EDA Homework 2 – Technical Documentation

Team 13: Divya Upadhyay, Muyu Yan, Phoebe Chang, Samira Palagiri, Siddhesh Kothari, Sydney Jiang

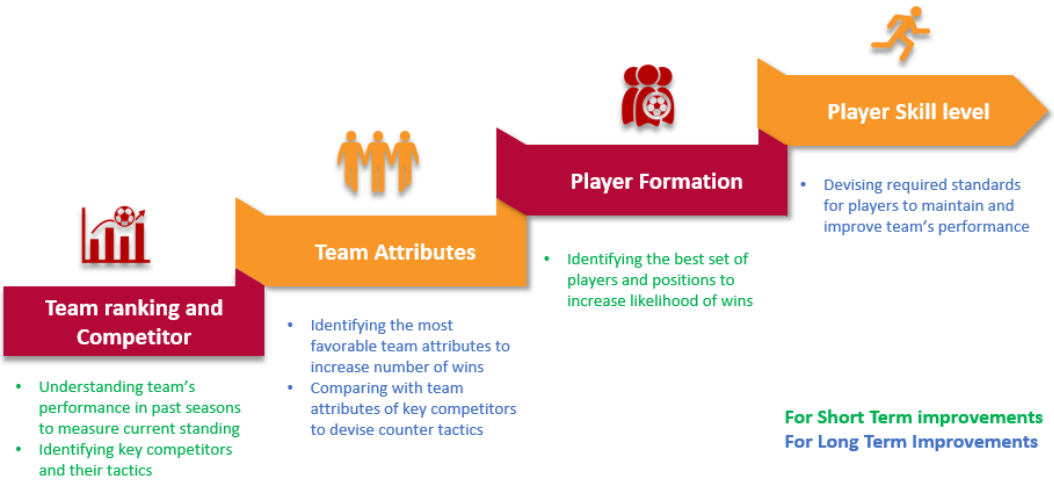
October 3, 2018

Introduction:

Associazione Sportiva Roma (A.S. Roma) has always been amongst the fan favorites to top the Italian football league *Serie A* [1]. In spite of this perception, Roma has topped the league only three times till date [2]. Roma often finishes amongst the top three teams with the likes of Juventus (32 titles), Inter Milan (18 titles) and A.C. Milan (18 titles) [3]. The title counts of these teams far surpasses Roma's count, even though Roma often ranks higher than most of these teams on the final leaderboard. Our objective of this analysis is to investigate and dig out insights that will inform the game strategy, team composition and strong player attributes to increase the number of wins in the coming seasons. These wins will translate into more points and accelerate the team to the top of the leaderboard thereby enabling us to bring the title home to *Stadio Olimpico*.

Approach and Assumptions:

Based on the match-level, player-level and team-level data available we attempted to identify a two-step approach to improve the team wins. The short term recommendations that would include immediate changes we can implement while the long term recommendations demand a long term investment in time and training to improve the overall team performance and translate into future wins and titles.



Key Assumptions:

1. Based on the matches present in the dataset from 2008/2009 season 2015/2016 season, we are assuming that our recommendations will be helping the coach of AS Roma to take informed decisions for the upcoming season of 2016/2017
2. Since, AS Roma is playing under the Italy Serie A league, we are analyzing matches for the Italy Serie A league only. As different country leagues have different playing tactics, official rankings, fans, or world recognition, we are looking into improving AS Roma's performance as compared to other Italian clubs in this analysis. The dataset does not include matched from UEFA cup which led us to assume that the scope of our recommendations is limited within Italy Serie A.

Apart from these assumptions, the assumptions for each analysis have been listed in the respective sections.

Analysis process and Details:

Overall Analysis Process:

1. Data import
2. Identifying AS Roma specifications in data
3. Data cleaning
4. Analysis 1: Identifying Competitors of Roma
5. Team Attributes:
    a. Analysis 2: Identifying best team attributes to beat the competitors:
6. Player Combination and Positioning:
    a. Analysis 3: Identifying best player combinations based on roles
    b. Analysis 4: Using player combinations for each role and finding their most effective positions
7. Benchmarking Player attributes:
8. Analysis 5: Identifying required skill level for the identified player formation
9. Summary of conclusive recommendations

Data Import:

The dataset was provided in the form of a SQLite database export. After setting a connection with sqlite database, 7 tables were extracted in R environment.

```
# loading libraries
library(dplyr)
library(magrittr)
library(RSQLite)
library(tidyr)
library(arules)
library(reshape2)
library(tidyr)
library(ggplot2)
library(XML)
library(matplot)

setwd('C:/Users/upadh/Desktop/Umin/2_Fall/MSBA 6410 - Exploratory Data Analysis/EDA HW/HW 2')

con                   <- src_sqlite("euro_soccer.sqlite")
country               <- data.frame(collect(tbl(con, "country")))
league                <- data.frame(collect(tbl(con, "league")))
match                 <- data.frame(collect(tbl(con, "match")))
player                <- data.frame(collect(tbl(con, "player")))
player_attributes     <- data.frame(collect(tbl(con, "player_attributes")))
team                  <- data.frame(collect(tbl(con, "team")))
team_atts             <- data.frame(collect(tbl(con, "team_attributes")))
```

Identifying specifications for AS Roma from data:

We began with understanding identifiers for corresponding to AS Roma in our datasets. This would help us filter and process datasets to view matched played by AS Roma.

```
# Collecting team specification for team Roma
long_team_name <- 'Roma'
romaId         <- team$id[team$team_long_name == long_team_name]
romaTeamApiId  <- team$team_api_id[team$team_long_name == long_team_name]
romaFifaApiId  <- team$team_fifa_api_id[team$team_long_name == long_team_name]
romaShortName  <- team$team_short_name[team$team_long_name == long_team_name]
rm(team)
```

```
# AS Roma is an Italian football club, plays under the Italian League
# Top 20 Italian League teams play in Italian Serie A League
# Winning team earns 3 points # Losing team earns 0 # In case of a draw 1 point each
# At the end of the league, when all matches are over, the team with highest
# points wins the league
```

Dataset seems to have country league details only. This led us to question if matches including AS Roma fall under Italy Serie A only. This was important to concentrate our focus on Roma's performances and associated players.

From the below test, we verified that AS Roma matches in the matches table were only played within Italy Serie A league and corresponding clubs under it.

```
# Checking if AS Roma plays for any other league apart from Italy Serie A
Romaleague <- unique(match$league_id[match$home_team_api_id == romaTeamApiId | match$away_team_api_id ==
romaTeamApiId])
league$name[Romaleague == league$country_id]
```

[1] "Italy Serie A"

Further, we looked at the total number of matches per season and number of matches with Roma as home or away team. This was to confirm the quality of dataset as well since online literature guided us that there are 20 teams in Serie A and a league season involves every team to play two matches against each team. Hence, every team played 38 matches in a league. Dataset shows 38 matches played by AS Roma in each season except 2011/2012 season. According to the 2011/2012 season details online, it was observed that there were a total of 38 matches played by Roma, which led us to conclude that there is **one match entry missing in 'match' table for 2011/2012 Serie A season**.

```
# Checking if how many games has AS Roma played the Italian league
# Subsetting matches for Italy Serie A league
matchesItalySerieA <- subset(match, match$league_id == league$id[league$name == "Italy Serie A"])
table(matchesItalySerieA$season)
```

```
2008/2009 2009/2010 2010/2011 2011/2012 2012/2013 2013/2014 2014/2015 2015/2016
      380       380       380       358       380       380       379       380
```

Multiple derived fields were created after sub-setting for the Roma matches. The original data table has goals scored by the home team and the away team, therefore we created 'winner' columns by looking at the difference in goals scored by the two teams in each match.

```
# Derived fields in RomaMatches table
RomaMatches           <- subset(match,match$home_team_api_id==romaTeamApiId|match$away_team_api_id ==
romaTeamApiId)
RomaMatches$Roma      <- ifelse(RomaMatches$home_team_api_id == romaTeamApiId, "H","A")
RomaMatches$Opp       <- ifelse(RomaMatches$Roma == "H",RomaMatches$away_team_api_id,
RomaMatches$home_team_api_id)
RomaMatches$goal_diff <- RomaMatches$home_team_goal - RomaMatches$away_team_goal
RomaMatches$result    <- ifelse(RomaMatches$goal_diff > 0, "H",ifelse(RomaMatches$goal_diff < 0,"A","D"))
RomaMatches$winner    <- ifelse(RomaMatches$result == RomaMatches$Roma,"Roma",
                           ifelse(RomaMatches$result == "D","Draw","Opp"))
# Number of Roma matches over all seasons
table(RomaMatches$season)
```

```
2008/2009 2009/2010 2010/2011 2011/2012 2012/2013 2013/2014 2014/2015 2015/2016
       38        38        38        37        38        38        38        38
```

```
# Among 3017 matches played in Italy Serie A from 2008/2009 to 2015/2016 (8 leagues)
# AS Roma played 303 matches, 38 in each league except 2011/2012
```

```
# Getting a table with player id and Player name
playeridNames <- player[,c("player_api_id","player_name")]
```

Dataset Cleaning:

Among the 7 tables of data we have, only the **'match'** table showed NA values. As we have stated towards the beginning of the document that we are performing most of our analysis on the matches that involve AS Roma to find areas of improvement, we are performing quality checks on this subset below.

