Dimension Reduction with Principal Component Analysis

This document will give a simple explanation and examples of PCA in R. The code can be used as for your own projects.

PCA is an unsupervised technique for continuous data that creates new variables called principal components.

PCA golden rule

$$Variation = Information$$

Our goal is to reduce the dimensionality of the data while still retaining most of the variability from de original data.

We will try to keep $q < p$ dimensions, where q are the principal components and p are the original variables.

Some important points

- PCA is useful to get rid of multicollinearity.
- It can be used with a correlation or a covariance matrix.
- The number of PC to retain can be found with these methods:
  - Proportion of variation.
  - Cattell's method.
  - Kaiser's method.
- No model or distribution is assumed.
- Outliers highly affect PCA.

R code example

We will take a look at a dataset on wine. This data set gives the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.The analysis determined the quantities of 13 variables found in each of the three types of wines. The data were sourced from the UCI Machine Learning Repository and you can find more information about them here.

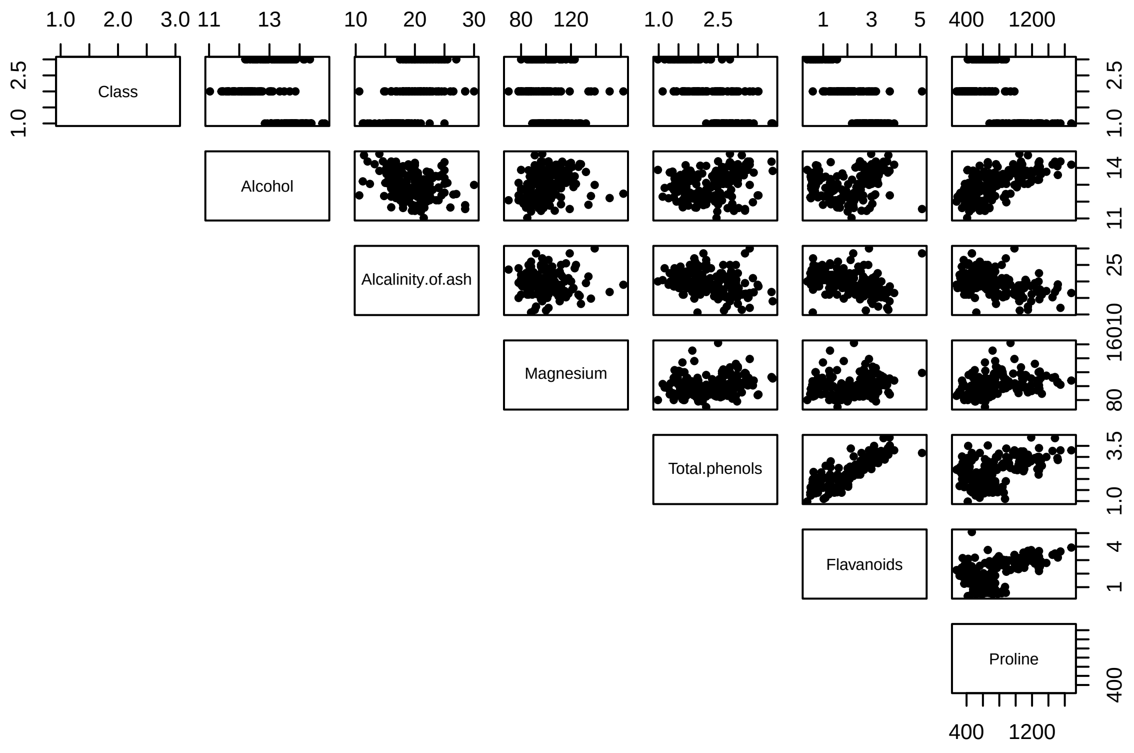
First we will read the data into R and review the different variables we will be working with:

```r
wine<-read.csv("wine.data.csv")
str(wine,vec.len=2)
```

```
## 'data.frame':    178 obs. of  14 variables:
##  $ Class                       : int  1 1 1 1 1 ...
##  $ Alcohol                     : num  14.2 13.2 ...
##  $ Malic.acid                  : num  1.71 1.78 2.36 1.95 2.59 ...
##  $ Ash                         : num  2.43 2.14 2.67 2.5 2.87 ...
##  $ Alcalinity.of.ash           : num  15.6 11.2 18.6 16.8 21 ...
##  $ Magnesium                   : int  127 100 101 113 118 ...
##  $ Total.phenols               : num  2.8 2.65 2.8 3.85 2.8 ...
##  $ Flavanoids                  : num  3.06 2.76 3.24 3.49 2.69 ...
##  $ Nonflavanoid.phenols        : num  0.28 0.26 0.3 0.24 0.39 ...
##  $ Proanthocyanins            : num  2.29 1.28 2.81 2.18 1.82 ...
##  $ Colour.intensity           : num  5.64 4.38 5.68 7.8 4.32 ...
##  $ Hue                        : num  1.04 1.05 1.03 0.86 1.04 ...
##  $ OD280.OD315.of.diluted.wines: num  3.92 3.4 3.17 3.45 2.93 ...
##  $ Proline                    : int  1065 1050 1185 1480 735 ...
```

