Homework 2: Association Rules and Sports Analytics

Jesse Sprinkel
Haojin Jia
Chandrakanth Tolupunoori
Anshuman Vijayvargia
Priyanka Singhal

10/03/2018

Abstract

This is the white paper and technical details of our analysis and recommendations.

Contents

1 One Main Problem and Our Approach:

1.1 Situation:

Our objective is to analyze soccer data to find insights and non-obvious patterns that will help A.S. Roma achieve greater success on the field. We have been specifically asked to utilize association rules in our analysis.

1.2 Complication:

The data available is comprehensive, covering more than 8 leagues and thousands of teams and players. Converting this data into a format suitable for rule mining is difficult, as well as knowing where to look for rules that will provide value for the team.

1.3 Our Approach

1.3.1 How we define the success for team and player

We have chosen two ways to define success:

1. The final result (win, loss, draw), specifically **Win/loss Percentage**, to define the success of the team.
2. **Mean Goal Difference (MGD)** to define the success of the team.

1.3.2 What are the KPIs we define:

Based on our research, we assume that all these factors:

1. Team attributes
2. Team player attributes
3. Player Positions
4. player combinations

actually do have an affect on the final result of a match.

These factors are not exhaustive. However, limited by resources, we decided to concentrate on these factors.

2 Preparedness

2.1 Brief Summary of the Dataset

2.1.1 Data Overview

What does the data look like?:

- +25,000 matches
- +10,000 players
- 11 European Countries with their lead championship
- Seasons 2008 to 2016
- Players and Teams' attributes* sourced from EA Sports' FIFA video game series, including the weekly updates
- Team line up with squad formation (X, Y coordinates)
- Betting odds from up to 10 providers
- Detailed match events (goal types, possession, corner, cross, fouls, cards etc...) for +10,000 matches

2.1.2 Original Data Source:

1. Scores, lineup, team formation and events
2. Betting odds
3. players and teams attributes from EA Sports FIFA games

2.2 Data Cleaning

2.3 R Library

Please make sure successfully install the packages before reproducing the research

```
library(dplyr)
library(magrittr)
library(arules)
library(tidyr)
library(arulesViz)
library(ggplot2)
library(stringr)
library(lubridate)
```

Set working directory

```
knitr::opts_knit$set(root.dir = 'C:/2018Fall/6410/HW 2')
```

Load packages and data to local machine

```
con <- src_sqlite("euro_soccer.sqlite")

country_tbl <- tbl(con, "country")
league_tbl <- tbl(con, "league")
match_tbl <- tbl(con, "match")
player_tbl <- tbl(con, "player")
player_atts_tbl <- tbl(con, "player_attributes")
team_tbl <- tbl(con, "team")
team_atts_tbl <- tbl(con, "team_attributes")

#write the data to local memory
country <- collect(country_tbl)
league <- collect(league_tbl)
```

```
match <- collect(match_tbl)
player <- collect(player_tbl)
player_atts <- collect(player_atts_tbl)
team <- collect(team_tbl)
team_att <- collect(team_atts_tbl)
```

3 Historical Performance:

3.1 Why historical performance matters? The effective way to delve into why we lose, why we win and how we improve.

Before diving into the strategy to increase the odds of winning, how was the historic performance of our client? Only with this mind can we study why we won in the past matches, why we were defeated by certain teams and how we can work on improving our competitiveness.

3.2 Our approach: Conducting association analysis but with special restrictions.

To answer this question, we uses association rules and filters the rules with the following characters:

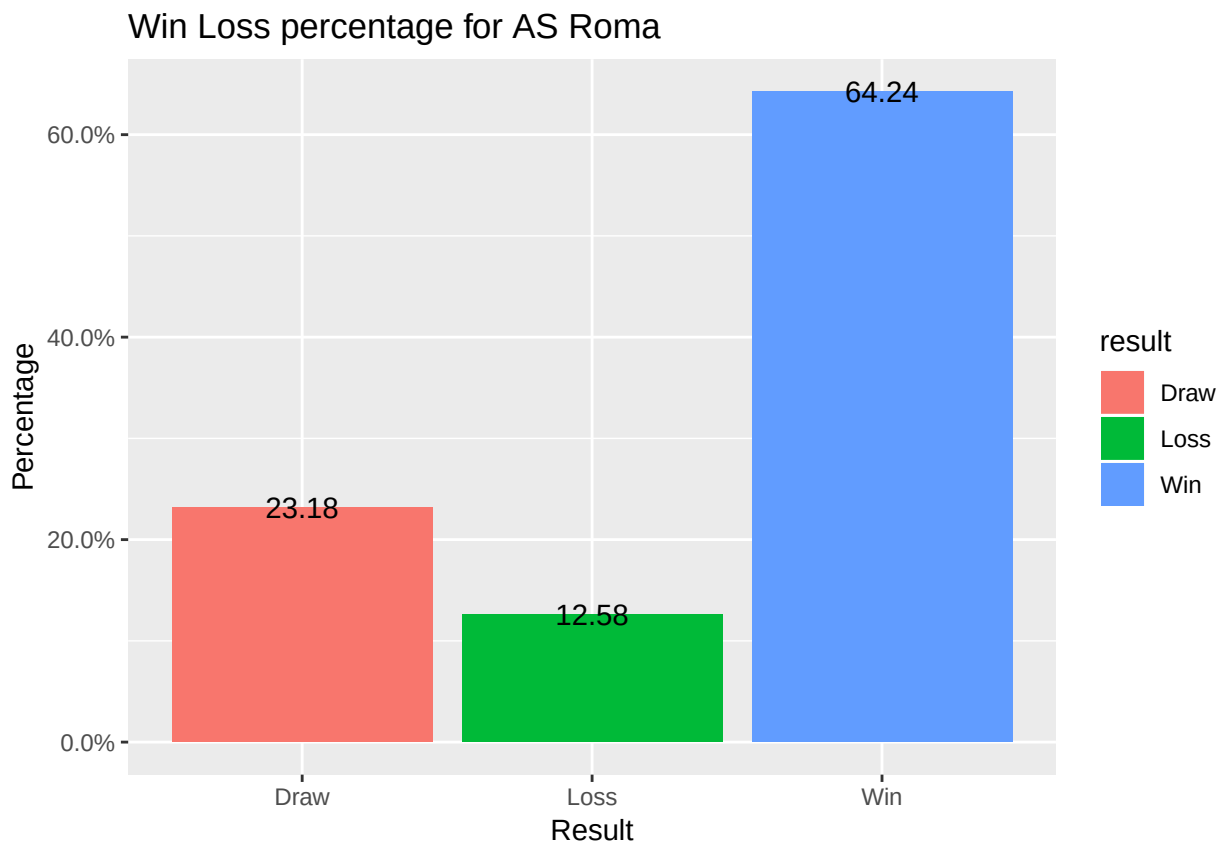
1. Match results are the sole item on the right hand.
2. Two team ID in the left-hand side, one of which is our client and the other is we competed with.