Table na_counts were created to observe which columns have NA values and how many. The table has been shown below for the RomaMatches table which has 303 matches played by AS Roma throughout 2008/2009 to 2015/2016 seasons.

```
## Data cleaning

# NA counts for each column in RomaMatches table
na_counts <- data.frame(sapply(RomaMatches, function(y) sum(length(which(is.na(y))))))
na_counts <- cbind(row.names(na_counts),na_counts)
row.names(na_counts) <- c(1:nrow(na_counts))
colnames(na_counts) <- c("column","na_counts")
na_counts <- na_counts[na_counts$na_counts>0,]
na_counts
```

column	na_counts	column	na_counts	column	na_counts	column	na_counts
home_player_2	1	shotoff	2	IWH	2	SJH	75
home_player_5	1	foulcommit	2	IWD	2	SJD	75
home_player_7	2	card	2	IWA	2	SJA	75
home_player_9	1	cross	2	LBH	1	VCH	2
home_player_10	1	corner	2	LBD	1	VCD	2
home_player_11	2	possession	2	LBA	1	VCA	2
away_player_3	1	B365H	1	PSH	151	GBH	115
away_player_4	1	B365D	1	PSD	151	GBD	115
away_player_10	3	B365A	1	PSA	151	GBA	115
away_player_11	3	BWH	2	WHH	1	BSH	116
goal	2	BWD	2	WHD	1	BSD	116
shoton	2	BWA	2	WHA	1	BSA	116

Based on the summary of NA counts and the types of analysis we wanted to perform later in the process, we decided to pay attention to missing player IDs for home and away players and missing betting odds for 4 specific channels (PS, SJ, GB, BS).

For the missing player IDs, we found that there are 16 out of 303 matches which have at most 1 player ID missing. The missingness did not seem extremely problematic for performing association rules later in the process, but we still dug a bit deeper to understand if these IDs were missing at random or conditionally at random.

```
# NAs in player id columns (home and away)
summary(rowSums(is.na(RomaMatches[,c(56:77)])))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.05281 0.00000 1.00000
```

Our initial hypothesis was that these missing player IDs could correspond to players who were given red cards due to severe fouls or inappropriate behavior. We looked at some random matches to see if the missing player ID was the one who received a red card during that match. We performed this test for about 10 matches, but we have shown only 3 below.

We also looked at online match summary for that exact match to compare if the player the missing player in the data corresponded to the one who was given a red card. However, this hypothesis was disproved since all the missing players were missing at random and they did not receive red cards.

```
rowsWithNaPlayers <- RomaMatches[rowSums(is.na(RomaMatches[,c(56:77)]))>0,]

# Hypothesis, are red card players missing?
# In order to verify this We looked at the xml cell for 'cards' columns and compared it with the match summary on
# ESPN.com. Seems like the NAs do not align with the red cards given for the match
xmlToDataFrame(rowsWithNaPlayers$card[2])
```

```
  comment stats event_incident_typefk elapsed card_type        subtype player1 sortorder team   n type     id
1       y     1                   73      37         y   serious_fouls   39626         1 9804 337 card 696248
2       r     1                   65      82         r emergency_brake   39626         2 9804 401 card 696879
3       y     1                   79      87         y    verbal_abuse   27693         1 8686 407 card 696926
4       y     1                  325      80         y         pushing   30453         4 8686 411 card 697018
```

```
xmlToDataFrame(rowsWithNaPlayers$card[3])
```

```
  comment stats event_incident_typefk elapsed card_type player1 sortorder team   n type      id
1       y     1                   70      57         y   73835         3 8533 280 card 1087559
```

```
xmlToDataFrame(rowsWithNaPlayers$card[4])
```

```
  comment stats event_incident_typefk elapsed card_type player1 sortorder team  n type      id
1       y     1                   70      45         y   41640         0 9858 16 card 1392934
2       y     1                   70      57         y   41670         0 9858 17 card 1393263
3       y     1                   70      67         y   30714         0 8686 20 card 1393497
4       y     1                   70      77         y   41643         0 9858 21 card 1393779
```

```
# There are only 16 matches out of 303, which have NA values for player ids
# Each of these 16 matches have only 1 player missing out of all 22 players from both teams

# Removing these matches from RomaMatches table would reduce the dataset by 5%.
# Conclusion: These 16 matches are not being dropped from the data in spite of the missingness of one player
# We are assuming that this missingness will not affect the Player formation analysis

# NAs in the betting odds columns
# Four betting channels out of 10 have a very high number of NA values as seen from the na_counts tables
# Dropping corresponding columns for these four betting channels => PS, SJ, GB, BS

RomaMatchesCleaned <- RomaMatches[,!colnames(RomaMatches) %in% c("PSH", "PSA","PSD","SJA","SJH","SJD", "GBH",
                                                                 "GBA","GBD","BSA","BSH","BSD")]

RomaMatchesCleaned <- RomaMatchesCleaned[rowSums(is.na(RomaMatchesCleaned[,c(86:103)]))==0,]

cor(RomaMatchesCleaned[,c("B365H", "BWH","IWH","LBH","WHH","VCH")])
```

```
          B365H       BWH       IWH       LBH       WHH       VCH
B365H 1.0000000 0.9916777 0.9747948 0.9833026 0.9833829 0.9879166
BWH   0.9916777 1.0000000 0.9807057 0.9880112 0.9910005 0.9917386
IWH   0.9747948 0.9807057 1.0000000 0.9836679 0.9824431 0.9757619
LBH   0.9833026 0.9880112 0.9836679 1.0000000 0.9885363 0.9886214
WHH   0.9833829 0.9910005 0.9824431 0.9885363 1.0000000 0.9911421
VCH   0.9879166 0.9917386 0.9757619 0.9886214 0.9911421 1.0000000
```

```
cor(RomaMatchesCleaned[,c("B365A", "BWA","IWA","LBA","WHA","VCA")])
```

```
          B365A       BWA       IWA       LBA       WHA       VCA
B365A 1.0000000 0.9847731 0.9801061 0.9829729 0.9834844 0.9782348
BWA   0.9847731 1.0000000 0.9752978 0.9794502 0.9817561 0.9751211
IWA   0.9801061 0.9752978 1.0000000 0.9793428 0.9769033 0.9641667
LBA   0.9829729 0.9794502 0.9793428 1.0000000 0.9867341 0.9795721
WHA   0.9834844 0.9817561 0.9769033 0.9867341 1.0000000 0.9834670
VCA   0.9782348 0.9751211 0.9641667 0.9795721 0.9834670 1.0000000
```

```
cor(RomaMatchesCleaned[,c("B365D", "BWD","IWD","LBD","WHD","VCD")])
```

```
          B365D       BWD       IWD       LBD       WHD       VCD
B365D 1.0000000 0.9664000 0.9580458 0.9560708 0.9592903 0.9693883
BWD   0.9664000 1.0000000 0.9371107 0.9563430 0.9492419 0.9712543
IWD   0.9580458 0.9371107 1.0000000 0.9356121 0.9256750 0.9375636
LBD   0.9560708 0.9563430 0.9356121 1.0000000 0.9393174 0.9722296
WHD   0.9592903 0.9492419 0.9256750 0.9393174 1.0000000 0.9506816
VCD   0.9693883 0.9712543 0.9375636 0.9722296 0.9506816 1.0000000
```

```
# Betting channels are observed to have highly correlated odds which suggests that these betting platforms
# have a similar perception of match outcomes
```

Analysis 1: Identifying Competitors of Roma

Technique used: Exploratory analysis

To understand improvement areas for Roma, our first step is to identify Roma's main competitors. After background research we found out the ranking of teams vary a lot across the past 8 years. Therefore, we choose the matches of Roma in last 3 seasons and try to find out its main competitors based on the outcome of the matches. To do that, we design a score indicator: when a team beats Roma, it gets 2 points, when it's a draw, it gets 1 point, when it loses, it gets 0 point. Then we calculate the score for each team in the past 3 seasons.

```
# ---------------------------------

# Analysis 1
# Find teams that competed with Roma in 2015/2016
table(RomaMatches$winner[RomaMatches$season == '2015/2016'])
```

```
Draw  Opp Roma
  11    4   23
```

```
win_list <- RomaMatches$Opp[RomaMatches$season == '2015/2016' & RomaMatches$winner == 'Opp']
draw_list <- RomaMatches$Opp[RomaMatches$season == '2015/2016' & RomaMatches$winner == 'Draw']

# Find teams that competed with Roma in 2014/2015
table(RomaMatches$winner[RomaMatches$season == '2014/2015'])
```

```
Draw  Opp Roma
  13    6   19
```

```
win_list_2 <- RomaMatches$Opp[RomaMatches$season == '2014/2015' & RomaMatches$winner == 'Opp']
draw_list_2 <- RomaMatches$Opp[RomaMatches$season == '2014/2015' & RomaMatches$winner == 'Draw']

# Find teams that competed with Roma in 2013/2014
table(RomaMatches$winner[RomaMatches$season == '2013/2014'])
```

```
Draw  Opp Roma
   7    5   26
```

```
win_list_3 <- RomaMatches$Opp[RomaMatches$season == '2013/2014' & RomaMatches$winner == 'Opp']
draw_list_3 <- RomaMatches$Opp[RomaMatches$season == '2013/2014' & RomaMatches$winner == 'Draw']

# Calculate the scores
score_list <- table(c(rep(win_list_2,2),rep(win_list,2),rep(win_list_3,2),draw_list_3,draw_list,draw_list_2))
score_list
```

```
7943 8524 8529 8530 8533 8534 8535 8540 8543 8564 8636 9804 9857 9875 9876 9882 9885 10167 10233
   3    5    1    2    2    1    1    3    2    5    6    3    2    5    3    5    9     1     2
```

```
# The top 2 teams got 9 and 6 scores: Juventus, Inter
team$team_long_name[team$team_api_id==8636]
```

```
[1] "Inter"
```

```
team$team_long_name[team$team_api_id==9885]
```

```
[1] "Juventus"
```

```
# Hence Juventus and Inter can be analyzed further to understand their winning strategy
```

```
# This can help us devise the strategy to beat them and increase the win probability.
```

Assumption: In the last three seasons, Roma played 6 matches with each team in the league, these 6 matches are sufficient to reflect the strengths of the teams.

Conclusion: We found that **Inter Milan** and **Juventus** are our main competitors based on the match results. And this is consistent with our background research on the Internet.

