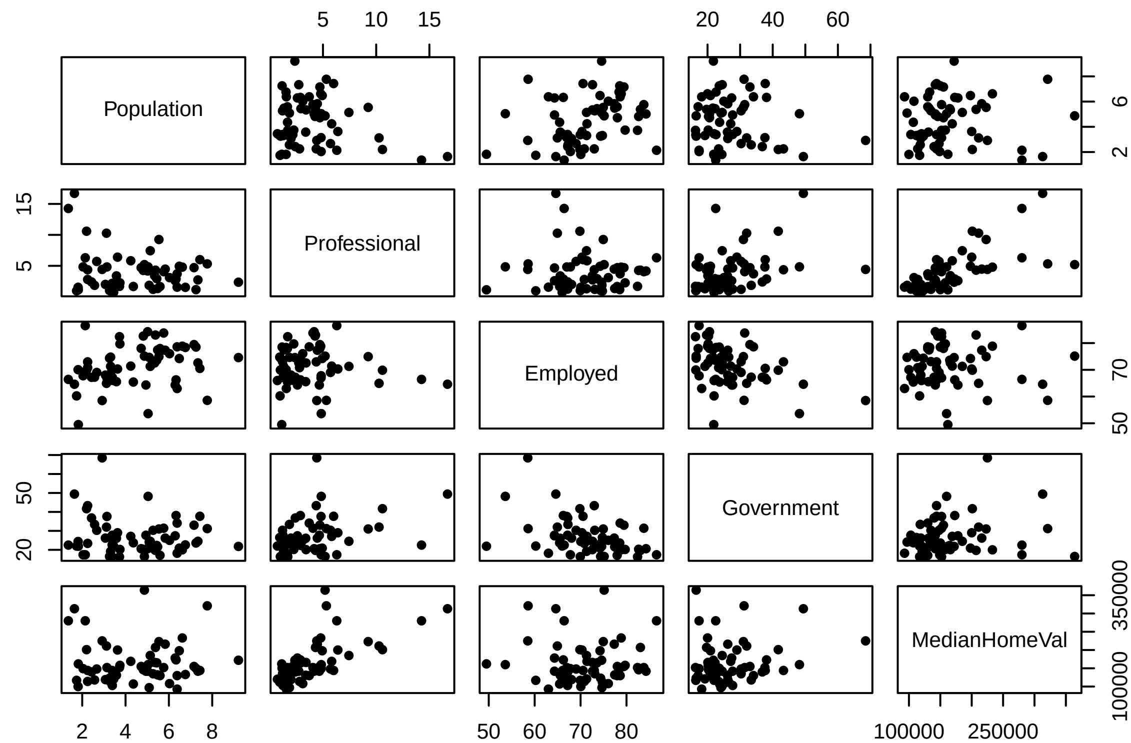
```r
# summary
summary(census)
```

```
##    Population     Professional       Employed       Government   
##  Min.   :1.360   Min.   : 0.720   Min.   :49.50   Min.   :16.30  
##  1st Qu.:3.120   1st Qu.: 1.670   1st Qu.:66.42   1st Qu.:20.60  
##  Median :4.720   Median : 3.380   Median :71.30   Median :24.40  
##  Mean   :4.469   Mean   : 3.962   Mean   :71.42   Mean   :26.91  
##  3rd Qu.:5.760   3rd Qu.: 4.830   3rd Qu.:77.33   3rd Qu.:31.00  
##  Max.   :9.210   Max.   :16.700   Max.   :86.54   Max.   :68.50  
##  MedianHomeVal  
##  Min.   : 93000  
##  1st Qu.:130000  
##  Median :149000  
##  Mean   :163557  
##  3rd Qu.:178000  
##  Max.   :364000
```

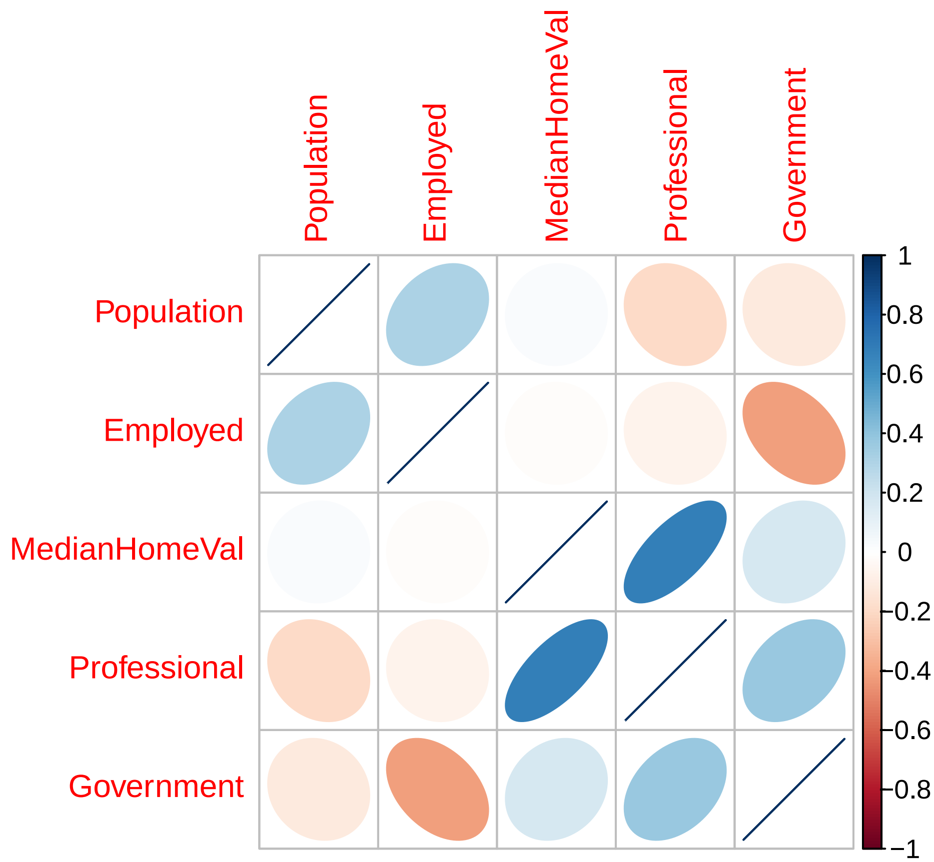
```r
# describe
describe(census)
```

```
##               vars  n      mean       sd   median   trimmed      mad      min
## Population       1 61      4.47     1.84 4.72e+00      4.41     2.16     1.36
## Professional     2 61      3.96     3.11 3.38e+00      3.42     2.30     0.72
## Employed         3 61     71.42     7.46 7.13e+01     71.64     7.50    49.50
## Government       4 61     26.91     9.44 2.44e+01     25.56     6.38    16.30
## MedianHomeVal    5 61 163557.38 56446.88 1.49e+05 154653.06 38547.60 93000.00
##                     max     range  skew kurtosis      se
## Population         9.21      7.85  0.24    -0.82    0.24
## Professional      16.70     15.98  1.94     4.56    0.40
## Employed          86.54     37.04 -0.38     0.19    0.95
## Government        68.50     52.20  1.85     4.72    1.21
## MedianHomeVal 364000.00 271000.00  1.56     2.34 7227.28
```

```r
# plot
plot(census, pch = 16)
```




```r
# corrplot
corrplot(cor(census), method = "ellipse",  order = "AOE")
```



Problem 4 a)

The output from a principal component analysis `PCA`, the first line, shows the standard deviations of the five variables used in the analysis, which measures the variation in the original data captured by each PC. The subsequent lines show the eigenvectors of each variable on the five principal components `PCs`.

The summary shows the cumulative proportion of the variance explained by each PC. For example, `PC1` explains 100% of the variance, so the cumulative proportion of variance explained by `PC1` is also 100%.

In conclusion, given that there is only one variable with a non-zero on PC1 `MedianHomeVal` this output confirms that PC1 is explaining all the variance in the analysis, explaining all of the variances in the data. The `MedianHomeVal` is over-showing the rest of the variables due to the large range or/and magnitude.

```r
# principal component analysis using the covariance
pCov <- prcomp(census)

# rotation
print(pCov)
```

```
## Standard deviations (1, .., p=5):
## [1] 56446.885008    10.206857     6.218887     2.246707     1.559823
##
## Rotation (n x k) = (5 x 5):
##                        PC1           PC2           PC3           PC4
## Population    -8.537905e-07 -4.108282e-02 -7.059713e-02 -4.826860e-01
## Professional  -3.775797e-05  7.080539e-02 -7.460074e-02  8.714029e-01
## Employed       1.367095e-06 -5.126328e-01 -8.542663e-01  1.524163e-02
```

```
## Government    -3.004471e-05  8.546967e-01 -5.095880e-01 -8.624903e-02
## MedianHomeVal -1.000000e+00 -2.901832e-05  1.701961e-05 -2.987813e-05
##                         PC5
## Population     8.719762e-01
## Professional   4.796648e-01
## Employed      -8.487872e-02
## Government    -4.873218e-02
## MedianHomeVal -1.750755e-05
```

```
# The first component contains 100% of the variance
summary(pCov)
```

```
## Importance of components:
##                          PC1   PC2   PC3   PC4  PC5
## Standard deviation     56447 10.21 6.219 2.247 1.56
## Proportion of Variance     1  0.00 0.000 0.000 0.00
## Cumulative Proportion      1  1.00 1.000 1.000 1.00
```

```
# PCs
round(pCov$rotation, 2)
```

```
##               PC1   PC2   PC3   PC4   PC5
## Population      0 -0.04 -0.07 -0.48  0.87
## Professional    0  0.07 -0.07  0.87  0.48
## Employed        0 -0.51 -0.85  0.02 -0.08
## Government      0  0.85 -0.51 -0.09 -0.05
## MedianHomeVal  -1  0.00  0.00  0.00  0.00
```

Problem 4b)

In summary, by dividing the `MedianHomeValue` field by 100K, rescale that variable and adjust the standard deviations for all five variables used in the analysis. Rescaling variable `MedianHomeValue` impacted PCA results because it did not dominate the variation captured by the first principal component and helped mitigate this issue and ensure that all variables contributed to the analysis.

For example, the first principal component is positive for `Employed`, and `Population` with contribution, and for `Professional`, `Government`, and `MedianHomeVal` are negative therefore acting in the opposite direction with contribution. This is a different interpretation between the 4a) because all the `PC1` on 4a) are zero except for `MedianHomeVal`, which is a massive difference because there are no contribution from other variables on PC1.

```
# update MedianHomeVal / 100K
census_100K <- census %>%
  select(names(census)) %>%
  mutate(MedianHomeVal = MedianHomeVal/100000)

# principal component analysis using the covariance
pCovS <- prcomp(census_100K)

# rotation summary
print(pCovS)
```

```
## Standard deviations (1, .., p=5):
```

```
## [1] 10.3448177  6.2985820  2.8932449  1.6934798  0.3933104
##
## Rotation (n x k) = (5 x 5):
##                       PC1         PC2         PC3         PC4          PC5
## Population     0.038887287 -0.07114494  0.18789258  0.97713524 -0.057699864
## Professional  -0.105321969 -0.12975236 -0.96099580  0.17135181 -0.138554092
## Employed       0.492363944 -0.86438807  0.04579737 -0.09104368  0.004966048
## Government    -0.863069865 -0.48033178  0.15318538 -0.02968577  0.006691800
## MedianHomeVal -0.009122262 -0.01474342 -0.12498114  0.08170118  0.988637470
```

```r
 # variance summary
summary(pCovS)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5
## Standard deviation     10.345 6.2986 2.89324 1.69348 0.39331
## Proportion of Variance  0.677 0.2510 0.05295 0.01814 0.00098
## Cumulative Proportion   0.677 0.9279 0.98088 0.99902 1.00000
```

```r
# PCs
round(pCovS$rotation, 2)
```

```
##                 PC1   PC2   PC3   PC4   PC5
## Population     0.04 -0.07  0.19  0.98 -0.06
## Professional  -0.11 -0.13 -0.96  0.17 -0.14
## Employed       0.49 -0.86  0.05 -0.09  0.00
## Government    -0.86 -0.48  0.15 -0.03  0.01
## MedianHomeVal -0.01 -0.01 -0.12  0.08  0.99
```

Problem 4 c)

There are three variables (`MedianHomeVal`, `Employed`, & `Government`) in particular need of scaling due to the range and its magnitude, but all the variables should be scaled because due to their different scales one some and ranges including magnitude which helps to ensure that each variable contributes equally to the analysis and prevents the model from being biased towards variables with larger values.