In the case of our rules, the confidence metric represents the probability that a win will occur. Lift and support metrics may have little meanings considering our interest of analysis in that we care about the odds of winning (with the same direction to the probability of winning) when our client team competes with rivals on average. It's useless to compare this specific format with, for example, more than 2 competing teams on the left-hand side.

3.3 Our analysis:

```
# Filter the mathches for Roma played at home
roma_record <- team %>%
                filter(team_long_name == 'Roma')

home_matches <- filter(match, home_team_api_id ==
                         roma_record$team_api_id)

# Filter out the required columns for converting
# match data to a transaction level
txn <- home_matches %>% select(id,home_team_goal, away_team_goal)

# Check for NA values and remove rows that
# contain NA values
txn <- na.omit(txn)

# convert the match result to win,loss or draw
txn$result <- txn$home_team_goal-txn$away_team_goal
txn$result <- ifelse(txn$result == 0, "Draw",
                     ifelse(txn$result > 0, "Win", "Loss"))

win_loss <- txn %>% select(result, id) %>% group_by(result) %>%
  summarise(totals = n())
```

```
ggplot(win_loss, aes(x= result, y = totals/sum(totals),
                     fill=result, legend = F)) +
  geom_bar(stat = 'identity') +
  scale_y_continuous(labels = scales::percent) +
  labs(title="Win Loss percentage for AS Roma", x = "Result",
       y = "Percentage") +
  geom_text(aes(label = round(totals*100/sum(totals),2)))
```




```
# data_preparedness
long_team_name <- 'Roma'
roma_record <- team_tbl %>%
                collect() %>%
                filter(grepl(long_team_name, team_long_name))

home_matches <- filter(match_tbl, home_team_api_id ==
                         roma_record$team_api_id)

match_outcomes_per_match <- match_tbl %>%
                    mutate(goal_diff =home_team_goal -
                             away_team_goal) %>%
                    select(id, goal_diff) %>%
                    rename(match_id = id)


roma_players_per_match <- select(home_matches,
                                 id,matches("home_player_[[:digit:]]")) %>%
```

```
                collect() %>%
                gather(player, player_api_id, -id) %>%
                rename(match_id = id)

team_tbl <- home_matches %>%
            mutate(goal_diff = home_team_goal - away_team_goal) %>%
            select(match_api_id,home_team_api_id,away_team_api_id,goal_diff) %>%
            mutate(result = ifelse(goal_diff >0, 'win',
                                   ifelse(goal_diff < 0,
                                          'loss', 'tie'))) %>%
            select(home_team_api_id,away_team_api_id,result)

write.csv(team_tbl,file = 'team.csv')
```

3.4 Which teams do we have advantages over?

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.01    0.1    1 none FALSE            TRUE       5   0.015      3
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[3 item(s)] done [0.00s].
## set transactions ...[186 item(s), 151 transaction(s)] done [0.00s].
## sorting and recoding items ... [26 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [22 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##     lhs             rhs   support    confidence lift     count
## [1] {8686,9888}  => {win} 0.01986755 1.000      1.556701 3
## [2] {7943,8686}  => {tie} 0.01986755 1.000      4.314286 3
## [3] {8686,9804}  => {win} 0.03311258 1.000      1.556701 5
## [4] {10233,8686} => {win} 0.05298013 1.000      1.556701 8
## [5] {8535,8686}  => {win} 0.04635762 0.875      1.362113 7
```

3.5 Visualize results

1. Rules Table:

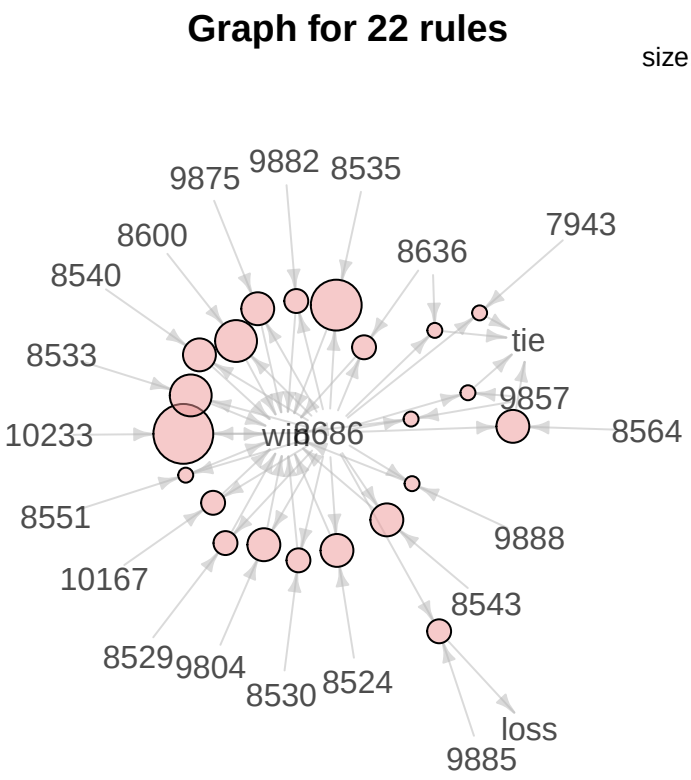
```
inspect(rules[1:5])
```

```
##     lhs             rhs   support    confidence lift     count
## [1] {8686,9888}  => {win} 0.01986755 1.000      1.556701 3
## [2] {7943,8686}  => {tie} 0.01986755 1.000      4.314286 3
## [3] {8686,9804}  => {win} 0.03311258 1.000      1.556701 5
```

```
## [4] {10233,8686} => {win} 0.05298013 1.000      1.556701 8
## [5] {8535,8686}  => {win} 0.04635762 0.875      1.362113 7
```

2. Interactive Visualization of Rules

