**Multivariate Analysis FIFA Analysis**

**Team Members:**

Nisha Pardeshi

Madhu Atyam

Saranya Murugan

Arjuna Menon

**Project Introduction:**

FIFA 2019 dataset was taken from Kaggle website, since soccer is an interesting international game and it has multiple dimensions to perform Multivariate Analysis between players, can identify cluster among players, performance and also from management (money, profitable player) and coach(performance) perspective.

**Attributes**: ID, Name, Nationality, Club, Position, Preferred Foot, Value, Wage, Age, International Reputation, Crossing, Heading Accuracy, Dribbling, Long Passing, Ball Control, Agility, Stamina, Interceptions, Penalties, Overall.

The attributes are self-explanatory.

Dataset columns = 21

Dataset rows = 18,206

Few libraries are loaded to perform the analysis and visualizations on FIFA dataset. The following libraries are:

Importing libraries:

# install.packages("lavaanPlot")  
# install.packages("sem")  
# install.packages("pca3d")  
# install.packages("ggfortify")  
# install.packages("magick")  
# install.packages("animation")  
library(ggplot2)  
library(plotly)

library(MVA)

library(scatterplot3d)  
library(sem)  
library(semPlot)

library(mclust)

library(igraph)

library(ResourceSelection)

library(KernSmooth)

library (lavaan)

library(lavaanPlot)  
library(visdat)

library(CCA)

library(pca3d)

library(rgl)

library(ggfortify)

library(magick)

library(purrr)

library(animation)  
library(semPlot)  
library(qgraph)

**Data Cleaning & Data Visualizations**

We dropped 68 unnecessary columns which are not required for our analysis. For example, we removed balance variable since it measures an attribute of the player that is included in the ball control variable. Also, we converted the wage variable into numerical values from the previous values that were specified in terms of thousands with the “K” symbol and “M” for million in dollars. Below are the functions to convert M’s and K’s to their numeric equivalent:

convert\_money <- function(x) {  
 # Check if M exist in x  
 if(grepl('M',x)) {  
 # Convert to numeric, replace M by blank and convert to its equivalent  
 x = as.numeric(gsub('M', '', x)) \* 1000000  
 # Check if K exist in x  
 } else if (grepl('K',x)) {  
 x = as.numeric(gsub('K', '', x)) \* 1000  
 }  
}

Function to fill NAs with medians of the respective columns:

fill\_NA\_with\_median <- function(x){  
 #First convert each column into numeric if it is from factor  
 x <- as.numeric(as.character(x))   
 #Convert the item with NA to median value from the columns  
 x[is.na(x)] = median(x, na.rm=TRUE)   
 #Display the column  
 x   
}

Function to partition wages (values considered based on the range of wages present in the dataset):

wage\_break = function(row) {  
 if (row['Wage'] < 100000) {  
 x = "0-100k"  
 } else if (100000 < row['Wage'] & row['Wage'] < 200000) {  
 x = "100k-200k"  
 } else if (200000< row['Wage'] & row['Wage'] < 300000)  
 x = "200k-300k"  
 else if (300000< row['Wage'] & row['Wage'] < 400000)  
 x = "300k-400k"  
 else if (400000< row['Wage'] & row['Wage'] < 500000)  
 x = "400k-500k"  
 else  
 x = "500k+"  
 x  
}

Function to partition Overall (since we’re considering the true clusters based on Overall\_break):

overall\_break = function(row) {  
 if (row['Overall'] < 81) {  
 x = 1  
 } else if (81 < row['Overall'] & row['Overall'] < 88) {  
 x = 2  
 } else  
 x = 3  
 x  
}

Reading the dataset and saving it to fifa dataframe:

fifa <- read.csv("Final\_FIFA\_new.csv")

Checking for duplicate values:

which(duplicated(fifa)) #No duplicates

## integer(0)

Creating wage brackets for the plot containing age, performance, and wage bracket:

wage\_breaks <- c(0, 100000, 200000, 300000, 400000, 500000, Inf)  
wage\_labels <- c("0-100k", "100k-200k", "200k-300k", "300k-400k", "400k-500k", "500k+")  
wage\_brackets <- cut(x = fifa$Wage, breaks = wage\_breaks, labels = wage\_labels, include.lowest = TRUE)

Creating a new column wage\_break as a categorical variable:

fifa$wage\_break <- apply(fifa, 1, wage\_break)

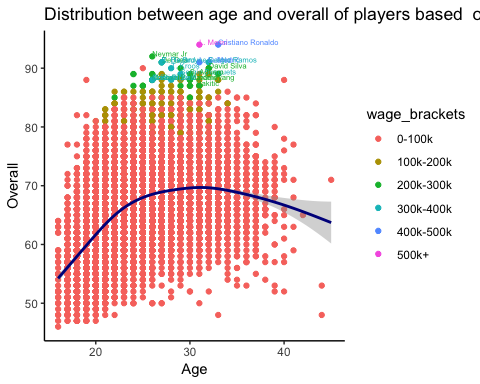
Few data visualizations are performed to understand the dataset, the plots attempted are:

* Scatterplot Matrix
* Line Plot
* Visualize Missing values by Viss Miss
* Bivariate Plot

Line Plot to identify players based on their age, performance, and wage:

g\_age\_overall <- ggplot(fifa, aes(Age, Overall, label = ifelse(Wage > 250000, Name, '')))  
g\_age\_overall + geom\_point(aes(color = wage\_brackets)) + geom\_smooth(color = "darkblue") + ggtitle("Distribution between age and overall of players based on wage bracket") + geom\_text(aes(label = ifelse(Wage > 250000,as.character(Name),''), color = wage\_brackets),hjust = 0,vjust = 0, cex = 2) +  
theme\_classic()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



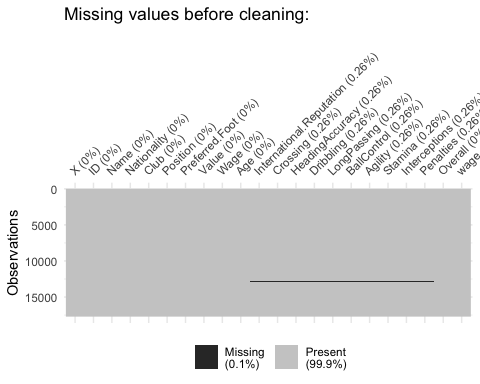
From this plot, we see that performance reduces for ages 35 and above. Hence, we only consider those players who are below the age of 35 years.