For example, the first principal component is positive `PC1` for `Population` & `Employee` are contribution in positive direction, but a negative `PC1` for `Professional`, `Government` , and `MedianHomeVal` are heading in an opposite direction. Also, this is a different interpretation between the 4a) because all the `PC1` on 4a) are zero except for `MedianHomeVal`, which is a massive difference because there are no contribution from other variables on PC1.

In conclusion, variables with the large range or magnitude do have huge impact on the analysis and transforming the data does spread out the variance which impacts on the interpretation from the analysis.

```r
# log transform
census_Log <- log(census)

# summary
summary(census_Log)
```

```
##    Population      Professional        Employed       Government
##   Min.   :0.3075   Min.   :-0.3285   Min.   :3.902   Min.   :2.791
```

```
##  1st Qu.:1.1378   1st Qu.: 0.5128   1st Qu.:4.196   1st Qu.:3.025
##  Median :1.5518   Median : 1.2179   Median :4.267   Median :3.195
##  Mean   :1.4023   Mean   : 1.1205   Mean   :4.263   Mean   :3.244
##  3rd Qu.:1.7509   3rd Qu.: 1.5748   3rd Qu.:4.348   3rd Qu.:3.434
##  Max.   :2.2203   Max.   : 2.8154   Max.   :4.461   Max.   :4.227
##  MedianHomeVal
##  Min.   :11.44
##  1st Qu.:11.78
##  Median :11.91
##  Mean   :11.96
##  3rd Qu.:12.09
##  Max.   :12.80
```

```r
# principal component analysis using the covariance
pCovL <- prcomp(census_Log)

# rotation summary
print(pCovL$rotation)
```

```
##                         PC1         PC2         PC3          PC4         PC5
## Population     5.270801e-02  0.98980155  0.10807219  0.001639567  0.07637061
## Professional  -9.375376e-01  0.05279537 -0.07088964 -0.329067793  0.07017788
## Employed       4.826652e-06  0.09195134 -0.15366600 -0.159207470 -0.97086801
## Government    -1.721226e-01 -0.07819159  0.93342758  0.235439702 -0.19375496
## MedianHomeVal -2.976893e-01  0.05419009 -0.29731260  0.900518088 -0.09548255
```

```r
 # variance summary
summary(pCovL)
```

```
## Importance of components:
##                           PC1    PC2     PC3    PC4     PC5
## Standard deviation     0.7694 0.4596 0.28392 0.1952 0.08661
## Proportion of Variance 0.6369 0.2273 0.08673 0.0410 0.00807
## Cumulative Proportion  0.6369 0.8642 0.95093 0.9919 1.00000
```

```r
# PCs
round(pCovL$rotation, 2)
```

```
##                 PC1   PC2   PC3   PC4   PC5
## Population     0.05  0.99  0.11  0.00  0.08
## Professional  -0.94  0.05 -0.07 -0.33  0.07
## Employed       0.00  0.09 -0.15 -0.16 -0.97
## Government    -0.17 -0.08  0.93  0.24 -0.19
## MedianHomeVal -0.30  0.05 -0.30  0.90 -0.10
```

Problem 4 d)

On b), we scaled down the magnitude of `MedianHomeVal`; on c), the entire dataset was log-transformed. Also, the PCA covariance matrix was used for both b) and c), but for d) PCA correlation matrix was used in PCA. Both correlation and covariance matrices are used in PCA to identify the principal components that capture the most variation in the data. The covariance matrix reflects both the variance and covariance between variables, while the correlation matrix only reflects the correlation between variables and is always standardized.

In the covariance matrix, PCs with the variance captured varied slightly, but in the correlation matrix, variance capture has a stark difference, but the meaning of the first component is the same since all three shows that `Professional`, `Government`, & `MedianHomeVal` going in a different direction as the `Population`, and `Employed`, but obviously eigenvectors are all different due to the variance explained by the first principal component.

```r
# principal component analysis using the covariance
pCor <- prcomp(census, scale = T)

# rotation
print(pCor)
```

```
## Standard deviations (1, .., p=5):
## [1] 1.4113534 1.1694129 0.9296006 0.7314787 0.4912604
##
## Rotation (n x k) = (5 x 5):
##                      PC1        PC2         PC3        PC4        PC5
## Population     0.2625829 -0.4629936  0.78390268  0.2169291 -0.2347882
## Professional  -0.5933541 -0.3256442 -0.16407255 -0.1446471 -0.7028828
## Employed       0.3256978 -0.6051419 -0.22487455 -0.6628689  0.1943206
## Government    -0.4792022  0.2524850  0.55070086 -0.5716730  0.2766497
## MedianHomeVal -0.4932213 -0.4996473 -0.06882436  0.4072024  0.5801162
```

```r
# summary
summary(pCor)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5
## Standard deviation     1.4114 1.1694 0.9296 0.7315 0.49126
## Proportion of Variance 0.3984 0.2735 0.1728 0.1070 0.04827
## Cumulative Proportion  0.3984 0.6719 0.8447 0.9517 1.00000
```

```r
# PCs
round(pCor$rotation, 2)
```

```
##                 PC1   PC2   PC3   PC4   PC5
## Population     0.26 -0.46  0.78  0.22 -0.23
## Professional  -0.59 -0.33 -0.16 -0.14 -0.70
## Employed       0.33 -0.61 -0.22 -0.66  0.19
## Government    -0.48  0.25  0.55 -0.57  0.28
## MedianHomeVal -0.49 -0.50 -0.07  0.41  0.58
```

Problem 4 e)

Testing the significance of the correlation coefficient at a 95% confidence level is to determine the statistical significance at a 95% confidence level and indicate that the correlation is not due to chance. Therefore, this exercise can aid in determining if there is possible multicollinearity among the variables, including variables that are uncorrelated with other variables.

It is a useful tool to assess multicollinearity to leverage this information in factor analysis or address it in multiple ways, like combining and transforming variables or utilizing regularization techniques in machine learning models.

```r
# probability correlation on sample size
PCorrTestC <- corr.test(census, adjust="none")

# vectorized probability
P <- PCorrTestC$p

# if True probability at a 95% confidence level
PTestC <- ifelse(P < 0.05, T, F)

# how many significant correlations there are for each variable.
colSums(PTestC) - 1  # We have to subtract 1 for the diagonal elements (self-correlation)
```

```
##    Population  Professional      Employed    Government MedianHomeVal 
##             1             2             2             2             1
```

Problem 4 f)

The interpretability of the covariance matrix can be difficult if the variables are on a different scale and unlike the correlation matrix, which is the same scale and can be directly interpreted as correlation coefficients. This makes it easier to identify which variables are most strongly related to each other and to interpret the principal components. In addition to being easier to interpret, the correlation matrix has other advantages over the covariance matrix. For example, the correlation matrix provides information about the direction and strength of the linear relationship between variables, while the covariance matrix only provides information about the direction of the relationship.

5) (Principal Component Analysis, 20 points): The data given in the file 'Employment.txt' is the percentage employed in different industries in Europe countries during 1979. Techniques such as Principal Component Analysis (PCA) can be used to examine which countries have similar employment patterns. There are 26 countries in the file and 10 variables as follows:

Problem 5

Variable Names:

1. Country: Name of country
2. Agr: Percentage employed in agriculture
3. Min: Percentage employed in mining
4. Man: Percentage employed in manufacturing
5. PS: Percentage employed in power supply industries
6. Con: Percentage employed in construction
7. SI: Percentage employed in service industries
8. Fin: Percentage employed in finance
9. SPS: Percentage employed in social and personal services
10. TC: Percentage employed in transport and communications

Problem 5 a)

The variables are all on the same scale, but their relative magnitudes varies. For example, `Agr` (Percentage employed in agriculture) has a significant variance than other variables. It will contribute more to the overall variability of the data and dominate the PCA as a result, dataset should be scaled.

The `Agr` (Percentage employed in agriculture) is highly correlated to six out of eight according to testing the significance of the correlation coefficient at a 95% confidence level. It is unlikely by chance, and according to the `corr.test`, no variable should be its own component, and it is clear that every variable has some correlation with each other per `corrplot`.

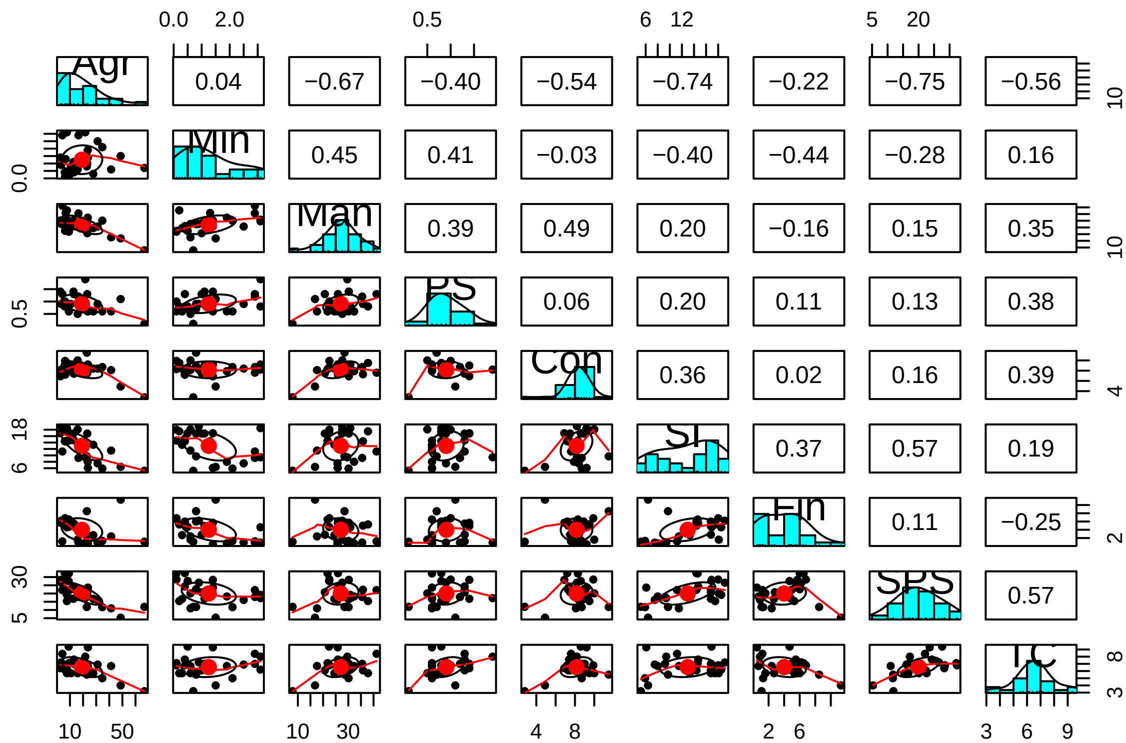
```r
# read tabular data
employ <- read.table("Employment.txt", header = T, sep = "\t", dec = ".")

# display
head(employ)
```

```
##      Country  Agr Min  Man  PS  Con   SI Fin  SPS  TC
## 1    Belgium  3.3 0.9 27.6 0.9  8.2 19.1 6.2 26.6 7.2
## 2    Denmark  9.2 0.1 21.8 0.6  8.3 14.6 6.5 32.2 7.1
## 3     France 10.8 0.8 27.5 0.9  8.9 16.8 6.0 22.6 5.7
## 4 W. Germany  6.7 1.3 35.8 0.9  7.3 14.4 5.0 22.3 6.1
## 5    Ireland 23.2 1.0 20.7 1.3  7.5 16.8 2.8 20.8 6.1
## 6      Italy 15.9 0.6 27.6 0.5 10.0 18.1 1.6 20.1 5.7
```