Team Attributes:

Analysis 2: Identifying best team attributes to beat the competitors:

Technique used: Exploratory analysis, Association Rules

In this part, we apply association rules to analyze the relationship between team attributes and win probability. There are totally 3 categories of attributes, which can be divided into 12 sub-attributes in total. In our preliminary analysis, we found that win probability of home team and away team are quite different (1.6 : 1). Thus, we divided the analysis into two parts - the analysis on home team and away team. Next, for each match in the whole dataset, take analyzing home team as an example, the transaction we define includes items: home team, its respective 12 attributes separately and the match results. Then we filter the rhs to be “Home team won” and find the best attributes combination with a highest lift and basic support. As a result, we have best attributes combination for both home team and away team. In the end, we combine the combinations together since there is no overlap between them, giving us a comprehensive attributes combination no matter we are a home team or away team.

```
# ---------------------------------
# Analysis 2


# processing team attributes
teamAttsRequired <-
att_competitors[,c("team_api_id","date","buildUpPlaySpeedClass","buildUpPlayDribblingClass",
                                       "buildUpPlayPassingClass", "buildUpPlayPositioningClass",
                                       "chanceCreationPassingClass","chanceCreationCrossingClass",
                                       "chanceCreationShootingClass","chanceCreationPositioningClass",
                                       "defencePressureClass","defenceAggressionClass",
                                       "defenceTeamWidthClass","defenceDefenderLineClass")]
teamAttsRequired$year <- sapply(teamAttsRequired$date, FUN = function(x){substr(x, 1, 4)})
teamAttsRequired$season <- sapply(teamAttsRequired$year, FUN = function(x){paste(x,'/',as.numeric(x)+1,sep =
'')})
teamAttsRequired <- teamAttsRequired[,c('team_api_id',
'season',"buildUpPlaySpeedClass","buildUpPlayDribblingClass",
                                        "buildUpPlayPassingClass", "buildUpPlayPositioningClass",
                                        "chanceCreationPassingClass","chanceCreationCrossingClass",
                                        "chanceCreationShootingClass","chanceCreationPositioningClass",
                                        "defencePressureClass","defenceAggressionClass",
                                        "defenceTeamWidthClass","defenceDefenderLineClass")]


# find out winner for each match
match$goal_diff <- match$home_team_goal - match$away_team_goal
match$winner <- ifelse(match$goal_diff > 0, 'H', ifelse(match$goal_diff < 0, 'A', 'D'))
match_want <- match[,c('home_team_api_id','away_team_api_id','season','league_id', 'winner')]
match_want <- merge(match_want, teamNames, by.x = 'home_team_api_id', by.y = 'team_api_id')
colnames(match_want)[6] <- 'home_team_name'
match_want <- merge(match_want, teamNames, by.x = 'away_team_api_id', by.y = 'team_api_id')
colnames(match_want)[7] <- 'away_team_name'
# columns of match_want : "away_team_api_id" "home_team_api_id" "season"  "league_id"  "winner"
# "home_team_name"   "away_team_name"
```

```
# merge the team attributes into the matches
match_with_ta <- merge(match_want, teamAttsRequired, by.x = c('home_team_api_id', 'season'), by.y =
c('team_api_id', 'season'))
colnames(match_with_ta)[c(8:19)] <- sapply(colnames(match_with_ta)[c(8:19)], FUN = function(x){paste('home_',
x, sep = '')})
match_with_ta <- merge(match_with_ta, teamAttsRequired, by.x = c('away_team_api_id', 'season'), by.y =
c('team_api_id', 'season'))
colnames(match_with_ta)[c(20:31)] <- sapply(colnames(match_with_ta)[c(20:31)], FUN =
function(x){paste('away_', x, sep = '')})
match_with_ta_all <- match_with_ta[,c(5,8:31)]
match_with_ta_IA <- subset(match_with_ta, match_with_ta$league_id == league$id[league$name == "Italy Serie
A"])
match_with_ta_IA <- match_with_ta_IA[,c(5,8:31)]
```

```
# calculate the win probability for home team and away team, they are quite different, so we analyze home
#team and away team separately.
table(match_with_ta_all[,'winner'])
# probability of winning for home team
p_home <- 7922 / (7922 + 5041 + 4408) # p_home = 0.46
p_away <- 5041 / (7922 + 5041 + 4408) # p_away = 0.29


# find the team attributes combination which won the most matches in Europe all over 8 years
match_with_ta_all_home <- match_with_ta_all[,c(1:13)]
match_with_ta_all_away <- match_with_ta_all[,c(1,14:25)]
for (i in colnames(match_with_ta_all_home)) {
  match_with_ta_all_home[,i] <- as.factor(match_with_ta_all_home[,i])
}
for (i in colnames(match_with_ta_all_away)) {
  match_with_ta_all_away[,i] <- as.factor(match_with_ta_all_away[,i])
}
match_with_ta_all_home_T <- as(match_with_ta_all_home, 'transactions')
match_with_ta_all_away_T <- as(match_with_ta_all_away, 'transactions')
rules_home <- apriori(match_with_ta_all_home_T, parameter=list(supp = 0.01, conf=0.5, maxlen = 12))
summary(rules_home)
rules_home = rules_home %>% subset(subset = rhs %pin% "winner=H")  %>%
sort(by=c('lift','confidence'),decreasing=TRUE)
inspect(rules_home[1])
```

```
> inspect(rules_home[1])
    lhs                                                rhs        support confidence lift count
[1] {home_buildUpPlaySpeedClass=Balanced,
     home_buildUpPlayPassingClass=Short,
     home_buildUpPlayPositioningClass=Free Form} => {winner=H}    0.01       0.73  1.6    175
```

```
rules_away <- apriori(match_with_ta_all_away_T, parameter=list(supp = 0.01, conf=0.5, maxlen = 12))
summary(rules_away)
rules_away = rules_away %>% subset(subset = rhs %pin% "winner=A")  %>% sort(by=c("lift",
"confidence"),decreasing=TRUE)
inspect(rules_away[1:5])
```

```
> inspect(rules_away[1])
    lhs                                                rhs        support confidence lift count
[1] {away_buildUpPlayPassingClass=Short,
     away_chanceCreationPositioningClass=Free Form,
     away_defenceDefenderLineClass=Cover}           => {winner=A}   0.011       0.51  1.7    198
```

Conclusion:

The result shows:

1. For home team, team with attributes combination of (build-up-play-speed = Balanced, build-up-play-passing = Short, build-up-play-positioning = free form) has the highest confidence (0.73), meaning this team has a chance of 73% to win when it's a home team, 1.6 times larger than that of teams in general. (support = 1%, lift = 1.6)
2. For away team, team with attributes combination of (build-up-play-passing = short, chance-creation-positioning = free form, defense-defender-line = cover) has the highest confidence (0.51), meaning this team has 51% chance to win when it's an away team, 1.7 times larger than that of teams in general. (support = 1.1%, lift = 1.7)
3. In conclusion, matches data from the whole Europe shows that training the team with attributes combination of balanced build-up-play speed, short build-up-play passing, free-form build-up-play positioning, free-form chance-creation-positioning and cover defense defender-line would mostly help the team to win.

Assumption:

1. The preference for team attributes of strong teams doesn't deviate the result a lot. Since the dataset is large (17371 valid observations), we believe it can eliminate the bias.
2. The result based on the matches of all the leagues is applicable to Italy Serie A, there is homogeneity among leagues.

However, we dive further to find out the result if we analyze only on the data of Italy Serie A. The analysis is following and we find that the result is biased by strong teams. So we choose to analyze the whole dataset.

```
# Try applying association rules on data of Italy Series A and compare the result
match_with_ta_IA_home <- match_with_ta_IA[,c(1:13)]
match_with_ta_IA_away <- match_with_ta_IA[,c(1,14:25)]
for (i in colnames(match_with_ta_IA_home)) {
  match_with_ta_IA_home[,i] <- as.factor(match_with_ta_IA_home[,i])
}
for (i in colnames(match_with_ta_IA_away)) {
  match_with_ta_IA_away[,i] <- as.factor(match_with_ta_IA_away[,i])
}
match_with_ta_IA_home_T <- as(match_with_ta_IA_home, 'transactions')
match_with_ta_IA_away_T <- as(match_with_ta_IA_away, 'transactions')
rules_home <- apriori(match_with_ta_IA_home_T, parameter=list(supp = 0.01, conf=0.5, maxlen = 12))
summary(rules_home)
rules_home = rules_home %>% subset(subset = rhs %pin% "winner=H")  %>%
sort(by=c('confidence'),decreasing=TRUE)
inspect(rules_home[1:5])

rules_away <- apriori(match_with_ta_all_away_T, parameter=list(supp = 0.01, conf=0.5, maxlen = 12))
summary(rules_away)
rules_away = rules_away %>% subset(subset = rhs %pin% "winner=A")  %>%
sort(by=c("confidence"),decreasing=TRUE)
inspect(rules_away[1:5])
# Result shows that a specific combination of team attributes can reach a confidence of 0.92, which is
# unreasonably high, and the count is just 35, meaning it probably just indicates the team attributes
# of the typically strong teams, which means the result is not generally applicable.
```

Player Combination and Positioning:

Analysis 3: Identifying best player combinations based on roles :

Technique used: Descriptive statistics and Association Rules

Most players will play in a limited range of positions throughout their career, as each position requires a particular set of skills and physical attributes. Based on our research about the player positions and their roles in soccer, we are assuming that players are positioned based on their performance and capability for a role. This analysis focuses on finding the most favorable player formation for each of the four roles: Goalkeeper, Defenders, Midfielders and Forwards. Association rules technique has been used here such that we can identify the player combinations which apply to a large number of

matches (support), should be the most occurring player combination when AS Roma has won (confidence) and these wins with such player combinations were is not just coincidences (lift).

https://en.wikipedia.org/wiki/Association_football_positions

Our main assumptions for this analysis are as follows:

1. Player combinations that have high support, lift and confidence associated with AS Roma-wins are the favorable player combinations and would most likely improve the team's performance in the future matches. We are looking into skill levels of these players in the next set of analysis
2. We have chosen to look at all the 303 matches played by AS Roma over the course of 8 seasons to find the best player combinations. Post this analysis, we looked at the player contract details from their respective Wikipedia pages to check if they are still playing for AS Roma after the 2015/2016 season (which is the last season in our dataset)
3. Player combinations such identified constitute the recommended line up for the next season i.e. 2016/2017 Italy Serie A league