```
plot(rules,method="graph", shading=NA)
```



3.6 Findings and insights

1. Our client won 64% of matches and tied in another 24%.
2. Our Team (Id: 8686) has overwhelming advantages over team Lecce (Id: 9888), Sassuolo (Id: 7943), Torino (Id:9804) and Genoa (Id:10233) in that we don't lose historically to them.
3. For other 15 teams, we have over 50% of chance to win on average. These are a good start to research why we won sometimes and we lost the other.
4. Admittedly, the number of records for some rules is not large enough to approximate probability based on proportion. However, our purpose, again, is to find a rational perspective to delve into. And we don't recommend to use this number to predict future odds of success.

3.7 Conclusion

Based on historical performance, we believe, our client has some advantages over other teams considering the changes in our teammates. We defeated them in the majority of past matches.

Next step, as we mentioned earlier, is to delve into what key performance indicators can significantly influence the result of a match.

4 Our Solutions

4.1 Solution 1: Optimizing player combinations

4.1.1 Why does the combination of players matter?

The question we would like to answer here is if we can find a combination of player that is more likely to make the team Win. The variable we can control while selecting a team is the players in the playing XI for each match. Hence we decided to start our analysis with a key factor that can be leveraged to maximize winning probability.

4.1.2 Our approach for home matches:

First, we looked at home matches for this analysis because there are significant differences in winning percentages due to hidden factors betweem home and away, and we want to keep the analysis consistent (**Assumption**). Based on the number of goals scored by the home team and away team in each match, we decided if the result was a Win, Loss or Draw. Then, applied association rule mining to find out if there are any occurrences for players in the match that result in a win. To do so we defined our transaction as each match played and the item set as the list of players in playing XI plus match result.

4.1.3 Analysis for home matches:

```
# Filter the mathches for Roma played at home
roma_record <- team %>%
               filter(team_long_name == 'Roma')

home_matches <- filter(match, home_team_api_id == roma_record$team_api_id)

# Filter out the required columns for converting match data to a transaction level
txn <- home_matches %>% select(home_team_goal, away_team_goal,
                               home_player_1:home_player_11)

# Check for NA values and remove rows that contain NA values
txn <- na.omit(txn)

# convert the match result to win,loss or draw
txn$result <- txn$home_team_goal-txn$away_team_goal
txn$result <- ifelse(txn$result == 0, "Draw",
                     ifelse(txn$result > 0, "Win", "Loss"))

# Removing redundant columns from data
txn <- txn[,-c(1,2)]

# write the dataframe to csv inorder to read as transactions
write.csv(txn, "txn.csv")

# read back as transactions
txn_final <- read.transactions("txn.csv",format=c("basket"),sep=",",
                               rm.duplicates=TRUE)
```

```
# Checking for player id combination which results in a WIN
rules <- apriori(txn_final,appearance = list(rhs = c('Win')),
                 parameter=list(supp = 0.05, conf=0.8,
                                minlen = 5, maxlen = 7))
```

```
## Apriori
```

```
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5    0.05      5
##  maxlen target   ext
##       7  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 7
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[259 item(s), 151 transaction(s)] done [0.00s].
## sorting and recoding items ... [65 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.00s].
## writing ... [118 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Restricting the output format for lift, confidence and support to 2 decimals
options(digits=2)

# Sorting the rules by parameter lift
rules <- sort(rules,by="lift",decreasing=TRUE)
```

```
# Filtering rules which are of interest
inspect(head(rules,5))
```

```
##     lhs                          rhs   support confidence lift count
## [1] {30714,32746,37950,46875} => {Win} 0.053   1          1.6  8
## [2] {30714,32746,46875,96398} => {Win} 0.053   1          1.6  8
## [3] {22337,30714,34305,37950} => {Win} 0.053   1          1.6  8
## [4] {30714,34305,37950,96398} => {Win} 0.060   1          1.6  9
## [5] {22337,30714,32746,34305} => {Win} 0.053   1          1.6  8
```

```
# Checking for player id combination which result in a LOSS
rules <- apriori(txn_final,appearance = list(rhs = c('Loss')),
                 parameter=list(supp = 0.015, conf=0.5, minlen = 4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5   0.015      4
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[1 item(s)] done [0.00s].
```

```
## set transactions ...[259 item(s), 151 transaction(s)] done [0.00s].
## sorting and recoding items ... [80 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.00s].
## writing ... [45 rule(s)] done [0.02s].
## creating S4 object  ... done [0.00s].
```

```
# Restricting the output format for lift, confidence
# and support to 2 decimals
options(digits=2)

# Sorting the rules by parameter lift
rules <- sort(rules,by="lift",decreasing=TRUE)
inspect(rules[1:5])
```

```
##     lhs                         rhs    support confidence lift count
## [1] {177874,195190,267365} => {Loss} 0.02    1          7.9  3
## [2] {177874,267365,40165}  => {Loss} 0.02    1          7.9  3
## [3] {177874,267365,30714}  => {Loss} 0.02    1          7.9  3
## [4] {177874,195190,40165}  => {Loss} 0.02    1          7.9  3
## [5] {177874,195190,30714}  => {Loss} 0.02    1          7.9  3
```

4.1.4 Findings and insights for home matches:

Varying the parameters, support, and confidence, while using the rule mining algorithm apriori we got an optimal number of rules in order to further investigate which combinations of players results in a win and loss. When we look at the top rules based on the lift, we are able to find the combination of players who can either increase or decrease the chance of winning a match. As the number of data points we have was limited, we had to trade-off support in order to get association rules of interest which have match result as the RHS in the rule.

4.1.5 Recommendation for home matches:

Assuming that this roster of players(who played from 2008-2016) will be available for future team selection, now we can use arules to make sure we have certain combinations of players that will increase the probability of the team winning a match and also avoid combinations of players that would increase the chance of losing a match.

4.1.6 Our Approach for away matches:

Now that we know how we can help improve our team's win probability at home, we also wanted to look at the combination of players that will improve the teams chance of winning away from home. We wanted to see if there are different combinations of players while playing away from home compared to at home that result in a win. So we went ahead to analyze the matches which AS Roma played away from home. (*Hypothesis*) Away matches are different from home matches when selecting the best team players

4.1.7 Our Analysis for away matches:

```
# Filter the mathches for Roma played at home
roma_record <- team %>%
               filter(team_long_name == 'Roma')

away_matches <- filter(match, away_team_api_id == roma_record$team_api_id)

# Filter out the required columns for converting match data to a transaction level
```

```r
txn <- away_matches %>% select(home_team_goal, away_team_goal,
                               away_player_1:away_player_11)

# Check for NA values and remove rows that contain NA values
txn <- na.omit(txn)

# convert the match result to win,loss or draw
txn$result <- txn$away_team_goal-txn$home_team_goal
txn$result <- ifelse(txn$result == 0, "Draw", ifelse(txn$result > 0,
                                                     "Win", "Loss"))

# Removing redundant columns from data
txn <- txn[,-c(1,2)]

# write the dataframe to csv inorder to read as transactions
write.csv(txn, "txn.csv")

# read back as transactions
txn_final <- read.transactions("txn.csv",format=c("basket"),sep=",",
                               rm.duplicates=TRUE)
```

