Module 4 Homework – PCA

Mike Hankinson

2021-10-19

# =====================================================  
# Module 4 Homework - Principal Component Analysis  
# Mike Hankinson  
# October 19, 2021  
# =====================================================  
  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# Assignment  
# 1. Perform Principal Component Analysis to analyze the exercise performance variables in the dataset. Include the following steps in your analysis:  
# a. Produce a correlation matrix of the variables.  
# b. Perform PCA and analyze the VAF by each factor. Include an elbow plot.  
# c. Interpret the first loading using loadings plot.  
# d. Use a bi plot of the first two factors and loadings and/or other visualizations to identify unique individuals.  
  
# 2. Discuss the following:  
 # . Do you think dimension reduction was helpful for this dataset?  
 # . Can you draw any connections between your results from clustering analysis and PCA?  
 # . What are your overall insights regarding these observations and variables   
# after performing multiple unsupervised modeling techniques?  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# Solution  
  
# Follow the Process:  
# 1. Load and Scale Data   
# 2. Perform PCA   
# 3. Create Covariance and Correlation Matrices  
# 4. Select the Number of Factors  
# 5. a. Plot Factors and Loadings  
# b. Extract Insights  
# 6. Bi plot of Loadings and Component scores  
# 7. Discussion

**# 1. Load and Scale Data**   
library(openxlsx)  
dat <- read.xlsx("Assignment\_3\_data.xlsx")  
dim(dat) # [1] 490 9

head(dat)

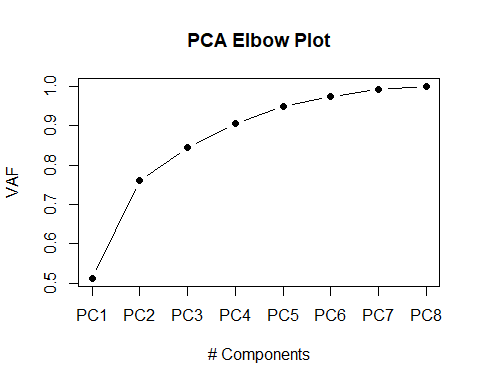
## Athlete Total.Distance Acceleration.Load Speed.Load Metabolic.Exertion  
## 1 1 2574.810 1377.210 1031.6881 365.8638  
## 2 2 2344.107 1281.303 646.6954 282.9880  
## 3 3 1974.224 1203.923 771.0174 286.4080  
## 4 4 2541.314 1279.254 1077.7205 408.4255  
## 5 5 1443.002 1131.579 441.9285 263.9749  
## 6 6 2288.777 1213.791 922.5933 374.6434  
## Sprints.per.Minute High.Accelerations.per.Minute Maximum.Speed  
## 1 5.703392 0.5190550 7.53  
## 2 6.333377 1.0775983 7.83  
## 3 6.358295 1.3840039 7.36  
## 4 5.923127 0.9097642 7.55  
## 5 6.115174 0.9939915 7.11  
## 6 6.548649 1.4467184 7.88  
## Maximum.Acceleration  
## 1 4.58  
## 2 5.85  
## 3 4.69  
## 4 4.76  
## 5 5.51  
## 6 4.71

pca.dat <-scale(dat[,2:9]) # Remove 1st column (athlete number) from data set and scale  
dim(pca.dat) # [1] 490 8

## [1] 490 8

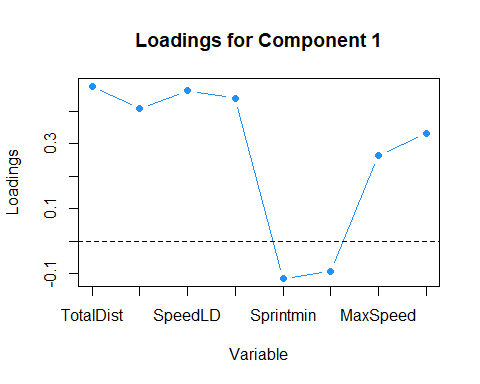
**# 2. Perform PCA**   
athlete.pca <- princomp(pca.dat)   
head(athlete.pca)

# Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6  
# 1 0.874306031 -1.722311753 -0.1637680374 0.591605013 -0.218925312 -0.2383615512  
# 2 0.247416237 1.189354395 -1.5105905447 -0.577412791 -0.866003169 -0.1176195863  
  
# Comp.7 Comp.8  
# 1 0.341293857 -1.762676e-01  
# 2 0.100999877 3.626711e-01  
  
  
  