This analysis is exploratory in nature and mainly intends to test our hypothesis that certain players are best suited for a specific role in the team and their performance in these roles increases the likelihood for the team's win. According to our understanding, this analysis attempts to confirm this hypothesis and to find patterns in the players lineups for each match that have strong association with team's win.

```
## Finding the best player combination
## Winning combination would only matter based on player positions
## Example Player X,Y,Z are the best Attackers, A,B,C are the best Defenders etc.
colnames(RomaMatches)

## Subsetting match outcome, Roma(Home/Away), Y positions of Home and Away Players, Player IDs of Home and
Away players
RomaMatchesPlayerPositions <- RomaMatches[,c(120,116,34:55,56:77)]

# Creating Home and Away tables for Roma Players and match out comes
RomaMatchesPlayerPositionsH <- subset(RomaMatchesPlayerPositions, RomaMatchesPlayerPositions$Roma == "H")
RomaMatchesPlayerPositionsH <-
select(RomaMatchesPlayerPositionsH,winner,Roma,home_player_1:home_player_11,home_player_Y1:home_player_Y11)
colnames(RomaMatchesPlayerPositionsH) <-
c("outcome","RomaHorA","player1","player2","player3","player4","player5","player6","player7",

"player8","player9","player10","player11","player1_Y","player2_Y","player3_Y","player4_Y",

"player5_Y","player6_Y","player7_Y","player8_Y","player9_Y","player10_Y","player11_Y")

RomaMatchesPlayerPositionsA <- subset(RomaMatchesPlayerPositions, RomaMatchesPlayerPositions$Roma == "A")
RomaMatchesPlayerPositionsA <-
select(RomaMatchesPlayerPositionsA,winner,Roma,away_player_1:away_player_11,away_player_Y1:away_player_Y11)
colnames(RomaMatchesPlayerPositionsA) <- c("outcome","RomaHorA","player1","player2","player3","player4",
"player5","player6","player7","player8","player9","player10","player11",
"player1_Y","player2_Y","player3_Y","player4_Y",
"player5_Y","player6_Y","player7_Y","player8_Y","player9_Y","player10_Y","player11_Y")

# Also a combined table with all Roma matches (Both Home and Away)
RomaMatchesPlayerPositions <- rbind(RomaMatchesPlayerPositionsH,RomaMatchesPlayerPositionsA)
```

Based on research from online sources (source link mentioned below), we have mapper player Y positions as below: (https://the18.com/soccer-learning/soccer-positions-explained-names-numbers-and-roles)

Player Y Positions	Player Role	Mapping used in the Code
1	Goalkeeper	G

2, 3, 4	Defenders	D
5, 6, 7, 8	Midfielders	MF
9, 10, 11	Forwards	F

```
# Creating a named data vector to map the positions
positions <- c("1"="G", "2"="D", "3"="D",
"4"="D","5"="MF","6"="MF","7"="MF","8"="MF","9"="F","10"="F","11"="F")

# Using the named vector, Replacing the Y positions by their mapped positions names
for (i in 14:24) {
  RomaMatchesPlayerPositions[,i] <- positions[RomaMatchesPlayerPositions[,i]]
}

# Replacing Player Ids by Player names
# Getting player names in the RomaMatchesPlayer table
playerNames <- unlist(split(as.character(playeridNames$player_name), playeridNames$player_api_id))

for (i in 3:13) {
  RomaMatchesPlayerPositions[,i] <- playerNames[as.character(RomaMatchesPlayerPositions[,i])]
}

# Creating columns for Goalkeeper, Defenders, MidFeilders, Forwards
gk <- c()
def <- c()
mf <- c()
f <- c()

for (i in 1:nrow(RomaMatchesPlayerPositions)) {
  gk <-
append(gk,paste(unlist(RomaMatchesPlayerPositions[i,c(3:13)][RomaMatchesPlayerPositions[i,c(14:24)]=="G"])
                          ,collapse = "+"))
  def <-
append(def,paste(unlist(RomaMatchesPlayerPositions[i,c(3:13)][RomaMatchesPlayerPositions[i,c(14:24)]=="D"])
                         ,collapse = "+"))
  mf <-
append(mf,paste(unlist(RomaMatchesPlayerPositions[i,c(3:13)][RomaMatchesPlayerPositions[i,c(14:24)]=="MF"])
                         ,collapse = "+"))
  f <-
append(f,paste(unlist(RomaMatchesPlayerPositions[i,c(3:13)][RomaMatchesPlayerPositions[i,c(14:24)]=="F"])
                         ,collapse = "+"))

}

RomaMatchesPlayerPositions$GK <- gk
RomaMatchesPlayerPositions$Def <- def
RomaMatchesPlayerPositions$MF <- mf
RomaMatchesPlayerPositions$F <- f
```

Goalkeeper:

Once the dataset is prepared for the analysis, we subset it for goalkeepers. Since the role of goalkeeper is very distinct from any other player in the team, we decided to perform descriptive analysis to compare team performance with different goalkeepers.

```
# Finding out the best Goalkeeper
Goalkeeper <- RomaMatchesPlayerPositions[,c("outcome","GK")]
GoalkeeperPerformance <- as.data.frame.matrix(table(Goalkeeper$GK,Goalkeeper$outcome))
GoalkeeperPerformance <- cbind(GK = row.names(GoalkeeperPerformance),GoalkeeperPerformance)
GoalkeeperPerformance$matched_played = (GoalkeeperPerformance$Draw +
```

```
                                                GoalkeeperPerformance$Opp + GoalkeeperPerformance$Roma)
GoalkeeperPerformance$win_pct <- GoalkeeperPerformance$Roma / (GoalkeeperPerformance$Draw +
                                                                 GoalkeeperPerformance$Opp +
GoalkeeperPerformance$Roma)


GoalkeeperPerformance[order(-GoalkeeperPerformance$matched_played,-GoalkeeperPerformance$win_pct),]
```

	GK	Draw	Opp	Roma	matched_played	win_pct
Morgan De Sanctis	21	9	45		75	0.6000000
Doni	11	16	22		49	0.4489796
Julio Sergio	12	9	27		48	0.5625000
Maarten Stekelenburg	9	18	21		48	0.4375000
Wojciech Szczesny	10	2	22		34	0.6470588
Bogdan Lobont	5	2	11		18	0.6111111
Mauro Goicoechea	2	5	6		13	0.4615385
Artur	2	2	7		11	0.6363636
Lukasz Skorupski	0	4	1		5	0.2000000
Gianluca Curci	1	1	0		2	0.0000000

```
table(Goalkeeper$GK)
```

```
                Artur         Bogdan Lobont                  Doni       Gianluca Curci         Julio Sergio     Lukasz Skorupski
                   11                    18                    49                    2                   48                    5
Maarten Stekelenburg      Mauro Goicoechea     Morgan De Sanctis     Wojciech Szczesny
                   48                    13                    75                   34
```

player api id	Roma GoalKeeper	# of Matches Played For Roma (2008-2016)	Roma win % (2008-2016)	Status till 2016 (wikipedia)
32746	Morgan De Sanctis	75	60%	Signed by Monaco in 2016
39351	Doni	49	45%	Signed by Liverpool in 2011
39401	Julio Sergio	48	56%	Left Roma in 2013
30841	Maarten Stekelenburg	48	44%	Signed by Fulham (EPL) in 2013
169718	Wojciech Szczesny	34	65%	Present
39725	Bogdan Lobont	18	61%	Present
206711	Mauro Goicoechea	13	46%	Signed by Romanian club Oțelul Galați in 2013
19344	Artur	11	64%	Signed by Portuguese club Braga in 2010
178732	Lukasz Skorupski	5	20%	Present
27691	Gianluca Curci	2	0%	Signed by German Club Mainz in 2015

Assumption: Higher win % for Roma would include goalkeeper's contribution and higher number of matches played by a goalkeeper can signify player's consistent presence in subsequent matches.

Based on the descriptive analysis and online research, we recommend Wojciech Szczesny with his experience with the team and win% for the matches.

Mid Fielders:

Midfielders (originally called half-backs) are players whose position of play is midway between the attacking forwards and the defenders. Their main duties are to maintain possession of the ball, taking the ball from defenders and feeding it to the strikers, as well as dispossessing opposing players. They are more often the players that initiate attacking play for a team.

Source: https://en.wikipedia.org/wiki/Association_football_positions#Midfielder

We are using Association rules to determine a combination of midfielders which show high support, confidence and lift with Roma's wins. Assuming that such a combination has proved to be favorable in the past, we can then further analyze and recommend the best combination.

```
# Finding best Midfielder Combination
MidFielder <- RomaMatchesPlayerPositions[,c("outcome","MF")]
MidFielder$outcome <- as.factor(MidFielder$outcome)
MidFielder$MF <- as.factor(MidFielder$MF)

MidFielderT <- as(MidFielder,"transactions")

rules <- apriori(MidFielderT,parameter=list(supp = 0.01, conf=0.005))
```

```
summary(rules)
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
      0.005    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 3

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[198 item(s), 303 transaction(s)] done [0.00s].
sorting and recoding items ... [10 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [22 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

```
rules <- sort(rules,by=c("lift","support","confidence"),decreasing=TRUE)
```

```
set of 22 rules

rule length distribution (lhs + rhs):sizes
 1  2
10 12

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.545   2.000   2.000

summary of quality measures:
    support         confidence          lift            count
 Min.   :0.01320   Min.   :0.01320   Min.   :0.9844   Min.   :  4.00
 1st Qu.:0.01980   1st Qu.:0.04620   1st Qu.:1.0000   1st Qu.:  6.00
 Median :0.02640   Median :0.06222   Median :1.0000   Median :  8.00
 Mean   :0.06871   Mean   :0.21184   Mean   :1.1050   Mean   : 20.82
 3rd Qu.:0.04290   3rd Qu.:0.30827   3rd Qu.:1.1983   3rd Qu.: 13.00
 Max.   :0.53465   Max.   :0.75000   Max.   :1.4028   Max.   :162.00

mining info:
        data ntransactions support confidence
 MidFielderT           303    0.01      0.005
```

```
wins <- rules %>% subset(subset = rhs %pin% "outcome=Roma")  %>%
  subset(subset = lift > 1) %>%
  inspect()
```