Including only players below the age of 35: So we subset the data below age 35.

fifa\_subset = subset(fifa, fifa$Age < 35)

Visual representation of the missing values before cleaning:

vis\_miss(fifa\_subset, sort\_miss = FALSE, show\_perc = TRUE, warn\_large\_data = TRUE) + labs(title = "Missing values before cleaning:")



Cleaning the dataset:

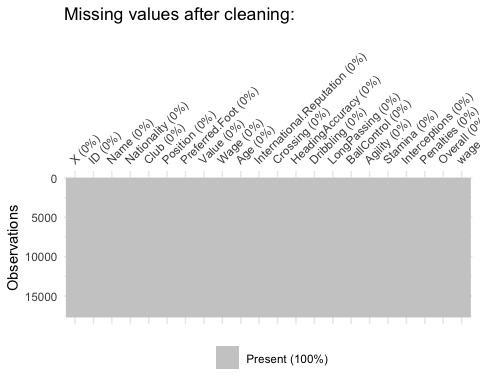
# Applying the convert\_money function to covert Ms and Ks to number format:  
fifa\_subset$Value <- sapply(fifa\_subset$Value, FUN=convert\_money)  
  
# Replacig all the NAs of the numeric columns with the median values of their respective columns:  
fifa\_subset[,c(8:21)] <- data.frame(apply(fifa\_subset[,c(8:21)],2,fill\_NA\_with\_median))

## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion  
  
## Warning in FUN(newX[, i], ...): NAs introduced by coercion

# converting only the numerical columns in the dataset to matrix format to assign the row names as their indices:  
mtrx\_fifa <- as.matrix(fifa\_subset[,c(8:21)])  
rownames(mtrx\_fifa) <- fifa\_subset$Name  
  
# Converting the matrix to data frame:  
fifa\_num\_data <- as.data.frame(mtrx\_fifa)  
  
# Scaling this data frame:  
fifa\_num\_data\_scaled <- scale(fifa\_num\_data)

Visual representation of the missing values after cleaning:

vis\_miss(fifa\_subset, sort\_miss = FALSE, show\_perc = TRUE, warn\_large\_data = TRUE) + labs(title = "Missing values after cleaning:")

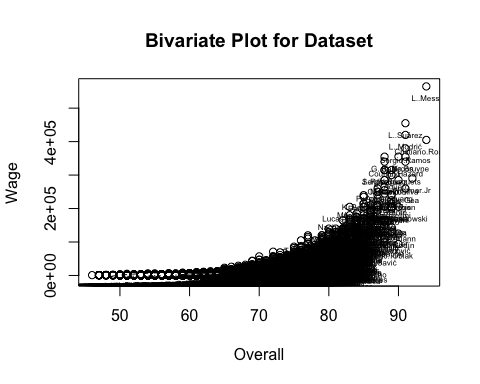


Thus, we verify that the data is a 100% clean now.

# Detecting outliers:

Bivariate boxplot for detecting outliers:

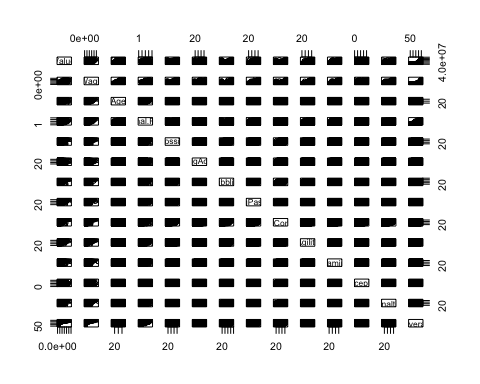
sub\_data = fifa\_num\_data[, c('Overall', 'Wage')]  
bvbox(sub\_data, main = "Bivariate Plot for Dataset", xlab = "Overall", ylab = "Wage")   
text(sub\_data, labels = rownames(sub\_data), cex = 0.5, pos = 1)



The extreme outliers here are L.Messi, L.Suarez, L.Modric, Christiano Ronaldo, Sergio Ramos, T.Kroos, K. De Bruyne, and E.Hazard. We want to analyze our data with outliers present since these are the top players and furthermore, there is not much of an improvement in the RMSE value obtained before and after the removal of outliers (as shown under factor analysis).

Scatterplot matrix:

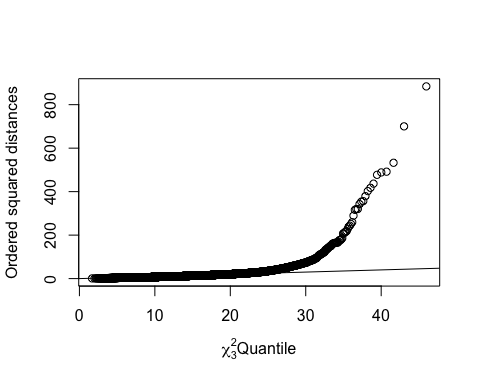
plot(fifa\_num\_data)



From this matrix of plots as well, we see that there are a few outliers.

# Testing multivariate normality:

x = fifa\_num\_data  
xbar = colMeans(x)  
s = cov(x)  
d2 = mahalanobis(x, xbar, s)  
quantiles = qchisq((1:nrow(x)-1/2)/nrow(x),df=ncol(x))  
sd2 = sort(d2)  
plot(quantiles, sd2, xlab = expression(paste(chi[3]^2,"Quantile")), ylab = "Ordered squared distances")  
abline(a = 0, b = 1) # 45 degree line originating from the origin



The data is not perfectly multivariate normally distributed due to the presence of outliers that cause deviations of the points from the line. In any case, most data in real life is not perfectly multivariate normally distributed.

Moving forward, we do not consider the variable Overall in our analysis of the dataset since it is a measure of the other varaibles that is used for representing the true clusters. We use Overall to compare our analysis with the true clusters.

**Dimension Reduction Analysis**

Multiple multivariate techniques are applied to the FIFA dataset to reduce the dimensionality of the multivariate FIFA dataset to visualize in two or three dimensions using Principal Component Analysis, Canonical Correlation Analysis, Multi-Dimensional Scaling and Exploratory Factor Analysis.

**Principal Component analysis:**

Principal Component analysis components are uncorrelated to each other and the first two or three components account for most of the variation in all the original variables. The set of variables in each component determines the interrelationships. The outcomes of the PCA are explained further.

pc <- princomp(fifa\_num\_data[, 1:13], cor = T)  
gr <- factor(fifa\_num\_data[,14])  
scores <- pc$scores  
summary(pc, loadings= T, cut = 0.2)

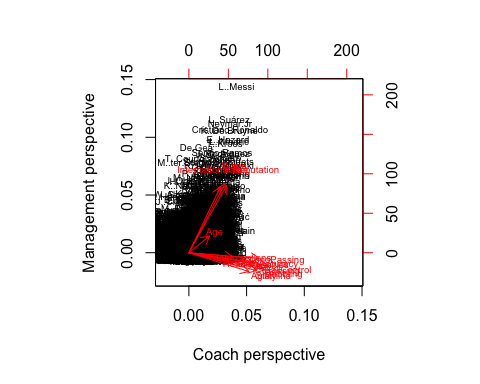
## Importance of components:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5  
## Standard deviation 2.4974208 1.4798817 1.1657424 0.93795847 0.82958975  
## Proportion of Variance 0.4797778 0.1684654 0.1045350 0.06767431 0.05293994  
## Cumulative Proportion 0.4797778 0.6482431 0.7527781 0.82045245 0.87339238  
## Comp.6 Comp.7 Comp.8 Comp.9  
## Standard deviation 0.64002030 0.58597332 0.47049687 0.44986881  
## Proportion of Variance 0.03150969 0.02641267 0.01702825 0.01556784  
## Cumulative Proportion 0.90490208 0.93131475 0.94834300 0.96391084  
## Comp.10 Comp.11 Comp.12 Comp.13  
## Standard deviation 0.42390721 0.36434662 0.335710937 0.209789017  
## Proportion of Variance 0.01382287 0.01021142 0.008669372 0.003385495  
## Cumulative Proportion 0.97773371 0.98794513 0.996614505 1.000000000  
##   
## Loadings:  
## Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7  
## Value 0.540 0.312  
## Wage 0.556 0.329  
## Age -0.419 -0.847   
## International.Reputation 0.529 -0.286 -0.771  
## Crossing 0.350 0.275   
## HeadingAccuracy 0.273 -0.322 -0.666   
## Dribbling 0.362 0.235   
## LongPassing 0.335 0.296 -0.493 0.249  
## BallControl 0.378   
## Agility 0.286 0.365 0.336 0.415 -0.263  
## Stamina 0.322 0.567   
## Interceptions 0.210 -0.617 0.335 0.228   
## Penalties 0.305 0.248 -0.279 -0.452   
## Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13  
## Value 0.693   
## Wage -0.665   
## Age   
## International.Reputation   
## Crossing -0.714 0.453   
## HeadingAccuracy -0.427 0.239 0.320   
## Dribbling -0.202 -0.504 0.677   
## LongPassing 0.484 0.325 0.294   
## BallControl 0.211 -0.491 -0.718   
## Agility -0.474 0.357   
## Stamina 0.653   
## Interceptions -0.537 -0.257   
## Penalties 0.293 0.248 -0.599