```r
# Checking for player id combination which results in a WIN
rules <- apriori(txn_final,appearance = list(rhs = c('Win')),
                 parameter=list(supp = 0.04,
                                conf=0.7, minlen = 5, maxlen = 7))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.04      5
##  maxlen target   ext
##       7  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 6
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[263 item(s), 153 transaction(s)] done [0.00s].
## sorting and recoding items ... [69 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7
```

```
## Warning in apriori(txn_final, appearance = list(rhs = c("Win")), parameter
## = list(supp = 0.04, : Mining stopped (maxlen reached). Only patterns up to
## a length of 7 returned!
```

```
##  done [0.00s].
## writing ... [49 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
# Restricting the output format for lift,
# confidence and support to 2 decimals
options(digits=2)

# Sorting the rules by parameter lift
rules <- sort(rules,by="lift",decreasing=TRUE)
inspect(rules[1:5])
```

```
##     lhs                                       rhs   support confidence
## [1] {161328,169718,292462,41433}           => {Win} 0.046   1.00
## [2] {154249,22337,264842,96398}            => {Win} 0.046   1.00
## [3] {22337,24622,264842,30682}             => {Win} 0.046   0.88
## [4] {22337,264842,30682,46875,96398}       => {Win} 0.046   0.88
## [5] {22337,264842,30682,32746,46875,96398} => {Win} 0.046   0.88
##     lift count
## [1] 2.4  7
## [2] 2.4  7
## [3] 2.1  7
## [4] 2.1  7
## [5] 2.1  7
```

```r
# Checking for player id combination which result in a LOSS
rules <- apriori(txn_final,appearance = list(rhs = c('Loss')),
                 parameter=list(supp = 0.03, conf=0.7, minlen = 4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.03      4
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 4
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[263 item(s), 153 transaction(s)] done [0.00s].
## sorting and recoding items ... [74 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.00s].
## writing ... [80 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
# Restricting the output format for lift, confidence
# and support to 2 decimals
options(digits=2)

# Sorting the rules by parameter lift
rules <- sort(rules,by="lift",decreasing=TRUE)
```

```
# Filtering rules which are of interest
inspect(head(rules,5))
```

```
##     lhs                      rhs    support confidence lift count
## [1] {184822,18814,30841} => {Loss} 0.033   1          3.1  5
## [2] {184822,30841,39345} => {Loss} 0.033   1          3.1  5
## [3] {162497,30841,35112} => {Loss} 0.033   1          3.1  5
## [4] {162497,18814,30841} => {Loss} 0.039   1          3.1  6
## [5] {162497,30841,39345} => {Loss} 0.033   1          3.1  5
```

4.1.8 Findings and insights for away matches:

Looking at the results it is clear that the combination of players for an away match is different from home matches. This is in support of the hypothesis made before starting this analysis.

4.1.9 Recommendation for away matches:

Clearly, the performance of players away from home is different when compared to home matches. We can leverage arules to make sure we have the right combinations of players that will increase the probability of the team winning a match and also avoid combinations of players that would increase the chance of losing a match while choosing a team for matches played away from home.

4.2 Solution 2: Pinpoint the position on the field for key players

4.2.1 Why does the player on field position matter?

It is common in football that a player usually has multiple on-field positions while playing for a match. Does one position increase/decrease the chance of winning a match compared to other? To answer this we combined player id with the player on field position and applied association rules.

4.2.2 Our approach

The question we are going to answer here is, can we find a combination of player and position that is more likely to make the team win. The scenario we can control while selecting a team is the players and their position in the playing XI for each match. We are expecting to get arules which differ in support, confidence and lift for a player with different on-field position indicating which is the most/least favorable position.

4.2.3 Analysis

```
home_matches <- home_matches %>%
  mutate(player_Combo1 = paste(str_trim(home_player_1),sep = "-",
                               str_trim(home_player_X1)))
home_matches <- home_matches %>%
  mutate(player_Combo2 = paste(str_trim(home_player_2),sep = "-",
                               str_trim(home_player_X2)))
home_matches <- home_matches %>%
  mutate(player_Combo3 = paste(str_trim(home_player_3),sep = "-",
                               str_trim(home_player_X3)))
home_matches <- home_matches %>%
  mutate(player_Combo4 = paste(str_trim(home_player_4),sep = "-",
                               str_trim(home_player_X4)))
home_matches <- home_matches %>%
  mutate(player_Combo5 = paste(str_trim(home_player_5),sep = "-",
                               str_trim(home_player_X5)))
home_matches <- home_matches %>%
```

```
  mutate(player_Combo6 = paste(str_trim(home_player_6),sep = "-",
                               str_trim(home_player_X6)))
home_matches <- home_matches %>%
  mutate(player_Combo7 = paste(str_trim(home_player_7),sep = "-",
                               str_trim(home_player_X7)))
home_matches <- home_matches %>%
  mutate(player_Combo8 = paste(str_trim(home_player_8),sep = "-",
                               str_trim(home_player_X8)))
home_matches <- home_matches %>%
  mutate(player_Combo9 = paste(str_trim(home_player_9),sep = "-",
                               str_trim(home_player_X9)))
home_matches <- home_matches %>%
  mutate(player_Combo10 = paste(str_trim(home_player_10),sep = "-",
                                str_trim(home_player_X10)))
home_matches <- home_matches %>%
  mutate(player_Combo11 = paste(str_trim(home_player_11),sep = "-",
                                str_trim(home_player_X11)))

# Filter out the required columns
txn <- home_matches %>%
  select(home_team_goal, away_team_goal, player_Combo1:player_Combo11)
txn <- na.omit(txn)
txn$result <- txn$home_team_goal-txn$away_team_goal
# convert to win loss draw
txn$result <- ifelse(txn$result == 0, "Draw",
                     ifelse(txn$result > 0, "Win", "Loss"))
summary(txn)
```

```
##  home_team_goal away_team_goal player_Combo1      player_Combo2
##  Min.   :0      Min.   :0.0    Length:151         Length:151
##  1st Qu.:1      1st Qu.:0.0    Class :character   Class :character
##  Median :2      Median :1.0    Mode  :character   Mode  :character
##  Mean   :2      Mean   :0.9
##  3rd Qu.:3      3rd Qu.:2.0
##  Max.   :5      Max.   :4.0
##  player_Combo3      player_Combo4      player_Combo5
##  Length:151         Length:151         Length:151
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  player_Combo6      player_Combo7      player_Combo8
##  Length:151         Length:151         Length:151
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  player_Combo9      player_Combo10     player_Combo11
##  Length:151         Length:151         Length:151
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
```

```
##
##
##     result
##  Length:151
##  Class :character
##  Mode  :character
##
##
##
```

```
txn <- txn[,-c(1,2)]
#read the table to csv

# read back as transactions
txn_final <- read.transactions("txn.csv",
                               format=c("basket"),sep=",",
                               rm.duplicates=TRUE)
rules <- apriori(txn_final,appearance = list(rhs = c('Win','Loss')),
                 parameter=list(supp = 0.015,
                                conf=0.5,minlen = 2, maxlen=8))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5   0.015      2
##  maxlen target   ext
##       8  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2
##
## set item appearances ...[2 item(s)] done [0.00s].
## set transactions ...[263 item(s), 153 transaction(s)] done [0.00s].
## sorting and recoding items ... [81 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.02s].
## writing ... [8441 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#We are taking support to be less so as to capture the losses
# as loss percentage of Roma is low.

