

# Homework 1: AS Roma - Association Rules for Competitive Advantage

*Brady Engelke, Adithya Ramesh, Jie Liu, Priya Priya, Sharang Jindal, Zixuan Wang*

*09/29/2019*

## Contents

0.1	Situation Overview . . . . .	1
0.2	What is the competitive landscape of the ISA? . . . . .	1
0.2.1	How does Roma stack up against the rest of the teams in the ISA? . . . . .	2
0.2.2	How have the top 5 ISA teams been trending in terms of win percentage ? . . . . .	6
0.2.3	How does Roma fare against the top teams in the ISA? . . . . .	7
0.3	How can Roma beat Juventus? . . . . .	9
0.3.1	Are there stages throughout the match where Juventus' defense is weak? . . . . .	9
0.3.2	Can we validate Juventus' defensive weaknesses using association rules? . . . . .	12
0.4	What player attributes are most indicative of success in the ISA? . . . . .	15
0.4.1	How does Roma's player skillset compare to the skillset of the best players in the ISA? . . . . .	22
0.5	Final Recommendations . . . . .	23

## 0.1 Situation Overview

The coach of AS Roma has hired the team to derive value from of a dataset relating to matches played amongst European soccer teams from 2008 to 2016. The dataset contains detailed information for over 25,000 matches as well as information regarding the team strategies employed and the skillsets of the players on the field. The dataset under analysis can be found below:

Link

The purpose of this analysis is to provide the coach of AS Roma with actionable insights that will help solve the most important and pressing issues. We assume the coach strives to become the number one team in AS Roma's league and this analysis will detail actions that can be taken to accomplish just that.

The success of this analysis will be dependent upon the identification of strategies that the coach can employ to solve pertinent issues the team faces. The coach's comprehension of the team's recommendations is critical to ensure this analysis proves worthwhile. Thus, the team has provided visualizations of all crucial findings accompanied by detailed explanations of how the team arrived at these insights.

Below are the three key questions this analysis answers: 1. What is the competitive landscape of AS Roma's league - Italy Serie A (ISA)? 2. How can AS Roma beat Juventus? 3. What player skillset is most indicative of success in the ISA and how does AS Roma's skillset compare?

The primary software used to answer these questions was R. Specifically, R's Tidyverse to wrangle the raw data into meaningful visualizations and the popular Apriori algorithm was utilized to identify association rules. Lets dive in.

## 0.2 What is the competitive landscape of the ISA?

Before extracting association rules, the outcomes of interest had to be defined in order to obtain meaningful results from the Apriori algorithm. The team set out to define the crucial questions the coach of Roma likely has.

First, we compared how Roma stacks up against all other teams in the ISA in terms of the percentage of games played that resulted in wins, losses, or ties. Roma's past performance was then compared to other ISA

teams on a slightly different metric - mean goal differential.

It was found that Roma is indeed a top performer in the ISA. With this information in hand, we then sought to understand how the performance of the top teams in the ISA has been trending year of year as well as how Roma has performed against these teams.

Assumptions: 1. Each league is idiosyncratic. Thus, only trends found in ISA will be relevant to Roma's future success. 2. Mean goal differential is a quality metric to judge a team's performance across many matches.

### 0.2.1 How does Roma stack up against the rest of the teams in the ISA?

To gather a baseline understanding of the teams within the ISA, the team created a match outcome distribution bar plot and a mean goal differential bar plot that compared all of the teams in the ISA.

#### Execution and Results:

*Load necessary packages and munge database*

```
library("dplyr")
library(magrittr)
library(tidyverse)
library(arules)
library(dplyr)
library(naniar)
library(RColorBrewer)
library(xml2)
library("purrr")
library("XML")
library("methods")
library(knitr)
```

```
database <- src_sqlite("euro_soccer.sqlite")
country_tbl <- data.frame(tbl(database, "country"))
league_tbl <- data.frame(tbl(database, "league"))
match_tbl <- data.frame(tbl(database, "match"))
player_tbl <- data.frame(tbl(database, "player"))
player_atts_tbl <- data.frame(tbl(database, "player_attributes"))
team_tbl <- data.frame(tbl(database, "team"))
team_atts_tbl <- data.frame(tbl(database, "team_attributes"))
```