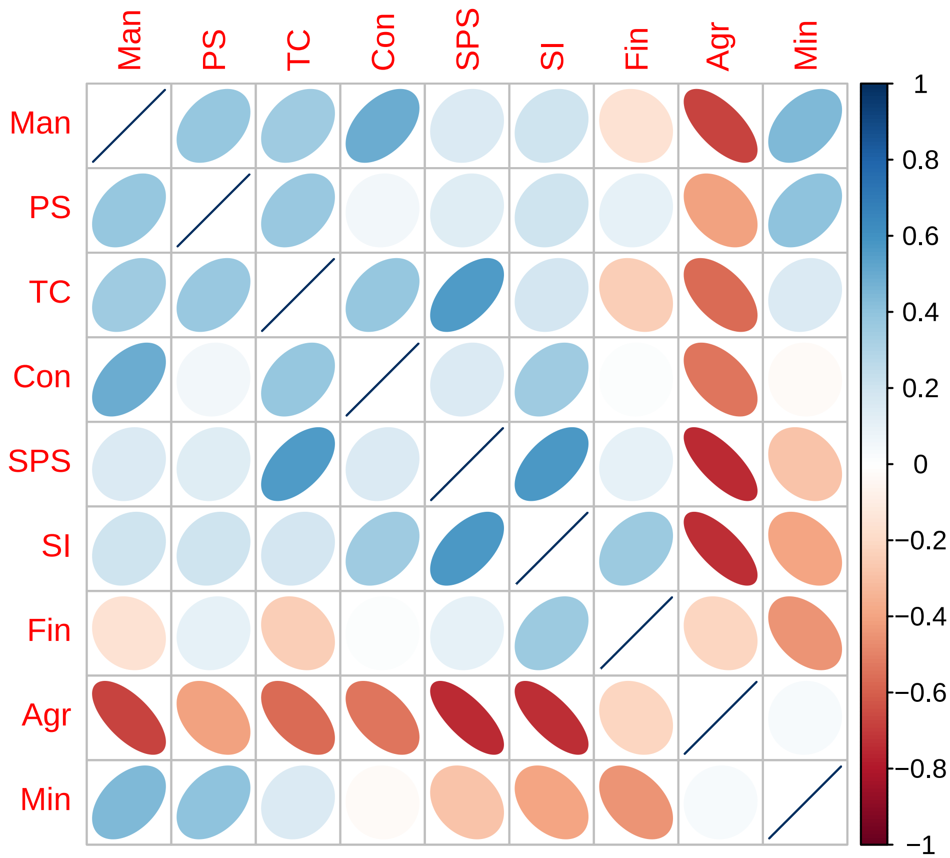
```r
# use country as rownames
rownames(employ) <- employ$Country
# remove country column
employ <- employ[,-1]

# summary
summary(employ)
```

```
##       Agr             Min             Man              PS        
##  Min.   : 2.70   Min.   :0.100   Min.   : 7.90   Min.   :0.1000  
##  1st Qu.: 7.70   1st Qu.:0.525   1st Qu.:23.00   1st Qu.:0.6000  
##  Median :14.45   Median :0.950   Median :27.55   Median :0.8500  
##  Mean   :19.13   Mean   :1.254   Mean   :27.01   Mean   :0.9077  
##  3rd Qu.:23.68   3rd Qu.:1.800   3rd Qu.:30.20   3rd Qu.:1.1750  
##  Max.   :66.80   Max.   :3.100   Max.   :41.20   Max.   :1.9000  
##       Con               SI             Fin              SPS       
##  Min.   : 2.800   Min.   : 5.20   Min.   : 0.500   Min.   : 5.30  
##  1st Qu.: 7.525   1st Qu.: 9.25   1st Qu.: 1.225   1st Qu.:16.25  
##  Median : 8.350   Median :14.40   Median : 4.650   Median :19.65  
##  Mean   : 8.165   Mean   :12.96   Mean   : 4.000   Mean   :20.02  
##  3rd Qu.: 8.975   3rd Qu.:16.88   3rd Qu.: 5.925   3rd Qu.:24.12  
##  Max.   :11.500   Max.   :19.10   Max.   :11.300   Max.   :32.40  
##        TC       
##  Min.   :3.200  
##  1st Qu.:5.700  
##  Median :6.700  
##  Mean   :6.546  
##  3rd Qu.:7.075  
##  Max.   :9.400
```

```r
# pairs panels plot
pairs.panels(employ)
```




```r
# corrplot
corrplot(cor(employ), method = "ellipse", order = "AOE")
```




```r
# determine variables should be their own components
PCorrTestEmp <- corr.test(employ, adjust="none")

# vectorized probability
PEmp <- PCorrTestEmp$p

# if True probability at a 95% confidence level
PEmpTest <- ifelse(PEmp < 0.05, T, F)

# how many significant correlations there are for each variable.
cat("\nProbability at a 95% Confidence Level: \n")
```

```
##
## Probability at a 95% Confidence Level:
```

```r
colSums(PEmpTest) - 1  # subtract 1 for the diagonal elements (self-correlation)
```

```
## Agr Min Man  PS Con  SI Fin SPS  TC
##   6   4   3   2   2   3   1   3   2
```

Problem 5 b)

The var = 1 plot show ambiguity since on the scree plot with v = 1, the PC4 is right on the variance equals one. However, the knee plot also could be a conflict since PC3 does appear to be the knee, and PC5 also could be the knee. The change in variance between PC3 & PC4 is relatively small, which adds additional ambiguity, but PC4 does not appear to be a knee. The variance after PC5 and beyond does level out, which adds even more ambiguity.

But in conclusion, examining both scree plots, the PC4 where var = 1 is on the borderline, and the line scree plots, the "knee" could be both PC3 or PC5. Since the var = 1 scree plot, PC3 is definite, and PC5, according to the var = 1, is not significant. Therefore, three PCs are considered with caution. Further exploration in the factor analysis is required to decide if additional or fewer PCs are required from the initial exploration.

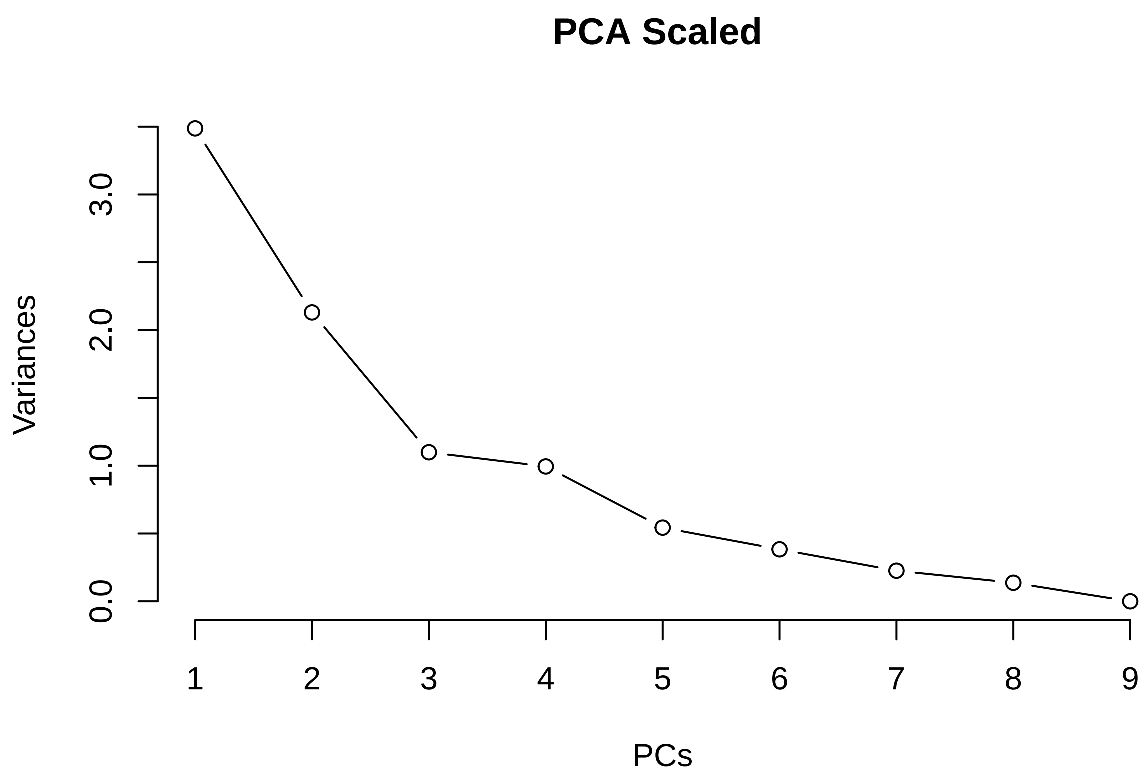
```r
# principal component analysis
pEmpScale <- prcomp(employ, scale = T)

# print
print(pEmpScale)
```

```
## Standard deviations (1, .., p=9):
## [1] 1.867391569 1.459511268 1.048311791 0.997237674 0.737033056 0.619215363
## [7] 0.475135828 0.369851221 0.006754636
##
## Rotation (n x k) = (9 x 9):
##              PC1         PC2         PC3         PC4        PC5        PC6
## Agr  0.523790989  0.05359389  0.04867439 -0.02879285  0.2127026  0.1533066
## Min  0.001323458  0.61780714 -0.20110021 -0.06408495 -0.1637431 -0.1005897
## Man -0.347495131  0.35505360 -0.15046308  0.34608821 -0.3849576 -0.2881523
## PS  -0.255716182  0.26109606 -0.56108325 -0.39330897  0.2951715  0.3572641
## Con -0.325179319  0.05128845  0.15332114  0.66832395  0.4715934  0.1303542
## SI  -0.378919663 -0.35017206 -0.11509551  0.05015651 -0.2835681  0.6148287
## Fin -0.074373583 -0.45369785 -0.58736130  0.05156652  0.2795682 -0.5255581
## SPS -0.387408806 -0.22152120  0.31190350 -0.41223019 -0.2203514 -0.2629097
## TC  -0.366822713  0.20259185  0.37510601 -0.31437188  0.5129356 -0.1239760
##             PC7          PC8        PC9
## Agr  0.02132116  0.007922069 0.80641788
## Min -0.72571894  0.088362816 0.04856307
## Man  0.47936298  0.125818308 0.36595728
## PS   0.25564699 -0.341228167 0.01938500
## Con -0.22069499 -0.355733906 0.08257219
## SI  -0.22943536  0.387536806 0.23829861
## Fin -0.18745525  0.174329338 0.14517064
## SPS -0.19130212 -0.506154178 0.35094226
## TC   0.06819331  0.544562381 0.07205520
```

```r
# summary
summary(pEmpScale)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5    PC6     PC7
## Standard deviation     1.8674 1.4595 1.0483 0.9972 0.73703 0.6192 0.47514
## Proportion of Variance 0.3875 0.2367 0.1221 0.1105 0.06036 0.0426 0.02508
## Cumulative Proportion  0.3875 0.6241 0.7462 0.8568 0.91711 0.9597 0.98480
##                           PC8      PC9
## Standard deviation     0.3699 0.006755
## Proportion of Variance 0.0152 0.000010
## Cumulative Proportion  1.0000 1.000000
```

```r
# line screeplot
screeplot(pEmpScale, npcs = 9, type = 'lines',
          main = "PCA Scaled ") + title(xlab = "PCs")
```

