University of Cape Town

STA5069Z

An Exploration of Multivariate Analysis Techniques Employed to Analyze Football Players' Attributes, Market Value and Rating

Author:
Edward Baleni
Thabo Dube

Student Number:
BLNEDW003
DBXTHA030

May 17, 2023

Contents

Abstract

The use of multivariate analysis (MVA) techniques have not been exhaustively used in the domain of football. In more recent publications, the use of clustering, factor analysis and optimisation have been used in the domain of football. The literature often does not extend much further than this at the mention of multivariate analysis techniques. This has introduced one fundamental idea behind this study, which is to explore the potential benefits of interlacing these techniques in the analysis of football related data. Another crucial point is the ability to objectively assess the market value and rating of football players. Market value in prior years has been evaluated by professionals and more recently by crowdsourcing, which are both very stringent approaches to the evaluation of market value, but are also two very subjective views, making them both irreproducible. A major aim of this study is to use data driven methods to invoke reproducibility of these measures and to suggest ways in which to ultimately remove the bias in this prediction of market value. Player rating is also a very important factor as it interlinks very closely to what a players' true market value should be. This study makes use of principle component analysis and robust principle component analysis to reduce the dimensionality of the data; following this, hierarchical clustering and k-means clustering are use to examine the behaviour of the observations; biclustering, a method, which has not been used in football before is employed to assess modules present in the data; finally, methods of prediction are utilised to reproduce the valuations made by crowdsourcing and FIFA.

1 Introduction

Football is a team sport that involves a team of 11 players kicking a ball towards a goal while being opposed by another team. Professional football, is a career in which people are paid to take part in this sport. A club is an organization composed of players, individuals who manage players and those who manage the organization. In the world of professional football, team owners and managers are always making important decisions about the future of the club. Among these decisions are those concerning both the current players in the club and those they wish to transfer in the near future. For this process to take place, they must be able to determine the market value of these players using all the information available about them. In the late 90s and early 2000s the primary method of player valuation was conducted by football experts and team management. Since the start of the 4th industrial revolution, the increase in inter-connectivity meant more people have the means of keeping up to date on football from anywhere in the world. As a result, a crowdsourcing approach has become the most popular method of player valuation. The leading website on all football related statistics, news, transfer rumours as well as the market value of all players in most of the top leagues in the world is Transfermarkt, www.transfermarkt.com. Founded in Germany by Matthias Seide in 2000, the main aim of the website is to continuously track players and transfer targets. It allows users to create accounts and add their opinions on players to the data base in terms of their performance and their estimated market value. The final market value for each player is a result of the analysis of each users opinion. This paper will go over the limitations of this crowdsourcing method and investigate how using a machine learning based approach can identify these limitations and provide a guideline on how one may go about overcoming them in determining a players market value and FIFA rating, using indicators from the 21/22 football season.

2 Literature Review

2.1 Market Value

Football clubs are businesses where each player is an asset or liability (an asset if they perform well, a liability if they are struggling), professional football clubs take part in the purchasing or selling of player contracts to improve club performance. This purchasing and selling is called a transfer and is a process that occurs between clubs. The transfer window is a period that occurs twice in a football season that permits clubs to perform these transfers; this window occurs in the middle of a season and at the end of a season. As players are to be transferred between clubs, the sale or purchase of each player is decided by a negotiated transfer fee. It is important that the club does not overestimate the value of players that they are keen to purchase, or underestimate players they mean to sell; either of these mistakes could result in great losses to the club. As the transfer fee is negotiated by both parties the transfer fee is variable. The transfer fee is the physical price that the player will cost to transfer from one club to the next. It is essential for clubs to have a gauge on on this transfer fee before negotiations; market value is one such estimate of transfer fee.

Historically, market value has been determined by experts. These being the club itself or sports journalists. Crowdsourcing, is a more recently used method in determining market value. It is an open forum approach to evaluating this market value, where any person can log-in and give their opinion biased or otherwise on what they presume a player is worth. In particular the crowd-based approach on websites like Transfermarkt use a hierarchical structure to classify the importance of the input of each user, i.e. notable users such as football experts and critics have opinions that carry more weight and are ultimately the decider on the market value of a player (Müller et al., 2017). These members have the task of going through all the information in the database , filtering and weighing the users' input using their own discretion, as well as the users' reasoning for their chosen estimates (Herm et al., 2014). The system works this way so as to counteract the input from users who are trying to manipulate the system due to personal interests or from users who simply lack the proper knowledge to make informed decisions about the football players (Herm et al., 2014).

2.2 Indicators of Market Value

There is an array of variables that are used as indicators player market value. These can generally be broken down into characteristics, skills and performance. Characteristics are the attributes like the age, which is seen as one of the most important indicators as it signifies both experience and potential for the club (Müller et al., 2017), as well as height, which correlates to a players heading ability and that increases their odds of scoring and/or preventing aerial goals (Fry et al., 2014), along with footedness, which represents the talent and flexibility of football players as they can be used in different positions on opposite sides of the pitch, and this positively impacts their market value (Herm et al., 2014). Player performance is comprised of statistical indicators of how well the players did in all the games they played. The number of matches a player has taken part in, whether it be by starting the game or coming in as a substitution, has been considered important in previous research.(Müller et al., 2017) The statistic not only shows how prone a player is to injury but their ability perform at the same level consistently. Goals and assists are the main indicators used in most research as they are an unambiguous measure of goal contributions which positively affects their market value. Skills are the players ability to perform different actions in the game such as shooting, defending and ball control and these are all given on a scale of 0-100. While there are some general indicators that apply to every single player, due to the nature of the game, not every position in the team requires the same set of skills.(Behravan and Razavi, 2021)

Strikers, whose main objective is to score goals, possess a different set of skills from defenders, whose objective is to ensure that no goals are conceded.

3 Data

The data used in this paper is an amalgamation of market value determinants, market values and FIFA ratings from the 2021/2022 football season. These were sourced from www.transfermarkt.co.uk, www.fbref.com and www.kaggle.com/datasets/stefanoleone992/fifa-22-complete-player-dataset. The scope of the dataset is limited to the top 5 European leagues: Bundesliga (Germany), Premier League (England), La Liga (Spain), Ligue 1 (France) and Serie A (Italy). The dataset only considered matches played domestically (i.e only games between teams in the same league, which amounts to 38 matches). The dataset consists of 327 observations of outfield players that were transferred at the end of the 2021/2022 season. Goalkeepers were excluded from the data as their value and performance are measured in a different standard in comparison to infield players. The vast number of player positions have been simplified into 3 main positions: defenders (DF), midfielders (MF) and strikers (FW). Figure 1 shows player distribution by position and by league. By position, the classes are almost balanced, with each position representing at least 30% of the data. All the leagues make up at least 20% of the dataset except for La liga which only makes up 9% of the players.

(a) Player position

(b) League

Figure 1: Player distribution by position and league

Figure 6 shows how the players in the dataset are distributed according to age, rating and market value. The age of the players are positively skewed, indicating that most of the players are younger than the average. This is expected as there has been a recent focus on utilising young players due to the longevity they provide (Leitch, 2019). The FIFA player ratings are calculated by combining each players' skill ratings and calculating one overall rating. The distribution is very close to normal, this observation is helpful in determining the distribution of link function when the multivariate general linear model is to be implemented. Finally, we observe that the distribution of the market values to be exponential, this may be due to presence of such great outliers, but also may be the case if the dataset was larger.

This section is simply a precursor to the study to gain some understanding of the data being used.

4 Methodology and Results

4.1 Linear Dimensionality Reduction

4.1.1 Principal Component Analysis (PCA)

Dimension reduction is a method utilised to move from a high-dimensional plane of data to a lower-dimensional subspace (Izenman, 2008). PCA reduces dimensionality by transforming potentially correlated variables of a matrix into a number of linearly uncorrelated principle components (PCs) while maintaining as much information as possible. A short derivation on how PCA is performed can be found in subsection B.1.

The scree plot, as shown in Figure 9, is used to select the number of PCs. Figure 9 has an elbow around the 4th, 6th and 10th PC, however, upon closer inspection it can be shown that the amount of explained variation at these respective points are given as 44%, 55% and 73%. A satisfactory cut-off of explained variance has been debated to be above 70% (Jolliffe and Cadima, 2016), which would that 10 PCs is suitable. Other studies suggest a cut-off above 90% if there is no obvious elbow (Izenman, 2008), which relates to the first 16 PCs. This PCA has reduced the number of dimensions but still leaves a substantial number of variables that explain a great deal of information. In most applications this is not ideal in our analysis.

This spread of variability can be explained by various attributes of the original data. The first being that PCA is sensitive to outliers (Hubert et al., 2005). Figure 7 illustrates the presence of outliers for almost every variable in the predictor space. This is problematic. The solution to this is to use robust principle component analysis.
PCA is a linear technique so it is clear that problems would arise if the lower-dimensional subspace of the data is non-linear, this is another potential reason for poor dimensionality reduction (Izenman, 2008). To remedy this one would use non-linear dimensionality reduction.
It is possible that PCA is a poor method to utilise in conjunction with the data present, in such a case, a different method may be able to better reduce the dimensionality of our data.

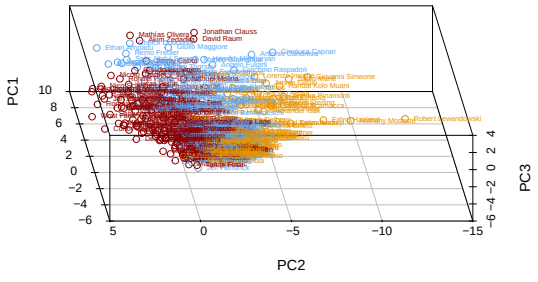


Figure 2: First 3 principle components

In Figure 2 the PC scores are separated into orange (FW), blue (MF) and red (DF). Figure 2 shows some degree of separation between the players. This separation in the first two PCs illustrates that certain features are able to characterize player positioning. As the loadings capture the correlation of the features on the principle components, it is possible to observe the relationship between the

features and the players present through a biplot; as a part of our analysis this may be able to inform our assessment of player roles. However, as the PCA does not reduce well, this will not be done at this stage.

4.1.2 Robust Principle Component Analysis (RPCA)

Outliers heavily weight on the information explained by the covariance matrix. PCA is done using this covariance matrix. Since this covariance matrix is sensitive to outliers, it would seem that the first few, most important, PCs will also be attracted towards these outliers, which would make it such that the PCs do not capture the true underlying structure of the observations creating a reliability issue. The Robust PCA (RPCA) is one way to combat outliers. There are a number of RPCA methods that could be used, this study will focus on the method detailed by Hubert et al. (2005). This method is a combination of two known robust methods developed by Croux and Ruiz-Gazen (1996) and Li and Chen (1985) who both focused on "projection pursuit" (PP) techniques to robust PCA, and the second method is developed by Croux and Haesbroeck (1999), Davies (1987), Wainer (1988) and Rousseeuw (1984), this second method is an older idea that replaces the covariance matrix with a robust covariance matrix. Hubert et al. (2005) combines the two by employing PP to reduce dimensionality followed by implementing a number of concepts based on minimum covariance determinent (i.e. a method to estimate the robust covariance). How RPCA is conducted is defined in subsection B.2

The RPCA was conducted with the "PcaHubert" function in the package by Todorov and Filzmoser (2009). Figure 10, is a scree plot demonstrating the proportion of explained variation. Here it is shown that the first 3 PCs are sufficient to explain majority of the variation in the data without losing too much information. This is a much improved number from the the classical PCA's 10 PCs as shown in Figure 9.

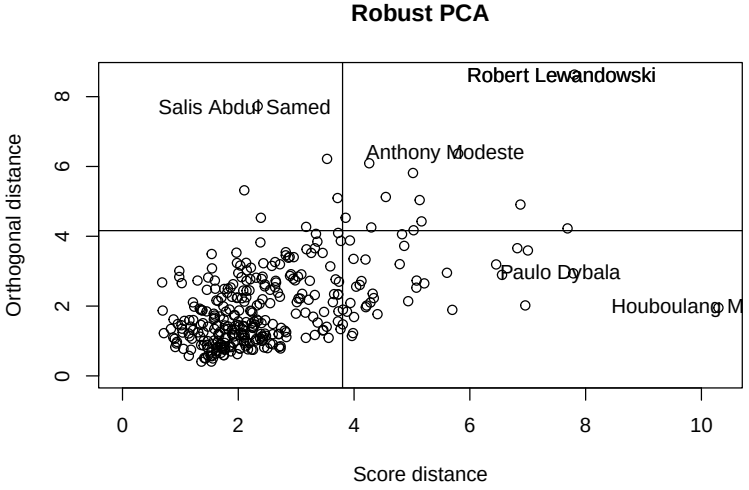


Figure 3: Outlier map of robust principle component analysis

Figure 3, depicts 4 quadrants indicating to a level of outlyiness of the data (Rousseeuw and Hubert, 2018). The bottom left quadrant corresponds to players who do not deviate, regular observations.

In the top left and bottom right portions of the map there exists good leverage points, or orthogonal leverage points; more specifically the top left seems to be pointing towards some players who may be talents, most of these players happen to be young but also happen to have good performance measures for their positions, these are the players in Table 3. The talents seem to often have ratings or valuations that do not reflect their true value and are relatively young (below 30). The top right quadrant corresponds to bad outliers, these happen to correspond to players with incredible player performance.