**# 3. Create Covariance and Correlation Matrices**cumulative\_variation\_VAF <- cumsum(athlete.pca$sdev^2/sum(athlete.pca$sdev^2)) # VAF  
  
# Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8   
# 0.5121450 0.7611089 0.8438478 0.9048600 0.9494903 0.9735950 0.9919680 1.0000000   
  
  
**# 4. Select the Number of Factors**  
plot(cumulative\_variation\_VAF, type="b", pch=16, xlab="# Components", ylab="VAF", main="PCA Elbow Plot", xaxt="n")  
axis(1, at=1:8, labels=paste0("PC",1:8))



# Argument could be made for either 2 or 3 principal components  
# based upon breaking point within the elbow plot:   
# With 2 Principal Components:   
# - Dimension reduction from 8 to 2  
# - Still accounting for 76.1% of the variance  
# With 3 Principal Components:   
# - Dimension reduction from 8 to 3  
# - Still accounting for 84.4% of the variance  
  
# However, I would like to account for at least 90% variance within the data set.   
# Therefore, I will select 4 principal components resulting in the following:  
# With 5 Principal Components:   
# - Dimension reduction from 9 to 4  
# - Still accounting for 90.4% of the variance  
  
  
**# 5.a. Plot Factors and Loadings**

# The key to interpreting our principal components will often be found in the LOADINGS.   
# The LOADINGS (Eigen VALUES/VECTORS) are the values that explain how factors are connected to the original variables that we are attempting to analyze.  
# ORIGINAL DATA = FACTORS \* transpose[LOADINGS]  
  
# --Component 1 Loading Plot--  
plot(athlete.pca$loadings[,1], type="b", xaxt="n", col="dodgerblue", pch=16,   
 xlab="Variable", ylab="Loadings", main="Loadings for Component 1")  
axis(1,1:8, c("TotalDist", "AccelLD", "SpeedLD", "MetExert", "Sprintmin", "Accelmin", "MaxSpeed", "MaxAccel"))  
# axis(1,1:8, c("1", "2", "3", "4", "5", "6", "7", "8"))  
abline(h=0, lty=2) # abline() adds line to an existing graph



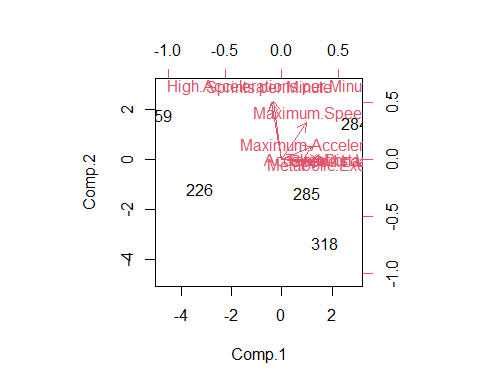
# Results of Component Loading Plot:  
# Component 1 is positively correlated with the following 6 variables:  
# 1. Total Distance  
# 2 Acceleration Load  
# 3. Speed Load  
# 4. Metabolic Exertion   
# 7. Maximum Speed  
# 8. Maximum Acceleration  
# Component 1 is negatively correlated with the following 2 variables:  
# 5. Sprints per Minute  
# 6. High Accelerations per Minute  
  
# An increase in the value of component 1 will   
# - increase variables 1, 2, 3, 4, 7 and 8.   
# - decrease variables 5 and 6.   
  
# This appears to be an ideal solution to improve a good bit of athlete performance.  
  
  
**# 5.b. Extract Insights**

# I want to study the following athletes:  
# - 5 athletes who have the HIGHEST MAXIMUM SPEED PCA SCORES  
  
  
# This sorts column 7 ascending  
five.athletes.to.study <- sort(athlete.pca$scores[,7])  
tail(five.athletes.to.study)

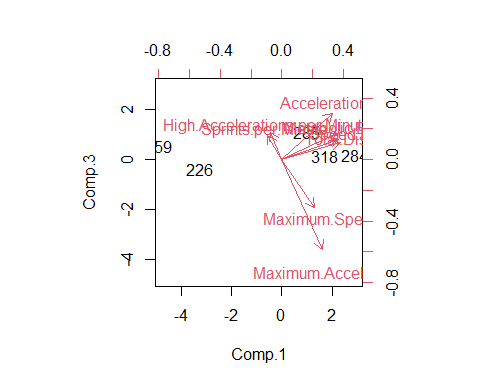
# 174 284 226 59 318 285   
# 0.9329706 0.9434466 0.9511316 0.9735648 0.9804337 1.0266568   
  
five.highest.speed <- sort(athlete.pca$scores[,7], index.return=T)$ix[486:490]   
dat[five.highest.speed,]