#formatting the results and view the summary
options(digits=2)
summary(rules)
```

```
## set of 8441 rules
##
## rule length distribution (lhs + rhs):sizes
##    2    3    4    5    6    7    8
##   37  422 1535 2517 2237 1241  452
```

```
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     2.0     5.0     5.0     5.4     6.0     8.0
##
## summary of quality measures:
##     support        confidence        lift          count
##  Min.   :0.020   Min.   :0.50   Min.   :1.18   Min.   : 3
##  1st Qu.:0.020   1st Qu.:0.60   1st Qu.:1.41   1st Qu.: 3
##  Median :0.020   Median :0.75   Median :1.77   Median : 3
##  Mean   :0.025   Mean   :0.75   Mean   :1.90   Mean   : 4
##  3rd Qu.:0.026   3rd Qu.:1.00   3rd Qu.:2.35   3rd Qu.: 4
##  Max.   :0.209   Max.   :1.00   Max.   :3.12   Max.   :32
##
## mining info:
##       data ntransactions support confidence
##  txn_final           153   0.015        0.5
```

```
# depending on the measure we care about,
# we can sort the rules before looking
# at the top candidates. This command sorts the rules based on lift.
# we then reinspect the top 5.
rules <- sort(rules,by="lift",decreasing=TRUE)
```

```
inspect(rules[1:20])
```

```
##      lhs                     rhs    support confidence lift count
## [1]  {39206}              => {Loss} 0.020   1          3.1  3
## [2]  {27693,39206}        => {Loss} 0.020   1          3.1  3
## [3]  {30996,39206}        => {Loss} 0.020   1          3.1  3
## [4]  {39206,39351}        => {Loss} 0.020   1          3.1  3
## [5]  {27695,39206}        => {Loss} 0.020   1          3.1  3
## [6]  {30682,39206}        => {Loss} 0.020   1          3.1  3
## [7]  {30747,96643}        => {Loss} 0.020   1          3.1  3
## [8]  {174850,206711}      => {Loss} 0.020   1          3.1  3
## [9]  {39447,41329}        => {Loss} 0.020   1          3.1  3
## [10] {184822,41023}       => {Loss} 0.020   1          3.1  3
## [11] {162497,30841}       => {Loss} 0.046   1          3.1  7
## [12] {162497,41023}       => {Loss} 0.020   1          3.1  3
## [13] {18814,30747}        => {Loss} 0.026   1          3.1  4
## [14] {18814,41023}        => {Loss} 0.026   1          3.1  4
## [15] {27693,30996,39206}  => {Loss} 0.020   1          3.1  3
## [16] {27693,39206,39351}  => {Loss} 0.020   1          3.1  3
## [17] {27693,27695,39206}  => {Loss} 0.020   1          3.1  3
## [18] {27693,30682,39206}  => {Loss} 0.020   1          3.1  3
## [19] {30996,39206,39351}  => {Loss} 0.020   1          3.1  3
## [20] {27695,30996,39206}  => {Loss} 0.020   1          3.1  3
```

4.2.4 Findings and insights for away matches:

We can see that there are certain combination of players which implies a loss or win with a probability defined by the confidence. The High lift can be noticed as well which means a loss/win x times as compared to them playing at a random position.

4.2.5 Recommendation for away matches:

We can be cautious about who all should be rested and which players need to play at what positions so as to increase the chances of winning against other teams.

4.3 Solution 3: Select new players on key player attributes and training team to improve them

4.3.1 Why player attributes?

We want to identify what non-obvious player attribute combinations have worked successfully for AS Roma. The Coach can use these patterns for training players on attributes that appear to be important. This can also be used towards selecting players which have these successful attribute combinations.

4.3.2 Our approach

We want to identify players who have played for Roma between 2008 and 2016 and measure their average attribute-level performance by taking the median score for every attribute. *Assumption* We are assuming here that the players' dominant attributes remain similar over the years. To capture how these player attributes translate to success, what should be the optimum success metric?

We do not want to look at win-loss percentage as that does not capture the difference in goals scored by opponents. The main premise of a win is to score as many goals as possible and prevent as many goals of the opponent. To measure the performance of a player with particular attributes across matches, we can see how his team performed when he was paying. We use the goal difference for all the matches he played. If a player has played 10 matches between 2008-16, his average goal difference tells us how many goals were scored vs prevented for matches where this player was in our team. (https://talksport.com/football/389411/world-cup-groups-goal-difference-head-to-head-england-belgium/) This is similar to the plus/minus metric used in games like Basketball. (https://content.iospress.com/articles/journal-of-sports-analytics/jsa225)

We could also take weighted mean goal difference with a win rate of opponents as weights to account for the difficulty of the match but we take simple mean goal difference to keep the model simple for now. That could be an extension of this work in the future.

Player attributes work in combinations. There are primarily grouped into three categories - Mental, Attack, and Defense. There are several attributes under each and its impossible to train players on all. We use Association Rules to identify non-obvious combinations have worked for AS Roma in the past by increasing mean goal difference at player level and which can be used by the Coach to increase mean goal difference; eventually wins and higher chances of qualifying for further stages.

4.3.3 Analysis

- There are over 700 missing items most of which are in betting related columns which we will not be using for player attribute analysis. We drop these columns.Of the remaining 9, 8 are for away team players which we will look at later. 1 is for home_team_player_5. We check the player that played most as home_player_5 in this Season 2015/16 and map it to the missing value after ensuring that this player wasn't amongst the other players.