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

*Data transformation to create visualization of match outcome distribution for all teams in ISA*

```
outcomes <- match_tbl %>%
  filter(league_id == 10257) %>%
  select(league_id, home_team_api_id, away_team_api_id, home_team_goal, away_team_goal) %>%
  mutate(goal_diff = home_team_goal - away_team_goal) %>%
  mutate(results = ifelse(goal_diff > 0, 'home', ifelse(goal_diff < 0, 'away', 'draw'))))

attach(outcomes)
df1 <- table(home_team_api_id, results)
df1 <- as.data.frame(df1)
df1 <- spread(df1, results, Freq)
```

```

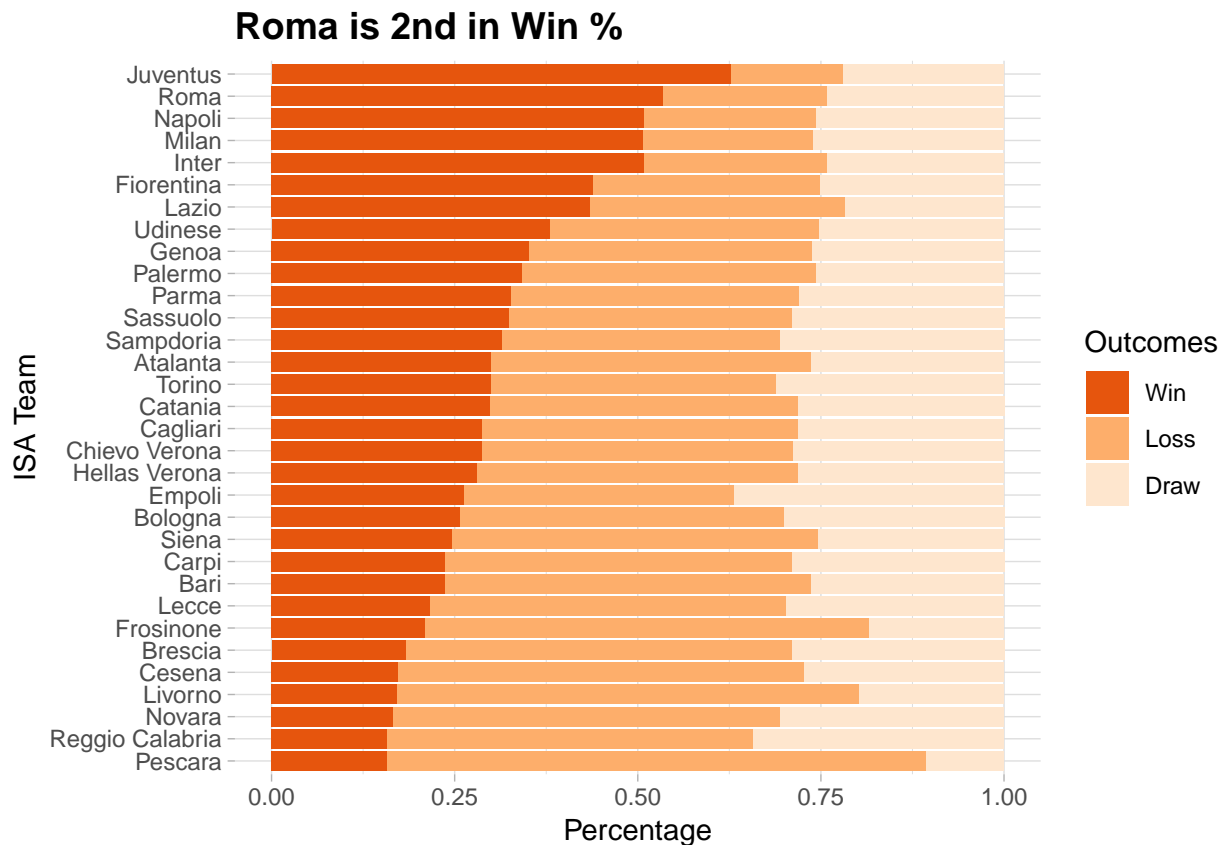
names(df1) <- c('Team', "Loss", "Draw", "Win")
df2 <- table(away_team_api_id, results)
df2 <- as.data.frame(df2)
df2 <- spread(df2, results, Freq)
names(df2) <- c('Team', 'Win', 'Draw', 'Loss')
df <- merge(df1, df2, by= 'Team')
df$total = rowSums(df[2:7])

df <- df %>%
  mutate(Win = (Win.x + Win.y)/total,
         Draw = (Draw.x + Draw.y)/total,
         Loss = (Loss.x + Loss.y)/total) %>%
  arrange(Win)

df <- gather(df[,c(1,9:11)], results, new_value, -Team)
df_ord <- df %>% filter(results == 'Win')
df <- merge(df, df_ord, by = 'Team')
df <- merge(df, team_tbl, by.x = 'Team', by.y = 'team_api_id')
df <- df %>%
  mutate(highlight_flag = ifelse(team_long_name == 'Roma', T, F))

myColors <- brewer.pal(3, "Oranges")
ggplot(df) + geom_bar(aes(reorder(team_long_name, desc(-new_value.y)),
                        y = new_value.x, fill = results.x), stat='identity') +
  labs(x = 'ISA Team',
       y = 'Percentage',
       fill = 'Outcomes',
       title = 'Roma is 2nd in Win %') +
  guides(fill = guide_legend(reverse=TRUE))+
  coord_flip() + scale_fill_manual(values = myColors) +
  theme_light() +
  theme(panel.border = element_blank(),
        plot.title = element_text(size = 14, face = 'bold'))