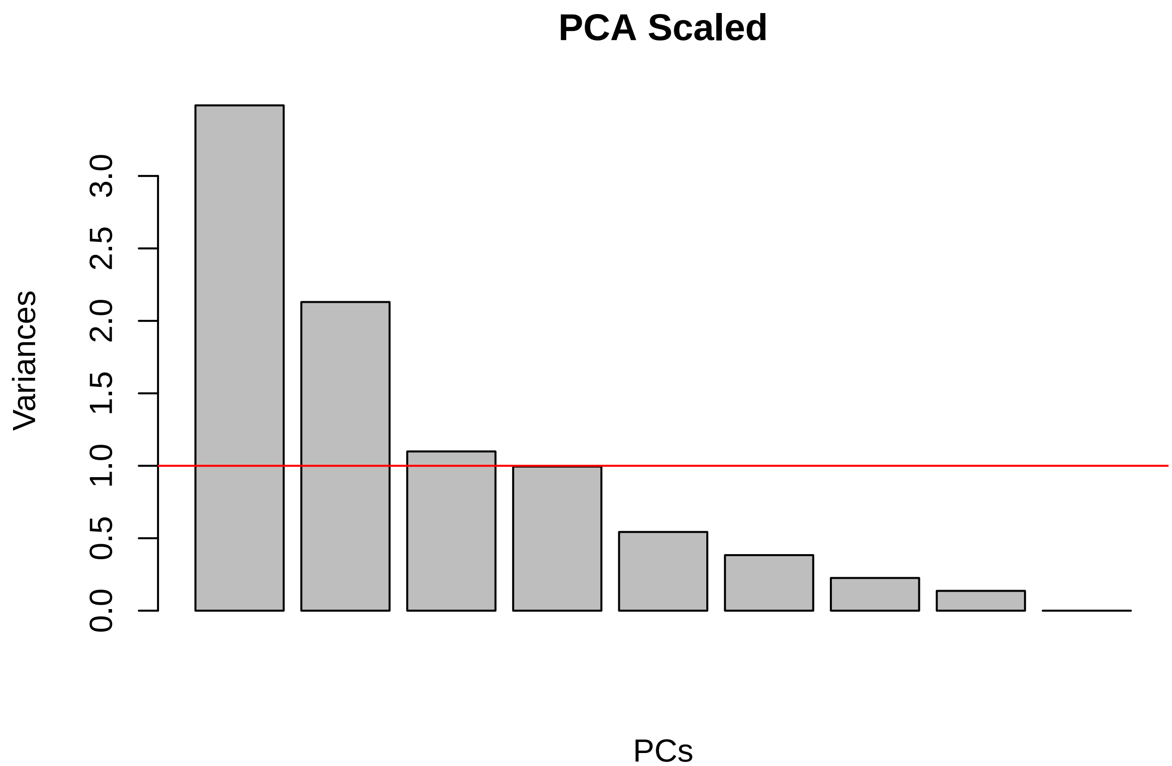



```
## integer(0)
```

```r
# bar scree plot
screeplot(pEmpScale, npcs = 9, type = 'barplot',
          main = "PCA Scaled") + title(xlab = "PCs")
```

```
## integer(0)
```

```r
abline(1, 0, col = "red")
```



Problem 5 c)

No, "VARIMAX" refers to the technique of maximizing the variance of the squared loadings (sqrt(eigenvalues) multiplied by eigenvectors)) of the variables on each factor. This is achieved by rotating the original factor matrix in a way that emphasizes the differences between the variables and the factors and minimizes the correlations between the factors. In b) PCA or principal component analysis was utilized as exploratory analysis for factory discovery.

Problem 5 d)

PC1 = .52Agr + .00Min - .35Man - .26PS - .33Con - .38SI - .07Fin - .39SPS - .37TC

The `Agr` and `Min` are heading in the same direction, but `Agr` has the most significant impact in the positive direction, and `Man`, `PS`, `Con`, `SI`, `Fin`, `SPS`, `TC` is acting in the opposite direction meaning these variable are acting together in the same direction. The ones in the negative direction except for `Fin` all seemed to be fairly balanced, and `Fin` had a lesser impact on the negative direction.

PC2 = .05Agr + .62Min + .36Man + .26PS + .05Con - .35SI - .45Fin - .22SPS + .20TC

The `Agr`, `Min`, `Man`, `PS`, and `Con` act positively, with `Min` having the most significant impact and `Agr` having the least impact. On the other hand, the `SI`, `Fin`, `SP`, and `TC` are in the negative direction and are relatively balanced.

PC3 = .05Agr - .20Min - .15Man - .56PS + .15Con - .12SI - .59Fin + .31SPS + .38TC

The `Agr`, `SPS`, and `TC` are heading in a positive direction, with `Agr` having the least impact in that direction, with the other two being balanced. The negative direction includes `Min`, `Man`, `PS`, `SI`, and `Fin` but `Fin` & `PS` has the balanced largest impact, and the rest has lesser impact and are fairly balanced.

```r
# recasting or rotation
round(pEmpScale$rotation, 2)
```

```
##       PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8  PC9
## Agr  0.52  0.05  0.05 -0.03  0.21  0.15  0.02  0.01 0.81
## Min  0.00  0.62 -0.20 -0.06 -0.16 -0.10 -0.73  0.09 0.05
## Man -0.35  0.36 -0.15  0.35 -0.38 -0.29  0.48  0.13 0.37
## PS  -0.26  0.26 -0.56 -0.39  0.30  0.36  0.26 -0.34 0.02
## Con -0.33  0.05  0.15  0.67  0.47  0.13 -0.22 -0.36 0.08
## SI  -0.38 -0.35 -0.12  0.05 -0.28  0.61 -0.23  0.39 0.24
## Fin -0.07 -0.45 -0.59  0.05  0.28 -0.53 -0.19  0.17 0.15
## SPS -0.39 -0.22  0.31 -0.41 -0.22 -0.26 -0.19 -0.51 0.35
## TC  -0.37  0.20  0.38 -0.31  0.51 -0.12  0.07  0.54 0.07
```

Problem 5 e)

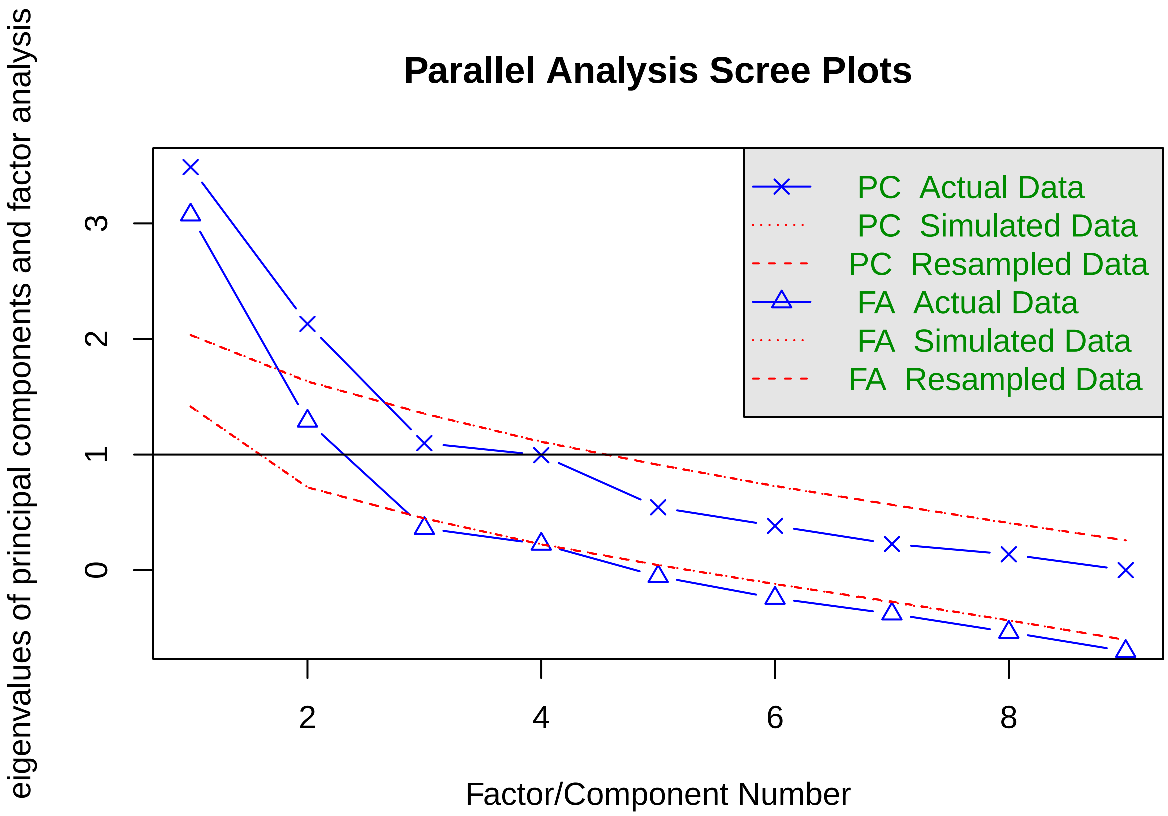
The parallel analysis for this dataset to compute a suggested number of components caused an `ultra-Heywood case was detected` due to the sample size with a warning stating that the scores are probably wrong. This analysis provided the graph with two components compared to scree plots results of three components. According to the parallel analysis, two components were chosen and did explain 62% of which is a suitable percentage of variance captured for the factor analysis.

The decision to select three PCs for Factor analysis because var = 1 and the "knee" plot analysis possibly corresponded with three PCs, which captured 86% of the variance. Finally, depending on the domain knowledge and percentage of captured variance required by the business and between two to five PCs may be acceptable, but we are deciding to choose three PCs to avoid being near the minimum variance requirement.

```r
# parallel analysis with 500 iterations
parallel_PFA = fa.parallel(employ, n.iter=500)
```

```
## 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
```




```
## Parallel analysis suggests that the number of factors =  2  and the number of components =  2
```

Problem 5 f)

RC1 = -.87Agr + .61Con + .64SI + .82SPS + .75TC + .47Man

RC1 has high negative loading on `Agr` and `Con`, `SI`, `SPS`, `TC`, and `Man`, which represents a factor related to the opposite of `Agr`. By examing the corrplot above, `Agr` is negatively correlated with all the variables in this RC.