```
sum(is.na(home_matches))
```

```
## [1] 726
```

```
home_matches <- select(home_matches, c(1:77))
home_matches$home_player_5[home_matches$season == "2015/2016"]
```

```
##  [1] 210111 210111 210111 210111 210111  30682 210111 276738  34305 210111
## [11] 210111 210111 210111     NA 210111 210111  71500 210111 210111
```

```
210111 %in%
  home_matches[(home_matches$season ==
                  "2015/2016")&(home_matches$id ==13173 ),]
```

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

```
home_matches$home_player_5[(home_matches$season ==
                              "2015/2016")&(home_matches$id ==
                                              13173 )] <- 210111

roma_players_per_match <- select(home_matches,
                                 id,
                                 matches("home_player_[[:digit:]]")) %>%
  collect() %>%
  gather(player, player_api_id, -id) %>%
  rename(match_id = id)
```

- DATA TRANSFORMATION:- We combine the player attributes and match table filtered for Roma matches and players, to get player attribute details and the corresponding mean goal difference.

1. We filter data for home players, gather by players to convert to long format. This gives us 1661 observations (151*11).
2. We then group player attributes using median for the 94 unique players that played for Roma in this period.
3. We merge the data back on matchId to get player attributes and calculate the mean goal_difference.

```
home_matches2 <- rename(home_matches,match_id= id)
roma_player_match <- merge(roma_players_per_match,home_matches2,
                           by.x = "match_id",by.y="match_id", all.x=TRUE)
roma_player_match <- mutate(roma_player_match,
                            goal_diff = home_team_goal - away_team_goal)
#1661 obs

goal_player <- roma_player_match%>%
  group_by(player_api_id)%>%
  summarise(goal_mean = round(mean(goal_diff),2))%>%
  select(player_api_id, goal_mean)


player_atts_tbl <- collect(player_atts_tbl)
roma_players_all <-unique(roma_players_per_match$player_api_id)
roma_players_all <- tbl_df(roma_players_all)
colnames(roma_players_all) <- "player_api_id"
roma_players_att <- tbl_df(player_atts_tbl[
  (player_atts_tbl$player_api_id %in%
     roma_players_all$player_api_id),])



roma_players_att2 <- roma_players_att %>%
  select(c(-player_fifa_api_id,-id,-date,-preferred_foot,-attacking_work_rate,
           -defensive_work_rate))%>%
  group_by(player_api_id) %>%
  summarise_all(funs(if(is.numeric(.)) median(., na.rm = TRUE) else max(.)))
```

```
sum(is.na(roma_players_att2))
```

```
## [1] 14
```

```
roma_players_att2$player_api_id[is.na(roma_players_att2$sliding_tackle)]
```

```
## [1] 27693 39410
```

```
# Some attributes are missing for only two players. We use the median
# attribute value across players of Roma to fill this value. Assumption
# is that these players have median # performance on these attributes.

roma_players_att2$sliding_tackle[
  is.na(roma_players_att2$sliding_tackle)] <-
  median((roma_players_att2$sliding_tackle),na.rm = TRUE)

roma_players_att2$volleys[is.na(roma_players_att2$volleys)] <-
  median((roma_players_att2$volleys),na.rm = TRUE)

roma_players_att2$curve[is.na(roma_players_att2$curve)] <-
  median((roma_players_att2$curve),na.rm = TRUE)

roma_players_att2$agility[is.na(roma_players_att2$agility)] <-
  median((roma_players_att2$agility),na.rm = TRUE)

roma_players_att2$balance[is.na(roma_players_att2$balance)] <-
  median((roma_players_att2$balance),na.rm = TRUE)

roma_players_att2$jumping[is.na(roma_players_att2$jumping)] <-
  median((roma_players_att2$jumping),na.rm = TRUE)

roma_players_att2$vision[is.na(roma_players_att2$vision)] <-
  median((roma_players_att2$vision),na.rm = TRUE)

Roma_player_match_att <- merge(goal_player,roma_players_att2,
                               by.x = "player_api_id",
                               by.y = "player_api_id",
                               all.x= TRUE)
```

- We then transform the data to transaction level data by discretizing the numerical variables which are all the selected attributes by breaking them into two categories. A player is high on an attribute if he is better than the median.

```
df <- as.data.frame(Roma_player_match_att)
df2 <- select(df, -c(33:37))
```

```
fn <- function(x)discretize(x,"frequency",
                            categories = 2,
                            label =c(0,1))
df3 <- tbl_df(cbind(df2[1], sapply(df2[2:32],fn)))
df3 <- select(df3, -player_api_id)
```

```
#Converting the attributes to numerical values
df4 <- sapply(df3, as.numeric)
df5 <- df4[, 1:31]-1
```

- Converting to transactions Matrix

```
txn_Matrix <- as(df5, "itemMatrix")
```

```
txn_p_attr <- apriori(data = txn_Matrix,
                      appearance = list(rhs = c('goal_mean') ),
                      parameter = list(support= 0.1,
                                       confidence = 0.70,
                                       maxlen =4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target   ext
##       4  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[31 item(s), 94 transaction(s)] done [0.00s].
## sorting and recoding items ... [31 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [261 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
#Inspect rules only where rhs is goal_mean

txn_p_attr <- sort(txn_p_attr,
                   by = c("lift", "confidence"),
                   decreasing = TRUE)