# Athlete Total.Distance Acceleration.Load Speed.Load  
# 174 174 2651.1492 1068.3449 1058.1902  
# 284 284 3761.1394 1713.9127 1499.6586  
# 226 226 914.0313 773.2589 407.6776  
# 59 59 343.4050 559.8534 260.9583  
# 318 318 3288.2450 1379.6117 1356.6456  
# 285 285 3493.6994 1705.5140 1086.7038  
# Metabolic.Exertion Sprints.per.Minute  
# 174 339.74392 6.045984  
# 284 432.43867 6.576628  
# 226 80.09827 5.957809  
# 59 40.07995 6.350976  
# 318 394.92513 5.283015  
# 285 299.38815 5.711018  
# High.Accelerations.per.Minute Maximum.Speed  
# 174 0.9490760 7.03  
# 284 1.0282222 7.95  
# 226 0.7576669 7.03  
# 59 1.8850593 6.88  
# 318 0.2718949 7.23  
# 285 0.7999550 7.37  
  
  
**# 6. Bi Plot of Loadings and Component Scores**

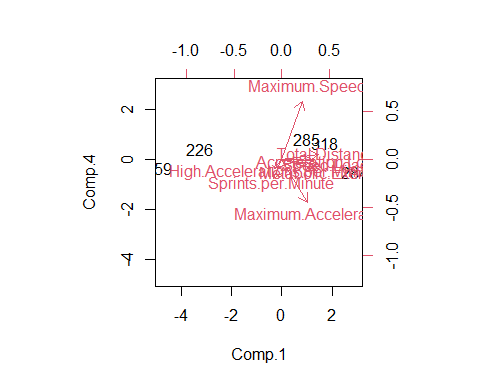
# Create bi plots containing the five athletes we decided to study.   
  
  
# Generating a bi plot of loading values for each variable and factor scores   
# for observations allows to directly evaluate PCA results, and to understand   
# differences between particular athletes.  
  
  
# By looking at the left (vertical) and bottom (horizontal) axes,   
# we can see the component scores for athletes, represented within the graph by   
# bold black numbers.  
# By looking at the right (vertical) and top (horizontal) axes,   
# we can observe the loadings components we are using.   
# for each variable, represented within the graph by bold red text and arrows.  
  
  
# biplot for 10 athletes for components 1 and 2  
biplot(athlete.pca$scores[five.highest.speed, c(1,2)], athlete.pca$loadings[,c(1,2)])



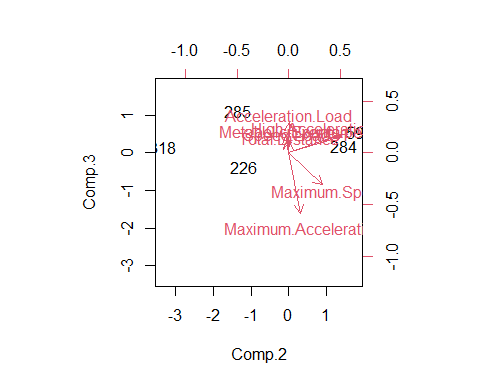
# biplot for 10 athletes for components 1 and 3  
biplot(athlete.pca$scores[five.highest.speed, c(1,3)], athlete.pca$loadings[,c(1,3)])



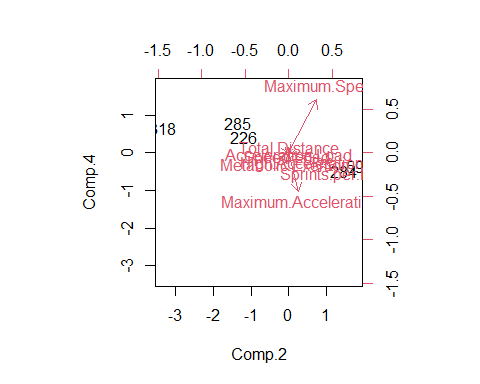
# biplot for 10 athletes for components 1 and 4  
biplot(athlete.pca$scores[five.highest.speed, c(1,4)], athlete.pca$loadings[,c(1,4)])