RC2 = .74Min + .69Man + .81PS

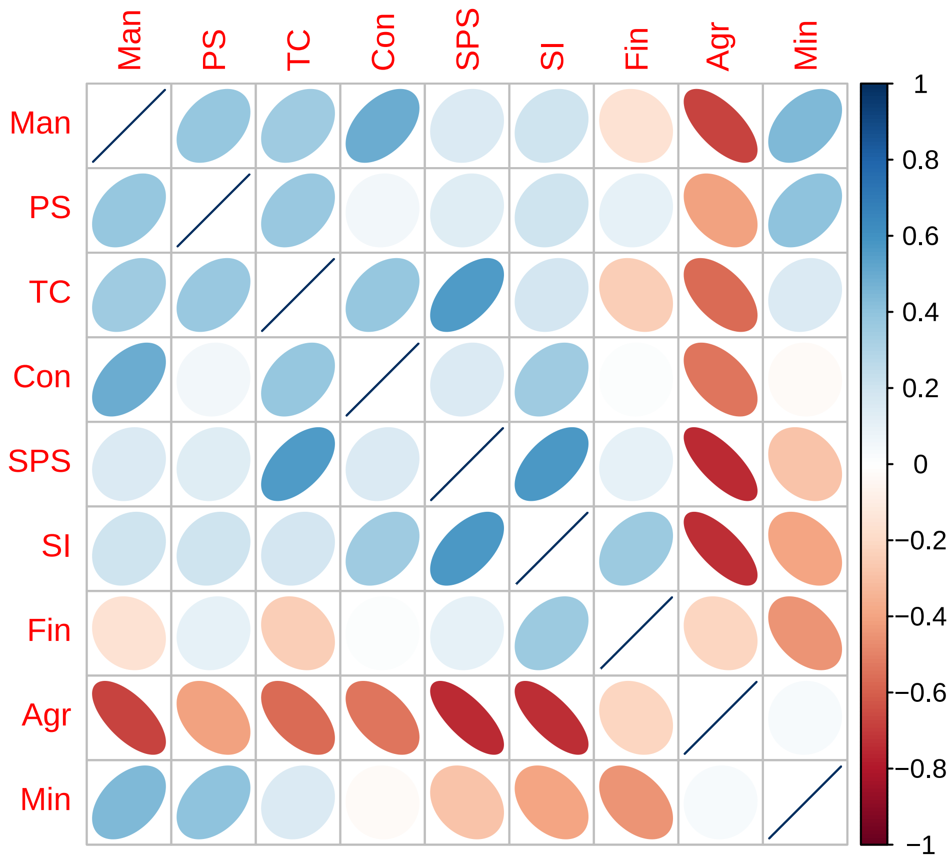
RC2 has all positive fairly evenly distributed positive loadings, and `corrplot` above coincides with all positive relationships.

RC3 = .60SI - .52Min + .91Fin

RC3 has evenly loaded negative loadings on `Min` and high positive loading on `Fin1` & `SI`, which represents a factor opposite of `Min` loading, and `corrplot` corresponds with those factors.

In conclusion, factor analysis with VARIMAX rotation has improved the ability by providing a breakdown into three factors and providing how each variable interacts on each RCs the three factors provide a helpful examination of the underlying features in the data, with each factor capturing a distinct aspect of the data.

```
# corrplot
corrplot(cor(employ), method = "ellipse", order = "AOE")
```




```r
# principal factor analysis (PFA - VARIMAX)
pVARIMAX = principal(employ, rotate="varimax", nfactors=3)

# using cutoff at 0.4
print(pVARIMAX$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##     RC1    RC2    RC3
## Agr -0.871
## Con  0.607
## SI   0.644         0.602
## SPS  0.824
## TC   0.750
## Min         0.743 -0.520
## Man  0.465  0.692
## PS          0.809
## Fin                0.913
##
##                  RC1   RC2   RC3
## SS loadings     3.06 1.902 1.754
## Proportion Var 0.34 0.211 0.195
## Cumulative Var 0.34 0.551 0.746
```

Problem 5 g)

See output below

```r
# RCs & PCs
rcs <- as.data.frame(pVARIMAX$scores[, c(1:3)])
pcs <- as.data.frame(pEmpScale$x[, c(1:3)])

# highest RC1
rc1_h <- tail(arrange(rcs, RC1), 1)
# lowest RC1
rc1_l <- head(arrange(rcs, RC1), 1)
# highest RC2
rc2_h <- tail(arrange(rcs, RC2), 1)
# lowest RC2
rc2_l <- head(arrange(rcs, RC2), 1)
# highest RC3
rc3_h <- tail(arrange(rcs, RC3), 1)
# lowest RC3
rc3_l <- head(arrange(rcs, RC3), 1)

# list of country names: higtest & lowest
country.h <- c(rownames(rc1_h), rownames(rc2_h), rownames(rc3_h))
country.l <- c(rownames(rc1_l), rownames(rc2_l), rownames(rc3_l))

# build dataframe for country
df_country <- cbind(country.h, country.l)
# provide column & row names
colnames(df_country) <- c('Country.High','Country.Low')
rownames(df_country) <- c('RC1', 'RC2', 'RC3')

# display country
df_country
```

```
##     Country.High Country.Low
## RC1 "Norway"     "Yugoslavia"
## RC2 "Hungary"    "Turkey"
## RC3 "Yugoslavia" "USSR"
```

```r
# gather all country names
country <- c(rownames(rc1_h), rownames(rc2_h), rownames(rc3_h),
             rownames(rc1_l), rownames(rc2_l), rownames(rc3_l))
# remove duplicates
country <- country[!duplicated(country)]

# gather PCs for each country
df_PCs <- data.frame(pcs[country[1],])
df_PCs <- rbind(df_PCs, pcs[country[2],],
                pcs[country[3],],
                pcs[country[4],],
                pcs[country[5],])

# print PCs
df_PCs
```

```
##                     PC1        PC2        PC3
```

```
## Norway     -1.65374572 -1.0548269  1.2942914
## Hungary    -0.56711319  3.0824016 -0.9057690
## Yugoslavia  3.87334753 -0.7981284 -3.0518966
## Turkey      6.22427511 -1.0454410  0.9080548
## USSR       -0.04945568  1.2419373  2.3759454
```

Problem 5 h)

According to RC2 & RC4, both Factors are acceptable by examining the `coorplot` from above. In conclusion, determining factors requires domain knowledge, including what percentage of variance captured is required and where this analysis is being used.

Since RC2, RC3, and RC4 are acceptable since the minimum variance captured is 60% as being useful, but it would depend on the business unit's requirements. For example, RC4 would be ideal for prediction accuracy since the 85.7% variance captured by the factor analysis, and for a simpler explanation, it would be possible to use the RC2 model. The result for selecting three components did provide 74.6% cumulative variance, which is not at the bottom end and made sense with the correlation.

```r
# principal factor analysis (PFA - VARIMAX) 2 factors
pVARIMAX_2 = principal(employ, rotate="varimax", nfactors=2)

# using cutoff at 0.4
print(pVARIMAX_2$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##     RC1    RC2
## Agr -0.905
## Man  0.778
## PS   0.572
## Con  0.600
## SPS  0.587  0.532
## TC   0.743
## Min        -0.858
## SI   0.514  0.706
## Fin         0.673
##
##                  RC1   RC2
## SS loadings    3.356 2.261
## Proportion Var 0.373 0.251
## Cumulative Var 0.373 0.624
```

```r
# principal factor analysis (PFA - VARIMAX) 3 factors
pVARIMAX_3 = principal(employ, rotate="varimax", nfactors=3)

# using cutoff at 0.4
print(pVARIMAX_3$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##     RC1    RC2    RC3
## Agr -0.871
## Con  0.607
```

```
## SI   0.644         0.602
## SPS  0.824
## TC   0.750
## Min        0.743 -0.520
## Man  0.465  0.692
## PS         0.809
## Fin               0.913
##
##                  RC1   RC2   RC3
## SS loadings     3.06 1.902 1.754
## Proportion Var 0.34 0.211 0.195
## Cumulative Var 0.34 0.551 0.746
```

```
# principal factor analysis (PFA - VARIMAX) 3 factors
pVARIMAX_4 = principal(employ, rotate="varimax", nfactors=4)

# using cutoff at 0.4
print(pVARIMAX_4$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##     RC1    RC4    RC3    RC2
## Agr -0.688 -0.569
## SPS  0.932
## TC   0.770
## Man         0.750         0.489
## Con         0.898
## SI   0.530         0.621
## Fin                0.912
## Min               -0.550  0.701
## PS                        0.892
##
##                  RC1   RC4   RC3   RC2
## SS loadings    2.363 1.880 1.799 1.668
## Proportion Var 0.263 0.209 0.200 0.185
## Cumulative Var 0.263 0.472 0.671 0.857
```

```
# principal factor analysis (PFA - VARIMAX) 5 factors
pVARIMAX_5 = principal(employ, rotate="varimax", nfactors=5)

# using cutoff at 0.4
print(pVARIMAX_5$loadings, cutoff=.4, sort=T)
```

```
##
## Loadings:
##     RC1    RC3    RC4    RC2    RC5
## Agr -0.833        -0.420
## SI   0.798
## SPS  0.846                       0.448
## Min         0.719        -0.535
## PS          0.883
## Man         0.540  0.589
## Con                0.949
```

```
## Fin                        0.909
## TC                                0.839
##
##                  RC1   RC3   RC4   RC2   RC5
## SS loadings    2.412 1.726 1.539 1.465 1.112
## Proportion Var 0.268 0.192 0.171 0.163 0.124
## Cumulative Var 0.268 0.460 0.631 0.794 0.917
```

Problem 6

6) (20 points, Common Factor Analysis)
   For this problem, you will analyze partial from intelligence tests given to children. Each child was given 11 tests on which they were rated.

Data definition:

1. info = 'Information'
2. comp = 'Comprehension'
3. arith = 'Arithmetic'
4. simil = 'Similarities'
5. vocab = 'Vocabulary'
6. digit = 'Digit Span'
7. pictcomp = 'Picture Completion'
8. parang = 'Paragraph Arrangement'
9. block = 'Block Design'
10. object = 'Object Assembly'
11. coding = 'Coding'

Problem 6 a)

Scaling is an important step in PCA because it helps to standardize the variables and ensure that they are on a comparable scale. However, variables seemed already on the same scale so scaling may be optional. Still, scaling would be essential to standardize the variables to a comparable scale since we need domain knowledge on the test result scale.