inspect(txn_p_attr[1:5])
```

```
##     lhs                  rhs         support confidence lift count
## [1] {ball_control,
##      aggression,
##      marking}         => {goal_mean}    0.13       0.86  1.5    12
## [2] {short_passing,
##      ball_control,
##      aggression}      => {goal_mean}    0.16       0.83  1.5    15
## [3] {ball_control,
##      aggression,
##      sliding_tackle}  => {goal_mean}    0.15       0.82  1.5    14
## [4] {ball_control,
##      aggression,
##      standing_tackle} => {goal_mean}    0.14       0.81  1.5    13
## [5] {ball_control,
##      aggression,
##      interceptions}   => {goal_mean}    0.14       0.81  1.5    13
```

4.3.4 Findings and insights

1. The Magic 3 as we could call it is the 3 player attributes combination that leads to better player performance and eventually higher chances of a win in home_matches for AS Roma. These are ball control, aggression, marking; a combination of attacking, mental and defensive abilities. [1] {ball_control,aggression,marking} => {goal_mean} 0.128 0.857 1.549 12 This means players who have high ball control, aggression and marking have ~55% higher mean goal difference, 85.7% of the times they have been used. This means that more goals have been scored or/and more goals from opponents have been prevented when players with these attribute combinations were on the field. This translates directly into ~55% more chance of winning. The Coach has used them in only 12.8% of the matches.
2. Crossing and short passing is a combination that has been used most (28.7% of the matches) but has resulted in 28.4% higher mean goal difference 71% of the times. On the other hand, Crossing along with ball control and agression are far more rewarding 81% of the times they have been used in ~14% of the matches. {crossing,ball_control,aggression} => {goal_mean} 0.138 0.813 1.469 13
3. An obvious pattern is ball control and a sliding tackle. Sliding tackle is a tricky defensive skill as it trades off balance and reduces the reaction time. This skill, when combined with good ball control has proven to be rewarding with 40% higher goal mean for matches ~78% of the times when this combination was used. (https://www.myactivesg.com/sports/football/training-methods/football-for-beginners/when-to-use-a-slide-tackle-in-football). (https://www.guidetofm.com/players/attribute-combinations) [22] {ball_control,sliding_tackle} => {goal_mean} 0.149 0.778 1.406 14

The non-obvious one is sliding tackle and ball control with long shots. This combination of attack and defense has resulted in a mean goal difference that is 42% higher than if this attribute combination was absent 78.6% of the times. The Coach has used this combination only 11.7% of the times. [21] {ball_control,long_shots,sliding_tackle} => {goal_mean} 0.117 0.786 1.420 11

4. Marking does require the obvious vision and ball control but when the player is able to intercept a pass made by the opposite team, the goal difference can increase by 42% in ~79% of the cases. This combination has been used only 11.7% of the times, [18] {ball_control,interceptions,marking} => {goal_mean} 0.117 0.786 1.420 11

4.3.5 Recommendation:

1. We recommend that the Coach continues to build on the Magic 3 player attributes.
2. Crossing and short passing has been used often but has not been rewarding. We suggest an alternative combination of Crossing, aggression (attack) and ball control (defense). This has not been tried as often but has proven to be very successful in the past.
3. Long shots and sliding tackle can be used by midfielders to attack and also prevent the opposite team from coming too close to the goal.
4. Players must be able to intercept the passes made towards the opponent player to whom they have been marked. This would reduce ball possession of the away team. These attribute combinations can be used by the Coach to train players and to select players that posses these combinations to increase goal difference and consequently win rate.

There are several attributes that are important for the game but it is not possible to focus on all at the same time or customize all based on opponents. We try and identify the non-obvious combination of attributes that have worked for AS Roma in the past by increasing mean goal difference at the player level. We use association rules to explore this combination which could be used by the Coach of AS Roma to maintain and increase mean goal difference which would eventually lead to more wins and higher chances of qualifying for further stages.

4.4 Solution 4: Strengthen advantageous team attributes

We have data regarding the attributes and tactics that different teams have used over the years. We what to see what combinations of tactics and general team features result in wins most often.

4.4.1 Why team attributes matter?

Knowing what team attributes and tactics that have the strongest association with winning will allow you to format your team in a way that has the greatest chance for success First, we need to collect the two tables we will be using and clean it to the appropriate format.

4.4.2 Analysis

```
team_atts_tbl <- tbl(con, "team_attributes") %>% collect()

# Attach the column names to the values so we can
# differentiate them easily when we find association rules

team_atts_tbl$buildUpPlaySpeedClass <-
  paste(team_atts_tbl$buildUpPlaySpeedClass,
        "buildUpPlaySpeedClass", sep = '_')

team_atts_tbl$buildUpPlayDribblingClass <- paste(
  team_atts_tbl$buildUpPlayDribblingClass,
  "buildUpPlayDribblingClass", sep = '_')

team_atts_tbl$buildUpPlayPassingClass <- paste(
  team_atts_tbl$buildUpPlayPassingClass,
  "buildUpPlayPassingClass", sep = '_')

team_atts_tbl$buildUpPlayPositioningClass <- paste(
  team_atts_tbl$buildUpPlayPositioningClass,
  "buildUpPlayPositioningClass", sep = '_')

team_atts_tbl$chanceCreationPassingClass <- paste(
  team_atts_tbl$chanceCreationPassingClass,
  "chanceCreationPassingClass", sep = '_')

team_atts_tbl$chanceCreationCrossingClass <- paste(
  team_atts_tbl$chanceCreationCrossingClass,
  "chanceCreationCrossingClass", sep = '_')

team_atts_tbl$chanceCreationShootingClass <- paste(
  team_atts_tbl$chanceCreationShootingClass,
  "chanceCreationShootingClass", sep = '_')

team_atts_tbl$chanceCreationPositioningClass <- paste(
  team_atts_tbl$chanceCreationPositioningClass,
  "chanceCreationPositioningClass", sep = '_')

team_atts_tbl$defencePressureClass <- paste(
  team_atts_tbl$defencePressureClass,
  "defencePressureClass", sep = '_')

team_atts_tbl$defenceAggressionClass <- paste(
  team_atts_tbl$defenceAggressionClass,
  "defenceAggressionClass", sep = '_')

team_atts_tbl$defenceTeamWidthClass <- paste(
```

```
  team_atts_tbl$defenceTeamWidthClass,
  "defenceTeamWidthClass", sep = '_')
team_atts_tbl$defenceDefenderLineClass <- paste(
  team_atts_tbl$defenceDefenderLineClass,
  "defenceDefenderLineClass", sep = '_')

# Now we select only the columns we want to merge with the matches table
team_atts_tbl <- team_atts_tbl %>%
  select(team_fifa_api_id, team_api_id, date,
         buildUpPlaySpeedClass,
         buildUpPlayPassingClass,
         buildUpPlayDribblingClass,
         buildUpPlayPositioningClass,
         chanceCreationCrossingClass,
         chanceCreationPassingClass,
         chanceCreationShootingClass,
         chanceCreationPositioningClass,
         defencePressureClass,
         defenceAggressionClass,
         defenceTeamWidthClass,
         defenceDefenderLineClass)

# Convert to date time for easier merge
team_atts_tbl$date <- ymd_hms(team_atts_tbl$date)

# Calculate the match results for all matches
league_matches <- collect(match_tbl) %>%
  filter(league_id == 10257) %>%
  select(home_team_goal,
         away_team_goal,
         away_team_api_id,
         home_team_api_id,
         date)

league_matches$result <- league_matches$home_team_goal-
  league_matches$away_team_goal

# Convert result column to win, loss, draw
league_matches$result <- ifelse(league_matches$result == 0,
                                "Draw", ifelse(league_matches$result > 0,
                                               "Win", "Loss"))

# Convert to date time for easier merge
league_matches$date <- ymd_hms(league_matches$date)

# Merge the team attributes with the match table using home
# team id and date to keep things consistent
team_atts_tbl <- dplyr::rename(team_atts_tbl,
                               home_team_api_id = team_api_id)
team_atts_tbl$date <- year(team_atts_tbl$date)
league_matches$date <- year(league_matches$date)
new_table <- merge(league_matches, team_atts_tbl,
                   by = c("home_team_api_id","date"),
```

```
                    all.x=FALSE, all.y=FALSE)