```



*Data transformation to create visualization of mean goal differential for all teams in ISA*

```
Italy_Serie_A_matches <- match_tbl %>%
  filter(league_id == 10257) %>%
  select(season, date, match_api_id:away_team_goal)
```

```
Italy_Serie_A_matches$season <- substr(Italy_Serie_A_matches$season, start = 1, stop = 4)
```

```
home_teams <- Italy_Serie_A_matches %>%
  select(home_team_api_id, home_team_goal, away_team_goal) %>%
  mutate(goal_differential = home_team_goal - away_team_goal) %>%
  rename(goals = home_team_goal,
         opponent_goals = away_team_goal,
         team_api_id = home_team_api_id)
```

```
away_teams <- Italy_Serie_A_matches %>%
  select(away_team_api_id, home_team_goal, away_team_goal) %>%
  mutate(goal_differential = away_team_goal - home_team_goal) %>%
  rename(goals = away_team_goal,
         opponent_goals = home_team_goal,
         team_api_id = away_team_api_id)
```

```
teams <- rbind(home_teams, away_teams)
```

```
teams_avg <- teams %>%
  group_by(team_api_id) %>%
  summarise(mean_goals = mean(goals),
            mean_opponent_goals = mean(opponent_goals),
```

```

mean_goal_differential = mean(goal_differential))

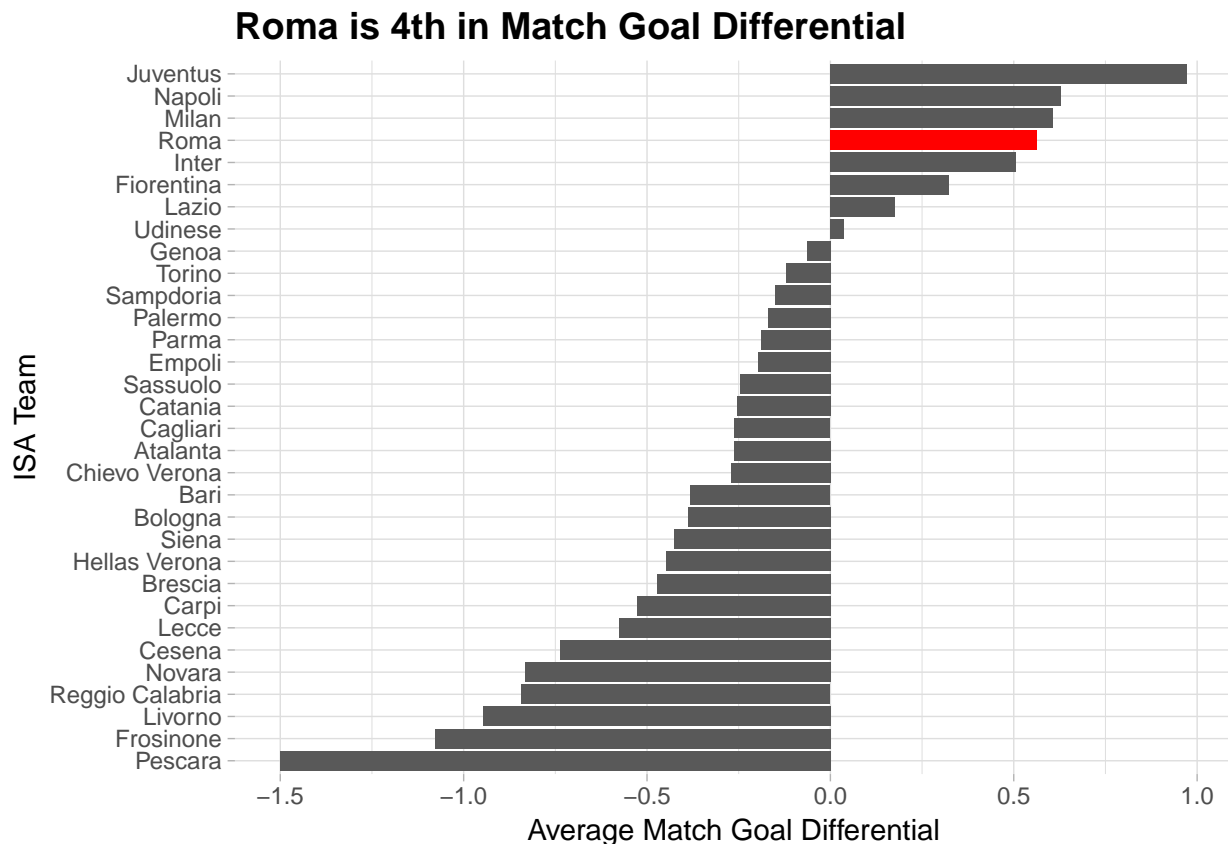
isa_team_names <- team_tbl %>%
  filter(team_api_id %in% teams_avg$team_api_id) %>%
  select(team_api_id, team_long_name)

teams_avg <- merge(teams_avg, isa_team_names, by = 'team_api_id') %>%
  select(team_long_name, mean_goals:mean_goal_differential)

teams_avg$team_long_name <- factor(teams_avg$team_long_name,
  levels = teams_avg$team_long_name[order(teams_avg$mean_goal_differential)])
teams_avg <- teams_avg %>%
  mutate(highlight_flag = ifelse(team_long_name == 'Roma', T, F))

ggplot(teams_avg, aes(x = team_long_name, y = mean_goal_differential)) +
  geom_col(aes(fill = highlight_flag)) +
  scale_fill_manual(values = c('#595959', 'red')) +
  coord_flip() + theme_light() +
  theme(legend.position = 'None',
    plot.title = element_text(size = 14, face = 'bold'),
    panel.border = element_blank()) +
  labs(x = 'ISA Team', y = 'Average Match Goal Differential',
    title = 'Roma is 4th in Match Goal Differential')