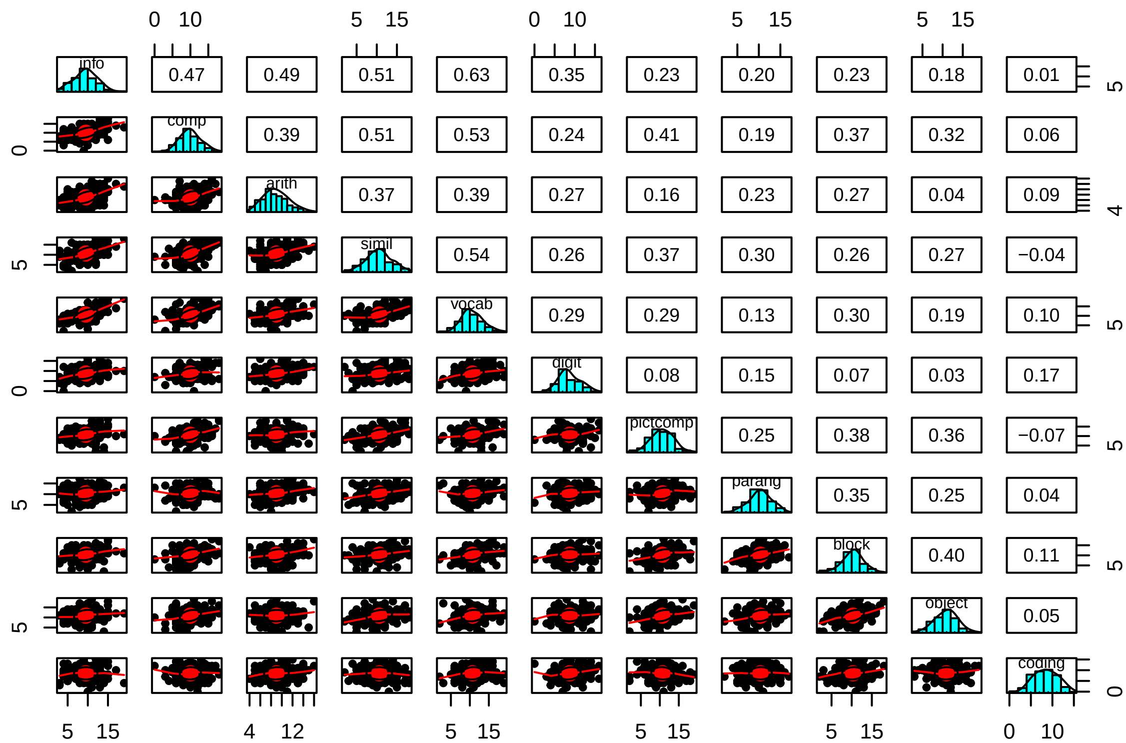
```r
# load data
intell <- read.csv("wiscsem.csv")

# display head
head(intell)
```

```
##   client agemate info comp arith simil vocab digit pictcomp parang block object
## 1      3       3    8    7    13     9    12     9        6     11    12      7
## 2      4       3    9    6     8     7    11    12        6      8     7     12
## 3      5       3   13   18    11    16    15     6       18      8    11     12
## 4      6       3    8   11     6    12     9     7       13      4     7     12
## 5      7       2   10    3     8     9    12     9        7      7    11      4
## 6      8       3   11    7    15    12    10    12        6     12    10      5
##   coding
## 1      9
## 2     14
## 3      9
## 4     11
## 5     10
## 6     10
```

```r
# delete first column
intell <- intell[, -(1:2)]

# summary
summary(intell)
```

```
##       info             comp        arith          simil           vocab
##  Min.   : 3.000   Min.   : 0   Min.   : 4.0   Min.   : 2.00   Min.   : 2.0
##  1st Qu.: 8.000   1st Qu.: 8   1st Qu.: 7.0   1st Qu.: 9.00   1st Qu.: 9.0
##  Median :10.000   Median :10   Median : 9.0   Median :11.00   Median :10.0
##  Mean   : 9.497   Mean   :10   Mean   : 9.0   Mean   :10.61   Mean   :10.7
##  3rd Qu.:11.500   3rd Qu.:12   3rd Qu.:10.5   3rd Qu.:12.00   3rd Qu.:12.0
##  Max.   :19.000   Max.   :18   Max.   :16.0   Max.   :18.00   Max.   :19.0
##      digit            pictcomp         parang          block
##  Min.   : 0.000   Min.   : 2.00   Min.   : 2.00   Min.   : 2.00
##  1st Qu.: 7.000   1st Qu.: 9.00   1st Qu.: 9.00   1st Qu.: 9.00
##  Median : 8.000   Median :11.00   Median :10.00   Median :10.00
##  Mean   : 8.731   Mean   :10.68   Mean   :10.37   Mean   :10.31
##  3rd Qu.:11.000   3rd Qu.:13.00   3rd Qu.:12.00   3rd Qu.:12.00
##  Max.   :16.000   Max.   :19.00   Max.   :17.00   Max.   :18.00
##      object         coding
##  Min.   : 3.0   Min.   : 0.000
##  1st Qu.: 9.0   1st Qu.: 6.000
##  Median :11.0   Median : 9.000
##  Mean   :10.9   Mean   : 8.549
##  3rd Qu.:13.0   3rd Qu.:11.000
##  Max.   :19.0   Max.   :15.000
```

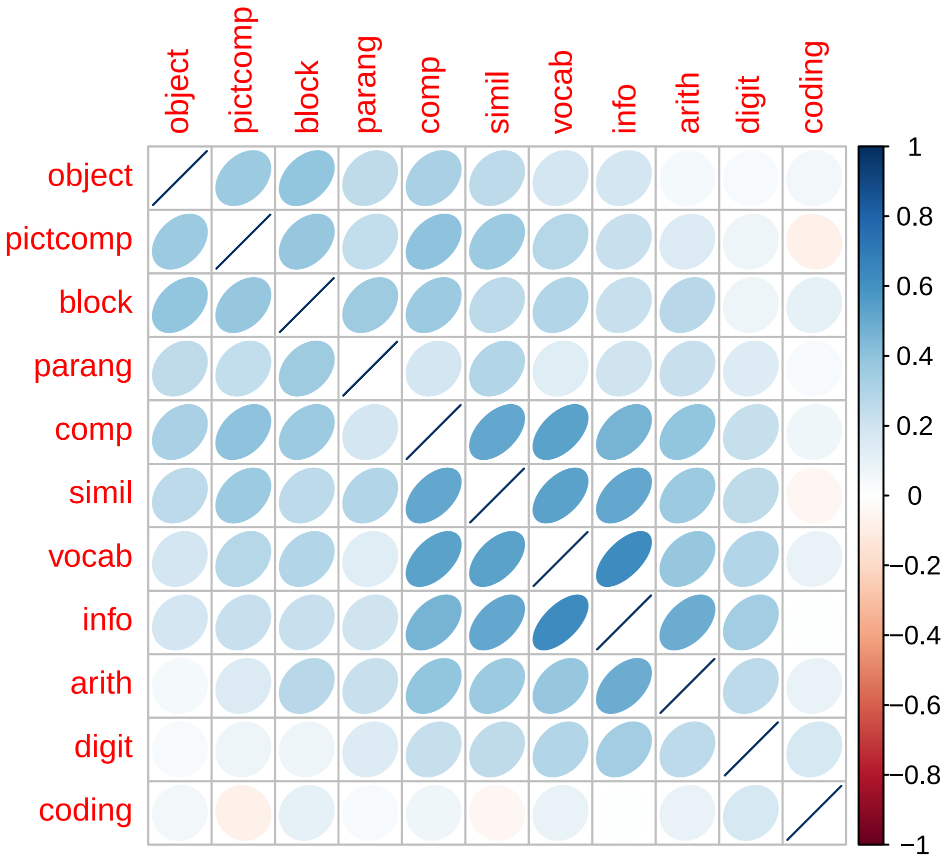
```r
# pairs panel plot
pairs.panels(intell)
```



Problem 6 b)

According to the `corrplot`, every variable has some correlation with each other and `corr.test` at probability at a 95% confidence level, there are significant correlations between all the variables in the dataset; therefore, no variable will likely be a single-variable factor. We are select three components because scree bar plot, parallel analysis scree plot at FA actual vs resampled data, and 59% cumulative variance.

```
# corrplot
corrplot(cor(intell), method = "ellipse", order = "AOE")
```




```r
# determine variables should be their own components
PCorrTestIntell <- corr.test(intell, adjust="none")

# vectorized probability
PCorIntell <- PCorrTestIntell$p

# if True probability at a 95% confidence level
PIntellTest <- ifelse(PCorIntell < 0.05, T, F)

# how many significant correlations there are for each variable.
cat("\nProbability at a 95% Confidence Level: \n")
```

```
##
## Probability at a 95% Confidence Level:
```

```r
colSums(PIntellTest) - 1  # subtract 1 for the diagonal elements (self-correlation)
```

```
##     info     comp    arith    simil    vocab    digit pictcomp   parang
##        9        9        8        9        8        6        8        7
##    block   object   coding
##        8        7        1
```

```r
# PCA NOT Scaled
pCovIntell <- prcomp(intell)
```

```r
# print
pCovIntell
```

```
## Standard deviations (1, .., p=11):
##  [1] 5.633401 3.347877 3.022781 2.570196 2.338059 2.282329 2.220914 2.038211
##  [9] 1.922074 1.840692 1.574071
##
## Rotation (n x k) = (11 x 11):
##                PC1          PC2         PC3         PC4          PC5
## info     0.38004352  0.321280168 -0.12941297  0.03151105  0.182129690
## comp     0.40279885 -0.004006781 -0.02424810 -0.33518717  0.010706479
## arith    0.22804986  0.200526144  0.03479953  0.15327661  0.308519993
## simil    0.43826191  0.079506647 -0.25434385  0.12703267 -0.126841037
## vocab    0.39362475  0.273620054 -0.05144879 -0.27113863  0.169639971
## digit    0.19361908  0.364952081  0.21857426  0.32631020 -0.669968098
## pictcomp 0.29801014 -0.435987208 -0.12637030 -0.14291290 -0.390104198
## parang   0.20016727 -0.237422564  0.16215216  0.75599334  0.163096138
## block    0.26510224 -0.351964084  0.25469222  0.04963212  0.398874984
## object   0.23633706 -0.491184973  0.17731997 -0.15477496 -0.199686653
## coding   0.04242157  0.174863633  0.85309293 -0.22059564 -0.006949277
##                  PC6         PC7          PC8        PC9         PC10
## info      0.161487765 -0.27594632 -0.365672811 -0.3714394  0.113408272
## comp     -0.142134120 -0.03379482  0.728786007 -0.2272717 -0.312549961
## arith    -0.203424815 -0.11782385  0.171924129 -0.2703414  0.570540927
## simil     0.258319653  0.64471194  0.101625606  0.3343590  0.299348807
## vocab     0.034675655 -0.01506809 -0.373404592  0.2464783 -0.488976733
## digit    -0.079276563 -0.38859350  0.107911570  0.2373103 -0.002808166
## pictcomp -0.579838030  0.11029915 -0.360041952 -0.2119011  0.096039438
## parang    0.006591808  0.17475153 -0.002232641 -0.2492613 -0.428448300
## block    -0.210631508 -0.30525795  0.022090768  0.6011478  0.146947521
## object    0.675403767 -0.24978289 -0.022921480 -0.1521481  0.125540404
## coding   -0.036472559  0.37903263 -0.116379534 -0.1298891  0.071223664
##                PC11
## info     -0.56094930
## comp     -0.15123113
## arith     0.55178779
## simil    -0.10282270
## vocab     0.47385706
## digit     0.03086562
## pictcomp  0.01990928
## parang    0.05346310
## block    -0.24166313
## object    0.22274006
## coding   -0.10325051
```

```r
# summary
summary(pCovIntell)
```