We can also make a pair plot to review patterns, outliers, etc.

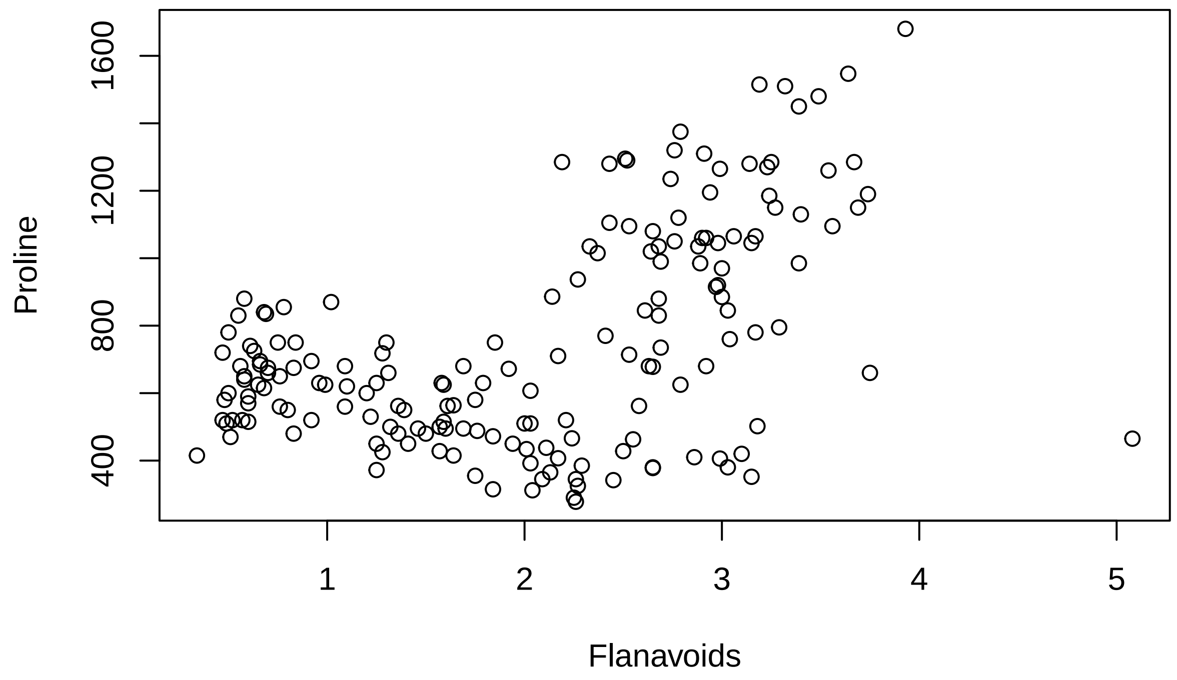
```r
pairs(wine[,c(1:2,5:8,14)],pch=20,lower.panel = NULL)
```



From this we can see that the variable "Class"is not continuous. Also, we can see some correlation between "Flavanoids" — "Total.phenols" and "Alcohol" — "Proline". There a couple outliers so we need to take care of them.