```
    lhs                                                  rhs              support    confidence lift     count
[1] {MF=Radja Nainggolan+Daniele De Rossi+Miralem Pjanic} => {outcome=Roma} 0.01980198 0.7500000  1.402778 6
[2] {MF=Miralem Pjanic+Daniele De Rossi+Kevin Strootman}  => {outcome=Roma} 0.02970297 0.6428571  1.202381 9
[3] {MF=Miralem Pjanic+Seydou Keita+Radja Nainggolan}     => {outcome=Roma} 0.02640264 0.5714286  1.068783 8
```

```
# Best Midfielder combination
# {MF=Radja Nainggolan+Daniele De Rossi+Miralem Pjanic} => {outcome=Roma} 0.01980198 0.7500000  1.402778 6
```

STATUS:	
Radja Nainggolan+Daniele De Rossi	Present for the lext league
Miralem Pjanic	He left after 2016 season and went to Juventus
Kevin Strootman	Still in the team
Seydou Keita	He left after 2016 and went to Milan

From the arules results, the first combination of Midfielders : **Radja Nainggolan + Daniele De Rossi + Miralem Pjanic** has roughly 2% support due to being a comparatively new in the team. With 75% confidence and 1.4 lift, this combination seems to have a strong association with Roma's wins (rhs). Hence we recommend this combination for the midfielder positions.

Defenders:

Defenders play behind the midfielders and their primary responsibility is to provide support to the team and to prevent the opposition from scoring a goal.

Source: https://en.wikipedia.org/wiki/Association_football_positions#Midfielder

We are using Association rules to determine a combination of defenders which show high support, confidence and lift with Roma's wins. Assuming that such a combination has proved to be favorable in the past, we can then further analyze and recommend the best combination.

```
# Finding best Defender combination
Defender <- RomaMatchesPlayerPositions[,c("outcome","Def")]
Defender$outcome <- as.factor(Defender$outcome)
Defender$Def <- as.factor(Defender$Def)

DefenderT <- as(Defender,"transactions")

rules <- apriori(DefenderT,parameter=list(supp = 0.01, conf=0.006))
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
      0.006    0.1    1 none FALSE            TRUE       5    0.01      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 3

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[153 item(s), 303 transaction(s)] done [0.00s].
sorting and recoding items ... [19 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 done [0.00s].
writing ... [41 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

```
summary(rules)
```

```
set of 41 rules

rule length distribution (lhs + rhs):sizes
 1  2
19 22

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.537   2.000   2.000

summary of quality measures:
    support          confidence          lift            count
 Min.   :0.01320   Min.   :0.01320   Min.   :0.9352   Min.   :  4.00
 1st Qu.:0.01650   1st Qu.:0.02469   1st Qu.:1.0000   1st Qu.:  5.00
 Median :0.01980   Median :0.03960   Median :1.0000   Median :  6.00
 Mean   :0.04443   Mean   :0.21243   Mean   :1.2121   Mean   : 13.46
 3rd Qu.:0.02640   3rd Qu.:0.40000   3rd Qu.:1.2469   3rd Qu.:  8.00
 Max.   :0.53465   Max.   :1.00000   Max.   :1.8704   Max.   :162.00

mining info:
      data ntransactions support confidence
 DefenderT           303    0.01      0.006
```

```
rules <- sort(rules,by=c("lift","support","confidence"),decreasing=TRUE)
inspect(rules)

wins <- rules %>% subset(subset = rhs %pin% "outcome=Roma")  %>%
  subset(subset = lift > 1) %>%
  inspect()
```

```
    lhs                                                                                rhs                 support confidence     lift count
[1] {Def=Maicon+Mehdi Benatia+Leandro Castan+Federico Balzaretti}                   => {outcome=Roma} 0.01980198  1.0000000 1.870370     6
[2] {Def=Marco Motta+Phillippe Mexes+Juan+John Arne Riise}                          => {outcome=Roma} 0.01320132  1.0000000 1.870370     4
[3] {Def=Ivan Piris+Marquinhos+Leandro Castan+Federico Balzaretti}                  => {outcome=Roma} 0.01980198  0.7500000 1.402778     6
[4] {Def=Marco Cassetti+Nicolas Burdisso+Juan+John Arne Riise}                      => {outcome=Roma} 0.03960396  0.6666667 1.246914    12
[5] {Def=Marco Cassetti+Juan+Nicolas Burdisso+John Arne Riise}                      => {outcome=Roma} 0.01980198  0.6666667 1.246914     6
[6] {Def=Alessandro Florenzi+Konstantinos Manolas+Antonio Ruediger+Lucas Digne} => {outcome=Roma} 0.02310231  0.5833333 1.091049     7
[7] {Def=Marco Cassetti+Phillippe Mexes+Juan+John Arne Riise}                       => {outcome=Roma} 0.01980198  0.5454545 1.020202     6
```

Checking player contract status from the above rules:

STATUS:	
Maicon	Left Roma after 2016 league
Mehdi Benatia	Bought by Bayern Munich in 2014
Leandro Castan	Present for next League
Federico Balzaretti	Retired in 2015
Marco Motta	Bought by Juventus in 2010
Phillippe Mexes	Bought by Milan in 2011
Juan	Bought by Internazionale in 2013
John Arne Riise	Bought by Fulham in 2011
Ivan Piris	Only played for Roma in 2012 on loan contract
Marquinhos	Bought by French club Paris Saint Germane
Marco Cassetti	Bought by Udinese in 2012
Nicolas Burdisso	Bought by Genoa in 2012
Alessandro Florenzi	Present but injured in 2016
Konstantinos Manolas	Present for next League
Antonio Ruediger	Present for next League
Lucas Digne	Only played for Roma in 2015 on loan contract

From the arules results, the sixth combination of defenders: **Alessandro Florenzi+Konstantinos Manolas+Antonio Ruediger+Lucas Digne** has roughly 2.3% support. With 59% confidence and 1.04 lift, this combination seems to have a strong association with Roma's wins (rhs). However, the top 5 rules do not qualify in spite of better lift and confidence since the players in these combinations do not play for AS Roma anymore after the 2015/2016 season and hence are not available for the next season of 2016/2017. But these have been the strongest defender combinations and hence we will be looking at their skill levels in the next section of analysis to find key patterns that build a good fence lineup.

Forwards:

Forwards (or strikers) are the players who are positioned nearest to the opposing team's goal. The primary responsibility of forwards is to score goals and to create scoring chances for other players. Forwards may also contribute defensively by harrying opposition defenders and goalkeepers whilst not in possession.

Source: https://en.wikipedia.org/wiki/Association_football_positions#Midfielder

We are using Association rules to determine a combination of forwards which show high support, confidence and lift with Roma's wins. Assuming that such a combination has proved to be favorable in the past, we can then further analyze and recommend the best combination.

```
# Finding the Best forwards
Forward <- RomaMatchesPlayerPositions[,c("outcome","F")]
Forward$outcome <- as.factor(Forward$outcome)
Forward$F <- as.factor(Forward$F)

ForwardT <- as(Forward,"transactions")

rules <- apriori(ForwardT,parameter=list(supp = 0.01, conf=0.03))
summary(rules)
```

```
set of 34 rules

rule length distribution (lhs + rhs):sizes
 1  2
 8 26

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   2.000   1.765   2.000   2.000

summary of quality measures:
    support         confidence          lift            count
 Min.   :0.01320   Min.   :0.03086   Min.   :0.6235   Min.   :  4.00
 1st Qu.:0.01320   1st Qu.:0.05580   1st Qu.:1.0000   1st Qu.:  4.00
 Median :0.01980   Median :0.23267   Median :1.0955   Median :  6.00
 Mean   :0.05154   Mean   :0.30912   Mean   :1.2248   Mean   : 15.62
 3rd Qu.:0.03300   3rd Qu.:0.50000   3rd Qu.:1.3836   3rd Qu.: 10.00
 Max.   :0.53465   Max.   :1.00000   Max.   :1.8704   Max.   :162.00

mining info:
     data ntransactions support confidence
 ForwardT           303    0.01       0.03
```

```
rules <- sort(rules,by=c("lift","support","confidence"),decreasing=TRUE)
inspect(rules)

wins <- rules %>% subset(subset = rhs %pin% "outcome=Roma")  %>%
  subset(subset = lift > 0.9) %>%
  inspect()
```

```
     lhs                                                  rhs            support    confidence lift      count
[1]  {F=Alessandro Florenzi+Mattia Destro+Gervinho}    => {outcome=Roma} 0.01650165 1.0000000  1.8703704   5
[2]  {F=Edin Dzeko}                                    => {outcome=Roma} 0.01320132 0.8000000  1.4962963   4
[3]  {F=Luca Toni}                                     => {outcome=Roma} 0.01320132 0.8000000  1.4962963   4
[4]  {F=Mohamed Salah+Diego Perotti+Stephan El Shaarawy} => {outcome=Roma} 0.01650165 0.7142857  1.3359788   5
[5]  {F=Mirko Vucinic+Francesco Totti}                 => {outcome=Roma} 0.01320132 0.6666667  1.2469136   4
[6]  {F=Francesco Totti+Marco Borriello}               => {outcome=Roma} 0.01980198 0.6000000  1.1222222   6
[7]  {F=Alessandro Florenzi+Francesco Totti+Gervinho}  => {outcome=Roma} 0.01320132 0.5714286  1.0687831   4
[8]  {}                                                => {outcome=Roma} 0.53465347 0.5346535  1.0000000 162
[9]  {F=Francesco Totti}                               => {outcome=Roma} 0.04950495 0.5000000  0.9351852  15
[10] {F=Francesco Totti+Mirko Vucinic}                 => {outcome=Roma} 0.01980198 0.5000000  0.9351852   6
[11] {F=Gervinho+Francesco Totti+Adem Ljajic}          => {outcome=Roma} 0.01320132 0.5000000  0.9351852   4
```

STATUS:	
Alessandro Florenzi	Present but injured in 2016
Mattia Destro	Moved to Bologna in 2015 and then Milan in 2016
Gervinho	Moved to Chinese Super League in 2016
Mohamed Salah	Present for next League
Diego Perotti	Present for next League
Stephan El Shaarawy	Present for next League

In the analysis around Forwards, it was critical to look at 3 player combinations unlike other analyses. We observed from online research that AS Roma has an aggressive attacking technique over the years. Based on our analysis in the previous sections, we have identified 4 defenders and 3 midfielders, which in turn suggests a 3 player combination for the forward/attacker lineup.