Table 2, gives an indication of some of the more outstanding players as shown in the top right quadrant of Figure 3. These are players who all seem to have performed well in their position in the 2021/2022 season and should be considered by teams looking to purchase players. The principle component scores for both the classical and robust PCA approaches are shown in Figure 11 and Figure 12. In both cases it can be seen that players in the same positions tend to neighbour each other, indicating that these players should have similar scores.

Figure 4, manages to illustrate which of the original variables characterise which position. As noted in the classical PCA section above, orange denotes forwards, blue denotes midfielders and red denotes defenders. Forwards are characterised by variables such as: Offsides, age, SOT%, Goals, Shots on Target, Goals+Assists, Assists, Key Passes, Fouls Drawn and Total Shots; they also seem to have a negative relationship with pass rate. Midfielders have a strong relationship with: Assists, Key Passes, Fouls Drawn, Fouls Committed, Matches Played, Passes Blocked, Yellow Cards, Tackles Won, Tackles Made, Passes Made, Blocks, Passes Attempted and Ball Recovery. Finally, it can be seen that defenders have a strong relationship with: Yellow Cards, Tackles Won, Tackles Made, Passes Made, Blocks, Passes Attempted, Ball Recovery, Clearances, Red Cards, Defensive Errors, Interceptions, Shots Blocked and Pass Rate; they also seem to have negative relationship with age.

In Figure 4, the performance measures that are strongly related to the defensive players are almost orthogonal to the performance measures that are related to the forward players. This does make sense since within a team the two roles do not often perform any of the same tasks. This goes to show that a defensive player's performance cannot be ranked on the same metric as a forward player. Midfield players happens to be right in the middle of these two. How well these types of players are performing should be slightly more difficult to characterise as they have some similar attributes to forwards and to defenders, this is easily seen in how their strong performance metrics are related in some way to both forward player performance metrics and defensive metrics.

Figure 4, also illustrates that as the scores of the first PC increase, then we will likely get players with better perfomance. The direction of the arrows of the loadings all seem to suggest an increase in each statistic of the original dataset as you move to the right. The 2nd PC illustrates the positioning of the player. If the 2nd PC value is around 0, one can expect to observe a midfielder, as it increases in value a defender is expected and as it decreases in value a striker will likely be found.

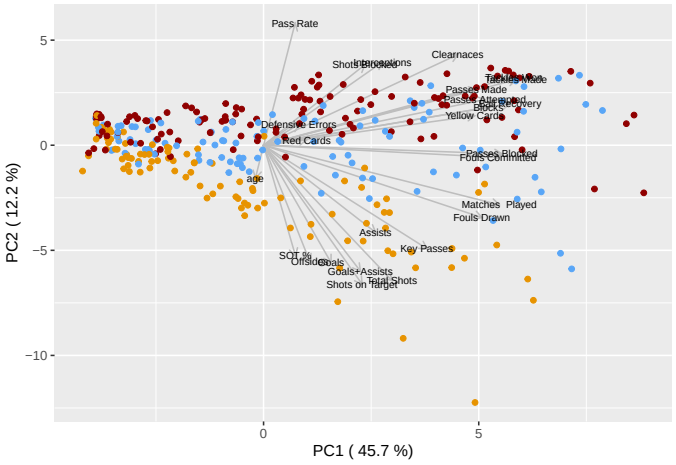


Figure 4: Robust principle component biplot

Finally, it can be seen that age, Red Cards and Defensive Errors are 3 variables that have relatively weak loadings in comparison to the rest of the data, these points are weak enough that one might expect them to disappear when performing a robust sparse principle component analysis. These particular points do not provide much information to the structure of the data relative to the the other features.

clustering these players may be able to distinguish the players by performance. It may also be able to better illustrate some of the suggestions made in accordance with the outliers as in Table 2 and Table 3. Following this, biclustering may be able to show specifically which players are strong or weak in their field, or similar results as is to be expected from the clustering. This would be informative to a team in identifying potentially overlooked and undervalued players that don't have as much fame as some other more notable individuals.

4.2 Non-Linear Dimensionality Reduction

Linear dimensionality reduction techniques are typically suited for finding the low-dimensional structure of data when this lies in a manifold. This has been explored above, however, these findings may be incorrect if the structure of the lower-dimension is indeed non-linear. When linear methods present inadequate findings, non-linear methods can be considered to identify the true structure of the manifold.

It is impossible to identify the low-dimensional structure hidden in the data at higher-dimensions (Izenman, 2008). To explore this both linear and non-linear dimensionality reduction techniques can be used to discover whether the low-dimensional structure is a manifold or a non-linear manifold.

The exploration of non-linear dimensionality techniques have not been explored in the field of football and should be considered as a possibility. Both KPCA and t-SNE are two of these techniques that can be used to assess if there is indeed a non-linear manifold present.

4.2.1 Kernal Principle Component Analysis (KPCA)

Kernal principle component analysis is an extension of PCA, to accomadate non-linear manifolds. KPCA maps the feature space of the data to a higher dimensional feature space, this is done using the kernal trick, a classical PCA is then performed at this stage to find PCs, having found these PCs a transformation back to low-dimensional space is done. As discussed above, the classical PCA is unable to adjust for outliers, this is a flaw that is present in both classical PCA and by consequence in KPCA (Nguyen and Torre, 2008; Huang et al., 2009). For this reason, KPCA is not explained in depth as both this and classical PCA happen to be poor methods to use in relation to football data. However, it is still possible to compare the KPCA to the PCA to assess whether the low-dimensional data is indeed in a non-linear manifold.

The KPCA was performed using the function "kpca" in the library by Karatzoglou et al. (2023). The kernals that worked best with the football data were the gaussian and the laplace, which both happened to produce the same output as in Figure 13. If the KPCA does not give a better representation of the data than the PCA then this would suggest that the low-dimensional representation of the data exists in a manifold not a non-linear manifold. Figure 13, represents the data in a very similar way to how it has been represented by the first PCs shown in Figure 2. This may be an indication that the lower-dimension is linear, and that non-linearity may not something to consider with football data. This can be further explored by the t-distributed stochastic neighbor embedding (t-SNE) below. Unlike KPCA, t-SNE is able to correct for outliers (Li et al., 2017).

4.2.2 t-Distributed Stochastic Neighbor Embedding (t-SNE)

The t-SNE, a modified approach of stochastic neighbourhood embedding(SNE), is a technique used to visualise high dimensional data that lies on multiple low dimensional manifolds that are related (Van der Maaten and Hinton, 2008). The difference between the t-SNE and the SNE is the ease at which the cost function can be optimised and the ability of the t-SNE to solve the crowding problem, which is the SNE's tendency to crowd datapoints in the middle of the map. The first step in SNE is converting the Euclidean distances between observations to conditional probabilities that quantify their similarity. Where datapoints are close to each other the probability will be high , but for data points which are more spaced out from each other their probability will be very small. Similar conditional probabilities are then computed for the low dimensional datapoints and both these probabilities are used in the cost function. The aim is to minimise the difference between the two probabilities calculated. The cost function follows a Kullback- Liebler divergence with a gradient descent method.The full SNE technique can be found in Hinton and Roweis (2002)

Rather than using conditional probabilities, The t-SNE uses joint probability distribution P, for the high dimensional space, and Q, for the low dimensional space, to minimise a single Kullback-Liebler divergence. The Cost function is then given as:

$$C = \sum_i KL(P||Q) = \sum_i \sum_j p_{ji} log \frac{p_{ji}}{q_{ji}},$$

where p_{ij} and q_{ij} are the pairwise similarities for the high and low dimension space. The probabilities for the high dimensional space are given as follows :

$$p_{ij} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq l} exp(-||x_k - x_l||^2/2\sigma_i^2)},$$

where σ_i is the variance of the Gaussian with the centre x_i. Cases where $j = i$ are given a probability of zero by virtue of the interest in this technique being to model pairwise similarities (Van der Maaten and Hinton, 2008). The pairwise similarities for the low dimensional is given as:

$$q_{ij} = \frac{(1 + (||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}},$$

where the variance σ_i is set to $\frac{1}{\sqrt{2}}$ in the low dimension. Just as with the the high dimensional data, the low dimensional instance where $j = i$ is given a probability of zero. In this dimension, the probability distribution used is a student t-distribution with 1 degree of freedom whereas in the high dimension space a Gaussian probability was used. The use of a distribution with a heavier tail in the low dimension allows for medium distances in the high dimensional space to be modelled more accurately by removing undesired forces between map points that signify moderately dissimilar data points(Van der Maaten and Hinton, 2008). When there are outliers present in the high dimensional data, the values of the joint probabilities are small, which results in the points not being mapped correctly. This issue is dealt with by defining the joint probabilities of the high dimensional data using conditional probabilities as $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$.(Van der Maaten and Hinton, 2008) This method makes sure that the sum of the high dimension joint probabilities for all x_i is greater than $\frac{1}{2n}$, resulting in each data point making a significant contribution to the cost function. The gradient of the cost function is then given as:

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(1 + ||y_i - y_j||^2)^{-1}(y_i - y_j).$$

Figure 14, Figure 15, Figure 16 and Figure 17, illustrate that t-SNE gives an inadequate representation of the data compared to the methods mentioned prior. The best separation of data, as shown by Figure 17, is again very similar to Figure 2, which once again would suggest that the manifold is linear. At this stage linear dimensionality reduction techniques work a lot better with football data than non-linear dimensionality techniques.

4.3 Cluster Analysis

Cluster Analysis is a popular technique of unsupervised learning. The technique consists of various methods which all use different algorithms to sort observations into separate natural subgroups (Izenman, 2008). The difference in the clustering algorithms presents an opportunity for these techniques to produce different clustering results. Ideally, these clusters should be far enough from each other that their difference is not mistaken, but there is no guarantee that more than one cluster can even be found.

4.3.1 K-Means Clustering

K-means is a non hierarchical clustering approach where data is split into a pre-determined number of clusters (Izenman, 2008). The objective is to ensure to maximise the similarity between observations within the same cluster while also ensuring that items from different clusters are dissimilar. The algorithm searches for solutions that produce the lowest clustering error called the Euclidean Sum of Squares(ESS) which is given as follows:

$$ESS = \sum_{k=1}^{K} \sum_{c(i)=k} (x_i - \bar{x}_k)^{\tau}(x_i - \bar{x}_k),$$

where $\bar{x}_k$ is the centroid of the kth cluster and $c(i)$ is the cluster containing x_i (Izenman, 2008). The cluster centres are initially placed in arbitrary positions and iteratively reassign items to clusters if they reduce the ESS. The algorithm stops when observations no longer change clusters and the ESS can not be reduced further. Figure 18 uses 3 approaches to determine the optimal number of clusters to set for the k-means algorithm.

The silhouette approach takes the average silhouette width for each cluster at different specifications of the number of clusters. A large average silhouette width is ideal as it represents a strong structure that has been found within the data, and it can be concluded that the data is well clustered(Rousseeuw, 1987). Figure 18 shows that only 3 clusters are necessary for the K-means approach, with the highest average silhouette width of approximately 0.4, which signifies that the structure of the cluster is substantial but weak. The total within sum of squares (TWSS) is the sum of squared deviations from each observation to the centroid of its allocated cluster and it measures the variability of observations in each cluster. The lower the TWSS the better. As the number of cluster increases, the TWSS decreases. However, it is common practice to select the no. of clusters at the elbow of the plot because after this point the reduction in the TWSS becomes marginally insignificant . Figure 18 shows that, like the silhouette approach, the TWSS also recommends a 3 clusters. The final statistic used to evaluate the number of clusters that should be used is the Gap Statistic, this is a very popular and well known strategy. This method also indicates that 3 clusters are ideal. The results of the k-means clustering based on 3 clusters is shown in Figure 5.

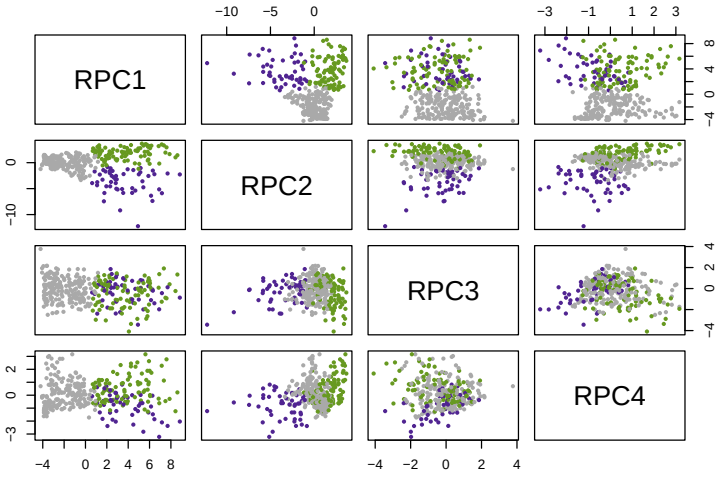


Figure 5: K-means clustering with robust principle component Scores

The first two PCs are able to capture the most information of the data. As previously mentioned RPCA1 shows the player talent/skill with the more talented players having higher scores and the less talented players having lower scores. RPCA2 provides a gradient of player positional ability. A high factor score means the player is good in terms of defence and a low factor score means the player is better in an attacking position. RPCA3 and RPCA4 do not provide much information. This will inform the interpretation of Figure 5 given in the list below:

- RPCA1 vs RPCA2 - The k-means clustering is able to separate the observations into players

who have quite poor performance in grey, players who are defensively strong in green and players who are attacking strong in purple. If the grey cluster is to be examined more closely, one could be able to identify players who are over-valued and over-rated. This understanding is very informative to those who buy and sell players as some players may be priced biasedly based on fame and not performance, such an analysis is capable of highlighting these type of players. If the green and purple clusters are to be examined closely, it is easy to identify the undervalued players and talents in each grouping, by searching for low-ranking, low-priced and young individuals who have been under-valued and under-rated.