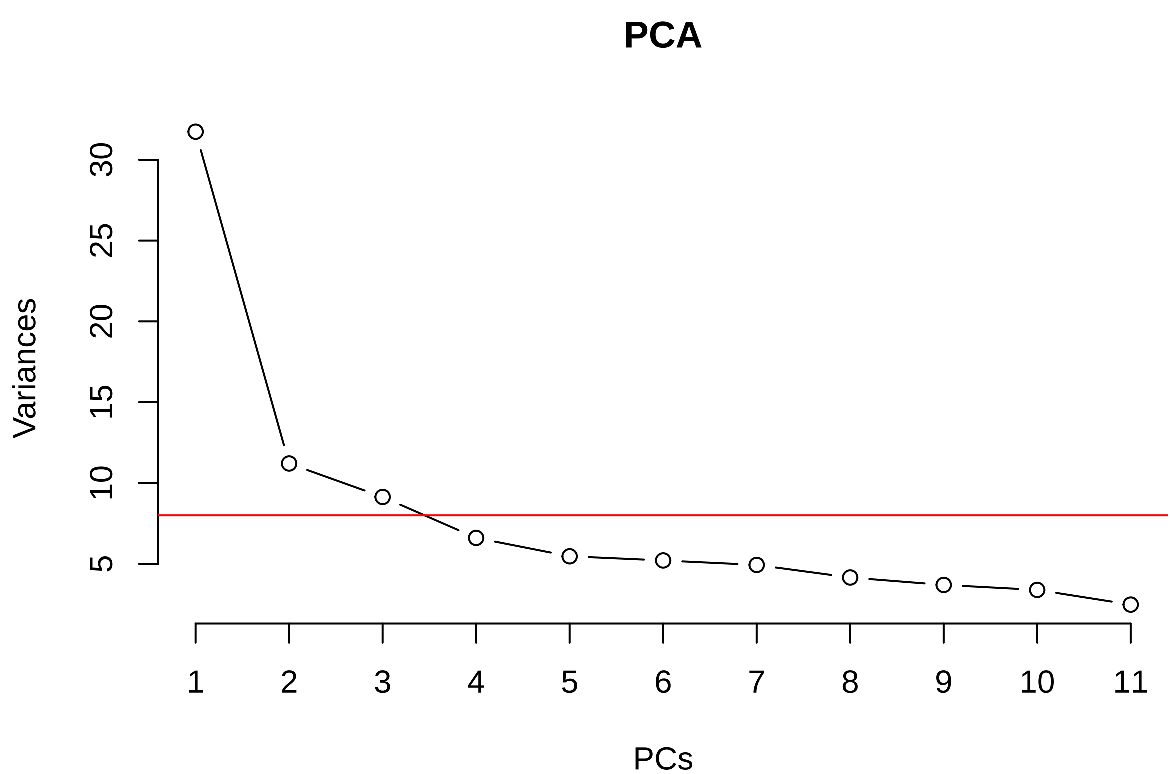
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     5.6334 3.3479 3.0228 2.57020 2.33806 2.28233 2.22091
## Proportion of Variance 0.3606 0.1273 0.1038 0.07506 0.06211 0.05919 0.05604
```

```
## Cumulative Proportion  0.3606 0.4879 0.5918 0.66682 0.72894 0.78812 0.84417
##                           PC8     PC9   PC10    PC11
## Standard deviation     2.0382 1.92207 1.8407 1.57407
## Proportion of Variance 0.0472 0.04198 0.0385 0.02815
## Cumulative Proportion  0.8914 0.93335 0.9718 1.00000
```

```
# average variance
avgVar = mean(pCovIntell$sdev^2)

# line screeplot
screeplot(pCovIntell, npcs = 11, type = 'lines',
          main = "PCA") + title(xlab = "PCs")
```

```
## integer(0)
```

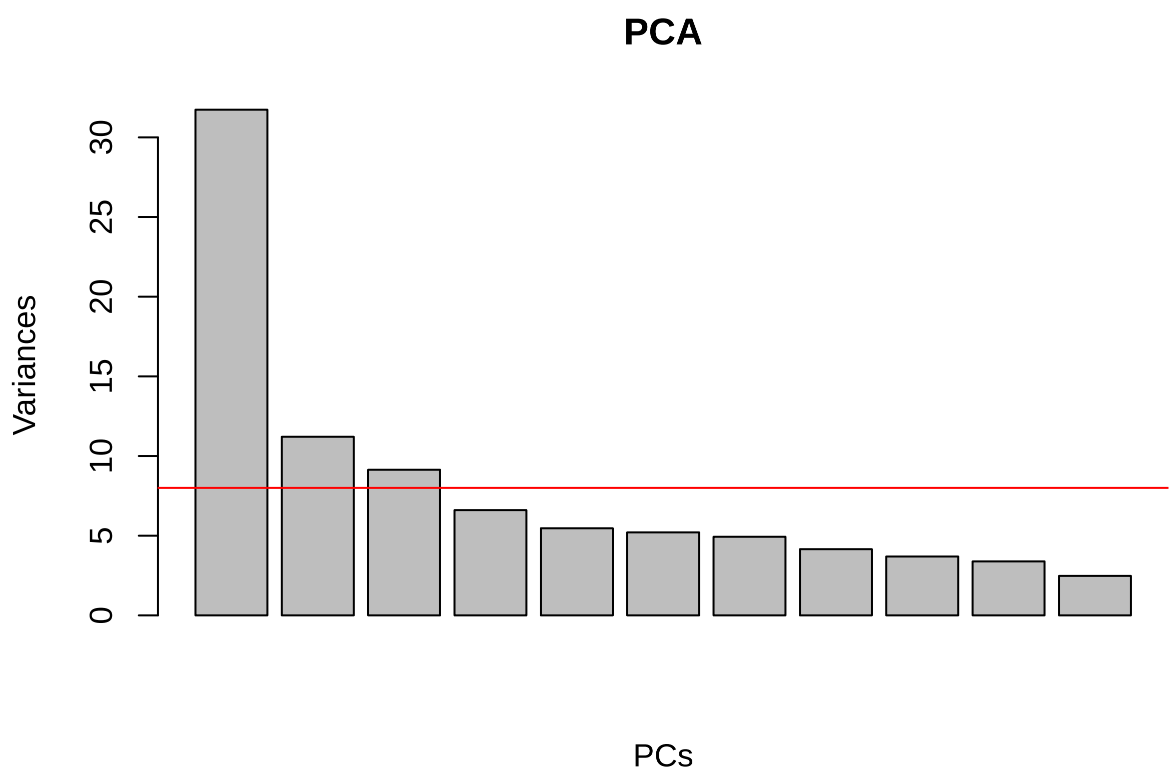
```
abline(avgVar, 0, col = "red")
```



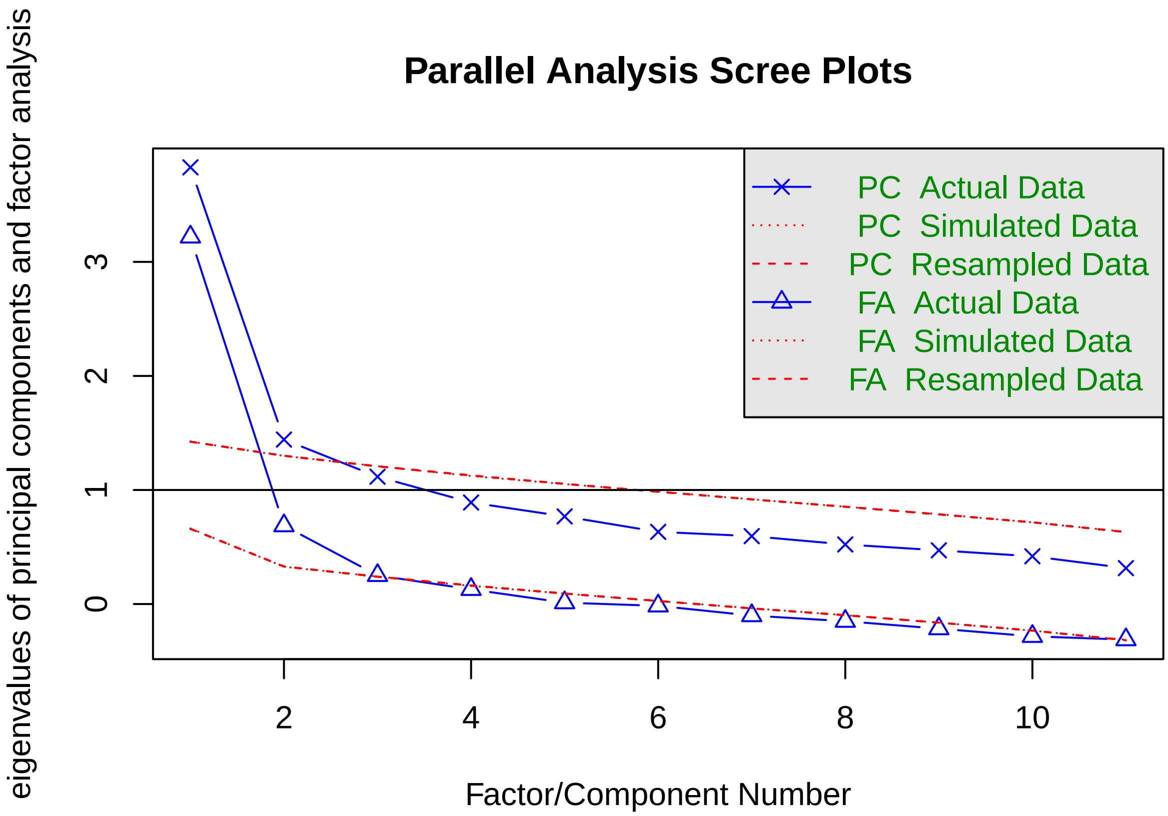

```
# bar scree plot
screeplot(pCovIntell, npcs = 11, type = 'barplot',
          main = "PCA") + title(xlab = "PCs")
```

```
## integer(0)
```

```r
abline(avgVar, 0, col = "red")
```




```r
# parallel analysis with 500 iterations
parallel_intellPFA = fa.parallel(intell, n.iter=500)
```




```
## Parallel analysis suggests that the number of factors =  2  and the number of components =  2
```

Problem 6 c)

The result of a factor analysis containing loadings of each variable on the first three principal components (RC1, RC2, RC3) represents the correlations between each variable and each component with a cutoff of 0.4.

According to the loadings, there is `comp` (Comprehension), `arith` (Arithmetic), `simil` (Similarities), `vocab` (Vocabulary), and `digit` (Digit Span) on RC1 all acting in the same direction with somewhat evenly loaded. In RC2, `pictcomp` (Picture Completion), `parang` (Paragraph Arrangement), `block` (Block Design), and `object` (Object Assembly) are also somewhat evenly loaded, including all the contributing heading in the same direction. On RC3, `coding` (coding) and `digit` contribute in the same direction, including correlation being significant, but `coding` is more significant in terms of contribution at RC3.

While examing the plot, there is a similarity between the plot versus Principal Factor Analysis with VARIMAX rotations loadings. For example, the plot displaying `comp`, `simil`, `vocab`, `arith`, and `digit` are one grouping of factors, and the second grouping contains `object`, `block`, `pictcomp`, and `parang`. Lastly, the third group contains `coding` and a small part of `digit`. RC1 = Math & Language intelligence & RC2 = Spatial intelligence

```r
# source code for PCA_Plot & PCA_Plot_Psyc
source("PCA_Plot.r")

# VARIMAX w/ all the variables
pIntellVARIMAX <- principal(intell, rotate = "varimax", nfactors = 3)
```

```r
# print
pIntellVARIMAX
```

```
## Principal Components Analysis
## Call: principal(r = intell, nfactors = 3, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            RC1   RC2   RC3   h2   u2 com
## info      0.83  0.11 -0.02 0.69 0.31 1.0
## comp      0.63  0.42 -0.05 0.58 0.42 1.7
## arith     0.67  0.08  0.18 0.49 0.51 1.2
## simil     0.69  0.32 -0.17 0.62 0.38 1.6
## vocab     0.78  0.18  0.01 0.64 0.36 1.1
## digit     0.53 -0.07  0.43 0.47 0.53 1.9
## pictcomp  0.25  0.65 -0.28 0.56 0.44 1.7
## parang    0.14  0.57  0.16 0.37 0.63 1.3
## block     0.17  0.74  0.14 0.60 0.40 1.2
## object    0.04  0.76 -0.03 0.57 0.43 1.0
## coding   -0.01  0.11  0.88 0.79 0.21 1.0
##
##                        RC1  RC2  RC3
## SS loadings           3.02 2.21 1.15
## Proportion Var        0.27 0.20 0.10
## Cumulative Var        0.27 0.48 0.58
## Proportion Explained  0.47 0.35 0.18
## Cumulative Proportion 0.47 0.82 1.00
##
## Mean item complexity =  1.3
## Test of the hypothesis that 3 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.09
##  with the empirical chi square  147.47  with prob <  2.5e-19
##
## Fit based upon off diagonal values = 0.91
```