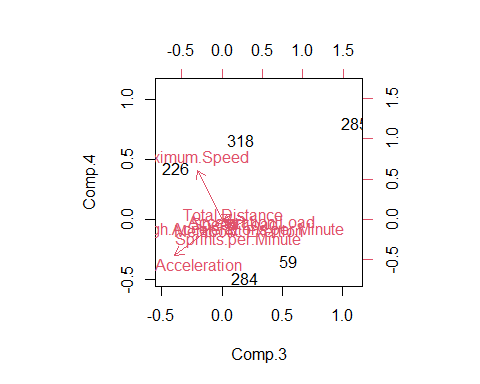
# biplot for 10 athletes for components 2 and 3  
biplot(athlete.pca$scores[five.highest.speed, c(2,3)], athlete.pca$loadings[,c(2,3)])



# biplot for 10 athletes for components 2 and 4  
biplot(athlete.pca$scores[five.highest.speed, c(2,4)], athlete.pca$loadings[,c(2,4)])



# biplot for 10 athletes for components 3 and 4  
biplot(athlete.pca$scores[five.highest.speed, c(3,4)], athlete.pca$loadings[,c(3,4)])



**7. Discussion**

In Modules 3 and 4, we worked on a provided data set containing results on 490 athletes. The data set contained the following 8 variables:

1. Total Distance
2. Acceleration Load
3. Speed Load
4. Metabolic Exertion
5. Sprints per Minute
6. High Accelerations per Minute
7. Maximum Speed
8. Maximum Acceleration

Analysis of this data falls under the umbrella of unsupervised learning.

In Module 3, we utilized both K-Means and Hierarchical clustering techniques to group similar athletes into "buckets". The algorithms for each were similar and are presented below:

\*\*\*K-Means Algorithm Methodology\*\*\*

1. Scale variables (0-1) to avoid bias created by variables with different scales.

2. Pick k number of clusters and randomly divide observations into those k groups.

3. Calculate the centroid for each cluster.

4. Calculate the Euclidean distance between every observation and every centroid.

Assign each observation to the cluster which has the closest centroid.

5. Iterate until the clusters stop changing.

6. Determine number of clusters to use by incorporating the "Elbow" plot.

7. Return centroid values to original scale for better result interpretation.

8. Plot cluster assignments for visual representation

9. Interpret results.

\*\*\*Hierarchical Algorithm Methodology\*\*\*

1. Determine linking method (single, average, complete or centroid)

2. Scale variables

3. Determine distances between ALL observations (Use a nx-n Dissimilarity Matrix)

4. Visualize the dendogram

5. Identify two most similar objects

6. Identify potential cutpoint(s)

7. Perform the cut and save the clusters

8. Interpret results

Using these techniques, we were able to find commonalities between the athletes based upon performance within the 8 features.

However, a problem occurs with clustering techniques when the data contains a high number of features. With high dimensionality, it is difficult to effectively differentiate the distances (between points--athletes) in high dimensional space.

In order to alleviate this concern, we learned Principal Component Analysis (PCA) technique in Module 4. PCA is used to alleviate the clutter in dimensional space. This technique allows for the reduction in features by "converting a set of possibly correlated values into a set of linearly uncorrelated factors"\*.

The process for PCA is fairly straightforward, however, interpretation is a bit tougher.

\*\*\*Principal Component Analysis Algorithm\*\*\*

1. Load and Scale Data

2. Perform PCA

3. Create Covariance and Correlation Matrices

4. Select the Number of Factors

5. a. Plotting Factors and Loadings

b. Extract Insights

6. Bi plot of Loadings and Component scores

I wonder if approaching the problem in reverse order would be beneficial. That is, perform PCA first followed by clustering. In this way, we could alleviate the dimensionality problem up front then look for commonalities through clustering.

\* Quote from Module 4 Course Material.

#Some Functions Used

# - var(),cor() - var, cov and cor compute the variance of x and the covariance or   
# correlation of x and y if these are vectors. If x and y are matrices  
# then the covariances (or correlations) between the columns of x and   
# the columns of y are computed   
  
# ?princomp() - princomp performs a principal components analysis on the given numeric   
# data matrix and returns the results as an object of class princomp.  
  
# cumsum() - Cumulative Sums  
# Returns a vector whose elements are the cumulative sums,   
# products, minima or maxima of the elements of the argument.  
  
# %\*% - Matrix Multiplication  
  
# t() - Matrix Transpose  
# abline() # Add Straight Lines to a current Plot  
  
# c() Combine Values into a Vector or List  
  
  
  
# rm(list = ls()) Removes global environment