loadings = pc$loadings

Hence, 3 principal components are chosen as they account for 75.28% of the variation in our data. Principal component 1 is a measure of crossing, dribbling, long passing, ball control,stamina, and penalties. Hence, we name this component as coach perspective. Principal component 2 is a measure of value, wage, and international reputation. Hence, we name this component as management perspective since these are the dimensions that the management cares about. Principal component 3 is a measure of the contrast between (age, heading accuracy, interceptions) and agility. Hence, we name this component as player experience since experience of a players improves with an increase in age, which also improves heading accuracy and interceptions, while agility reduces over time.

Biplot to detect outliers based on the first 2 principal components:

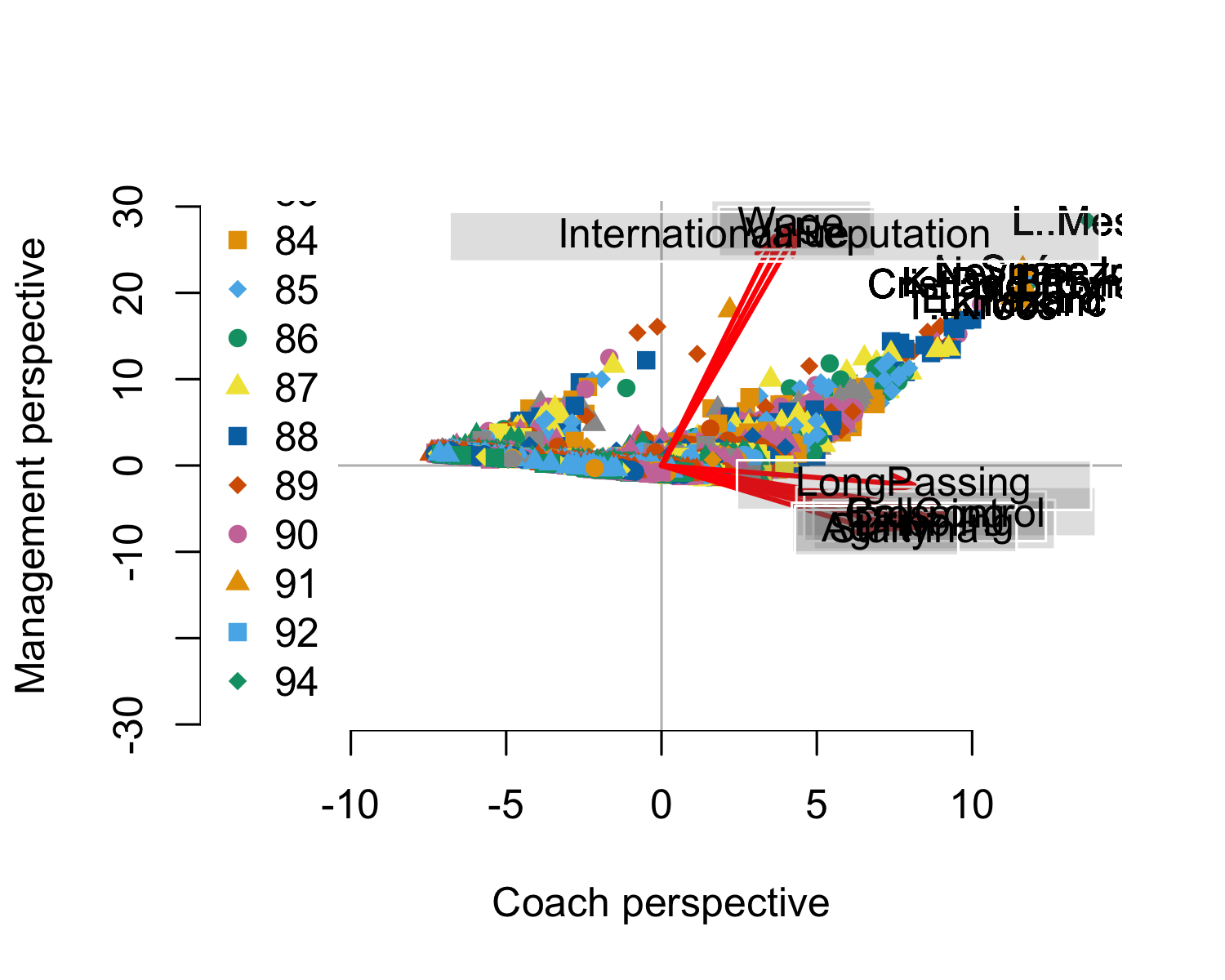
biplot(pc, col=c("black", "red"), cex = 0.6, xlab = "Coach perspective", ylab = "Management perspective")



The extreme outliers detected here are L.Messi, L.Suarez, Neymar.Jr, Christiano Ronaldo, K. De Bruyne, E.Hazard, L.Modric, and T.Kroos.

Biplot between the first 2 principal components wherein the points represent the overall of the players:

#lab <- c("L..Messi", "L..Su?.rez", "Neymar.Jr", "Cristiano.Ronaldo", "K..De.Bruyne", "E..Hazard", "L..Modri?.", "T..Kroos") # For windows  
lab <- c("L..Messi", "L..Suárez", "Neymar.Jr", "Cristiano.Ronaldo", "K..De.Bruyne", "E..Hazard", "L..Modrić", "T..Kroos") # For Mac  
out\_players <- match(lab, rownames(fifa\_num\_data))  
pca2d(pc, group=gr, biplot=TRUE, legend="bottomleft", show.labels = ifelse(row(fifa\_num\_data) %in% out\_players, rownames(fifa\_num\_data),''), axe.titles = c("Coach perspective", "Management perspective"))



3D plot of the principal components (This works only on MAC and not Windows):

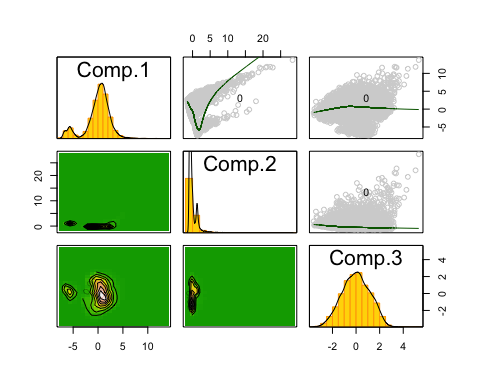
A close up of a map

Description automatically generated

Contour plot using kdepairs:

library(ResourceSelection)  
kdepairs(pc$scores[,1:3])

**kdepairs Contour Plot**



This shows us that the correlation between principal components is zero and that there are 2 clusters. However, during clustering we find that there are 3 clusters. This discrepancy is due to the fact that the first principal components account for only 75.28%, and not 100%, of the true variation in the data.