```r
# summary
summary(pIntellVARIMAX)
```

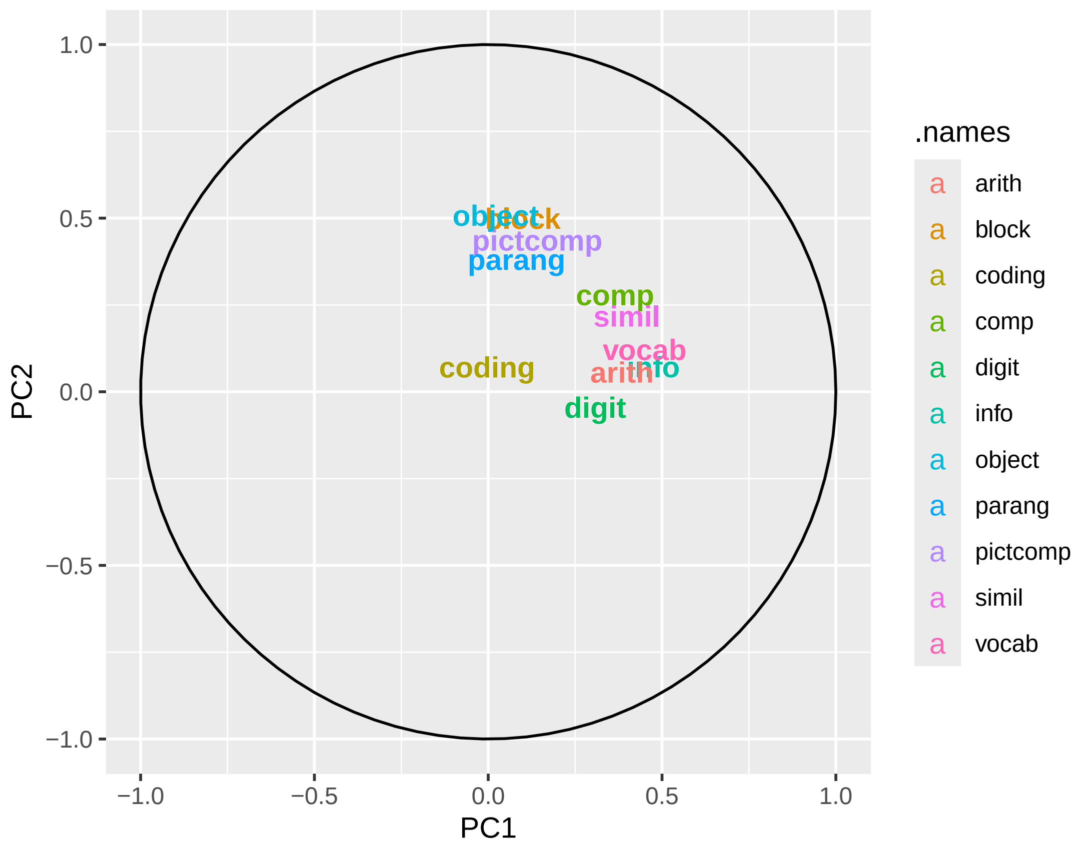
```
##
## Factor analysis with Call: principal(r = intell, nfactors = 3, rotate = "varimax")
##
## Test of the hypothesis that 3 factors are sufficient.
## The degrees of freedom for the model is 25  and the objective function was  0.73
## The number of observations was  175  with Chi Square =  121.51  with prob <  1.2e-14
##
## The root mean square of the residuals (RMSA) is  0.09
```

```r
# kind of loading structure
print(pIntellVARIMAX$loadings, cutoff=.4)
```

```
##
## Loadings:
##          RC1    RC2    RC3
```

```
## info      0.826
## comp      0.634  0.416
## arith     0.669
## simil     0.694
## vocab     0.782
## digit     0.535         0.428
## pictcomp         0.649
## parang           0.567
## block            0.743
## object           0.756
## coding                  0.883
##
##                   RC1   RC2   RC3
## SS loadings    3.022 2.211 1.154
## Proportion Var 0.275 0.201 0.105
## Cumulative Var 0.275 0.476 0.581
```

```
# PCA_Plot_Psyc
PCA_Plot_Psyc(pIntellVARIMAX)
```



Problem 6 d)

By examining the high scores on RC1 and RC2, high RC1 scores (Math & Language intelligence) tend to perform poorly on spatial intelligence. Similarly, high RC2 scores (Spatial intelligence) generally performed poorly in math/language intelligence.

While there are exceptions (such as the child with a high score on both RC1 and RC2), the general trend holds in this dataset. Finally, factor analysis makes interpreting the high-dimensional dataset easier by summarizing the information into a smaller number of components. This data has no surprises, but I wonder if this trend would hold among different gender.

```r
# copy PC1 & PC2
rc1n2Scoring <- as.data.frame(pIntellVARIMAX$scores[,1:2])

#
tail(arrange(rc1n2Scoring, RC1),10)
```

```
##          RC1        RC2
## 166 1.713285  0.5268298
## 167 1.730310  0.3260319
## 168 1.756553  0.6754826
## 169 1.989139  1.7910913
## 170 2.022425  0.6860689
## 171 2.098106 -0.0242992
## 172 2.128616 -0.6852655
## 173 2.157673  2.1370419
## 174 2.336594  0.0238716
## 175 3.058884 -0.1727502
```

```r
#
tail(arrange(rc1n2Scoring, RC2),10)
```

```
##            RC1      RC2
## 166 -0.1824237 1.558315
## 167 -1.2790950 1.567952
## 168  0.3383830 1.593998
## 169 -2.1248127 1.595910
## 170 -1.1012058 1.636385
## 171  1.9891390 1.791091
## 172 -1.2019603 1.874698
## 173 -0.7992679 1.912309
## 174  2.1576727 2.137042
## 175 -0.8075214 2.512265
```

Problem 6 e)

Common Factor Analysis (CFA) is an exploratory factor analysis technique that assumes a common underlying factor affects all the variables. In contrast to principal component analysis, which seeks to explain the maximum amount of variance in the original variables, CFA aims to find a smaller number of factors that account for the maximum amount of covariance between the variables.

In the given reports, the loadings of the CFA show that Factor 1 is composed of variables `comp`, `arith`, `simil`, `vocab`, and `digit`. Factor 2 comprises variables `pictcomp`, `block`, `object`, and `comp`. Factor 3 includes nothing. The total cumulative variance explained by the three factors is 39.8%, lower than the 58% explained by the three factors in the previous Factor Analysis.

The exclusion of `parang` and `coding` in the CFA, but not in the previous factor analysis, indicates that these two variables may have a common underlying factor that affects them. On the other hand, the splitting of `comp` from CFA and its inclusion in Factor 2 may indicate that its relationship with the other variables is also related to the concept of spatial intelligence.

The lower cumulative variance explained by CFA compared to the previous Factor Analysis could be due to the assumption of a common underlying factor that affects all the variables, which may not always be accurate in real-life situations. Furthermore, the fact that `parang` is significantly correlated with six other variables, as shown by the `coor.test`, could contribute to the lost variance explained by the CFA. Lastly, `prang` and `coding` is not displayed in Factor 3 with a cutoff of 0.4 may be due to a lack of data since the CFA is a statistical analysis instead of a geometric one.

```
# common factor analysis and compare
fit_cfa3 = factanal(intell, 3, scores="regression", rotation = "varimax")

# print all
print(fit_cfa3)
```

```
##
## Call:
## factanal(x = intell, factors = 3, scores = "regression", rotation = "varimax")
##
## Uniquenesses:
##     info     comp    arith    simil    vocab    digit pictcomp   parang
##    0.363    0.494    0.599    0.455    0.415    0.797    0.556    0.805
##    block   object   coding
##    0.332    0.662    0.913
##
## Loadings:
##          Factor1 Factor2 Factor3
## info      0.779   0.156
## comp      0.551   0.449
## arith     0.556   0.140   0.269
## simil     0.620   0.366  -0.160
## vocab     0.721   0.252
## digit     0.431           0.134
## pictcomp  0.202   0.605  -0.194
## parang    0.154   0.392   0.135
## block     0.117   0.714   0.380
## object            0.573
## coding                    0.290
##
##                Factor1 Factor2 Factor3
## SS loadings      2.399   1.801   0.410
## Proportion Var   0.218   0.164   0.037
## Cumulative Var   0.218   0.382   0.419
##
## Test of the hypothesis that 3 factors are sufficient.
## The chi square statistic is 30.58 on 25 degrees of freedom.
## The p-value is 0.203
```

```
# print common factor analysis
print(fit_cfa3$loadings, cutoff = 0.4, sort=T)
```

```
##
## Loadings:
##          Factor1 Factor2 Factor3
## info      0.779
## comp      0.551   0.449
## arith     0.556
```

```
## simil     0.620
## vocab     0.721
## pictcomp          0.605
## block             0.714
## object            0.573
## digit     0.431
## parang
## coding
##
##                Factor1 Factor2 Factor3
## SS loadings      2.399   1.801   0.410
## Proportion Var   0.218   0.164   0.037
## Cumulative Var   0.218   0.382   0.419
```

```r
# kind of loading structure
print(pIntellVARIMAX$loadings, cutoff=.4)
```

```
##
## Loadings:
##          RC1    RC2    RC3
## info      0.826
## comp      0.634  0.416
## arith     0.669
## simil     0.694
## vocab     0.782
## digit     0.535         0.428
## pictcomp         0.649
## parang           0.567
## block            0.743
## object           0.756
## coding                  0.883
##
##                  RC1   RC2   RC3
## SS loadings    3.022 2.211 1.154
## Proportion Var 0.275 0.201 0.105
## Cumulative Var 0.275 0.476 0.581
```

Problem 7

7) (Paper review) An academic paper on principal component analysis in genetics research is posted in the "Supplimental Reading List" and also included in the homework materials. Read the paper and review the paper's use of PCA. In your analysis, you should address the following:

Problem 7 a)

The data used in the paper, which contains gene expression ratios across different time points during sporulation, is suitable for PCA analysis. They are applying PCA to extract underlying variables that explain the variation in the data, and their goal is dimensionality reduction and interpretation of the resulting components.

Problem 7 b)

In this report, the natural log transform to all ratios was used to transform the data, effectively reducing the influence of extreme values and making the data more symmetric and easier to analyze due to equalizing the variance.

Problem 7 c)

In this paper, a common rotation method is used, which is principal component analysis (PCA), to transform the original data into a new set of orthogonal variables, or principal components, which capture the maximum amount of variance in the data. In terms of factor rotation in this paper, they are rotating orthogonally, a type of factor rotation.

Problem 7 d)

The report suggests that the data can be summarized with just two principal components, as two eigenvalues lie above the 10% cutoff accounts for over 90% of the total variability; including the third component accounts for almost 95%.

Problem 7 e)

They perform a sensitivity analysis where they remove gene subsets and observe the principal components' stability. They also mention that the first two components are consistently recovered across multiple different preprocessing methods, suggesting that these components are robust.

Problem 7 f)

PCA allows the authors to draw several conclusions about the gene expression data in the context of sporulation. First, they find that the data can be summarized effectively using just two principal components, which account for over 90% of the total variability in the dataset.

Through further analysis of the coefficients associated with these components, the authors are able to draw more specific conclusions about the genes that are up- or down-regulated during sporulation, as well as those that exhibit positive or negative trends in expression over time. They are also able to identify a third component that measures concavity in gene expression profiles.

In conclusion, Principal components analysis is often used as a preprocessing step to clustering to identify several gene classes relevant to sporulation but they discovered that the genes are not located in clusters rather they are spread throughout this space.