```



**Interpretation & Conclusions** The above graphs clearly depict where Roma falls in the hierarchy of the league. Roma is top 5 in the league based on both metrics the team based team performance on. It is interesting that Roma is fourth in the league in terms of average match goal differential but second in the league in win percentage. It could be that Roma's players are particularly clutch or that Roma rarely blows

out teams. It was also inferred that there is a demonstrable difference between the performance of the top 5 teams and the rest of the league seeing that the mean goal differential and the win % dropped off after the fifth place team. This motivated the team to dig deeper into the performance of the top 5 teams.

## 0.2.2 How have the top 5 ISA teams been trending in terms of win percentage ?

To gather a better understanding of the performance of the top 5 teams in the ISA, the team plotted how each team has performed over time.

### Execution and Results:

```
home <- Italy_Serie_A_matches %>%
  select(season, home_team_api_id, home_team_goal, away_team_goal) %>%
  mutate(outcome = ifelse(away_team_goal - home_team_goal > 0, 'won',
                          ifelse(away_team_goal - home_team_goal < 0, 'loss', 'tie'))) %>%
  rename(goals = home_team_goal, opponent_goals = away_team_goal,
         team_api_id = home_team_api_id)

away <- Italy_Serie_A_matches %>%
  select(season, away_team_api_id, home_team_goal, away_team_goal) %>%