# 

# Canonical Correlation Analysis:

We perform Canonical Correlation Analysis for FIFA dataset to identify the correlation between the Coach perspective and management perspective of the player. Below is the analysis to identify the relationship.

# CCA:  
X <- fifa\_num\_data[, 1:4] # Management perspective considering value, wage, age and international reputation  
Y <- fifa\_num\_data[, 5:13] # Coach perspective considering all game skills  
  
cca\_fifa = cc(X,Y)  
cca\_fifa$xcoef

## [,1] [,2] [,3]  
## Value -6.599340e-08 -1.730775e-07 2.353602e-07  
## Wage -3.862173e-06 4.125367e-06 -2.811942e-05  
## Age -1.779777e-01 1.348751e-01 8.578304e-02  
## International.Reputation -2.818369e-01 1.530235e-01 -3.169737e+00  
## [,4]  
## Value 1.983501e-07  
## Wage -8.740481e-05  
## Age 9.087236e-03  
## International.Reputation 2.034867e+00

cca\_fifa$ycoef

## [,1] [,2] [,3] [,4]  
## Crossing -0.029142376 0.046013377 -0.009434021 -0.023484354  
## HeadingAccuracy -0.004442152 0.017148867 -0.012722547 -0.021187644  
## Dribbling 0.080465950 -0.058566050 -0.039608227 -0.012912349  
## LongPassing -0.038165449 -0.013541868 0.014475711 0.074446965  
## BallControl -0.031990871 -0.068212426 0.020078192 -0.056510888  
## Agility 0.004699544 0.004383247 0.008577519 0.008613359  
## Stamina -0.004532142 0.012023598 0.099860957 0.014765018  
## Interceptions -0.005571711 0.013659749 -0.032465459 -0.032074081  
## Penalties -0.042998657 0.036405620 -0.036653916 0.054371383

cca\_fifa$xcoef[,1]/min(cca\_fifa$xcoef[,1]) #Since xcoef[, 1] contains only negative values

## Value Wage Age   
## 2.341546e-07 1.370357e-05 6.314917e-01   
## International.Reputation   
## 1.000000e+00

cca\_fifa$ycoef[,1]/max(cca\_fifa$ycoef[,1]) #Since ycoef[, 1] contains both positive and negative values

## Crossing HeadingAccuracy Dribbling LongPassing   
## -0.36217028 -0.05520536 1.00000000 -0.47430559   
## BallControl Agility Stamina Interceptions   
## -0.39757029 0.05840413 -0.05632373 -0.06924309   
## Penalties   
## -0.53437084

cca\_fifa$cor[1]

## [1] 0.4299911

x = cca\_fifa$scores$xscores  
y = cca\_fifa$scores$yscores  
cor(x[,1],y[,1])

## [1] 0.4299911

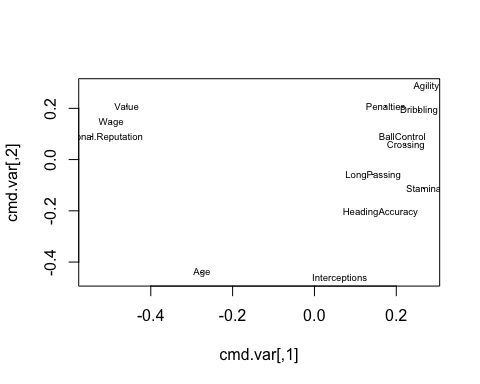
Hence, the maximum correlation between the management and coach’s perspective is 42.9% where from a management perspective, they are most concerned about the international reputation of the players, followed by age. From a coach’s perspective, the most important skill required of the players is dribbling. We can say that the correlation here is not greater than 50% since the management and coach do not always see eye to eye since they consider different criteria more important during the selection of players.

# 

# Multidimensional scaling between variables:

The Multidimensional Scaling constructs a map and an easier interpretation for showing the relationship between the variables.

dist.var <- 1 - cor(fifa\_num\_data[, 1:13])  
cmd.var = cmdscale(dist.var)  
plot(cmd.var, pch = ".")  
text(cmd.var, labels = colnames(fifa\_num\_data[, 1:13]), cex = 0.6)

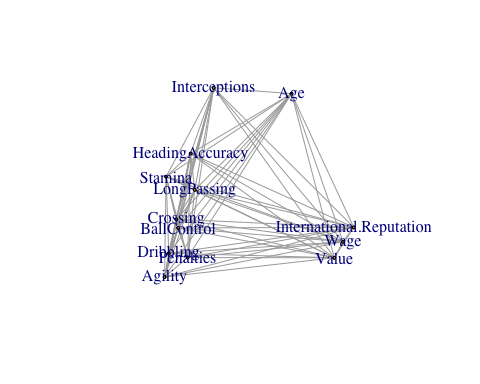


An easier to interpret visualization of the distance between the variables is as follows:

g <- graph.full(nrow(dist.var))  
V(g)$label <- colnames(fifa\_num\_data)

## Warning in vattrs[[name]][index] <- value: number of items to replace is  
## not a multiple of replacement length

layout <- layout.mds(g, dist = as.matrix(dist.var))  
plot(g, layout = layout, vertex.size = 3)



We see that the management perspective variables except for age are close to each other and the coach perspective variables except for interceptions are close to each other. Hence, we can say that the variables Age and Interceptions will have low loading coefficients on the first 2 principal components (as seen above during principal component analysis) and low factor loadings on the first 2 factors (as seen below in exploratory factor analysis).

# Exploratory factor analysis:

Exploratory Factor Analysis is used to identify the latent variables related to the manifest variables from the factors without assumptions. Based on this analysis few latent variables are determined.

fa\_fifa <- factanal(fifa\_num\_data[,1:13], factors = 3)  
fa\_fifa

##   
## Call:  
## factanal(x = fifa\_num\_data[, 1:13], factors = 3)  
##   
## Uniquenesses:  
## Value Wage Age   
## 0.153 0.128 0.928   
## International.Reputation Crossing HeadingAccuracy   
## 0.451 0.225 0.520   
## Dribbling LongPassing BallControl   
## 0.048 0.287 0.056   
## Agility Stamina Interceptions   
## 0.397 0.348 0.005   
## Penalties   
## 0.343   
##   
## Loadings:  
## Factor1 Factor2 Factor3  
## Value 0.155 0.902   
## Wage 0.116 0.919 0.115   
## Age 0.139 0.225   
## International.Reputation 0.727 0.101   
## Crossing 0.815 0.112 0.312   
## HeadingAccuracy 0.492 0.480   
## Dribbling 0.956 0.120 0.156   
## LongPassing 0.654 0.169 0.506   
## BallControl 0.915 0.157 0.288   
## Agility 0.773   
## Stamina 0.630 0.502   
## Interceptions 0.130 0.988   
## Penalties 0.796 0.150   
##   
## Factor1 Factor2 Factor3  
## SS loadings 4.780 2.326 2.004  
## Proportion Var 0.368 0.179 0.154  
## Cumulative Var 0.368 0.547 0.701  
##   
## Test of the hypothesis that 3 factors are sufficient.  
## The chi square statistic is 15089.03 on 42 degrees of freedom.  
## The p-value is 0

f\_loading = fa\_fifa$loadings[, 1:3]  
corHat = f\_loading %\*% t(f\_loading) + diag(fa\_fifa$uniquenesses)  
corr <- cor(fifa\_num\_data[,1:13])  
# Check the root mean square error:  
rmse = sqrt(mean((corHat-corr)^2))  
rmse