Here we plot flavanoids vs proline to identify the outlier:

```r
plot(wine$Flavanoids,wine$Proline,xlab="Flanavoids",ylab="Proline")
```



We identify the outlier by plotting the max value in flavanoids and then removing it. We also will delete the first column since it is not continuous.

```
print(max(wine$Flavanoids)) # It prints 5.08
```

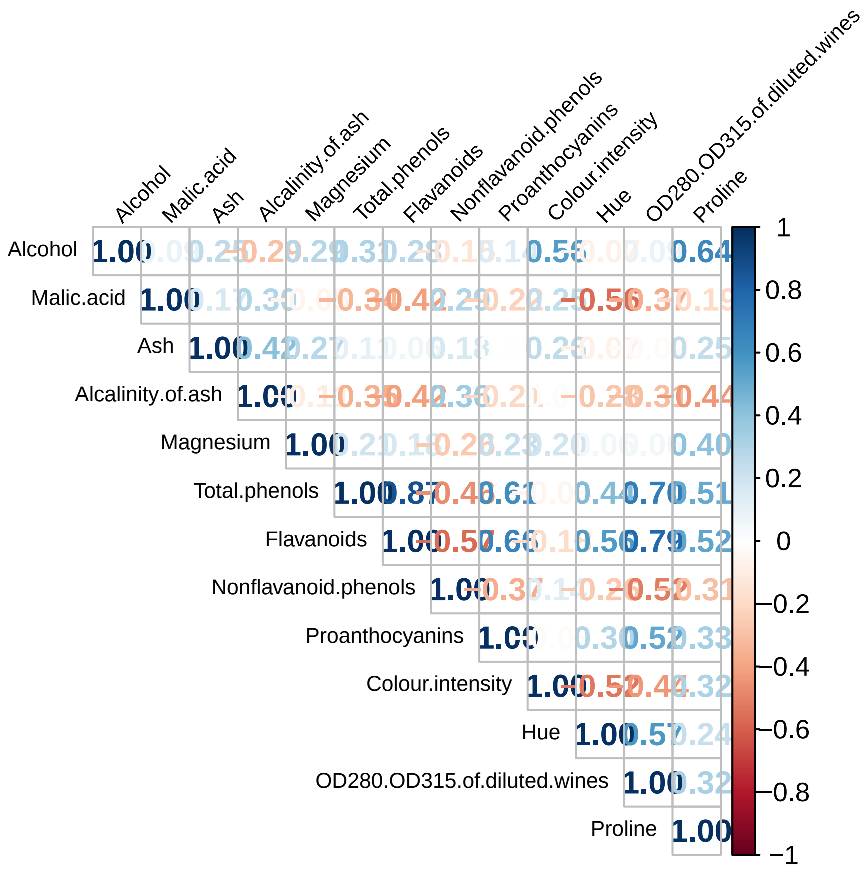
```
## [1] 5.08
```

```
wine.new <- wine[wine$Flavanoids < 5.08, -1] # delete outlier and 1st column
```

```
# Load the corrplot library for visualizing correlation matrices
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
# Compute the correlation matrix for the wine.new data frame
M <- cor(wine.new)
#Plot
corrplot(M, method = "number", type = "upper",
         tl.cex = 0.7, tl.col = "black", tl.srt = 45, addrect = 2)
```



You can see that some of the variables have high correlation, which means that some dimension reduction might be possible.

We need to review the variances to decide if we are using the correlation or covariance matrix in our PCA.

```
print(round(diag(var(wine.new)),2))
```

```
##                      Alcohol                   Malic.acid
##                         0.65                         1.25
##                          Ash            Alcalinity.of.ash
##                         0.07                        10.75
##                    Magnesium                Total.phenols
##                       203.03                         0.39
##                   Flavanoids         Nonflavanoid.phenols
##                         0.95                         0.02
##              Proanthocyanins              Colour.intensity
##                         0.33                         5.40
##                          Hue OD280.OD315.of.diluted.wines
##                         0.05                         0.50
##                      Proline
##                     99276.11
```