- RPCA1 vs RPCA3 - The separation of good and bad players is still pronounced here, however, distinguishing between attacking strong and defensively strong is not what RPCA3 does.
- RPCA2 vs RPCA3 and RPCA2 vs RPCA4 - The separation of players by position is maintained, which goes to show that RPCA2 is indeed robust at separating players by position.
- RPCA3 vs RPCA4 - There is no clear separation in the observations. These PC scores are also difficult to interpret, so it is not obvious what can be said about them. Both PCs have extremely low values of explained variation, this may demonstrate that the information displayed between these two PCs is something close to white noise.

4.3.2 Hierarchical Clustering

This report explores the agglomerative hierachical clustering method, which starts with each observation in its own cluster then iteratively combines the observations until only one cluster remains (Izenman, 2008). Hierarchical clustering is based on the distance between each observation known as the dissimilarity. The definition of distance used is the Euclidean distance which is defined as follows:

$$d(x_i, x_j) = \left(\sum_{k=1}^{r} (x_{ik} - x_{jk})^2 \right)^{\frac{1}{2}}$$

where i and j are the observations and k is the variable (Izenman, 2008). Upon finding the dissimilarity matrix the algorithm begins by clustering two items that are the closest to each other. A new dissimilarity matrix is created because of the new distances between the created cluster and every other cluster. The distance used depends on the linkage. There is single linkage which uses the minimum distance, complete linkage uses the maximum distance and average linkage which uses the average distance (Izenman, 2008). The process is repeated until one cluster is left. Figure 19 is a dendrogram that shows the result of the clustering. A specific number of clusters can be formed by cutting across the vertical lines of the dendrogram at a suitable height. Items that are similar to each other are combined at low heights, whereas items that are more dissimilar are combined higher up the dendrogram (Izenman, 2008). Therefore, the best height to cut this dendogram is where the longest because it means the items between clusters are the least similar to each other. Figure 19 shows the ideal height to cut the dendrogram as approximately 13. As this horizontal cut will intersect 3 vertical lines it means the solution will have 3 clusters and all the observations below each line will be the members of that cluster.

Another appropriate way to find the number of clusters is to use the same 3 strategies as proposed in the K-means section. Figure 20, illustrates that 3 clusters is a good number of clusters to have once again.

Figure 21 illustrates the clustering. It results in the same types of clusters as were found in the k-means clustering, however, there are some minute differences. There seem to be a lot less attacking strong players accounted for here than there were in Figure 5; it has been previously stated that what makes a MF good is complicated and that they have many defensive and attacking performance metrics. This clustering may be subsuming this ambiguous group into the defensively strong players. Essentially, allowing this clustering to show strictly attacking players who have performed well, other players who have performed well and bad players. This result has some benefits, teams that are short of attacking players can find specifically that and cut out everyone else, it essentially highlights the FW role.

4.3.3 Biclustering

Biclustering is data analysis tool to investigate local structures in data. It is a method to simulataneously cluster variables and observations together resulting in a number of sub-matrices called biclusters Izenman (2008). There are many biclustering techniques that can be used, but specifically for this football data, through trial and error, only the Plaid model, iteratve signature algorithm (ISA) and fable methods were capable of clustering well. At this stage only the Plaid and ISA methods will be considered, as factor analysis has not been considered in this study.

As mentioned in the RPCA, certain metrics are able to characterise the performance of players in each of the three positions. While using plaid, it was illustrated that defenders were being clustered together with forwards by performance measures that characterise an attacking position and vice versa. This was not ideal as a defender/forward's performance cannot be measured on attacking/defending indices. However, when the ISA was used, forwards were clustered by attacking indices only, defenders by defensive indices and midfielders by a mixture of other metrics. This is ideal as it allows one to check the within class performance, which is more informative than looking at between class performance for a number of random indices. The plaid model also used weak indicators that demonstrated low to no importance in the RPCA, indices such as: age, SOT%, red cards, shots blocked, interceptions and defensive errors. ISA was robust in making it's clusters in that it did not make use of such indices. Essentially, the Plaid model does not work well for football data in comparison to ISA.

Biclustering via ISA, is capable of finding correlated sub-matrices. It is resilient to noise and can handle overlapping modules (biclusters). The idea behind the biclustering by ISA is given in subsection D.3, with this as background, the ISA biclustering is performed in the following steps (Kasim et al., 2016):

1. Start with a randomly seeded observation score, ϕ^0. This can be used to calculate the variable score, γ^1, by using thresholding,

$$\gamma^1 = f_{t_C}(\boldsymbol{X}^{(norm)}\phi^0) \tag{1}$$

2. With this newly calculated γ^1, the initial observation score can be updated to,

$$\phi^1 = f_{t_G}(\boldsymbol{X}'^{(norm)}\gamma^1) \tag{2}$$

3. Equation 1 and 2 are iterated over until convergence or tolerance has been accepted, where the tolerance is defined as,

$$\frac{|\phi^* - \phi^n|}{|\phi^* + \phi^n|} < \epsilon$$

This is the ISA algortihm for discovering biclusters, the method is further extended to discover multiple biclusters by changing the seed and adjusting the thresholds.

The "isa" function in the package "isa2" by Csardi (2023) was used to perform this biclustering. There was a total of 44 clusters that were created when the R code seed was set to 3000, however, beyond the 19th bicluster, the results became difficult to interpret. A large number of these 19 biclusters were overlapping, as a result, similar biclusters were compared and the biclusters that provided the most information were selected visually. The intensity of the bicluster is determined by a gradient of colour from green, indicating a low/weak value, to red, which indicated a high/strong value.

Figure 22, is a bicluster that captured defensive ability the best. The 3rd, 11th, 17th and 19th did the same but not as well as this module. Here it can be seen that players like Zihno Vanheusden, Ozan Kabak, Matt Miazga and Jack Stephens are defensive players with relatively poor statistics. These same players are all part of the grey cluster that is shown in Figure 5 and Figure 21. Figure 22 has helped in confirming what the grey cluster in Figure 5 and Figure 21 illustrates. At this stage it can be seen that these players have much greener intensities relative to others on this diagram, such players can be seen to be having a bad season and may act as liabilities to their team. Managers and club owners may find it in their best interest to sell these types of players in order to purchase stronger players. All the other players have more red intensities, which would indicate that they are decent players to keep an eye on. These players are located in the green cluster in Figure 5 and Figure 21. This indicates that they are defensively strong, or at least reliable defenders to have on a team. Some of these players have much very red intensities, players such as: Wout Faes, Nico Schlotterbeck, Conor Coady, Sebastiano Luperto, Romain Saiss and Jan Bednareck. Most of these players are part of the outliers suggested in Table 2 and Table 3. This biclustering is able to show who amongst this group of good defenders stand out, it also manages to related this idea back to the outliers shown in Table 2 and Table 3 to confirm the idea suggested about the outlier map. These outlying players with strong performance metrics are the best defenders of the season, if managers and club owners are looking to purchase good defensive players they should consider these to name a few.

Figure 23 illustrates the strongest attacking players in the 5 leagues that were transferred in the 21/22 season. Robert Lewandowski, Erling Haaland and Gianluca Scamacca are three very outlying players that were also highlighted in Table 2 and Table 3. These players had a significantly better season than most players who entered the transfer market in this summer transfer window.

Like before Figure 24 will demonstrate a variety of players with differing levels of performance. Many players in this module have red intensities: Robert Lewandowski, Erling Haaland, Gianluca Scamacca, Anthony Modeste, Giovanni Simeone, Arnaud Kalimuendo and Taiwo Awoniyi. These players all relate to players found in Table 2 and Table 3 as well as players foung in the purple group of both Figure 5 and Figure 21. These players all being Figure 21 is significant, this cluster of players is smaller than in the case of K-means. As mentioned above the hierarchical clustering was able to separate players who are strictly good attackers from everyone else. This group also relates to a most of the "good" and "bad" outliers in the outlier maps in Table 2 and Table 3. Alexander Isak and Andrea Pinamonti are two players here that have a good intensity and are present in the green cluster of Figure 21 but are not outliers. Some players have green intensities like Romelu Lukaku, Timo Werner, Patrick Cutrone and Ignatius Ganago. These players have been tossed into the bad player category by all clustering methods. Romelu Lukaku and Timo Werner are two players that highlight the importance of this study. Both these players have a market value of 35 million and 70 million respectively, but they happen to have the worst performance measures of the players in this module. Players like Arnaud Kalimuendo and Taiwo Awoniyi are two players

that had significantly better perfomances in the season but were given lower player ratings and were undervalued in comparison. This perfectly illustrates the roll that fame has to play in the valuation of football players and biased measures of market value and player rating as perpetuated by transfermarkt and FIFA.

Figure 25, is a mix between DF and MF. This mix is present as many defensive metrics are shared with MF players. The players here can be seen as defensive MFs. Often in a team the midfield will be split between attacking midfielders and defensive midfielders but in other cases one might observe a central midfielder who is a mix between the two. In this diagram we have some players such as Mikkel Damsgaard seems to perform worse than all other players in this group. Mikkel Damsgaard is a winged player (essentially an attacking midfielder), this player being measured on defensive characteristics alone may not be able to fully capture the ability of the player, it is possible that this players valuation should increase due to his defensive role as an attacking player, this is not a trait that is commonly seen. Mathias Oliviera is a DF, he also happens to have some of the more red intensities across the board of metrics included in this module. He should be good at this tasks as he is a defender but it is possible that being grouped with these players demonstrates that he has the ability to play as a MF and a DF, this should increase his value as it illustrates a unique ability to play in more than one position. The remaining players are either central midfielders or defensive midfielders. The remaining players in this modeule are: Yves Bissouma (CM), Santiago Ascacibar (CDM), Remo Freuler (CM), Kalvin Phillips (CM), Jimmy Cabot (CDM), Xaver Schlager and Cheikhou Kouyate (CDM). Xaver Schlager in this group happens to have performed quite poorly in comparison to his counterparts. All the players in this module, except for Xaver and Mikkel Damsgaard, can be found in the green group of Figure 5 and Figure 21 but none of the actors in this module can be found as an outlier. The different positions in this module and the lack of outliers present begins to highlight the difficulty in rating MF players.

The final module examined attacking midfielders, Figure 26. This group is a lot more distinctive than defensive midfielders. Table 2 and Table 3 capture players such as, Antonio Candreva, David Raum and Jonatahn Clauss who are also present in Figure 26 and lie in the purple cluster of Figure 5. These are the strongest attacking MF you could have selected in the data and should be considered by managers and club owners. These two players also cost less than a large number of players in this same module with worse performance. These players are unambiguous and the two variables in the module capture them relatively well.

4.4 Multivariate General Linear Modelling (MVGLM)

MVGLM is an extension of the uni-variate case of a general linear model with more than one dependent variable. In the case of the data, the dependent variables are the market value of the player as well as their FIFA rating. Notable variables that affect player value were determined to be the categorical variables of the position and the league in which they play as well as their age(Felipe et al., 2020). The packages used to perform the the MVGLM were provided by Cornu et al. (2018). The multivariate from of the GLM can be written as:

$$Y_{nxp} = \beta 0_{n\times p} + X_{n\times k} \times \beta 1_{k\times p} + E_{n\times p}$$

$$\begin{pmatrix} Y_{11} & \dots & Y_{1p} \\ \vdots & . & \vdots \\ Y_{n1} & \dots & Y_{np} \end{pmatrix} = \begin{pmatrix} \beta 0_{11} & \dots & \beta 1_{1p} \\ \vdots & . & \vdots \\ \beta 0_{n1} & \dots & \beta 1_{np} \end{pmatrix} + \begin{pmatrix} X_{11} & \dots & X_{1k} \\ \vdots & . & \vdots \\ X_{n1} & \dots & X_{nk} \end{pmatrix} \times \begin{pmatrix} \beta 1_{11} & \dots & \beta 1_{1p} \\ \vdots & . & \vdots \\ \beta 1_{k1} & \dots & \beta 1_{kp} \end{pmatrix} + \begin{pmatrix} \epsilon_{11} & \dots & \epsilon_{1p} \\ \vdots & . & \vdots \\ \epsilon_{n1} & \dots & \epsilon_{np} \end{pmatrix},$$

where n is the number of observations, p is the number of dependent variables and k is the number

of independent variables. The β matrix values are the regression coefficients for each independent variable. The link functions used were the Gaussian function for both market value and rating.