## [1] 0.03889484

We chose 3 factors here since 2 factors gave an RMSE of greater than 0.05. The p-value of 0 obtained here could be due to the large sample size and hence, we rely on RMSE (which is not affected by the sample size) instead. **Factor 1 is the coach perspective**. **Factor 2 is the management perspective**. **Facotr 3 is the player experience.**

Performing factor analysis after the removal of extreme outliers found via biplot (We prefer the outliers found via biplot rather than those found via bivariate boxplot since the biplot considers almost all the variables while the bivariate boxplot only considers 2 variables for plotting):

#out\_lab <- c("L..Messi", "L..Su?.rez", "Neymar.Jr", "Cristiano.Ronaldo", "K..De.Bruyne", "E..Hazard", "L..Modri?.", "T..Kroos") #For Windows  
out\_lab <- c("L..Messi", "L..Suárez", "Neymar.Jr", "Cristiano.Ronaldo", "K..De.Bruyne", "E..Hazard", "L..Modrić", "T..Kroos") #For Mac  
out\_players <- match(out\_lab, rownames(fifa\_num\_data))  
out\_fifa <- fifa\_num\_data[-out\_players, 1:13]  
fa\_fifa1 <- factanal(out\_fifa, factors = 3)  
f\_loading1 = fa\_fifa1$loadings[, 1:3]  
corHat = f\_loading1 %\*% t(f\_loading1) + diag(fa\_fifa1$uniquenesses)  
corr <- cor(out\_fifa)  
# Check the root mean square error:  
rmse1 = sqrt(mean((corHat-corr)^2))  
rmse1

## [1] 0.03888102

The RMSE values before and after the removal of the extreme outliers is 0.038895 and 0.038881. Hence, there is not much of an improvement (only 0.000014 reduction in RMSE value).

# 

**Cluster Analysis:**

Cluster Analysis is performed to observe the groups that are homogeneous and separated from other groups. Some clustering methods are performed to understand the players significance groups. Below are the following cluster analysis performed:

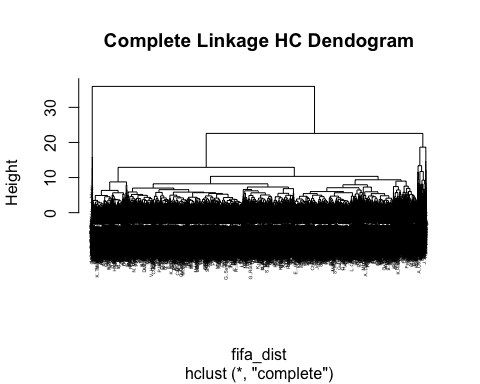
* Hierarchal Clustering
* K-Means Clustering
* Model based Clustering

Creating the categorical variable Overall\_break for true clusters:

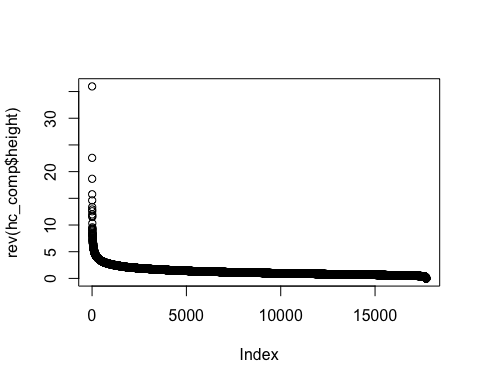
fifa\_num\_data$Overall\_break <- apply(fifa\_num\_data, 1, overall\_break)

1. **Hierarchical clustering by complete linkage:**

fifa\_dist = dist(fifa\_num\_data\_scaled[, 1:13])  
hc\_comp <- hclust(fifa\_dist)  
plot(hc\_comp, main = "Complete Linkage HC Dendogram", cex = 0.3) # Ineligible since too much data is present here.



plot(rev(hc\_comp$height)) #3 clusters



Three clusters are identified

ct\_comp = cutree(hc\_comp,3)  
table(ct\_comp)

## ct\_comp  
## 1 2 3   
## 15 463 17240

tb\_comp = table(ct\_comp, fifa\_num\_data$Overall\_break)  
tb\_comp

##   
## ct\_comp 1 2 3  
## 1 0 0 15  
## 2 166 215 82  
## 3 17154 51 35

chisq.test(tb\_comp)

## Warning in chisq.test(tb\_comp): Chi-squared approximation may be incorrect

##   
## Pearson's Chi-squared test  
##   
## data: tb\_comp  
## X-squared = 10436, df = 4, p-value < 2.2e-16

The high chi-squared value and the very low p-value less than 0.05 indicate that clustering is good here.

Interpreting the meanings of the clusters:

colMeans(subset(fifa\_num\_data\_scaled, ct\_comp ==1))

## Value Wage Age   
## 13.3848081 15.9826322 0.9653349   
## International.Reputation Crossing HeadingAccuracy   
## 7.8964883 1.7417812 0.8076190   
## Dribbling LongPassing BallControl   
## 1.7260069 1.8683440 1.9995118   
## Agility Stamina Interceptions   
## 1.5318429 1.0475833 0.2716045   
## Penalties Overall   
## 1.9524539 3.4944019

**Cluster 1** contains players having higher values in all variables. Thus, these players perform exceptionally both from a management and coach standpoint.

colMeans(subset(fifa\_num\_data\_scaled, ct\_comp ==2))

## Value Wage Age   
## 3.7209137 3.6786984 0.6515516   
## International.Reputation Crossing HeadingAccuracy   
## 4.1848444 0.9558951 0.8677750   
## Dribbling LongPassing BallControl   
## 1.0092069 1.1221745 1.1689919   
## Agility Stamina Interceptions   
## 0.5461391 0.6771873 0.6493720   
## Penalties Overall   
## 1.0174781 2.2327747

**Cluster 2** contains players having medium values in all variables. Thus, these players have an average performance both from a management and coach perspective.