As you can see, there are huge differences in the variances, so we proceed using correlation matrix.

```
# Perform principal component analysis (PCA) on the wine.new data frame
# Set cor = TRUE to perform PCA using correlation matrix instead of covariance matrix
wine.pca <- princomp(wine.new, cor = TRUE)
```

```r
# Output the results of PCA
print(wine.pca)
```

```
## Call:
## princomp(x = wine.new, cor = TRUE)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 2.1861050 1.5899422 1.1601299 0.9617078 0.9282584 0.8004340 0.7439816 0.5794238
##    Comp.9   Comp.10   Comp.11   Comp.12   Comp.13
## 0.5397900 0.5024390 0.4813398 0.4073751 0.2986393
##
##  13  variables and  177 observations.
```

It's anticipated that we obtain 13 components, given the initial 13 variables and 177 observations. Although the first component exhibits a notably larger standard deviation compared to the others, determining the number of components to retain isn't straightforward from this observation alone. To make an informed decision, we need to choose from one of three methods to determine the retained components. In this example, we'll explore all three methods, but it's important to note that, in practice, only one method should be selected.

Proportion of Variance: A standard number is to keep 80% of the variability. We need to see the cumulative proportions of variance for all the components which we get using the summary command.

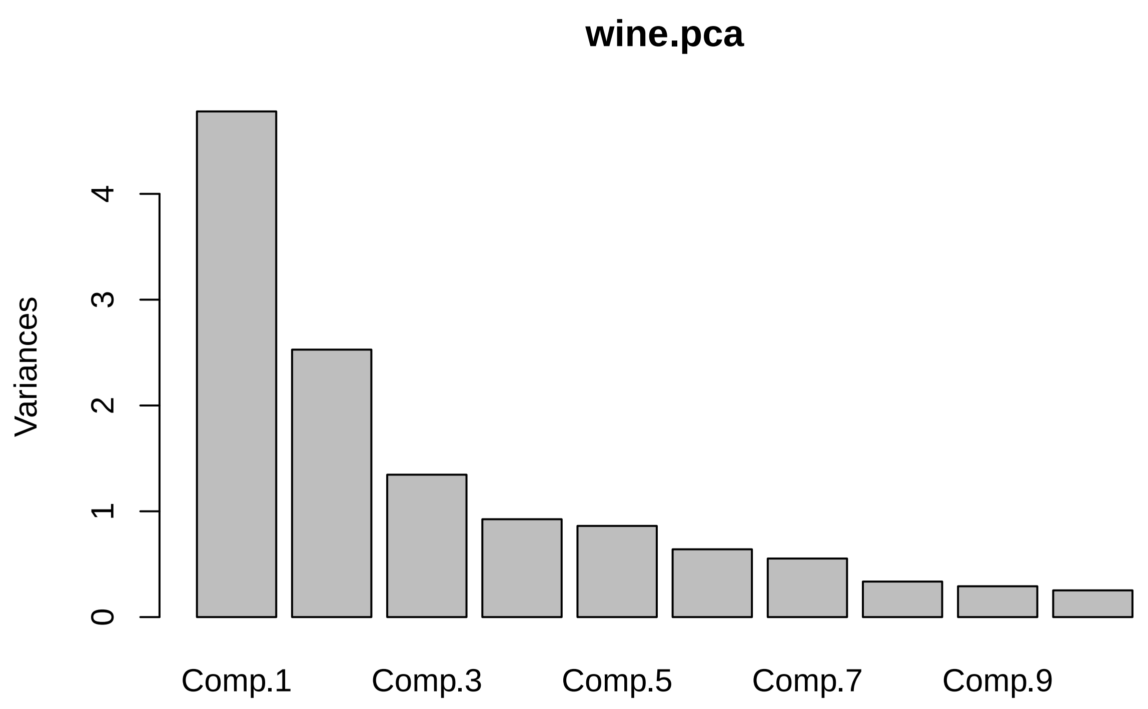
```r
summary(wine.pca)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     2.1861050 1.5899422 1.1601299 0.96170782 0.92825842
## Proportion of Variance 0.3676196 0.1944551 0.1035309 0.07114476 0.06628182
## Cumulative Proportion  0.3676196 0.5620747 0.6656056 0.73675038 0.80303220
##                           Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     0.8004340 0.74398164 0.57942382 0.53978997 0.50243903
## Proportion of Variance 0.0492842 0.04257759 0.02582554 0.02241332 0.01941884
## Cumulative Proportion  0.8523164 0.89489400 0.92071953 0.94313286 0.96255170
##                           Comp.11    Comp.12     Comp.13
## Standard deviation     0.48133977 0.40737507 0.298639313
## Proportion of Variance 0.01782215 0.01276573 0.006860418
## Cumulative Proportion  0.98037386 0.99313958 1.000000000
```

So if we want to retain 80% of the variability we would keep 5 principal components.