# Create a new table with only the factors we need
item_data_frame <- new_table %>%
  select(-home_team_api_id, -date,
         -home_team_goal,
         -away_team_goal,
         -away_team_api_id,
         -team_fifa_api_id)
```

We now have a table that has both team attributes and results of the match. We can then proceed to use this table to analyze what combinations of team attributes result in wins most often.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.3    0.1    1 none FALSE            TRUE       5    0.05      5
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 113
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[2308 item(s), 2261 transaction(s)] done [0.00s].
## sorting and recoding items ... [31 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.02s].
## writing ... [4445 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
## set of 4445 rules
##
## rule length distribution (lhs + rhs):sizes
##    5    6    7    8    9   10
##  853 1048 1019  809  496  220
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     5.0     6.0     7.0     6.9     8.0    10.0
##
## summary of quality measures:
##     support        confidence        lift          count
##  Min.   :0.050   Min.   :0.36   Min.   :0.79   Min.   :114
##  1st Qu.:0.103   1st Qu.:0.42   1st Qu.:0.92   1st Qu.:234
##  Median :0.136   Median :0.44   Median :0.95   Median :307
##  Mean   :0.135   Mean   :0.44   Mean   :0.96   Mean   :305
##  3rd Qu.:0.168   3rd Qu.:0.45   3rd Qu.:0.98   3rd Qu.:380
##  Max.   :0.300   Max.   :0.61   Max.   :1.33   Max.   :679
##
## mining info:
##       data ntransactions support confidence
```

```
##  team_atts          2261    0.05        0.3
```

```
inspect(rules[1:10])
```

```
##      lhs                                             rhs   support confidence lift count
## [1]  {Balanced_buildUpPlaySpeedClass,
##       Lots_chanceCreationShootingClass,
##       Medium_defencePressureClass,
##       Normal_defenceTeamWidthClass}              => {Win}   0.055       0.61  1.3   124
## [2]  {Cover_defenceDefenderLineClass,
##       Lots_chanceCreationShootingClass,
##       Normal_defenceTeamWidthClass,
##       Organised_buildUpPlayPositioningClass}     => {Win}   0.060       0.61  1.3   135
## [3]  {Balanced_buildUpPlaySpeedClass,
##       Lots_chanceCreationShootingClass,
##       Normal_chanceCreationCrossingClass,
##       Normal_defenceTeamWidthClass}              => {Win}   0.057       0.60  1.3   128
## [4]  {Balanced_buildUpPlaySpeedClass,
##       Lots_chanceCreationShootingClass,
##       Normal_defenceTeamWidthClass,
##       Organised_buildUpPlayPositioningClass}     => {Win}   0.052       0.60  1.3   117
## [5]  {Cover_defenceDefenderLineClass,
##       Lots_chanceCreationShootingClass,
##       Normal_chanceCreationCrossingClass,
##       Normal_defenceTeamWidthClass}              => {Win}   0.057       0.60  1.3   130
## [6]  {Balanced_buildUpPlaySpeedClass,
##       Lots_chanceCreationShootingClass,
##       Medium_defencePressureClass,
##       Organised_buildUpPlayPositioningClass}     => {Win}   0.054       0.60  1.3   122
## [7]  {Lots_chanceCreationShootingClass,
##       Normal_defenceTeamWidthClass,
##       Organised_buildUpPlayPositioningClass,
##       Organised_chanceCreationPositioningClass} => {Win}   0.053       0.59  1.3   120
## [8]  {Cover_defenceDefenderLineClass,
##       Lots_chanceCreationShootingClass,
##       Normal_chanceCreationCrossingClass,
##       Organised_buildUpPlayPositioningClass}     => {Win}   0.053       0.59  1.3   119
## [9]  {Lots_chanceCreationShootingClass,
##       Normal_chanceCreationCrossingClass,
##       Normal_defenceTeamWidthClass,
##       Organised_chanceCreationPositioningClass} => {Win}   0.052       0.59  1.3   118
## [10] {Lots_chanceCreationShootingClass,
##       Normal_chanceCreationCrossingClass,
##       Normal_defenceTeamWidthClass,
##       Organised_buildUpPlayPositioningClass}     => {Win}   0.062       0.59  1.3   140
```

4.4.3 Findings and Insights

We see that a value of "lots" in the chance creation column is associated with wins in the top 10 rules sorted by confidence. This means that teams with the highest average amount of shots frequently win. This may seem obvious that teams that shoot more win more, but the data suggests that teams should focus on shooting frequently over looking for the perfect scoring opportunity. Another finding is, teams employing a "balanced" approach to attacking often results in wins, as it appears in many of the top 10 rules.

Now that we know what team attributes are most successful I want to check Roma's attributes to see which

attributes most closely match what you are currently utilizing.

```
glimpse(filter(team_atts_tbl, home_team_api_id == 8686, date == '2015'))
```

```
## Observations: 1
## Variables: 15
## $ team_fifa_api_id               <int> 52
## $ home_team_api_id               <int> 8686
## $ date                           <dbl> 2015
## $ buildUpPlaySpeedClass          <chr> "Fast_buildUpPlaySpeedClass"
## $ buildUpPlayPassingClass        <chr> "Mixed_buildUpPlayPassingClass"
## $ buildUpPlayDribblingClass      <chr> "Little_buildUpPlayDribblingClass"
## $ buildUpPlayPositioningClass    <chr> "Organised_buildUpPlayPositioni...
## $ chanceCreationCrossingClass    <chr> "Normal_chanceCreationCrossingC...
## $ chanceCreationPassingClass     <chr> "Risky_chanceCreationPassingClass"
## $ chanceCreationShootingClass    <chr> "Normal_chanceCreationShootingC...
## $ chanceCreationPositioningClass <chr> "Free Form_chanceCreationPositi...
## $ defencePressureClass           <chr> "Medium_defencePressureClass"
## $ defenceAggressionClass         <chr> "Press_defenceAggressionClass"
## $ defenceTeamWidthClass          <chr> "Normal_defenceTeamWidthClass"
## $ defenceDefenderLineClass       <chr> "Cover_defenceDefenderLineClass"
```

4.4.4 Recommendations

Roma currently is in the "normal" category for chance creation shooting. We recommend you formulate your offense in a way in which getting shots off is prioritized. We also noticed that teams that took a balanced approach to attacking showed up in our top rules frequently, suggesting that this approach contributes to winning. Roma currently builds attacks quickly. We recommended taking a more balanced approach to this and slowing down the pace. In addition, slowing down the pace of attacks will help balance out the change we recommended to increase shots and will ease the transition to these new tactics.

5 Warnings or Potential Drawbacks of our analysis

1. We assume that player performance on attributes have not changed much over the last 8 years and a median would be a good indicator of how he fared on these attributes between 2008-16.
2. We treat all opponents equally in our analysis. One risk is that a goal difference of 2 means more when achieved against a tough opponent (like FC Barcelona) as against an easy opponent.