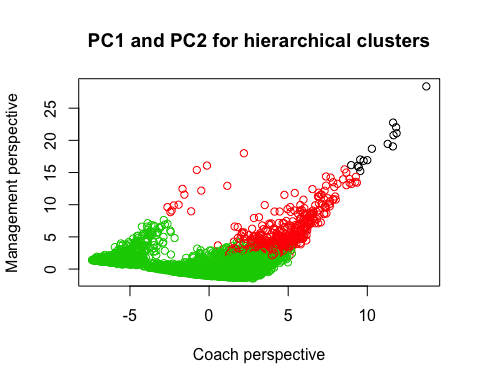
colMeans(subset(fifa\_num\_data\_scaled, ct\_comp ==3))

## Value Wage Age   
## -0.11157513 -0.11270167 -0.01833808   
## International.Reputation Crossing HeadingAccuracy   
## -0.11925930 -0.02718713 -0.02400778   
## Dribbling LongPassing BallControl   
## -0.02860516 -0.03176287 -0.03313433   
## Agility Stamina Interceptions   
## -0.01600000 -0.01909811 -0.01767595   
## Penalties Overall   
## -0.02902431 -0.06300410

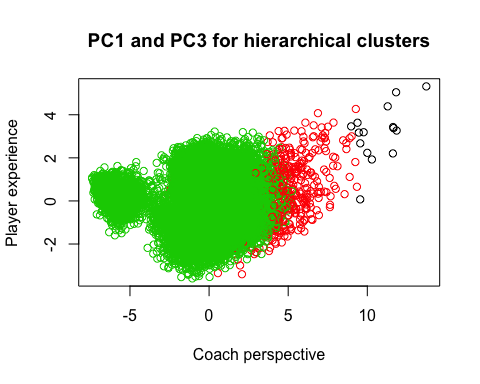
**Cluster 3** contains players having low values in all variables. Thus, these players perform poorly overall.

**Visualizing the clusters:**

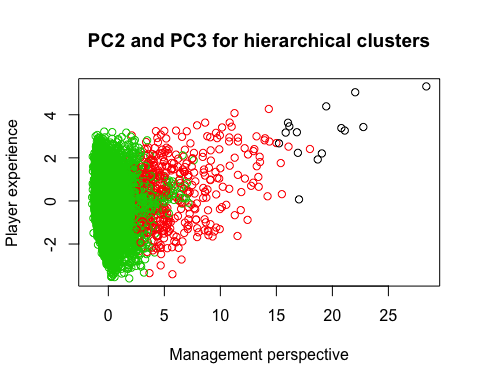
plot(pc$scores[, c(1,2)], col = ct\_comp, main = "PC1 and PC2 for hierarchical clusters", xlab = "Coach perspective", ylab = "Management perspective")



plot(pc$scores[, c(1,3)], col = ct\_comp, main = "PC1 and PC3 for hierarchical clusters", xlab = "Coach perspective", ylab = "Player experience")



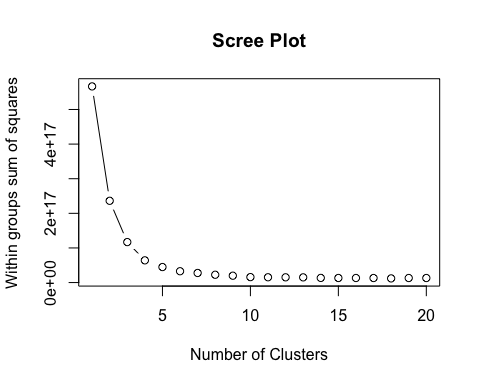
plot(pc$scores[, c(2,3)], col = ct\_comp, main = "PC2 and PC3 for hierarchical clusters", xlab = "Management perspective", ylab = "Player experience")

 There is proper separation between the clusters here in all cases.

1. **K-Means Clustering:**

# Elbow Test:  
plot.wgss = function(mydata, maxc) {  
 wss = numeric(maxc)  
 for (i in 1:maxc)   
 wss[i] = kmeans(mydata, centers=i, nstart = 10)$tot.withinss   
 plot(1:maxc, wss, type="b", xlab="Number of Clusters",  
 ylab="Within groups sum of squares", main="Scree Plot")   
}  
  
plot.wgss(fifa\_num\_data, 20)

## Warning: did not converge in 10 iterations



From this scree plot, we see that there are 3 clusters.

Applying k-means for 3 clusters:

km <- kmeans(fifa\_num\_data\_scaled[, 1:13], 3, nstart = 10)  
table(km$cluster)

##   
## 1 2 3   
## 1234 14477 2007

tb\_km = table(km$cluster, fifa\_num\_data$Overall\_break)  
tb\_km

##   
## 1 2 3  
## 1 874 236 124  
## 2 14474 1 2  
## 3 1972 29 6

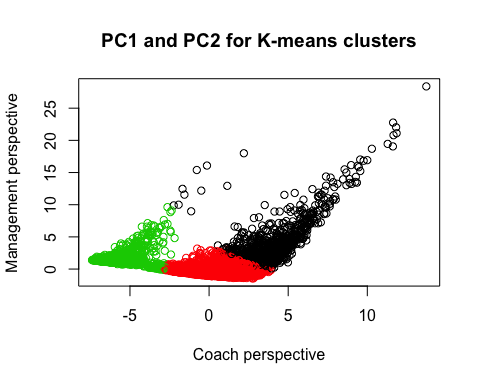
chisq.test(tb\_km)

##   
## Pearson's Chi-squared test  
##   
## data: tb\_km  
## X-squared = 4410.2, df = 4, p-value < 2.2e-16

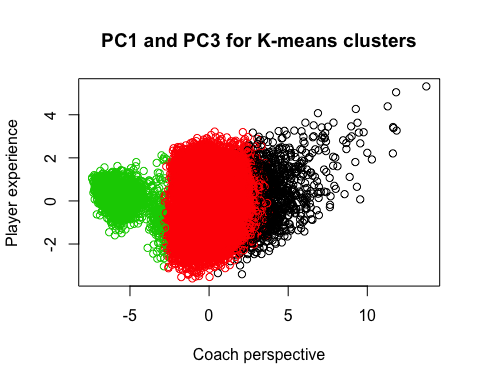
This has a lower chi-squared value than that obtained by hierarchical clustering.

Visualizing the clusters:

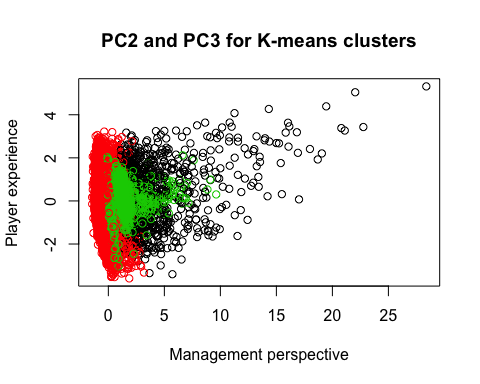
plot(pc$scores[, c(1,2)], col = km$cluster, main = "PC1 and PC2 for K-means clusters", xlab = "Coach perspective", ylab = "Management perspective")



plot(pc$scores[, c(1,3)], col = km$cluster, main = "PC1 and PC3 for K-means clusters", xlab = "Coach perspective", ylab = "Player experience")



plot(pc$scores[, c(2,3)], col = km$cluster, main = "PC2 and PC3 for K-means clusters", xlab = "Management perspective", ylab = "Player experience")

There is overlap between the clusters in the case of PC2 vs PC3 here. Hence, we can say that hierarchical clustering provides better results than K-means.