Cattell's method: To use this method, we need to produce a scree plot of the proportion of variance versus component number. We expect this to drop off (since the eigenvalues are in decreasing order) but what we are looking for in the plot is a bend, where the rate of decrease suddenly reduces.

```r
plot(wine.pca)
```



There are a couple of different place that we could say represent a change in rate of decrease: between components 3 and 4, or 5 and 6 or 7 and 8. If we wanted to really reduce the dimensionality we could argue the first one and say we are retaining 3 components.

Kaiser's method: Here we need to look at finding the average eigenvalue to discover which set of components have variation above it that we will retain. Because we used the correlation matrix, we know the average should be 1 but let's check.

```
#Extract the component standard deviations
sd.pca<-wine.pca$sdev #Find the average variance, take the mean after squaring them
ave.var<-mean((sd.pca**2))
ave.var
```

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

```
#Find which components have higher than average variance (TRUE)
sd.pca**2>ave.var
```

```
##  Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9 Comp.10
##    TRUE    TRUE    TRUE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE
## Comp.11 Comp.12 Comp.13
##   FALSE   FALSE   FALSE
```

So based on this, we would retain the first 3 PCs.

We will now examine if there is any meaningful interpretation that can be derived from the loadings of the PCA for our wine dataset. To accomplish this, we extract the loadings (or rotation) matrix from the fitted object. Interpreting these loadings can be somewhat subjective. We examine each column in turn, looking for variables with relatively large positive values and grouping them together. Similarly, we identify variables with relatively large negative values and group them together as well. The principal component (PC) is interpreted as the contrast between these groups of variables. If all variables with significant loadings have the same sign, then that PC is interpreted as the (weighted) average or overall score of these variables.

```r
wine.pca$loadings[,1:7]
```