Table 1: Coefficients of independent variables

Variable	Market Value	FIFA Rating
(Intercept)	25.70184614	56.674251842
age	-0.93333285	0.659592294
Position	0.70366140	-0.104621562
League	0.76545288	-0.056809070
Matches..Played	-0.14867920	0.030435086
Goals	2.57555799	0.417450466
Assists	2.25147716	0.272257766
SOT..	0.01244351	-0.006209522
Blocks	0.18296840	0.112628974
Red.Cards	-0.07880100	-0.341163114
Fouls.Committed	-0.20216399	-0.091131151

Table 1 shows a snippet of the coefficients for the different variables and how they affect the market value and FIFA rating. Goals and assists have some of the largest coefficients when predicting market value and rating, which is expected because goal contributions are the most important measures of performance. Other statistics which are commonly interpreted as negative, such as no. of red cards and no. of fouls committed, have negative coefficients. In ratings, red cards possesses the largest negative coefficient and fouls committed has the second largest coefficient for the market value prediction. Age affects market value negatively and rating positively, possessing the highest negative and positive coefficient in each case respectively. This can be interpreted as a trade off of longevity vs experience. In real world situations, as players get older they become more of a liability to the club because they are more prone to injuries and they do not perform as well compared to the younger players. However, the older players do provide a higher level of thinking due to their wisdom in the game. A more recent example is Cristiano Ronaldo, who is rated among the top 5 players in FIFA and regarded as one of the best footballers but his market value is 20.00 million euro (www.transfermarkt.com) and this is because he is 37 years old.

Table 6 and Table 7 show the predictive ability of the model in terms of the overall rating and market value. The deviation between the observed and predicted ratings is low for most observations and this results in a low Root Mean Squared Error (RMSE) of 4.74. This means the MVGLM is able to closely predict FIFA ratings. The deviation between the observed and predicted market values is higher, which resulted in a RMSE 12.43 that is almost 3 times larger than that of the FIFA ratings. This would suggest that the MVGLM does not perform regression on this dependent variable as accurately.This is most likely because there is a set method for calculating the ratings, which makes the rating method consistent. But when it comes to market value the final value is bias and is not as easily reproducible for all players.

5 Limitations and Recommendations

- Model does not consider economic factors such as inflation, tax, exchange rates, etc. These are major factors that contribute to the valuation of football players. This can be easily be seen by the increase in valuation over the years. The use of time series data (by looking into as

many seasons as possible), as opposed to one season, can overcome this limitation. Methods such independent component analysis, functional regression, recurrent neural networks to name a few would be interesting methods to explore with this type of data

- The clustering here did not give the strongest clustering, in future try model based clustering (e.g. Pocuca et al. (2022)) probabilistic and neural network based methods (e.g. SOM) can be considered.
- RPCA is being used to adjust for outliers, it may be worthwhile to explore the data and find levels of skewness on the features of interest. This would then suggest the possibility of moving from RPCA suggested by Hubert et al. (2005) to an adaptation of this that is able to correct for skewness as seen in Hubert et al. (2009)
- Regular MVGLM was used at this stage with the original dataframe. It may be worthwhile to explore robust principle component regression from first principles as there is no package that supports this type of regression, using MCD-regression or multivariate least trimmed squares should be considered in conjunction with the robust PCs (Filzmoser, 2001; Rousseeuw et al., 2004; Verboven and Hubert, 2002). Least trimmed squares is often the approach, however, this does not exactly work in the multivariate case (Agullo et al., 2002).
- As shown in subsubsection 4.1.2, players are characterized by different sets of variables. It may be wise to look into Factor analysis (FA) as a means to analyse the data as this is capable of grouping the data into sets with high within group correlation and poor between group correlation. Such an analysis has been conducted by Rodríguez (2021). The nice thing about using FA is that it pairs well with one of the more famous biclustering techniques "Fabia" (fator analysis for bicluster acquisition) (Kasim et al., 2016). Based on how well biclustering performed here, it would be interesting to see the results in the case of "Fabia". One would use O'Hara-Wild et al. (2023) in r to perform this. Following this, the study can be furthered by using latent variable models (Izenman, 2008).
- Our data-driven approach to estimating market value and player rating is in fact complementary to transfermarkt and FIFA in helping make the estimation of such values reproducible, which is the same approach taken by Müller et al. (2017). It would do well to extend this study so as to cut out these two, and remove the bias, especially transfermarkt as FIFA ratings are relatively consistent. In this study, market value was used to build the model to predict market value, however, in future transfer fees should be used, with data spanning as many years as possible of transfers made. The problem then becomes a time series problem and the methods provided as the first recommendation would be implemented.

6 Conclusion

The study sought out to provide a framework for MVA and the reprodcution of both football player valuation and rating. Data was collected as the transfers made in the summer 2021/2022 football transfer window. A PCA was performed, but was found to be insufficient as this did not account for outliers; following this a RPCA was done to remedy the previous problem. This RPCA provided a diagnostic plot of the outliers which allowed some insight into which players should be scouted by teams and which should be left alone, it also helped inform on which players are liabilities and should be sold from their respective team for the sake of that team. Following this the RPCA was able to classify players into position according to certain metrics, this idea informs the remainder of the study. Clustering was then used as a means to group the players into a respective clusters. 3 clusters were ultimately formed which helped separate attacking strong

players, defensive strong players and players who did not perform well in the 2021/2022 season. Biclustering was then utilised to confirm the ideas brought about by the outlier maps that were a part of the RPCA and the meaning of the clusters. It was also helpful in identifying players who were over-valued and under-valued in their position. The biclustering illustrated the difficulty of classifying MF players who were not attacking strong, essentially highlighting a limitation in our data that should be examined further.

These exploratory steps illustrative to buyers and sellers a way in which one may go about choosing players to buy and to sell. It has also helped identify the potential drawbacks of trusting crowdsourcing as a means to measure player valuation. It reveals the bias involved in the assessment of players that would arise due to fame and a neglect in the proper evaluation of lesser known players. This would suggest that there is a space to create a completely data-driven approach to the evaluation of football player value.

Following this, supervised learning was conducted in the form of a MVGLM and this revealed the difficulty of predicting market values using performance statistics because of the unstructured manner in which market values have been previously determined. On the other hand, the FIFA rating were more accurately predicted and the MVGLM proved to be a useful technique for that variable. This further suggests a need study market value in the future, it proposes a need for an unbiased evaluation of player valuation that will be able to remedy the drawbacks of crowdsourcing.

References

Agullo, J., Croux, C., and Van Aelst, S. (2002). The multivariate least trimmed squares estimator. *Journal of Multivariate Analysis*, 99:311–338.

Behravan, I. and Razavi, S. M. (2021). A novel machine learning method for estimating football players' value in the transfer market. *Soft Computing*, 25(3):2499–2511.

Cornu, G., Mortier, F., Trottier, C., and Bry, X. (2018). *SCGLR: Supervised Component Generalized Linear Regression*. R package version 3.0.

Croux, C. and Haesbroeck, G. (1999). Principal component analysis based on robust estimators of the covariance or correlation matrix: Influence functions and efficiencies. *Biometrika*, 87.

Croux, C. and Ruiz-Gazen, A. (1996). A fast algorithm for robust principal components based on projection pursuit. In Prat, A., editor, *COMPSTAT*, pages 211–216, Heidelberg. Physica-Verlag HD.

Csardi, G. (2023). *isa2: The Iterative Signature Algorithm*. R package version 0.3.6.

Davies, P. L. (1987). Asymptotic behaviour of s-estimates of multivariate location parameters and dispersion matrices. *The Annals of Statistics*, 15(3):1269–1292.

Felipe, J. L., Fernandez-Luna, A., Burillo, P., de la Riva, L. E., Sanchez-Sanchez, J., and Garcia-Unanue, J. (2020). Money talks: Team variables and player positions that most influence the market value of professional male footballers in europe. *Sustainability*, 12(9):3709.

Filzmoser, P. (2001). Robust principal component regression. *Computer data analysis and modeling. Robust and computer intensive methods. Belarusian State University, Minsk*, 132(7).

Fry, T. R., Galanos, G., and Posso, A. (2014). Let's get messi? top-scorer productivity in the european champions league. *Scottish Journal of Political Economy*, 61(3):261–279.

Herm, S., Callsen-Bracker, H.-M., and Kreis, H. (2014). When the crowd evaluates soccer players' market values: Accuracy and evaluation attributes of an online community. *Sport Management Review*, 17(4):484–492.

Hinton, G. E. and Roweis, S. (2002). Stochastic neighbor embedding. *Advances in neural information processing systems*, 15.

Huang, S.-Y., Yeh, Y.-R., and Eguchi, S. (2009). Robust kernel principal component analysis. *Neural Computation*, 21(11):3179–3213.

Hubert, M., Rousseeuw, P., and Verdonck, T. (2009). Robust pca for skewed data and its outlier map. *Computational Statistics & Data Analysis*, 53:2264–2274.

Hubert, M., Rousseeuw, P. J., and Branden, K. V. (2005). Robpca: A new approach to robust principal component analysis. *Technometrics*, 47(1):64–79.

Izenman, A. J. (2008). *Modern multivariate statistical techniques*, volume 1. Springer.

Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philos Trans A Math Phys Eng Sci*, 374(2065):20150202.

Karatzoglou, A., Smola, A., and Hornik, K. (2023). *kernlab: Kernel-Based Machine Learning Lab*. R package version 0.9-32.

Kasim, A., Shkedy, Z., Kaiser, S., Hochreiter, S., and Talloen, W. (2016). *Applied Biclustering Methods for Big and High-Dimensional Data Using R*. CRC Press.

Leitch, W. (2019). The era of the old athlete is over.

Li, G. and Chen, Z. (1985). Projection-pursuit approach to robust dispersion matrices and principal components: primary theory and monte carlo. *Journal of the American Statistical Association*, 80(391):759–766.

Li, W., Cerise, J. E., Yang, Y., and Han, H. (2017). Application of t-sne to human genetic data. *Journal of bioinformatics and computational biology*, 15(04):1750017.

Müller, O., Simons, A., and Weinmann, M. (2017). Beyond crowd judgments: Data-driven estimation of market value in association football. *European Journal of Operational Research*, 263(2):611–624.

Nguyen, M. and Torre, F. (2008). Robust kernel principal component analysis. *Advances in Neural Information Processing Systems*, 21.

O'Hara-Wild, M., Hyndman, R., and Wang, E. (2023). *fable: Forecasting Models for Tidy Time Series*. R package version 0.3.3.

Pocuca, N., Browne, R. P., and McNicholas, P. D. (2022). *mixture: Mixture Models for Clustering and Classification*. R package version 2.0.5.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rodríguez, M. S. (2021). Factor analysis of the market value of high-performance players for three major european association football leagues. *Managing Sport and Leisure*, 26(6):484–507.

Rousseeuw, P. (1984). Least median of squares regression. *Journal of The American Statistical Association - J AMER STATIST ASSN*, 79:871–880.

Rousseeuw, P. and Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41:212–223.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Rousseeuw, P. J., Aelst, S. V., Driessen, K. V., and Agulló, J. (2004). Robust multivariate regression. *Technometrics*, 46(3):293–305.

Rousseeuw, P. J. and Hubert, M. (2018). Anomaly detection by robust statistics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(2):e1236.

Todorov, V. and Filzmoser, P. (2009). An object-oriented framework for robust multivariate analysis. *Journal of Statistical Software*, 32(3):1–47.

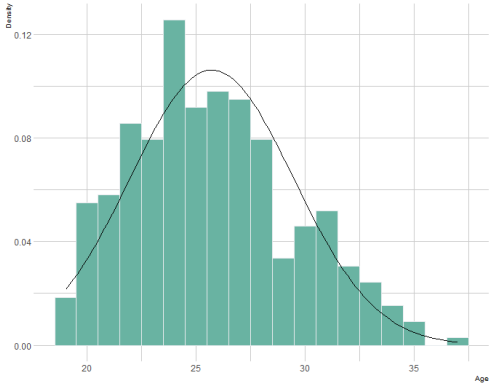
Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Verboven, S. and Hubert, M. (2002). Robust principal components regression. In Härdle, W. and Rönz, B., editors, *Compstat*, pages 515–520, Heidelberg. Physica-Verlag HD.

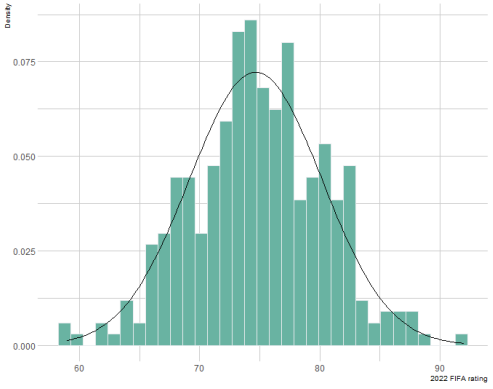
Wainer, H. (1988). Robust regression and outlier detection. *Journal of Educational Statistics*, 13(4):358–364.