1. **Model-based clustering:**

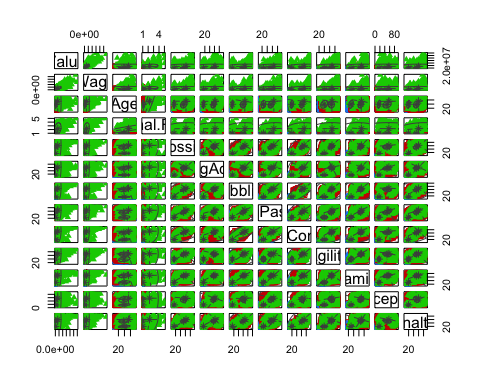
mc <- Mclust(fifa\_num\_data[,1:13], 3)  
summary(mc)

## ----------------------------------------------------   
## Gaussian finite mixture model fitted by EM algorithm   
## ----------------------------------------------------   
##   
## Mclust VEV (ellipsoidal, equal shape) model with 3 components:   
##   
## log-likelihood n df BIC ICL  
## -1074050 17718 290 -2150936 -2150939  
##   
## Clustering table:  
## 1 2 3   
## 1738 14467 1513

table(mc$classification, fifa\_num\_data$Overall\_break)

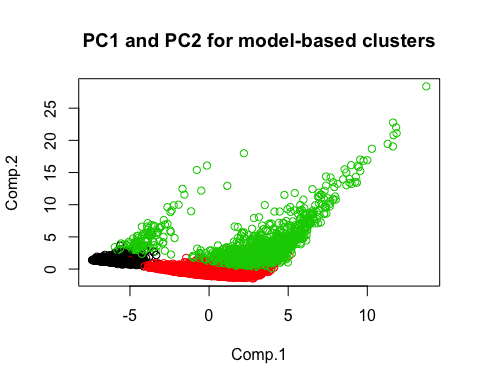
##   
## 1 2 3  
## 1 1737 1 0  
## 2 14454 8 5  
## 3 1129 257 127

plot(mc, what = "classification")

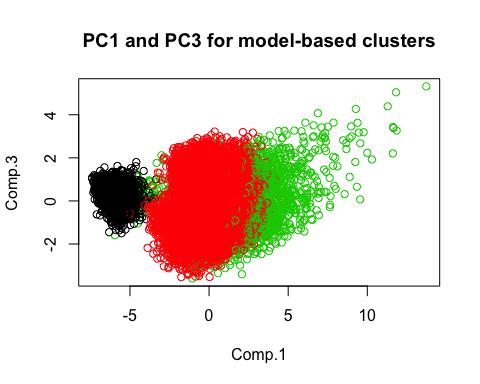
The very low values of log-likelihood and BIC indicate that this is a bad clustering model.

**Visualizing the clusters:**

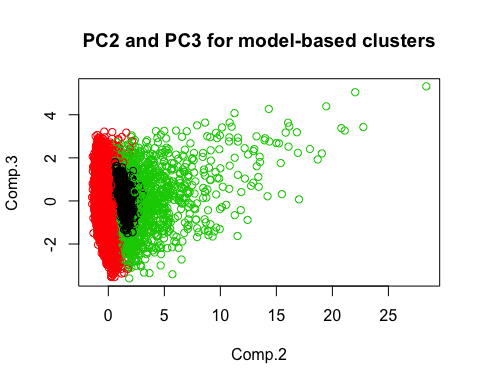
plot(pc$scores[, c(1,2)], col = mc$classification, main = "PC1 and PC2 for model-based clusters")



plot(pc$scores[, c(1,3)], col = mc$classification, main = "PC1 and PC3 for model-based clusters")

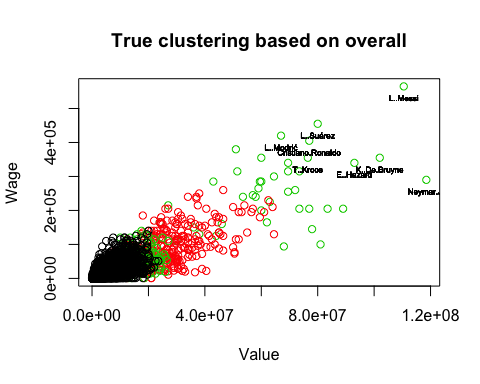


plot(pc$scores[, c(2,3)], col = mc$classification, main = "PC2 and PC3 for model-based clusters")

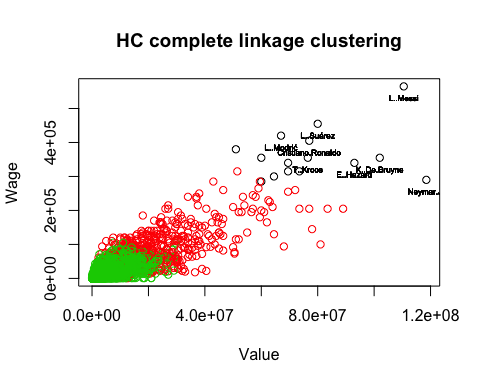
 There is overlap between the clusters in the cases of PC1 vs PC3 and PC2 vs PC3 here. Hence, this method of clustering is worse than both K-means and hierarchical clustering in this case.

Now, visually comparing the clusters obtained by the above 3 methods with the true clusters:

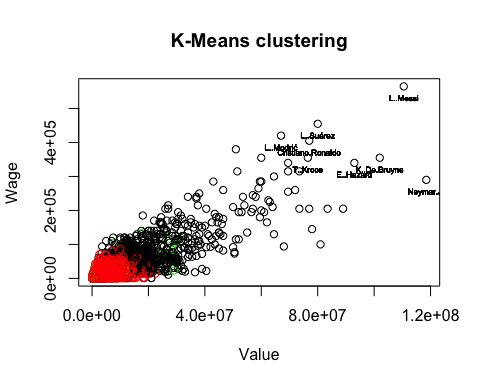
plot(fifa\_num\_data$Value,fifa\_num\_data$Wage, col = fifa\_num\_data$Overall\_break, main = "True clustering based on overall", xlab = "Value", ylab = "Wage")  
text(fifa\_num\_data, labels = ifelse(row(fifa\_num\_data) %in% out\_players, rownames(fifa\_num\_data),''), cex = 0.5, pos = 1)



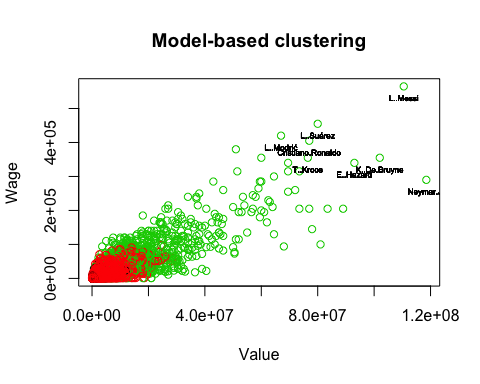
plot(fifa\_num\_data$Value,fifa\_num\_data$Wage, col = ct\_comp, main = "HC complete linkage clustering", xlab = "Value", ylab = "Wage")  
text(fifa\_num\_data, labels = ifelse(row(fifa\_num\_data) %in% out\_players, rownames(fifa\_num\_data),''), cex = 0.5, pos = 1)



plot(fifa\_num\_data$Value,fifa\_num\_data$Wage, col = km$cluster, main = "K-Means clustering", xlab = "Value", ylab = "Wage")  
text(fifa\_num\_data, labels = ifelse(row(fifa\_num\_data) %in% out\_players, rownames(fifa\_num\_data),''), cex = 0.5, pos = 1)



plot(fifa\_num\_data$Value,fifa\_num\_data$Wage, col = mc$classification, main = "Model-based clustering", xlab = "Value", ylab = "Wage")  
text(fifa\_num\_data, labels = ifelse(row(fifa\_num\_data) %in% out\_players, rownames(fifa\_num\_data),''), cex = 0.5, pos = 1)

 Hence, we visually verify that hierachical clustering is the best method for our dataset.