From the arules results, the sixth combination of defenders: **Mohamed Salah+Diego Perotti+Stephan El Shaarawy** has roughly 2% support. With 71% confidence and 1.3 lift, this combination seems to have a strong association with Roma's wins (rhs). However, the first rule which also has 3 players does not qualify despite better lift and confidence since the players in these combinations do not play for AS Roma anymore after the 2015/2016 season and hence are not available for the next season of 2016/2017. But these have been the strongest forward combinations and hence we will be looking at their skill levels in the next section of analysis to find key patterns that build a good fence lineup.

Analysis 4: Using player combinations for each role and finding their most effective positions:

Technique used: Association Rules

Once the most effective player combinations for each of the four roles (Goalkeeper, Defenders, Midfielders and Forwards) have been identified, the next area of focus is identifying optimum positions for each player within each zone. The following piece of analysis leverages association rules to identify player positions that contribute to a higher probability of a win.

This technique is most effective for this analysis due to the fact that there are several matches (303 in number) which have been played by Roma over the past seasons, with varying combinations of players and across several positions. Patterns in our data which have led to wins would be best obtained by association rules.

This analysis is an exploratory analysis, because the motivation to perform the analysis is to understand if, in fact, there are specific positions for each player which contribute to an increased chance of AS Roma wins. This is an exploration analysis.

One main assumption for this analysis is:

> The analysis identifies standalone optimum positions for each player (within a zone). It is assumed that players being stationed in their respective "effective" positions will be compatible with each other. A finding which supports this assumption is that the identified effective positions for each player are mutually exclusive.

```
# ------------------------------------------------------------
# Analysis 4
```

```r
Match <- match
Player <- player

##League = 10257
match_italy <- Match[Match$league_id == 10257,]

## Team AS Roma = 8686
match_roma <- match_italy[match_italy$home_team_api_id == 8686|match_italy$away_team_api_id == 8686,]

##Creating a win flag for home & away teams
match_roma$away_result <- ifelse(match_roma$away_team_goal > match_roma$home_team_goal,
                                 "win",ifelse(match_roma$away_team_goal <
match_roma$home_team_goal,"loss","draw"))
match_roma$home_result <- ifelse(match_roma$home_team_goal > match_roma$away_team_goal,
                                 "win",ifelse(match_roma$home_team_goal <
match_roma$away_team_goal,"loss","draw"))

##Analyzing Roma home matches first
match_roma_home <- match_roma[match_roma$home_team_api_id == 8686,]

##Creating necessary columns for analyzing players and positions associations
##Combining X and Y of all home players

for (i in c(12:22,34:44)){
  match_roma_home[,i] <- as.character(match_roma_home[,i])
}

match_roma_home_form <- mutate(match_roma_home, player_XY1 = paste(home_player_X1,home_player_Y1,sep=":"),
                               player_XY2 = paste(home_player_X2,home_player_Y2,sep=":"),
                               player_XY3 = paste(home_player_X3,home_player_Y3,sep=":"),
                               player_XY4 = paste(home_player_X4,home_player_Y4,sep=":"),
                               player_XY5 = paste(home_player_X5,home_player_Y5,sep=":"),
                               player_XY6 = paste(home_player_X6,home_player_Y6,sep=":"),
                               player_XY7 = paste(home_player_X7,home_player_Y7,sep=":"),
                               player_XY8 = paste(home_player_X8,home_player_Y8,sep=":"),
                               player_XY9 = paste(home_player_X9,home_player_Y9,sep=":"),
                               player_XY10 = paste(home_player_X10,home_player_Y10,sep=":"),
                               player_XY11 = paste(home_player_X11,home_player_Y11,sep=":"))

## Repeating the same for away matches
match_roma_away <- match_roma[match_roma$away_team_api_id == 8686,]

##Creating necessary columns for analyzing players and positions associations
##Combining X and Y of all away players

for (i in c(23:43,45:55)){
  match_roma_away[,i] <- as.character(match_roma_away[,i])
}

match_roma_away_form <- mutate(match_roma_away, player_XY1 = paste(away_player_X1,away_player_Y1,sep=":"),
                               player_XY2 = paste(away_player_X2,away_player_Y2,sep=":"),
                               player_XY3 = paste(away_player_X3,away_player_Y3,sep=":"),
                               player_XY4 = paste(away_player_X4,away_player_Y4,sep=":"),
                               player_XY5 = paste(away_player_X5,away_player_Y5,sep=":"),
                               player_XY6 = paste(away_player_X6,away_player_Y6,sep=":"),
                               player_XY7 = paste(away_player_X7,away_player_Y7,sep=":"),
                               player_XY8 = paste(away_player_X8,away_player_Y8,sep=":"),
                               player_XY9 = paste(away_player_X9,away_player_Y9,sep=":"),
                               player_XY10 = paste(away_player_X10,away_player_Y10,sep=":"),
                               player_XY11 = paste(away_player_X11,away_player_Y11,sep=":"))

##Filtering required columns in home
required_h <- c(7,8,117,56:66,118:128)
match_roma_home_form_filt <- match_roma_home_form[,required_h]
```

```r
##Filtering required columns in away
required_a <- c(7,9,116,67:77,118:128)
match_roma_away_form_filt <- match_roma_away_form[,required_a]

##R binding the away and home datasets
colnames(match_roma_away_form_filt) <- colnames(match_roma_home_form_filt)
match_roma_form_filt <- rbind(match_roma_home_form_filt,match_roma_away_form_filt)

##Mapping player ids to their names
require(plyr)
for (i in 4:14){
  match_roma_form_filt[,i] <- mapvalues(match_roma_form_filt[,i],
                                        from=Player$player_api_id,
                                        to=Player$player_name)
}

## Creating new columns for concatenating player & position
MRH_player_form <- mutate(match_roma_form_filt, PF1 = paste(home_player_1,player_XY1,sep="|"),
                          PF2 = paste(home_player_2,player_XY2,sep="|"),
                          PF3 = paste(home_player_3,player_XY3,sep="|"),
                          PF4 = paste(home_player_4,player_XY4,sep="|"),
                          PF5 = paste(home_player_5,player_XY5,sep="|"),
                          PF6 = paste(home_player_6,player_XY6,sep="|"),
                          PF7 = paste(home_player_7,player_XY7,sep="|"),
                          PF8 = paste(home_player_8,player_XY8,sep="|"),
                          PF9 = paste(home_player_9,player_XY9,sep="|"),
                          PF10 = paste(home_player_10,player_XY10,sep="|"),
                          PF11 = paste(home_player_11,player_XY11,sep="|"))

## Getting rid of unnecessary columns
reqd <- c(3,26:36)
MRH_player_form <- MRH_player_form[,reqd]

## Converting all columns to factor before conversion to matrix
for (i in 1:12){
  MRH_player_form[,i] <- as.factor(MRH_player_form[,i])
}

## Converting to matrix and then transactions for the apriori algorithm
m <- as.matrix(MRH_player_form)
l <- lapply(1:nrow(m), FUN = function(i) (m[i, ]))
ph <- as(l,"transactions")

inspect(RomaHomeT[1:5])

##Running association rules
itemFrequencyPlot(ph,topN=20,type="absolute")
rules1 <- apriori(ph,parameter=list(supp = 0.01, conf=0.8))

options(digits=2)
summary(rules1)
inspect(rules1[1:20])

## Sorting rules which lead to a win
rules31 <- sort(subset(rules1, subset = (rhs %pin% "win"),by="lift",decreasing=TRUE))
inspect(rules31[1:5)

## Sorting rules with optimum positions of forwards
rules33 <- sort(subset(rules31, subset = (lhs %pin% "Shaarawy"|lhs %pin% "Perotti"|lhs %pin% "Salah"),
                       by="lift",decreasing=TRUE))
inspect(rules33[1:5])

## Sorting rules with optimum positions of defenders
rules34 <- sort(subset(rules31, subset = (lhs %pin% "Alessandro"|lhs %pin% "Ruediger"|lhs %pin% "Manolas"|lhs
%pin% "Digne"),decreasing=TRUE))
inspect(rules34[1:5])
```

```
## Sorting rules with optimum positions of midfielders
rules35 <- sort(subset(rules31, subset = (lhs %pin% "Nainggolan"|lhs %pin% "Rossi"|lhs %pin% "Pjanic"),
                       by="lift",decreasing=TRUE))
inspect(rules35[1:5])
```

Results–

One of the outputs from the above analysis (positions for two forwards) -

```
    lhs                                                    rhs    support    confidence lift     count
[1] {Diego Perotti|5:8}                                 => {win} 0.01320132 1.0        1.870370 4
[2] {Diego Perotti|5:8,Wojciech Szczesny|1:1}           => {win} 0.01320132 1.0        1.870370 4
[3] {Mohamed Salah|3:10,Seydou Keita|5:7,Wojciech Szczesny|1:1} => {win} 0.01320132 0.8        1.496296 4
```

All the above rules had an average support of 0.02, an average confidence of 0.9 and an average lift of 1.4, signifying that the above positions increased the likelihood of wins by ~40%.

Overall, the following “effective” positions were obtained for each player, within a particular role/zone :

Zone	Player	Position (X,Y)
Forward	Mohamed Salah	(4,10)
	Diego Perotti	(5,8)
	Stephan El Shaarawy	(6,10)
Mid Fielder	Radja Nainggolan	(3,7)
	Daniele De Rossi	(5,7)
	Miralem Pjanic	(7,7)
Defender	Alessandro Florenzi	(2,3)
	Konstantinos Manolas	(4,3)
	Antonio Ruediger	(6,3)
	Lucas Digne	(8,3)

This would be a powerful addition to our “best” combination for each zone, identified in the previous analysis. A significant advantage of these insights is that it is quickly actionable, AS Roma can experiment with the above formation for the initial matches of the upcoming season to test its effectiveness.