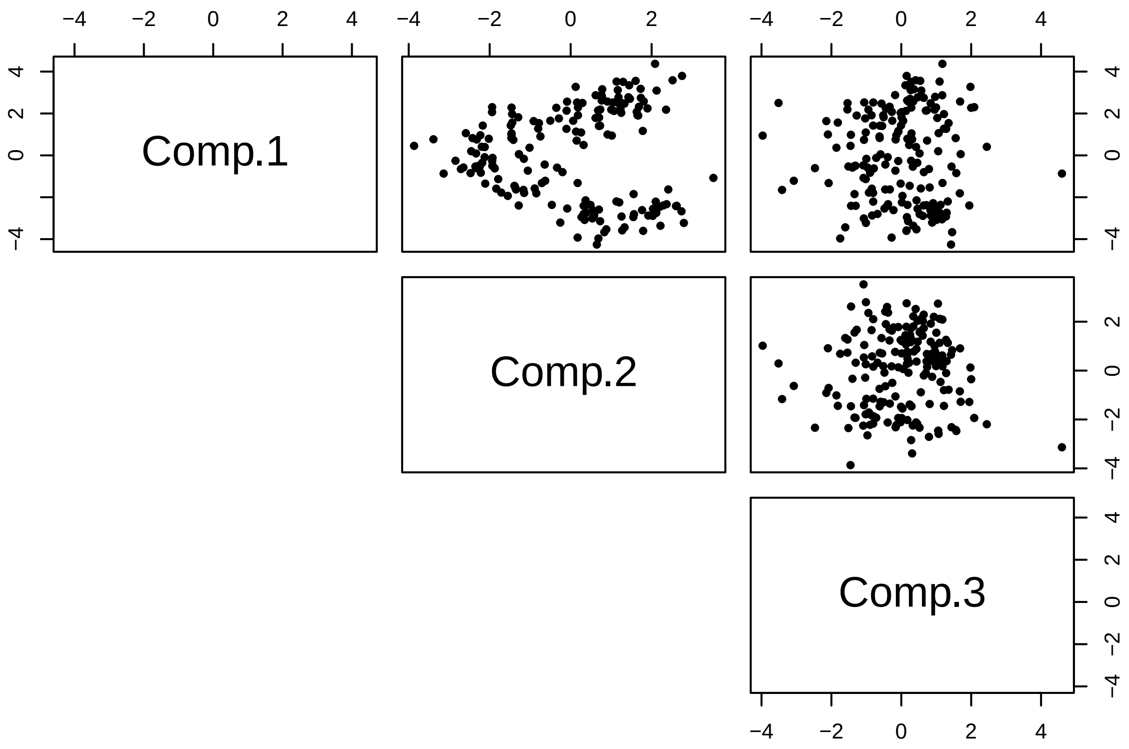
```
##                                   Comp.1       Comp.2      Comp.3       Comp.4
## Alcohol                       0.149122514  0.485555741  0.17407847  0.009290005
## Malic.acid                   -0.242463819  0.223443550 -0.08786790 -0.540664071
## Ash                          -0.009586478  0.328328599 -0.64011224  0.194647375
## Alcalinity.of.ash            -0.250346241 -0.008123314 -0.61163461 -0.083786308
## Magnesium                     0.138690160  0.299087400 -0.11171780  0.367268278
## Total.phenols                 0.391800881  0.063885614 -0.13714253 -0.205480320
## Flavanoids                    0.426727328 -0.005371811 -0.11224538 -0.157668691
## Nonflavanoid.phenols         -0.298883085  0.030423001 -0.15662737  0.183269298
## Proanthocyanins               0.309978301  0.037335041 -0.15316499 -0.399805096
## Colour.intensity             -0.087565645  0.524727427  0.17887557 -0.066407169
## Hue                           0.293857121 -0.275287273 -0.13045957  0.419157734
## OD280.OD315.of.diluted.wines  0.373092215 -0.164998633 -0.16727410 -0.188596177
## Proline                       0.287635764  0.363611710  0.09622339  0.227194542
##                                  Comp.5       Comp.6      Comp.7
## Alcohol                       0.26592996  0.206791287  0.06076710
## Malic.acid                   -0.05603543  0.516225841 -0.43803175
## Ash                           0.13900243  0.141761650  0.14255139
## Alcalinity.of.ash            -0.08370602 -0.125541975  0.30971341
## Magnesium                    -0.72229570  0.027466546 -0.32614912
## Total.phenols                 0.13876016 -0.085524119  0.02038041
## Flavanoids                    0.10673293 -0.006399587  0.04580641
## Nonflavanoid.phenols          0.50008515 -0.277102460 -0.60022016
## Proanthocyanins              -0.15844164 -0.566623853 -0.33126816
## Colour.intensity              0.07868683 -0.399170503  0.23274848
## Hue                           0.17838345  0.081005629 -0.22090863
## OD280.OD315.of.diluted.wines  0.09167382  0.267109729  0.02984656
## Proline                       0.16078018  0.108103550 -0.07375147
```

For the first principal component, we observe two sets of variables with significant weightings: one with positive values and another with negative values. Therefore, we interpret this component as the disparity between the average values of alcohol, magnesium, total phenols, flavanoids, proanthocyanins, hue, OD280/OD315 of diluted wines, and proline, and the average values of malic acid, alcalinity of ash, nonflavanoid phenols, and color intensity. This same approach can be applied to all other components that we have chosen to retain.

We are typically interested in visualizing the data in the new reduced space. The scores for the principal components on the original data are automatically generated by the **`princomp`** function. Therefore, we can create a pairs plot of the three principal components we have chosen to retain.

```r
# Extract the scores for the first three principal components
scores.wine <- wine.pca$scores[, 1:3]

# Create a pairs plot of the scores
# Use pch=20 to set the point shape to filled circles
# Set lower.panel=NULL to remove lower panels (we only want scatterplots in the upper triangle)
pairs(scores.wine, pch = 20, lower.panel = NULL)
```



Notably, we observe indications of potentially two or more distinct groups in the data, suggesting the presence of multiple clusters rather than a single homogeneous population. Additionally, there appear to be a couple of outliers. To refine our analysis, we could consider an iterative approach by identifying and removing these outliers from the data, then running PCA again. It's worth noting that PCA is often an iterative process, requiring multiple iterations to refine the analysis, rather than a one-time operation.