**Factory Analysis**

**Confirmatory Factor Analysis:**

Confirmatory Factor Analysis is performed using lavaan project. In Confirmatory Factor Analysis, we assume that the factors build in EFA are correlated to each other. Here, Factor 1 represents skills, Factor 2 represents Management and Factor 3 represents the Experience.

print(fa\_fifa$loading, cut = 0.2) # Looking at the factors obtained from EFA to help decide our factors for CFA

##   
## Loadings:  
## Factor1 Factor2 Factor3  
## Value 0.902   
## Wage 0.919   
## Age 0.225   
## International.Reputation 0.727   
## Crossing 0.815 0.312   
## HeadingAccuracy 0.492 0.480   
## Dribbling 0.956   
## LongPassing 0.654 0.506   
## BallControl 0.915 0.288   
## Agility 0.773   
## Stamina 0.630 0.502   
## Interceptions 0.988   
## Penalties 0.796   
##   
## Factor1 Factor2 Factor3  
## SS loadings 4.780 2.326 2.004  
## Proportion Var 0.368 0.179 0.154  
## Cumulative Var 0.368 0.547 0.701

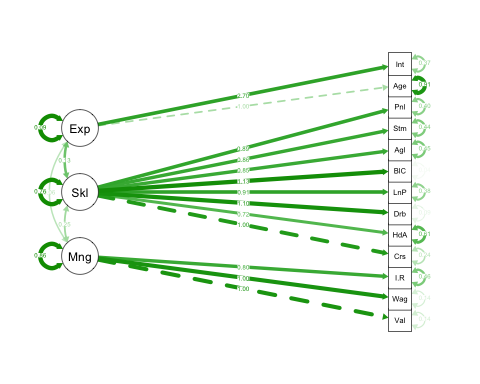
**Building the CFA model**

HS.model <- 'Management =~ Value + Wage + International.Reputation  
Skills =~ Crossing + HeadingAccuracy + Dribbling + LongPassing + BallControl + Agility + Stamina + Penalties  
Experience =~ Age + Interceptions'  
fifa\_sem <- cfa(HS.model, data=fifa\_num\_data\_scaled)  
summary(fifa\_sem, fit.measures=TRUE)

## lavaan 0.6-3 ended normally after 53 iterations  
##   
## Optimization method NLMINB  
## Number of free parameters 29  
##   
## Number of observations 17718  
##   
## Estimator ML  
## Model Fit Test Statistic 36295.906  
## Degrees of freedom 62  
## P-value (Chi-square) 0.000  
##   
## Model test baseline model:  
##   
## Minimum Function Test Statistic 207117.910  
## Degrees of freedom 78  
## P-value 0.000  
##   
## User model versus baseline model:  
##   
## Comparative Fit Index (CFI) 0.825  
## Tucker-Lewis Index (TLI) 0.780  
##   
## Loglikelihood and Information Criteria:  
##   
## Loglikelihood user model (H0) -241412.286  
## Loglikelihood unrestricted model (H1) -223264.333  
##   
## Number of free parameters 29  
## Akaike (AIC) 482882.571  
## Bayesian (BIC) 483108.259  
## Sample-size adjusted Bayesian (BIC) 483016.099  
##   
## Root Mean Square Error of Approximation:  
##   
## RMSEA 0.182  
## 90 Percent Confidence Interval 0.180 0.183  
## P-value RMSEA <= 0.05 0.000  
##   
## Standardized Root Mean Square Residual:  
##   
## SRMR 0.080  
##   
## Parameter Estimates:  
##   
## Information Expected  
## Information saturated (h1) model Structured  
## Standard Errors Standard  
##   
## Latent Variables:  
## Estimate Std.Err z-value P(>|z|)  
## Management =~   
## Value 1.000   
## Wage 1.004 0.006 175.347 0.000  
## Intrntnl.Rpttn 0.798 0.006 123.771 0.000  
## Skills =~   
## Crossing 1.000   
## HeadingAccurcy 0.720 0.007 96.149 0.000  
## Dribbling 1.099 0.005 202.899 0.000  
## LongPassing 0.909 0.007 136.628 0.000  
## BallControl 1.127 0.005 215.840 0.000  
## Agility 0.850 0.007 122.392 0.000  
## Stamina 0.857 0.007 124.083 0.000  
## Penalties 0.892 0.007 132.308 0.000  
## Experience =~   
## Age 1.000   
## Interceptions 2.699 0.134 20.175 0.000  
##   
## Covariances:  
## Estimate Std.Err z-value P(>|z|)  
## Management ~~   
## Skills 0.253 0.007 37.478 0.000  
## Experience 0.061 0.004 15.574 0.000  
## Skills ~~   
## Experience 0.130 0.007 19.516 0.000  
##   
## Variances:  
## Estimate Std.Err z-value P(>|z|)  
## .Value 0.144 0.004 38.730 0.000  
## .Wage 0.137 0.004 36.967 0.000  
## .Intrntnl.Rpttn 0.455 0.005 85.493 0.000  
## .Crossing 0.244 0.003 87.187 0.000  
## .HeadingAccurcy 0.608 0.007 92.735 0.000  
## .Dribbling 0.086 0.001 66.467 0.000  
## .LongPassing 0.376 0.004 90.497 0.000  
## .BallControl 0.040 0.001 39.579 0.000  
## .Agility 0.454 0.005 91.514 0.000  
## .Stamina 0.444 0.005 91.409 0.000  
## .Penalties 0.399 0.004 90.842 0.000  
## .Age 0.914 0.011 86.988 0.000  
## .Interceptions 0.374 0.030 12.668 0.000  
## Management 0.856 0.011 77.531 0.000  
## Skills 0.756 0.010 72.976 0.000  
## Experience 0.086 0.006 14.593 0.000

The SRMR value of 0.08 is greater than 0.05. Similarly, by performing CFA using SEM, we obtained a value greater than 0.05 for SRMR and values less than 0.95 for GFI and AGFI. Hence, this model is not supported by the data.

semPaths(fifa\_sem, rotation = 2, 'std', 'est')



It can be seen that the variable Age has the highest uniqueness and lowest loading value.

Below is the easy representation of the sem path with abbreviated variables.

A screenshot of a cell phone

Description automatically generated

A close up of a device

Description automatically generated

A picture containing screenshot

Description automatically generated

**Conclusion**

From the analysis, the players are identified by their Game Skills(Long Passing, Dribbling, Ball Control, Stamina) , Management Perspective(Value, Wage, International Reputation) and Player Experience(Age, Heading Accuracy, Interceptions). The top interested players from management and coach perspective are L..Messi, L..Suárez, Neymar.Jr, Cristiano.Ronaldo, K..De.Bruyne, E..Hazard, L..Modrić, T..Kroos.

All these players are top performers and few players like Messi and Cristiano Ronaldo have high wages. Using, Hierarchal clustering complete linkage, we could identify the proper clustering compare to other cluster models. These dimension reduction techniques performed in the above analysis has helped to visualize the relations among the variables for the multi dimension data.

As these analyses has found the latent variables which helps to understand the recruiter on what basis the players are selected for and what are the other correlated factors which can influence the overall balance of a team. Since for a team it is always required to have a reputation and as well performers.