Analysis 5: Identifying required skill level for the identified player formation:

Technique used: Descriptive statistics

After identifying the best combinations for each role (goalkeeper, midfielders, forwards, defenses), we would like to conduct further analysis of player attributes, which can help us identify the most important skills for the team and for different roles; as well as skills combination for each role. This micro analysis of the team will help us better understand the team dynamic as well as take advantage of each player's strength and create winning game strategy. In addition, we can identify skills for candidate players which will help trading and recruiting players more efficient. Since we have already identified the best player formation, descriptive statistics is most appropriate to conduct further analysis of individual player's skills.

Our main assumptions for this analysis are as follows:

1. All players attributes ratings are based on FIFA rating, which was generated based on players' performances from the past year of world football.
2. Player attribute ratings are consistent for each player across their career.
3. The identified 30 skills are the most important attributes to describe each player's performance.
4. Players who are being considered in the best formation are "top players" in the team.

Reference: FIFA Play [5]

```
# ------------------------------------------
# Analysis 5 - Player attributes

#copy to new tables
player_attr_clean <- player_attributes

# remove NA
player_attr_clean <-na.omit(player_attr_clean)
```

adding players' name to the table

```
playeridNames <- player[,c("player_api_id","player_name")]
playerNames <- unlist(split(as.character(playeridNames$player_name), playeridNames$player_api_id))
player_attr_bin$player_name <- playerNames[as.character(player_attr_bin$player_api_id)]
player_attr_clean$player_name <- playerNames[as.character(player_attr_clean$player_api_id)]
```

top players:

goalkeepers: 169718 Wojciech Szczesny, 39725 Bogdan Lobont, 178732 Lukasz Skorupski

```
tgk <- filter(player_attr_clean, player_api_id %in% c(169718,39725,178732))
```

mid-fielders: Radja Nainggolan Daniele De Rossi Miralem Pjanic Kevin Strootman Seydou Keita

```
tmf <- filter(player_attr_clean, player_name %in% c("Radja Nainggolan", "Daniele De Rossi", "Miralem Pjanic",
"Kevin Strootman", "Seydou Keita"))
```

defender: Maicon Mehdi Benatia Leandro Castan Federico Balzaretti Marco Motta Phillippe Mexes Juan John Arne Riise Ivan Piris Marquinhos Marco Cassetti Nicolas Burdisso Alessandro Florenzi Konstantinos Manolas Antonio Ruediger Lucas Digne

```
td <- filter(player_attr_clean, player_name %in% c("Maicon","Mehdi Benatia","Leandro Castan", "Federico
Balzaretti", "Marco Motta", "Phillippe Mexes","Juan","John Arne Riise","Ivan Piris","Marquinhos","Marco
Cassetti","Nicolas Burdisso","Alessandro Florenzi", "Konstantinos Manolas", "Antonio Ruediger","Lucas
Digne"))
```

forwards: Alessandro Florenzi Mattia Destro Gervinho Mohamed Salah Diego Perotti Stephan El Shaarawy

```
tf <- filter(player_attr_clean, player_name %in% c("Alessandro Florenzi",
                                                   "Mattia Destro",
                                                   "Gervinho",
                                                   "Mohamed Salah",
```

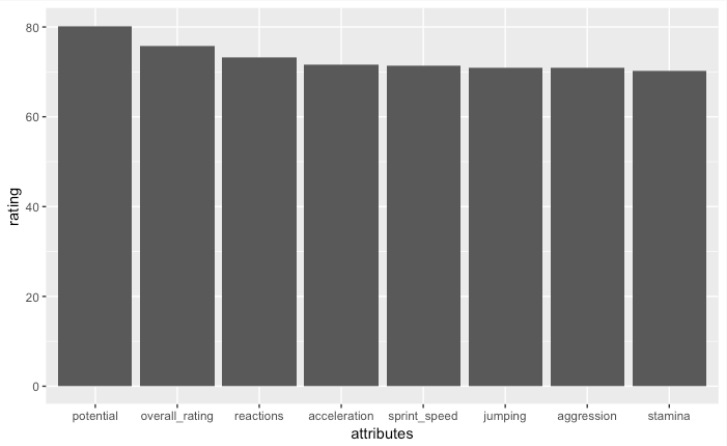
```
                                                "Diego Perotti",
                                                "Stephan El Shaarawy"))
```

all top players

```
top <- rbind(tgk,tmf,td,tf)
```

Important attributes for all top players

```
a<- top %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>% group_by(player_name) %>%
summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)

c <- t(b[,-1])

colnames(c) <- t(b[,1] )

rownames(c) <- colnames(b)[2:ncol(b)]

class(c) <- "numeric"

d <- data.frame(c)

e<- d[rev(order(d$mean)),]

f<-e[1]

f["attributes"]<-rownames(f)

g <- f[1:8,] %>% arrange(desc(mean))

ggplot(g,aes(x=reorder(attributes,-mean),mean))+geom_col()+labs(x = "attributes", y = "rating")
```



Looking across all top players, we identified the most important attributes, which are reactions, acceleration, sprint speed, jumping, aggression, and stamina.

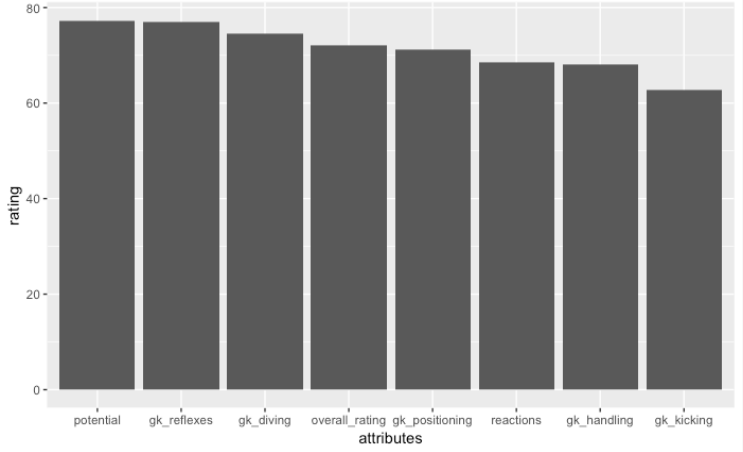
Recommendation:

When recruiting for players, the initial screening can focus on these attributes when not knowing which role the candidate is playing.

Important attributes for goalkeepers:

```
a<- tgk %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>% group_by(player_name) %>%
summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
```

```
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
f<-e[1]
f["attributes"]<-rownames(f)
g <- f[1:8,] %>% arrange(desc(mean))
ggplot(g,aes(x=reorder(attributes,-mean),mean))+geom_col()+labs(x = "attributes", y = "rating")
```



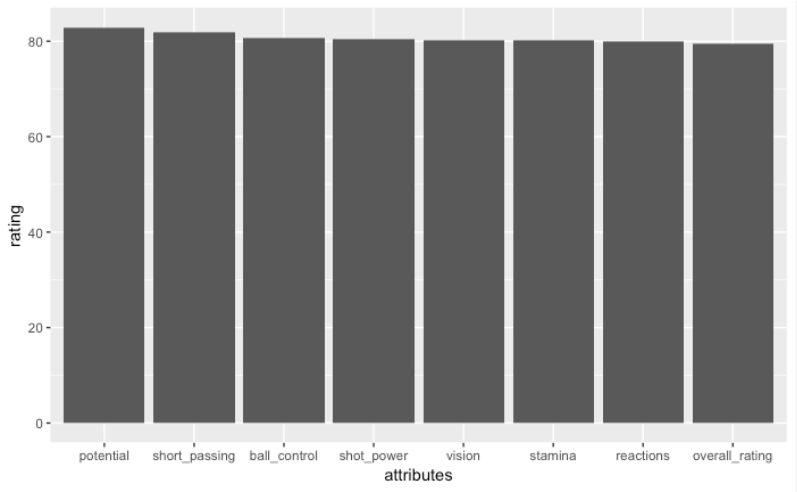
The identified player attributes for top goal keeper in the team are reflexes, diving, positioning, reactions, handling and kicking.

When comparing across the 3 present goalkeepers in the team, Wojciech Szczesny, Bogdan Lobont, Lukasz Skorupski, are all good at the required skills for goalkeepers, with Wojciech Szczesny better than the other two in all required skills. This can explain why when Wojciech Szczesny was present at the game, AC Roma winning odd is higher (65%), compared to when Bogdan Lobont was present (61%), and Lukasz Skorupski was present (20%).

Recommendation: Wojciech Szczesny is the best goalkeeper of the team, use Wojciech Szczesny as a benchmark when recruiting for goalkeeper.

Important attributes for midfielders:

```
a<- tmf %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>% group_by(player_name) %>%
summarise_at(vars(-player_name),funs(mean(.,)))



b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
f<-e[1]
f["attributes"]<-rownames(f)
g <- f[1:8,] %>% arrange(desc(mean))
ggplot(g,aes(x=reorder(attributes,-mean),mean))+geom_col()+labs(x = "attributes", y = "rating")
```



Best/recommended midfielder combination: Radja Nainggolan, Daniele De Rossi, Miralem Pjanic

```
mf_bestcobo <- filter(player_attr_clean, player_name %in% c("Radja Nainggolan", "Daniele De Rossi",  "Miralem Pjanic"))
a<- mf_bestcobo %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>% group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
```

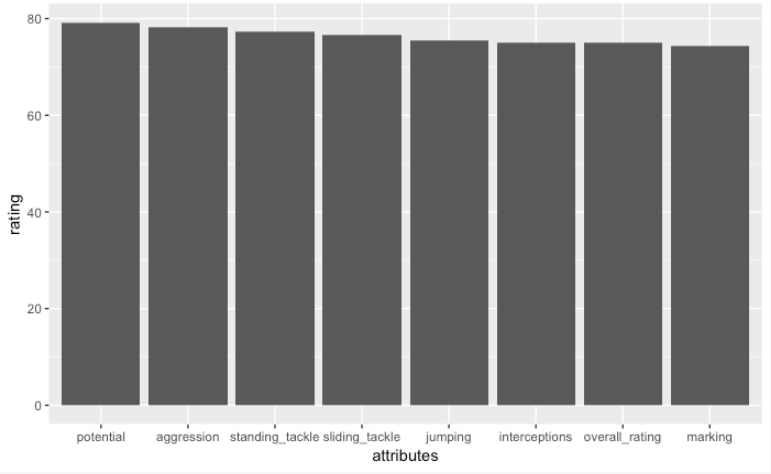
Recommendations for midfielder player attribute analysis:

1. Since players come and go frequently, we should recruit midfielders based on the most important attributes we identified in the graph above.
2. Looking more closely into individual player level, we can find appropriate substitute player without changing the team dynamic. Take Miralem Pjanic as an example, he is good at long shots dribbling, curve, and free kick accuracy, which none of the other midfielders are good at. Therefore, we should acquire midfielders with these three skills (long shots dribbling, curve, and free kick accuracy) in addition to the required skills as Miralem Pjanic's substitute.