A Exploratory Data Analysis

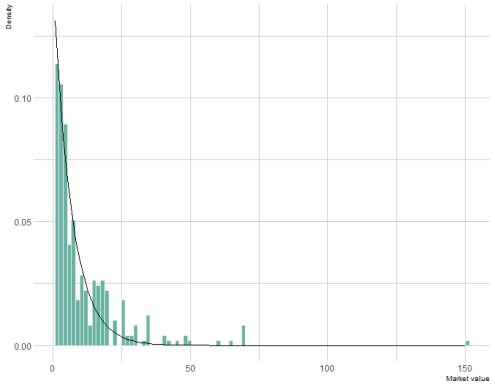
A.1 Univariate Exploratory Data Analysis



(a) Player Age



(b) FIFA Rating



(c) Player Market Value

Figure 6: Histogram and density distributions of Age, FIFA rating and Market Value

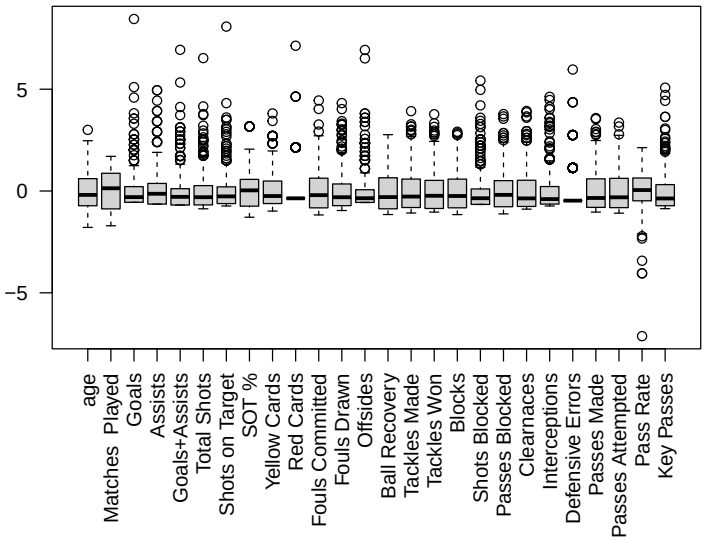


Figure 7: Boxplots of the predictor space

A.2 Bivariate Exploratory Data Analysis

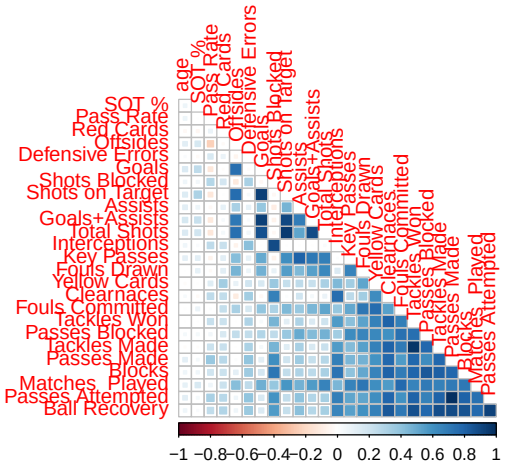


Figure 8: Correlations of Figure

B Linear Dimensionality Reduction

B.1 Principle Component Analysis

Short derivation of PCA:

The information of a dataset is captured by the covariance matrix, Σ, of the data. The PCs are found by performing spectral decomposition on this covariance matrix,

$$\Sigma = \boldsymbol{U}\Lambda\boldsymbol{U}'.$$

The eigenvalues, λ_i, which will be ordered from highest to lowest in Λ will illustrate the variation captured in the ordered PCs, and the PCs will be captured by the columns in $\boldsymbol{U}$. Hereafter, it is possible to select a number of PCs that explain a sufficent amount of variation.

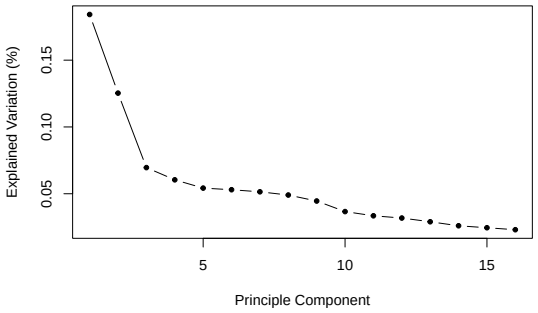


Figure 9: Scree plot of principle components

B.2 Robust Principle Component Analysis

An overview of how the RPCA is condcuted:

1. First, for a dataset where $p < n$, dimensionality reduction is performed by taking the singular value decomposition (SVD) of the mean-centered data matrix,

$$\boldsymbol{X}_{(n\times p)} - \mathbf{1}_{(n\times 1)}\hat{\boldsymbol{\mu}}_0' = \boldsymbol{U}_{(n\times r_0)}\boldsymbol{D}_{(n\times r_0)}\boldsymbol{V}'_{(r_0\times p)}$$

where $r_0 = \text{rank}(\boldsymbol{X} - \mathbf{1}\hat{\boldsymbol{\mu}}_0'\hat{\boldsymbol{\mu}}_0)$, $\boldsymbol{X}$ is the data matrix, $\hat{\boldsymbol{\mu}}_0$ is the sample mean of the original dataset, $\boldsymbol{UDV}'$ is the SVD. The subspace is now taken as $\boldsymbol{Z} = \boldsymbol{UD}$.

2. Next, we try to find the number of "least outlying" data points, $h < n$. The covariance matrix of these h data points is then used to create another subspace, k_0. The calculation of these h data points is as follows. For each data point, $\boldsymbol{x}_i$, an adaptation of the Stahel–Donoho affine-invariant outlyingness is computed,

$$outl_A(\boldsymbol{x}_i) = \max_{\boldsymbol{v}\in B} \frac{|\boldsymbol{x}_i'\boldsymbol{v} - t_{MCD}(\boldsymbol{x}_j'\boldsymbol{v})|}{s_{MCD}(\boldsymbol{x}_j'\boldsymbol{v})}$$

where B refers to all non-zero directional vectors and that this can be restricted to all directions through 2 observations, but not more than 250 vectors, so if there are $\binom{n}{2} > 250$, then we randomly select 250 of these vectors. This also explains that $\boldsymbol{v}$ is a vector created from the choice of two data points. In the formulation above t_{MCD} and s_{MCD} are the univariate MCD location and scale estimators respectively. By computing $outl_A\boldsymbol{x}_i$ for all data points

we end up with a reduced rank dataset, $\boldsymbol{X}_{(n\times r_1)}$, $r_1 < r_0$, (where the rank reduces due to "exact fit situations"), and a set of indices, H_0, that correspond to the h least outlying data points. The number of h is selected as,

$$h = max\left\{\alpha n, \frac{n + k_{max} + 1}{2}\right\}$$

where k_max is the max number of components, α is the robustness parameter (a value [0.5, 1] deciding how many outliers to include). The mean and covariance of $\boldsymbol{X}_{(n\times r_1)}$, are denoted by $\hat{\boldsymbol{\mu}}_1$ and $\boldsymbol{S}_0$. A spectral decomposition can then be performed on the covariance matrix, where like a PCA, the eigen values will be ordered in size along with their corresponding eigenvectors to indicate proportion of explained variation.

$$\boldsymbol{S}_0 = \boldsymbol{P}_0\boldsymbol{L}_0\boldsymbol{P}_0$$

where $\boldsymbol{L}_0 = diag(\tilde{l}_1, ..., \tilde{l}_r)$ and $r \leq r_1$. $\boldsymbol{P}_0$, are the principle components of $\boldsymbol{X}_{(n\times r_1)}$. The first $k_0 < r$ eigenvectors $\boldsymbol{P}_{(r_1\times k_0)}$ can then be selected via a scree plot. This essentially creates the subspace mentioned at the earlier. The data points can now be projected onto the subspace spanned by the first k_0 eigenvectors,

$$\boldsymbol{X}^*_{(n\times k_0)} = (\boldsymbol{X}_{(n\times r_1)} - \boldsymbol{1}_{(n\times 1)}\hat{\boldsymbol{\mu}}_1')\boldsymbol{P}_{(r_1\times k_0)}$$

3. This is the final step of this algorithm. The location (mean) and scatter (covariance) of the projected data, $\boldsymbol{X}^*_{(n\times k_0)}$, are found robustly with the MCD estimator, using an adaptation of the Fast-MCD algorithm introduced by (Rousseeuw and Driessen, 1999) (this adaptation is called the reweighted MCD estimator). The eigenvectors of this scatter matrix will give the robust principle components $P_{(p\times k)}$, where $k \ll k_0$.
4. The full derivation and a complete descriptive explanation of the reweighted MCD estimator can be found in (Hubert et al., 2005).

Scree Plot:

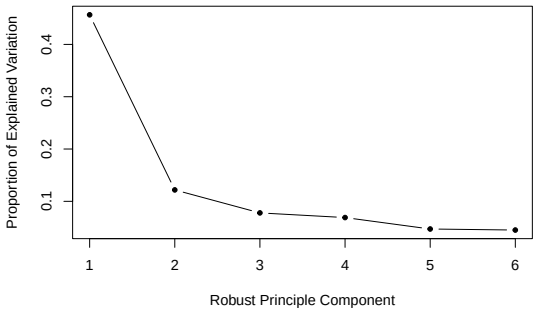


Figure 10: Scree plot of robust principle components

Table of Outliers:

Table 2: Bad outliers, make for incredible players

Name	Age	Position	Market Value	OVR
Anthony Modeste	34.00	FW	3.00	72
Antonio Candreva	35.00	MF	2.50	79
Conor Coady	29.00	DF	25.00	79
David Raum	24.00	DF	20.00	73
Erling Haaland	21.00	FW	150.00	88
Francesco Acerbi	34.00	DF	4.00	83
Giovanni Simeone	27.00	FW	17.00	75
Jan Bednarek	26.00	DF	22.00	76
Jules Koundé	23.00	DF	60.00	83
Nico Schlotterbeck	22.00	DF	33.00	72
Robert Lewandowski	33.00	FW	45.00	92
Wout Faes	24.00	DF	10.00	73

Table 3: Some good outliers, are undervalued up and coming talents

Name	Age	Position	Market Value	OVR
Arnaud Kalimuendo	20.00	FW	18.00	73
Dominik Kohr	28.00	MF	5.00	75
Ethan Ampadu	21.00	MF	13.00	68
Gianluca Scamacca	23.00	FW	30.00	74
Jonathan Clauss	29.00	DF	15.00	77
Paul Pogba	29.00	MF	48.00	87
Salis Abdul Samed	22.00	MF	3.00	60
Sebastiano Luperto	25.00	DF	3.00	71
Taiwo Awoniyi	24.00	FW	20.00	74

Table 4: Some other good outliers, but not all good outliers are good players

Name	age	Position	Market Value	OVR
Amine Gouiri	22.00	FW	42.00	78
Antonio Rüdiger	29.00	DF	40.00	83
Arkadiusz Reca	27.00	DF	3.00	70
Dedryck Boyata	31.00	DF	3.50	78
Francesco Acerbi	34.00	DF	4.00	83
Gianluca Caprari	28.00	MF	10.00	75
Hannibal Mejbri	19.00	FW	6.00	62
Harry Winks	26.00	MF	15.00	77
Houboulang Mendes	24.00	DF	3.00	67
Ilaix Moriba	19.00	MF	9.00	73
Jimmy Giraudon	30.00	DF	1.50	70
Joel Pohjanpalo	27.00	FW	2.50	73
Johan Mojica	30.00	DF	5.00	74
Jordan Beyer	22.00	DF	4.50	70
Jordan Veretout	29.00	MF	17.00	81
Kalidou Koulibaly	31.00	DF	35.00	86
Konstantinos Mavropanos	24.00	DF	15.00	73
Leandro Paredes	28.00	MF	17.00	81
Lorenzo Insigne	31.00	FW	25.00	86
Luca Kilian	22.00	FW	4.50	69
Mattia Viti	20.00	DF	7.00	64
Maxim Leitsch	24.00	DF	5.00	72
Maya Yoshida	33.00	DF	1.50	73
Mehdi Chahiri	25.00	DF	1.20	69
Nahuel Molina	24.00	DF	20.00	73
Nayef Aguerd	26.00	DF	12.00	76
Nemanja Matić	33.00	MF	5.00	79
Niklas Stark	27.00	DF	6.50	76
Nordi Mukiele	24.00	DF	20.00	81
Paulo Dybala	28.00	FW	35.00	87
Pervis Estupiñán	24.00	DF	20.00	79
Romain Saïss	32.00	DF	8.00	78
Romain Thomas	34.00	DF	1.00	76
Sadio Mané	30.00	FW	70.00	89
Sebastiano Luperto	25.00	DF	3.00	71
Shane Duffy	30.00	DF	5.00	72
Stefan Posch	25.00	DF	10.00	75
Tyler Adams	23.00	MF	17.00	77

Score Plots:

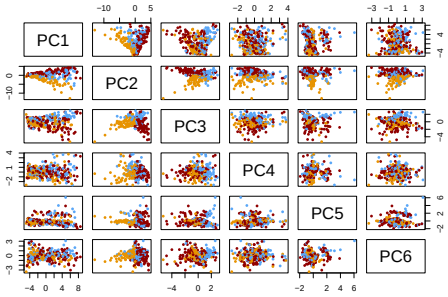


Figure 11: Principle component scores

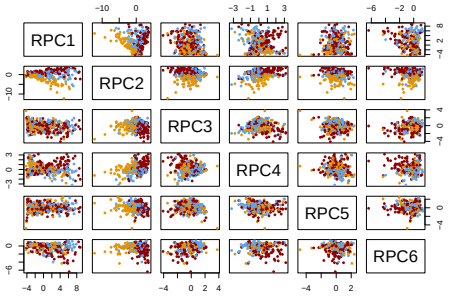


Figure 12: Robust principle component scores

C Non-Linear Dimensionality Reduction

C.1 Kernal Principle Component Analysis

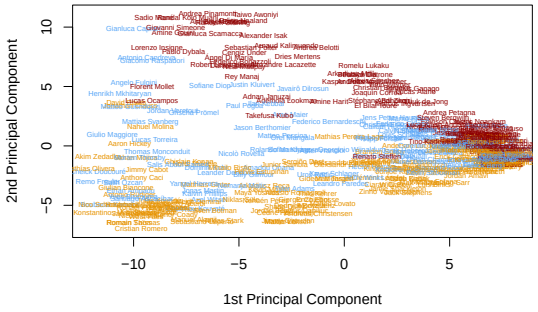


Figure 13: Radial basis kernal function, kernal principle component analysis, which looks the same as Laplacian kernal function

C.2 t-distributed stochastic neighbor embedding (t-SNE)

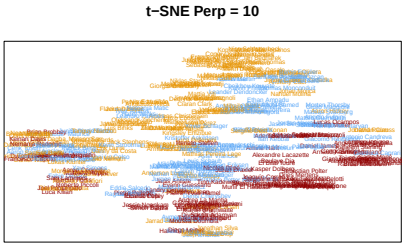


Figure 14: t-SNE at a perplexity of 10

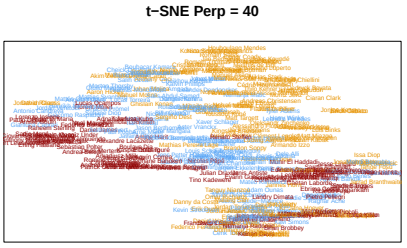


Figure 15: t-SNE at a perplexity of 40

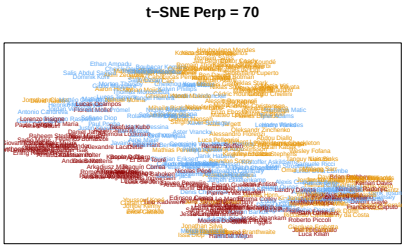


Figure 16: t-SNE at a perplexity of 70

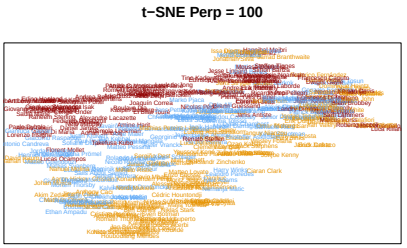
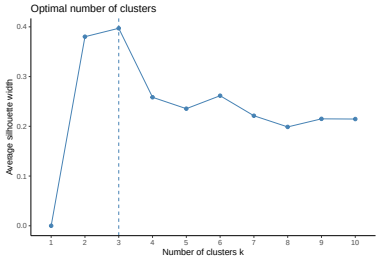


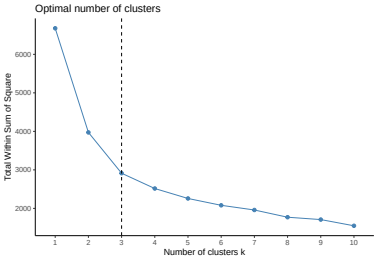
Figure 17: t-SNE at a perplexity of 100

D Clustering

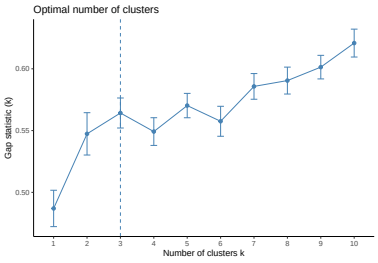
D.1 K-Means Clustering



(a) Silhouette Method



(b) Total Within Sum of Squares



(c) Gap Statistic

Figure 18: Number of clusters for the k-means approach

D.2 Hierarchical Clustering

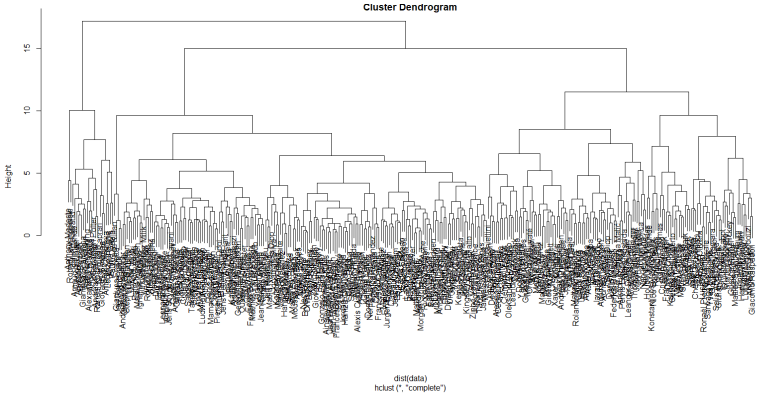
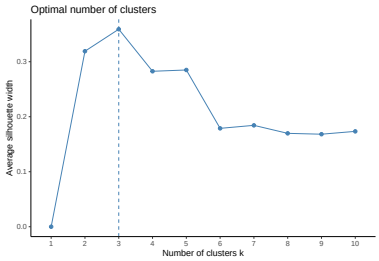
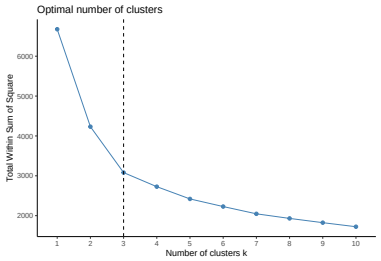


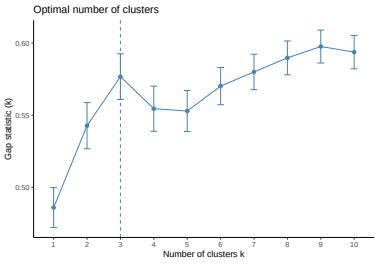
Figure 19: Dendrogram of Players using complete linkage



(a) Silhouette Method



(b) Total Within Sum of Squares



(c) Gap Statistic

Figure 20: Number of clusters for the agglomerative hierarchical clustering approach

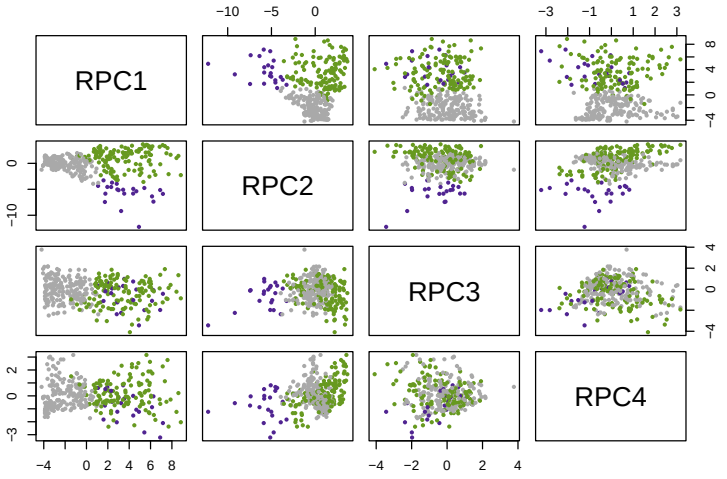


Figure 21: Agglomerative hierarchical clustering with robust principle component scores

D.3 Biclustering

ISA biclustering proposed by Kasim et al. (2016):

1. The data matrix is defined as $\boldsymbol{X}_{M\times N}$.

2. This data matrix has to be normalized, $\boldsymbol{X}^{(norm)}_{M\times N}$, as z-scores are capable of providing meaningful comparisons between pairs of variables (columns) and pairs of observations (rows).
3. Let ϕ and γ be vectors that represent row scores and column scores, respectively.
4. An observation, $\boldsymbol{X}_{i.}$, is a constituent of a module if its $\phi_i \neq 0$ (i.e. its row score is non-zero), and a variable, $\boldsymbol{X}_{.j}$, is a also a constituent of a module if its $\phi_j \neq 0$ (i.e. its column score is non-zero)
5. For random sets of ϕ and γ, both the variable projection scores, c^{proj}, and the observation projection scores, g^{proj}, can be obtained as a linear combination of $c^{proj} = \boldsymbol{X}^{(norm)}\phi$ and $g^{proj} = \boldsymbol{X}'^{(norm)}\gamma$ (This is a way in which the observations are related to the variables and vice versa).
6. Co-regulated observations can be recognized by thresholding observation projection scores and variable projection scores. This thresholding essentially turns an observation/variable projection score and converts it back to a variable/observation score. An example of one such binary threshold function are defined below,

$$\phi = f_{t_G}(g^{proj}) = \begin{cases} 1 & \text{if } g^{proj} > t_G \\ 0 & \text{otherwise} \end{cases}$$

in the case of the observation score, and,

$$\gamma = f_{t_C}(c^{proj}) = \begin{cases} 1 & \text{if } c^{proj} > t_C \\ 0 & \text{otherwise} \end{cases}$$

in the case of the variable score. These thresholds, t_ϕ and t_γ are selected so as to recreate these scores. The bicluster is then described jointly by the two scores. This threshold function does not have to be binary; strictly speaking it is a product of weight and step functions,

$$f_t(x) = \begin{pmatrix} w(x_1) \\ \vdots \\ w(x_N) \end{pmatrix} \begin{pmatrix} \Psi(\boldsymbol{X}^{(norm)}_{1.} - t) \\ \vdots \\ \Psi(\boldsymbol{X}^{(norm)}_{N.} - t) \end{pmatrix}$$

The step function, $\Psi(\boldsymbol{X}^{(norm)})$, uses the z-score and checks it against against the threshold, the output is dependent on whether the threshold is met and if it is not 0 is outputted. The weight function, $w(X)$, further constrains this step function.

This got a bit confusing but basically, there's a projection of the variable scores in the observation space and there is a projection of the observation scores onto the variable space. Separate thresholds are then used for both projections to identify biclusters.

Heatmap of modules:

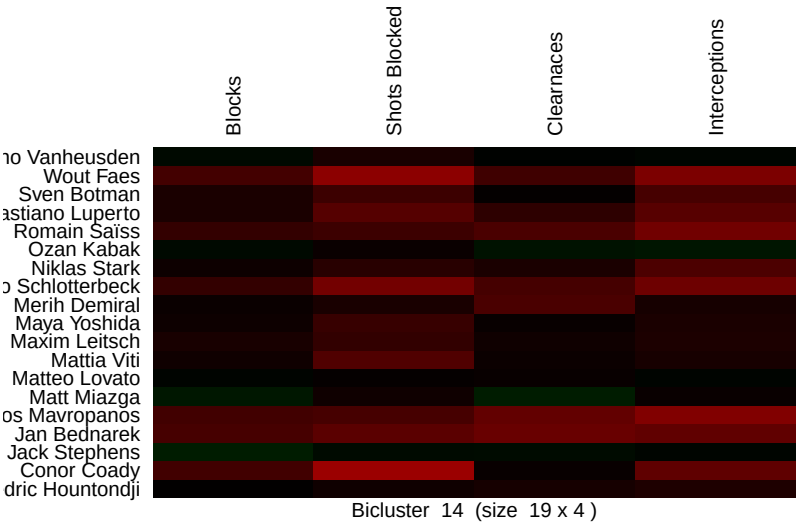


Figure 22: Defensive bicluster

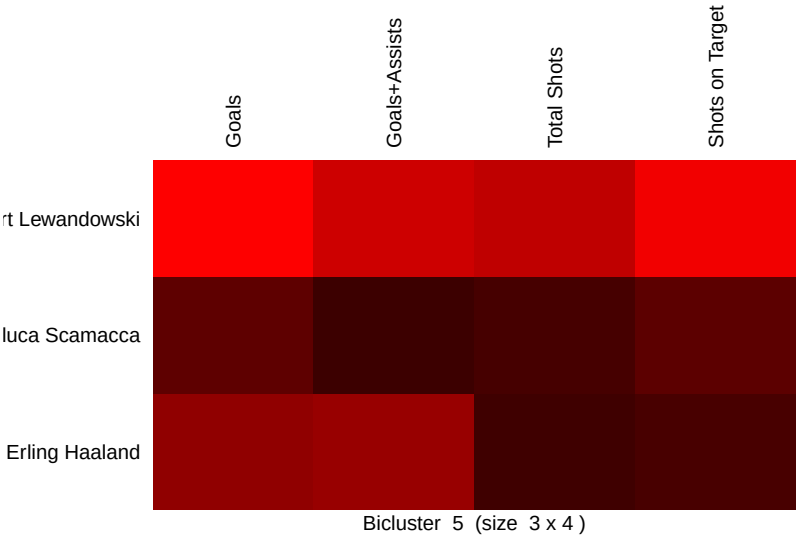


Figure 23: Attacking bicluster, a grouping of three strongest attacking players

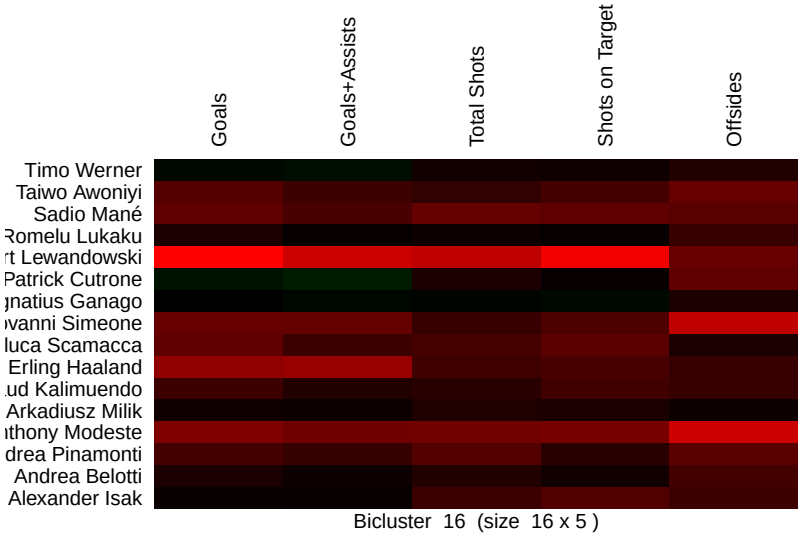


Figure 24: Attacking bicluster

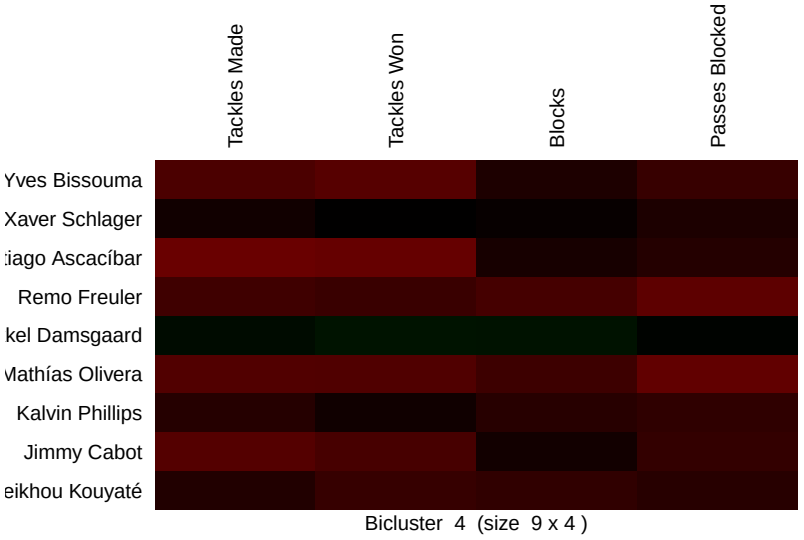


Figure 25: Defensive Midfielders

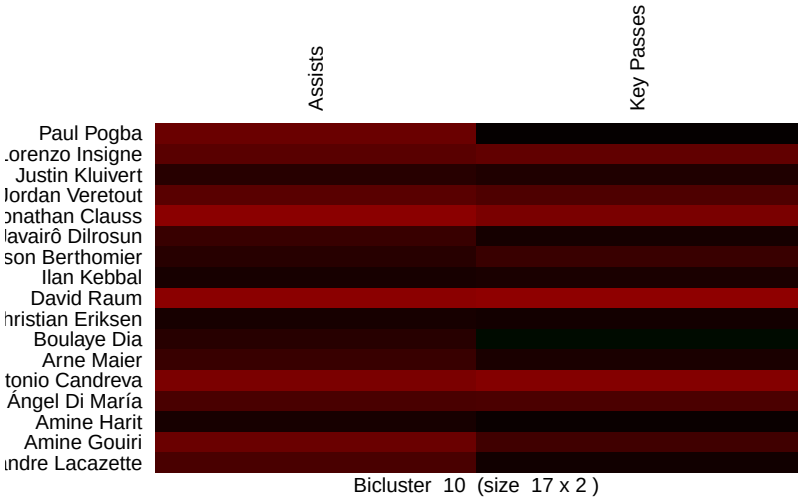


Figure 26: Attacking Midfielders

E Multivariate General Linear Models

Table 5: Coefficients of independent variables

Variable	Market Value	FIFA Rating
(Intercept)	25.70	56.67
age	-0.93	0.66
Position	0.70	-0.10
League	0.77	-0.06
Matches..Played	-0.15	0.03
Goals	2.58	0.42
Assists	2.25	0.27
SOT..	0.01	-0.01
Yellow.Cards	0.05	-0.03
Red.Cards	-0.08	-0.34
Fouls.Committed	-0.20	-0.09
Fouls.Drawn	-0.14	-0.03
Offsides	-0.42	-0.14
Ball.Recovery	0.09	0.02
Tackles.Made	0.06	-0.02
Tackles.Won	0.06	0.08
Blocks	0.18	0.11
Clearnaces	-0.23	-0.09
Interceptions	0.05	-0.00
Defensive.Errors	0.09	-0.13
Key.Passes	-0.23	0.02

Table 6: Observed and Predicted Market Values

Player	Position	Observed Market value	Predicted Market value
Ilaix Moriba	MF	9	11.686
Maxim Leitsch	DF	5	19.662
Leandro Paredes	MF	17	14.642
Shane Duffy	DF	5	9.451
Mathias Pereira Lage	DF	2.5	14.783
Yves Bissouma	MF	35	20.288
Konstantinos Mavropanos	DF	15	27.606
Julian Draxler	FW	18	7.222
Nicolò Casale	DF	7.5	12.402
Dominique Heintz	DF	2.5	0.749
Amine Gouiri	FW	42	38.484
Nahuel Molina	DF	20	28.235
Paulo Dybala	FW	35	31.066
Romelu Lukaku	FW	70	11.955
Jan Bednarek	DF	22	20.301
Mathías Olivera	DF	15	9.667
Jason Berthomier	MF	1.2	6.309
Ozan Kabak	DF	10	10.36
David Nemeth	DF	2.3	8.29
Antonio Rüdiger	DF	40	17.739
Aaron Hickey	DF	18	17.808
Andrea La Mantia	FW	1.3	1.331
Armando Izzo	DF	2.8	-1.062
Francesco Caputo	FW	2.5	0.289
Takumi Minamino	FW	12	11.747
Conor Coady	DF	25	28.679
Sebastiano Luperto	DF	3	10.367
Alessandro Florenzi	DF	5.5	2.217
Héctor Herrera	MF	5	-2.027
Arkadiusz Milik	FW	16	15.402
Erick Cabaco	DF	1.8	3.163
Steven Bergwijn	FW	18	12.488
Denis Zakaria	MF	27	13.23
Roberto Piccoli	FW	4	9.419
Matteo Lovato	DF	7	13.264

Table 7: Observed and Predicted Ratings

Player	Position	Observed Rating	Predicted Rating
Ilaix Moriba	MF	73	68
Maxim Leitsch	DF	72	75
Leandro Paredes	MF	81	77
Shane Duffy	DF	72	78
Mathias Pereira Lage	DF	74	74
Yves Bissouma	MF	79	75
Konstantinos Mavropanos	DF	73	76
Julian Draxler	FW	80	76
Nicolò Casale	DF	64	72
Dominique Heintz	DF	74	75
Amine Gouiri	FW	78	78
Nahuel Molina	DF	73	77
Paulo Dybala	FW	87	81
Romelu Lukaku	FW	88	79
Jan Bednarek	DF	76	74
Mathías Olivera	DF	76	73
Jason Berthomier	MF	72	82
Ozan Kabak	DF	76	71
David Nemeth	DF	69	71
Antonio Rüdiger	DF	83	78
Aaron Hickey	DF	69	72
Andrea La Mantia	FW	69	77
Armando Izzo	DF	79	74
Francesco Caputo	FW	82	79
Takumi Minamino	FW	75	75
Conor Coady	DF	79	81
Sebastiano Luperto	DF	71	72
Alessandro Florenzi	DF	81	77
Héctor Herrera	MF	81	78
Arkadiusz Milik	FW	81	77
Erick Cabaco	DF	73	74
Steven Bergwijn	FW	80	74
Denis Zakaria	MF	80	74
Roberto Piccoli	FW	63	70
Matteo Lovato	DF	72	71

F Code

(R Core Team, 2023)

```
---
title: "MVA Assignment"
author: "Edward Baleni, BLNEDW003, Thabo Dube, DBXTHA030"
date: "`r Sys.Date()`"
output:
  pdf_document:
    fig_caption: yes
```

```
    extra_dependencies:
    - float
    - subfig
    keep_md: yes
  html_document:
    df_print: paged
header-includes: \usepackage{amsmath}
always_allow_html: yes
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = F, fig.align="center", out.width = "65%", fig.pos = "H")
```

Introduction

```{r Packages, include=FALSE}
set.seed(16)
require(Rtsne)
require(scatterplot3d)
require(kernlab)
require(biclust)
require(cluster)
require(corrplot)
require(isa2)
require(sparsepca)
require(xtable)
require(ggfortify)
require(rrcov)
require(ordr)
require(ggplot2)
require(viridis)
require(reshape2)
require(factoextra)
require(SCGLR)
```


```{r Data}
# Load in data
load("FinData.RData")
# Change row names
rownames(FinData) <- FinData$Name
# Get X matrix that does not include potential response variables and categorical variables
X <- FinData
rownames(X) <- X$Name
X <- scale(X[,c(-1,-5, -6, -7, -8,-3,-4, -35, -34, -36, -37)])

# Look for outliers
par(mar = c(8, 4, 0.5, 2))
boxplot(X, las = 2)

# Define colour scheme
colors <- c("#900000", "#E79300", "#59A6F7") # DF, FW, MF
colors <- colors[FinData$Position]
```

Exploratory Data Analysis


```

```
68
69 # Methodology
70 ```{r Correlations}
71 # Quickly check correlations as this may inform our thinking on principle
       components
72 co <- cor(X)
73 corrplot((co),method = 'square', order = 'FPC', type = 'lower', diag = FALSE)
74
75 # Now we see how much of the data is correlated and how much of it is not
76 co[upper.tri(co)] <- NA
77 co <- co - diag(nrow(co))
78 # How many are correlated strongly
79 (pres <- sum(abs(co) > 0.5, na.rm = T))#/(ncol(combn(26, 2))))
80 # How many correlations were checked
81 ncol(combn(26, 2))
82
83 # The code above shows us that out of the 325 correlations not including
       correlation with itself that only 54 correlations are present in our dataset.
       This may indicate the possibility that the PCA may not be able to decorrelate
       our data very as it is already mostly uncorrelated.
84 ```
85
86
87 ```{r PCA}
88 # Run a Principle Component Analysis
89 PC <- prcomp(X, scale. = F)
90
91 # Check levels of information
92 plot(PC$sdev[1:16]/sum(PC$sdev), pch = 19, cex = 0.7, ylab = "Proportion of
       Explained Variation", xlab = "Principle Component", type = "b")
93
94 # Check elbows
95 sum(PC$sdev[1:4])/sum(PC$sdev)
96 sum(PC$sdev[1:6])/sum(PC$sdev)
97 sum(PC$sdev[1:10])/sum(PC$sdev)
98
99 # Plot most important Principle components
100 pairs(PC$x[,1:6], pch = 16, cex = 0.75, col = colors)
101
102 # Biplot for PCA
103 autoplot(PC, data=FinData, colour=colors, loadings=TRUE, loadings.label = TRUE,
       loadings.label.size = 2, loadings.colour = 'grey', loadings.label.colour="black
       ", loadings.label.angle = 90) +
104   theme_light()
105
106 # Loadings
107 melted_cormat <- melt(PC$rotation)
108 colnames(melted_cormat) <- c("Var", "PC", "Correlation")
109 melted_cormat$Correlation <- ifelse(abs(melted_cormat$Correlation) < 0.3, 0, melted
       _cormat$Correlation)
110 ggplot(data = melted_cormat, aes(x=Var, y=PC, fill=Correlation)) +
111   geom_tile() +
112   scale_fill_viridis(discrete=FALSE)+
113   theme(axis.text.x = element_text(angle = 90))+
114   xlab("")+ ylab("")
115 ```
116
117 ```{r Robust PCA}
118 # Perform Robust PCA
119 robPCA <- PcaHubert(X)
120
121 # Rename Columns of Scores
```

```r
colnames(robPCA@scores) <- paste0("RPC", 1:6)

# Get variable importance
varimp <- summary(robPCA)@importance[2,]

# Plot Scree Plot
plot(varimp, pch = 19, cex = 0.7, ylab = "Proportion of Explained Variation", xlab = "Robust Principle Component", type = "b")

# Plot Scores
pairs(robPCA@scores, col = colors, pch = 16, cex = 0.75)

# Proper visible biplot
ggplot(data=20*robPCA$loadings, aes(PC1, PC2))+
  geom_vector( col = "grey") +
  geom_point(data=robPCA$scores, aes(x=RPC1, y=RPC2), col=colors) +
  geom_text(label = rownames(robPCA$loadings), size = 2.5)+
  xlab(paste("PC1 (", round(varimp[1]*100,1), "%)"))+
  ylab(paste("PC2 (", round(varimp[2]*100,1), "%)"))

# Outlier Map
plot(robPCA)
# Obtain outliers
FinData[which(robPCA$flag == F),]
  # Score distance > 4
#robPCA$sd
  # Orthogonal distance > 4
#robPCA$od