  mutate(outcome = ifelse(away_team_goal - home_team_goal > 0, 'won',
                          ifelse(away_team_goal - home_team_goal < 0, 'loss', 'tie'))) %>%
  rename(goals = away_team_goal, opponent_goals = home_team_goal,
         team_api_id = away_team_api_id)

team_records <- rbind(home, away)

team_records <- team_records %>%
  mutate(win = ifelse(outcome == 'won', 1, 0),
         loss = ifelse(outcome == 'loss', 1, 0),
         tie = ifelse(outcome == 'tie', 1, 0))

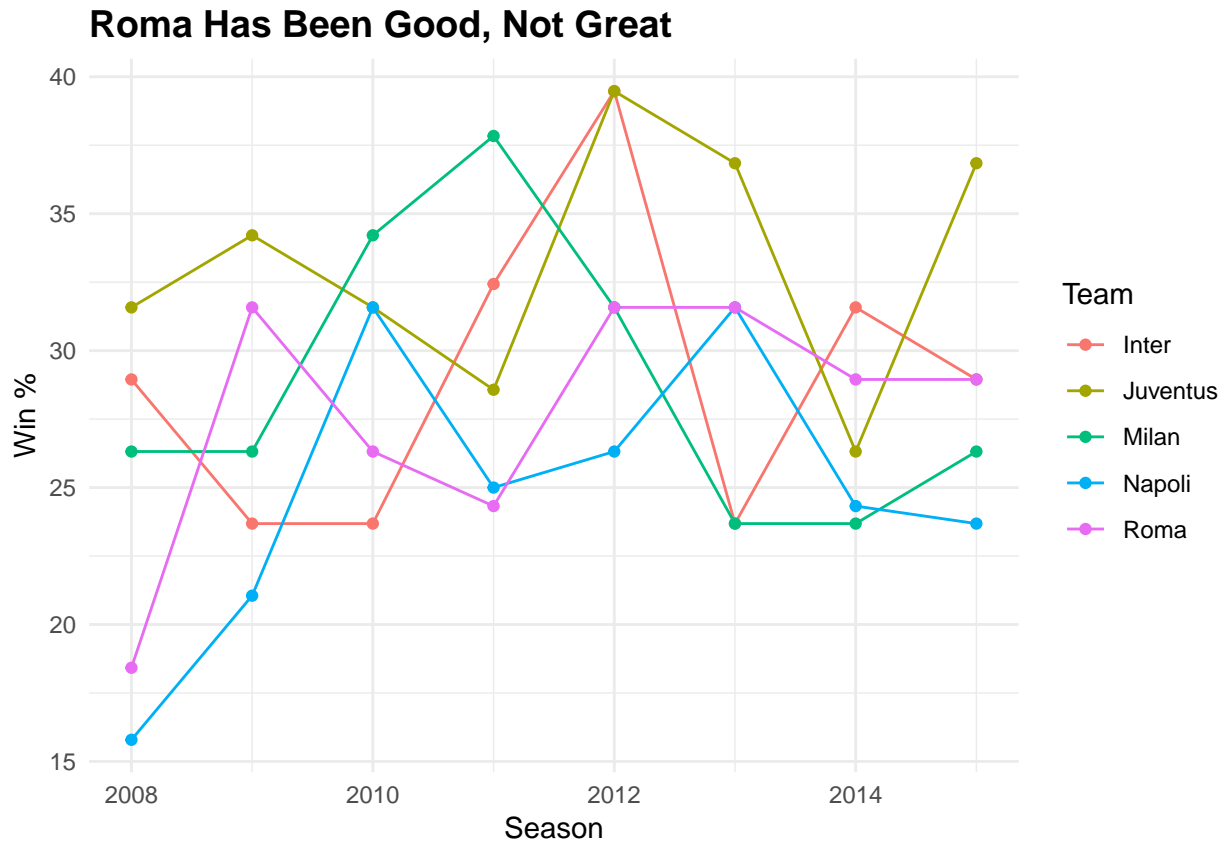
team_records <- merge(team_records, isa_team_names, by = 'team_api_id')
best_teams <- c('Juventus', 'Napoli', 'Milan', 'Roma', 'Inter')

team_records <- team_records %>%
  select(team_long_name, season:tie) %>%
  rename(Team = team_long_name) %>%
  filter(Team %in% best_teams)

team_records <- team_records %>%
  group_by(season, Team) %>%
  summarise(wins = sum(win),
            losses = sum(loss),
            ties = sum(tie))

team_records <- team_records %>%
  mutate(total_games = wins + losses + ties) %>%
  mutate(win_percentage = (wins / total_games) * 100)

team_records$season <- as.numeric(team_records$season)
ggplot(team_records, aes(x = season, y = win_percentage, color = Team)) +
  geom_line() + geom_point() +
  theme_minimal() + theme(plot.title = element_text(size = 14, face = 'bold')) +
  labs(y = 'Win %', x = 'Season', title = 'Roma Has Been Good, Not Great')
```