Important attributes for defenders:

```
a<- td %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>% group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))


b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
f<-e[1]
f["attributes"]<-rownames(f)
g <- f[1:8,] %>% arrange(desc(mean))
```

```
ggplot(g,aes(x=reorder(attributes,-mean),mean))+geom_col()+labs(x = "attributes", y = "rating")
```



Best defender combination: Maicon, Mehdi Benatia, Leandro Castan, Federico Balzaretti

```
d_bestcobo <- filter(player_attr_clean, player_name %in% c("Maicon", "Mehdi Benatia",  "Leandro Castan",
"Federico Balzaretti"))
a<- d_bestcobo %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>%
group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
```

2nd best defender combination: Marco Motta, Phillippe Mexes, Juan, John Arne Riise

```
d_bestcobo_2 <- filter(player_attr_clean, player_name %in% c("Marco Motta", "Phillippe Mexes",  "Juan", "John
Arne Riise"))
a<- d_bestcobo_2 %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>%
group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
```

3rd best defender combination: Ivan Piris, Marquinhos, Leandro Castan, Federico Balzaretti

```
d_bestcobo_3 <- filter(player_attr_clean, player_name %in% c("Ivan Piris", "Marquinhos",  "Leandro Castan",
"Federico Balzaretti"))
a<- d_bestcobo_3 %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>%
group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
```

```
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
```

4th best defender combination: Marco Cassetti, Nicolas Burdisso, Juan, John Arne Riise

```
d_bestcobo_4 <- filter(player_attr_clean, player_name %in% c("Marco Motta", "Nicolas Burdisso",  "Juan",
"John Arne Riise"))
a<- d_bestcobo_4 %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>%
group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
```

Selecting from who are currently presenting in the team, we recommend this combination for defenders: Alessandro Florenzi, Konstantinos Manolas, Antonio Ruediger, Lucas Digne.

```
d_recommend <- filter(player_attr_clean, player_name %in% c("Alessandro Florenzi", "Konstantinos Manolas",
"Antonio Ruediger", "Lucas Digne"))
a<- d_recommend %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>%
group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
print(e[5])
```

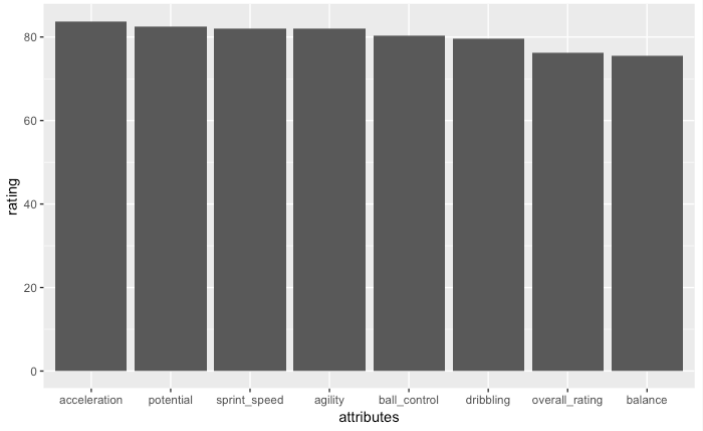
Recommendations for defender player attribute analysis:

Best combination for defenders: Maicon, Mehdi Benatia, Leandro Castan, and Federico Balzaretti. However, three of these four players no longer plays in the team. After analyzing the top 3 combinations for defenders, we found that the combination always have these top skills: standing tackle, sliding tackle, aggression, interceptions, jumping. Based on past performances and individual player's skills, we recommend a combination of defenders of Alessandro Florenzi, Konstantinos Manolas, Antonio Ruediger, Lucas Digne. Potential future acquisition should consider players who are good at stamina, sprint speed, acceleration, in addition to the required skills for defenders.

Important attributes for forwards:

```
a<- tf %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>% group_by(player_name) %>%
summarise_at(vars(-player_name),funs(mean(.,)))



b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
```

```
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
f<-e[1]
f["attributes"]<-rownames(f)
g <- f[1:8,] %>% arrange(desc(mean))
ggplot(g,aes(x=reorder(attributes,-mean),mean))+geom_col()+labs(x = "attributes", y = "rating")
```



Detail analysis for forward combination:

Best forward combination: Alessandro Florenzi, Mattia Destro, Gervinho However, Mattia Destro and Gervinho has already left the team.

```
f_bestcobo <- filter(player_attr_clean, player_name %in% c("Alessandro Florenzi", "Mattia Destro",
"Gervinho"))
a<- f_bestcobo %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>%
group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
```

Selecting from who are currently present in the team, we recommend forward combination: Mohamed Salah, Diego Perotti, Stephan El Shaarawy

```
f_recommend <- filter(player_attr_clean, player_name %in% c("Mohamed Salah", "Diego Perotti",  "Stephan El
Shaarawy"))
a<- f_recommend %>% select(player_name, overall_rating:potential, crossing:gk_reflexes) %>%
group_by(player_name) %>% summarise_at(vars(-player_name),funs(mean(.,)))

b <- rbind(c("mean",colMeans(a[,2:ncol(a)])),a)
c <- t(b[,-1])
colnames(c) <- t(b[,1] )
rownames(c) <- colnames(b)[2:ncol(b)]
class(c) <- "numeric"
d <- data.frame(c)
e<- d[rev(order(d$mean)),]
print(e)
```

Recommendations for defender player attribute analysis:

We recommend the combination of Mohamed Salah, Diego Perotti, Stephan El Shaarawy for forwards. Because this combination has a very high average rating for the required skills for forwards, also, these three players playing together improve the chance of AC Roma winning by 134%.

Summary of insights:

Areas of Analysis	Insights	Outcomes
AS Roma's competitors	List of competitors AS Roma has been consistently losing to	Juventus, Inter Milan
Team Attributes	Team attributes contributing to an increased likelihood of wins, based on if the match is an away/home match	**Home game strategy :** *Build-up-play speed:* Balanced *Build-up-play passing:* Short *Build-up-play positioning:* Free-form
		Away game strategy : *Defense defender-line:* Cover *Build-up-play passing:* Short *Chance-creation positioning:* Free-form
Player Formation	Most effective combination of players for each role, and their optimum formation on field	**Dream Team (Left/Center/Right) :** *Goalkeeper:* Wojciech Szczesny *Defenders:* Alessandro Florenzi (L), Konstantinos Manolas (LC), Antonio Ruediger (RC), Lucas Digne (R) *Mid-fielders:* Radja Nainggolan (L), Daniele De Rossi (C), Miralem Pjanic (R) *Forwards:* Mohamed Salah (L), Diego Perotti (C), Stephan El Shaarawy (R)
Player Attributes	Most effective player skills for each zone	*Goalkeeper:* Reflexes, Diving, Positioning, Reaction *Defenders:* Short passing, Ball control, Shot power, Vision *Mid-fielders:* Aggregation, Standing tackle, Sliding tackle, Interceptions *Forwards:* Acceleration, Agility, Sprint speed, Ball control

Recommendations:

Based on the insights obtained in the previous analyses, our recommendations for AS Roma can be divided into two broad umbrellas -

Short-term recommendations:

- *Dream team combination:* Based on the association of existing players at positions in tandem with wins, and player availability in the following season 2016-17, we recommend the following team composition and formation as the starting lineup for the upcoming season.
  - Goalkeeper: Wojciech Szczesny
  - Defenders: Alessandro Florenzi (L), Konstantinos Manolas (LC), Antonio Ruediger (RC), Lucas Digne (R)
  - Mid-fielders: Radja Nainggolan (L), Daniele De Rossi (C), Miralem Pjanic (R)
  - Forwards: Mohamed Salah (L), Diego Perotti (C), Stephan El Shaarawy (R)
- *Winning team formation:* Based on the above team combination and the positions these players excel at, we recommend a 4-3-3 starting formation for AS Roma

Long-term recommendations:

- *Game strategies to adopt, train, and strengthen*:
  - Build-up play speed: Balanced
  - Build-up play passing: Short
  - Build-up play positioning: Free-form
  - Chance-creation-positioning: Free-form
  - Defense defender-line: Cover
- *Player Attributes:* In alignment with the favorable player attributes that best suit the team at each position (*summarized in the table below*), identify and discover hidden talents that can be acquired and deployed as part of the team to ensure continued success

Importance	Goalkeeper	Midfielders	Defenders	Forwards
1	Goalkeeper Reflexes	Short Passing	Aggregation	Acceleration
2	Goalkeeper Diving	Ball Control	Standing Tackle	Agility
3	Goalkeeper Positioning	Shot Power	Sliding Tackle	Sprint Speed
4	Reaction	Vision	Interceptions	Ball Control

Web References:

Reference Number	Reference Name and Link
1	Oddschecker https://www.oddschecker.com/football/italy/serie-a/winner
2	Wikipedia https://en.wikipedia.org/wiki/List_of_Italian_football_champions
3	EuroSport https://www.eurosport.com/football/serie-a/palmares.shtml
4	ESPN.com http://www.espn.com/soccer/report?gameId=491415
5	FIFA play http://www.fifplay.com/encyclopedia/player-attributes/