# Get tables of good and bad outliers to compare with original data
FinData[which(robPCA$sd > robPCA$cutoff.sd & robPCA$od > robPCA$cutoff.od), c("Name", "age", "Position", "Market Value", "OVR")]
FinData[which((robPCA$sd < robPCA$cutoff.sd & robPCA$od > robPCA$cutoff.od)| (robPCA$sd > robPCA$cutoff.sd & robPCA$od < robPCA$cutoff.od)),c("Name", "age", "Position","Market Value", "OVR")]
FinData[which(robPCA$sd > robPCA$cutoff.sd & robPCA$od < robPCA$cutoff.od),c("Name", "age", "Position","Market Value", "OVR")]
FinData[which(robPCA$sd < robPCA$cutoff.sd & robPCA$od < robPCA$cutoff.od),]
```

```{r 3D_PCA}
# Plot the PCA in 3 dimensions
pl <- scatterplot3d(PC$x[,1], PC$x[,2], PC$x[,3], color = colors, angle = 275, xlab = "PC1", ylab = "PC2", zlab = "PC3")
zz.coords <- pl$xyz.convert(PC$x[,1], PC$x[,2], PC$x[,3])
text(zz.coords$x,
     zz.coords$y,
     labels = FinData$Name,
     cex = .5, col =colors,
     pos = 4)

# Plot the RPCA in 3 dimensions
pl <- scatterplot3d(robPCA$scores[,1], robPCA$scores[,2], robPCA$scores[,3], color = colors, angle = 275, xlab = "RPC1", ylab = "RPC2", zlab = "RPC3")
zz.coords <- pl$xyz.convert(robPCA$scores[,1], robPCA$scores[,2], robPCA$scores[,3])
text(zz.coords$x,
     zz.coords$y,
     labels = FinData$Name,
     cex = .5, col =colors,
     pos = 4)
```

```
176
177
178 ```{r KPCA-Radial}
179 # We've tested polydot, vanilladot, splinedot, these all do not give a very nice
      display of the data for KPCA, but both rbfdot and laplacedot give the same type
      . And it diffferentiates quite well.
180
181 # KPCA with radial basis
182 KPC2 <- kpca(~., data = data.frame(X), kernal = "rbfdot", kpar=list(sigma=0.01))
183
184 # Which components explain the most variation
185
186
187 # Plot of KPCA
188 plot(rotated(KPC2),col=FinData$Position,
189 xlab="1st Principal Component",ylab="2nd Principal Component", cex = 0)
190 text(rotated(KPC2), labels = FinData$Name, col = colors[FinData$Position], cex =
      0.5)
191 ```
192
193 ```{r KPCA-Laplace}
194 # # KPCA with laplace
195 KPC4 <- kpca(~., data = data.frame(X), kernal = "laplacedot", kpar=list(sigma=0.01)
      )
196
197 # Which components explain the most variation
198
199 # Plot of KPCA
200 plot(rotated(KPC4),col=FinData$Position,
201 xlab="1st Principal Component",ylab="2nd Principal Component", cex = 0)
202 text(rotated(KPC4), labels = FinData$Name, col = colors[FinData$Position], cex =
      0.5)
203 ```
204
205
206 ```{r Rtsne}
207 # Run various t-sne's
208 tsne10 <- Rtsne(X,
209                 dims = 2,
210                 perplexity = 10,
211                 verbose = TRUE,
212                 max_iter = 1000)
213 tsne40 <- Rtsne(X,
214                  dims = 2,
215                  perplexity = 40,
216                  verbose = TRUE,
217                  max_iter = 1000)
218
219 tsne70 <- Rtsne(X,
220                  dims = 2,
221                  perplexity = 70,
222                  verbose = TRUE,
223                  max_iter = 1000)
224
225 tsne100 <- Rtsne(X,
226                   dims = 2,
227                   perplexity = 100,
228                   verbose = TRUE,
229                   max_iter = 1000)
230
231 # Plot t-SNE's
232 plot(tsne10$Y,
```

```
     col = colors[FinData$Position],
     t = "n",
     xlab = "t-SNE Dimension 1",
     ylab = "t-SNE Dimension 2")
text(tsne10$Y, labels = FinData$Name, col = colors[FinData$Position], cex = 0.5)

plot(tsne40$Y,
     col = colors[FinData$Position],
     t = "n",
     xlab = "t-SNE Dimension 1",
     ylab = "t-SNE Dimension 2")
text(tsne40$Y, labels = FinData$Name, col = colors[FinData$Position], cex = 0.5)

plot(tsne70$Y,
     col = colors[FinData$Position],
     t = "n",
     xlab = "t-SNE Dimension 1",
     ylab = "t-SNE Dimension 2")
text(tsne70$Y, labels = FinData$Name, col = colors[FinData$Position], cex = 0.5)

plot(tsne100$Y,
     col = colors[FinData$Position],
     t = "n",
     xlab = "t-SNE Dimension 1",
     ylab = "t-SNE Dimension 2")
text(tsne100$Y, labels = FinData$Name, col = colors[FinData$Position], cex = 0.5)
```

```
# Cluster Analysis
```{r Hierarchical Agglomerative}
# Hieararchical Clustering (Agglomerative)
cols <- c("darkgrey", "#679920", "#502491")
# Obtain the number of clusters
fviz_nbclust(robPCA$scores, hcut, method = "silhouette") +
  theme_classic()
fviz_nbclust(robPCA$scores, hcut, method = "wss")+ theme_classic() +
  geom_vline(xintercept = 3, linetype = 2)
fviz_nbclust(robPCA$scores, hcut, method = "gap_stat") +
  theme_classic()

# Obtain hierachical clustering
clusters <- hclust(dist(robPCA$scores),method ="complete" )
plot(clusters)

# Cut hierarchical tree
clusterCut <- cutree(clusters, 3)

# Plot Clusters
pairs(robPCA$scores[,1:4], col = cols[clusterCut],pch = 16, cex = 0.7)
```


```{r Kmeans}
cols <- c( "#502491","#679920", "darkgrey")
# Obtain the number of clusters
fviz_nbclust(robPCA$scores, kmeans, method = "silhouette") +
  theme_classic()
fviz_nbclust(robPCA$scores, kmeans, method = "wss")+ theme_classic() +
  geom_vline(xintercept = 3, linetype = 2)
fviz_nbclust(robPCA$scores, kmeans, method = "gap_stat") +
```
```

```
  theme_classic()

# Obtain kmeans clustering
kmeans1 <- kmeans(robPCA$scores, 3)

# Plot clusters
pairs(robPCA$scores[,1:4],col=cols[kmeans1$cluster], pch=16, cex=0.7)
#points(kmeans1$centers,col=1:4,pch=8)

# Cluster 1 is purple
# Cluster 2 is green
# Cluster 3 is grey
```


```{r Plaid Biclustering, include=FALSE}
# Obtain bi-clustering
# Doesn't have to be plaid check other methods
  # Other methods don't work and neither does Plaid
  # BCCC, BCXmotifs, BCSpectral, BCBImax, BCQuest

# Perform plaid biclustering
bi <- biclust(X, method = BCPlaid(), cluster="b",fit.model=y~m+a+b)


# See how many biclusters there are
summary(bi)

# 3 clusters
parallelCoordinates(X, bi,number = 1)
parallelCoordinates(X, bi,number = 2)
parallelCoordinates(X, bi,number = 3)

# See how plaid clusters by unimportant variables that would have been removed if
    sparsePCA was explored
drawHeatmap(X, bi, number = 1)
drawHeatmap(X, bi, number = 2)
drawHeatmap(X, bi, number = 3)
```

```{r ISA}
# Choose between these 2 seeds # 3000 does work slightly better
set.seed(3000)

# Biclustering using ISA
isa.result <- isa(X)

# Turn result into a workable bicluster
biii <- isa.biclust(isa.result)

#plotclust(X, biii)

#parallelCoordinates(X, biii,number = 1)
#parallelCoordinates(X, biii,number = 2)
#parallelCoordinates(X, biii,number = 3)

############### Defensive Biclustering
  # 14 better than 3
#drawHeatmap(X,biii, 3)
  # Both 11 and 14 are informative but maybe just use 14
#drawHeatmap(X,biii, 11)
drawHeatmap(X,biii, 14)
```

```
  # 14 better than 17
#drawHeatmap(X,biii, 17)
  # Same as 17 but less info
#drawHeatmap(X,biii, 19)

############### Attacking Biclustering
drawHeatmap(X,biii, 5)
  # 16 is better than 12
#drawHeatmap(X,biii, 12)
  # 16 better than 12
#drawHeatmap(X,biii, 13)
drawHeatmap(X,biii, 16)
# Definitely include all above

############### Misc Biclustering
  # Not sure what yellow cards and red cards tell us. Maybe more aggressive players
drawHeatmap(X,biii, 2)

############### CDM Biclustering
  # 15 gives more information
drawHeatmap(X,biii, 4)
#drawHeatmap(X,biii, 15)
#drawHeatmap(X,biii, 18)

############### CAM Biclustering
  # 10 holds more information than 1
#drawHeatmap(X,biii, 1)
#drawHeatmap(X,biii, 6)
drawHeatmap(X,biii, 10)

#kmeans1$cluster
#which(clusterCut==3)
# From the 19th bicluster onwards, biclusters were more broad and were difficult to
    interpret due to the number of players involved. So although there are 44
   biclusters available it would not be worthwhile to look any further
```

```{r MultiFA, include=FALSE}
# library(FactoMineR)
#
# resMFA <- MFA(SPCA_scores,
# group = c(3,3,3),#c(2, 2, 2, 2,2,2,2 ,2 ,2 ,2 ,2 ,2 ,2 ),
# #type = c("c", "c"),
# #ncp = 2,
# name.group = c("Group 1", "Group 2", "Group 3"),
# graph=T
# )

# plot(resMFA)
#
# plot(resMFA, choix = "ind", partial="all", cex = 0.7)
# plot(resMFA, choix = "ind", habillage="Label")
# plot(resMFA, choix = "axes")
# liste = plotMFApartial(resMFA, cex = 0.3)
# plot(resMFA,choix="ind",habillage = "Terroir")
```

```{r Fable, include=FALSE}
#BiocManager::install("fable")
#
# if (!require("BiocManager", quietly = TRUE))
#     install.packages("BiocManager")
```

```
#
# BiocManager::install("fabia")
# install.packages("fabia")
# require(fabia)
# # p relating to 3 positions
# fab <- fabia(t(X),cyc = 1000, center = 0, norm = 0)
# summary(fab)
#
# # Tried Fabia and it did not work out
# bicF <- extractBicList(data = X, biclustRes = fab, p=5, bcMethod="fabia")
# show(bicF)
#
# # ppBC(bicF,eMat=t(X), bcNum=1)
# # ppBC(bicF,eMat=t(X), bcNum=2)
# # ppBC(bicF,eMat=t(X), bcNum=3)
#
# # heatmapBC2(X,bicF,bcNum=2, N=10)
#
# plotFabia(fab, bicF, bcNum=1, plot = 1)
# plotFabia(fab, bicF, bcNum=1, plot = 2)
#
# plotFabia(fab, bicF, bcNum=2, plot = 1)
# plotFabia(fab, bicF, bcNum=2, plot = 2)
#
# plotFabia(fab, bicF, bcNum=3, plot = 1)
# plotFabia(fab, bicF, bcNum=3, plot = 2)
```

```{r MVGLM}
# Obtain original data
Data<-FinData
# Change position to numeric
Data$Position<- as.numeric(Data$Position)
# Change league to numeric
Data$League<- as.numeric(as.factor(Data$League))
# Obtain unique names of variables
names(Data) <- make.names(names(Data), unique=TRUE)

# Data that will be used for modelling
all<-Data[,c(3,35,1,2,6,8:11,15:24,27:29,33)]
# Specify link function for both response variables
fam<- c("gaussian", "gaussian") #defining the distributions of dependent variables

# Names
n0<-names(all)
# Response names
ny_all<- n0[1:2]
# Covariate names
nx_all<- n0[4:length(n0)]

# Specify model formuula
form_all<-  multivariateFormula(ny_all,nx_all)
# Use a subset for CV
sub <- sample(1:nrow(all),35,replace=FALSE)
# Obtain subset (Training set)
sub_fit <- (1:nrow(all))[-sub]

# Specify covariate design matrix of data data
X<- model.matrix(form_all, data=all)[,-1]
# Specify covariate design matrix for test data
xnew <- model.matrix(form_all, data=Data[sub,])
```

```
# Specify repsonse matrix of data
Y<-all[,ny_all]

# Perform MVGLM on training set
player_glm<- multivariateGlm.fit(Y[sub_fit,,drop=FALSE],
                                 X[sub_fit,,drop=FALSE],
                                 family=fam,size=NULL)

# Obtain coefficients
coefs <- as.matrix(sapply(player_glm,coef))

# Predict for test set
pred.glm <- multivariatePredictGlm(xnew,family=fam,beta=coefs)

# Obtain RMSE and comparison of results
sqrt(mean((Y[sub,1]-pred.glm[,1])^2))
sqrt(mean((Y[sub,2]-pred.glm[,2])^2))
MKV<-cbind(Y[sub,1],pred.glm[,1])
OVR<-cbind(Y[sub,2],pred.glm[,2])
```