**Interpretation & Conclusions** Roma has been in the middle of the pack amongst the top five ISA teams from 2008 - 2015, placing third or tied for third the past 4 years. Juventus has enjoyed first place for a majority of the seasons. Inter has come on strong the past 2 years and even competed with Juventus at the height of Juventus' run in 2012. Milan may be capable of rising quickly in the coming years. Napoli is not as frightening as the other top teams. But which team is causing the biggest headache for Roma? The team hypothesized that Roma has a negative mean goal differential against one or more of these teams which is restricting Roma from becoming a front-runner in the ISA.

### 0.2.3 How does Roma fare against the top teams in the ISA?

The team wanted to identify the team or teams that Roma historically has a disadvantage against.

#### Execution and Results:

```
isa_best_teams_info <- team_tbl %>%
  filter(team_long_name %in% c('Inter', 'Juventus', 'Milan', 'Napoli', 'Roma'))
```

```
Italy_Serie_A_matches <- match_tbl %>%
  filter(league_id == 10257) %>%
  select(season, date, match_api_id:away_team_goal)
```

```
isa_best_team_matches <- Italy_Serie_A_matches %>%
  filter(home_team_api_id %in% isa_best_teams_info$team_api_id) %>%
  filter(away_team_api_id %in% isa_best_teams_info$team_api_id)
```

```
roma_facing_best_teams <- isa_best_team_matches %>%
  filter(home_team_api_id == 8686 | away_team_api_id == 8686) %>%
  mutate(home_team = ifelse(home_team_api_id == 9885, 'Juventus',
    ifelse(home_team_api_id == 8564, 'Milan',
```

```

        ifelse(home_team_api_id == 8686, 'Roma',
        ifelse(home_team_api_id == 9875, 'Napoli',
        ifelse(home_team_api_id == 8636, 'Inter', NA)))))) %>%
mutate(away_team = ifelse(away_team_api_id == 9885, 'Juventus',
        ifelse(away_team_api_id == 8564, 'Milan',
        ifelse(away_team_api_id == 8686, 'Roma',
        ifelse(away_team_api_id == 9875, 'Napoli',
        ifelse(away_team_api_id == 8636, 'Inter', NA))))))

roma_facing_best_teams['Opponent'] <- ifelse(roma_facing_best_teams$home_team != 'Roma',
      roma_facing_best_teams$home_team,
      ifelse(roma_facing_best_teams$away_team != 'Roma',
      roma_facing_best_teams$away_team, NA))

roma_facing_best_teams['goals'] <- ifelse(roma_facing_best_teams$home_team == 'Roma',
      roma_facing_best_teams$home_team_goal,
      ifelse(roma_facing_best_teams$away_team == 'Roma',
      roma_facing_best_teams$away_team_goal, NA))

roma_facing_best_teams['opponent_goals'] <- ifelse(roma_facing_best_teams$home_team != 'Roma', roma_fac

roma_facing_best_teams['roma_outcome'] <- ifelse(roma_facing_best_teams$goals -
      roma_facing_best_teams$opponent_goals > 0, 'Win',
      ifelse(roma_facing_best_teams$goals -
      roma_facing_best_teams$opponent_goals < 0, 'Loss', 'Tie'))

roma_facing_best_teams_clean <- roma_facing_best_teams %>%
  select(roma_outcome, Opponent, goals, opponent_goals, season, date) %>%
  mutate(win = ifelse(roma_outcome == 'Win', 1, 0),
    loss = ifelse(roma_outcome == 'Loss', 1, 0),
    tie = ifelse(roma_outcome == 'Tie', 1, 0))

roma_facing_best_teams_clean['goal_differential'] <- roma_facing_best_teams_clean$goals - roma_facing_b

roma_record_df <- roma_facing_best_teams_clean %>%
  group_by(Opponent) %>%
  summarize(wins = sum(win),
    losses = sum(loss),
    ties = sum(tie))

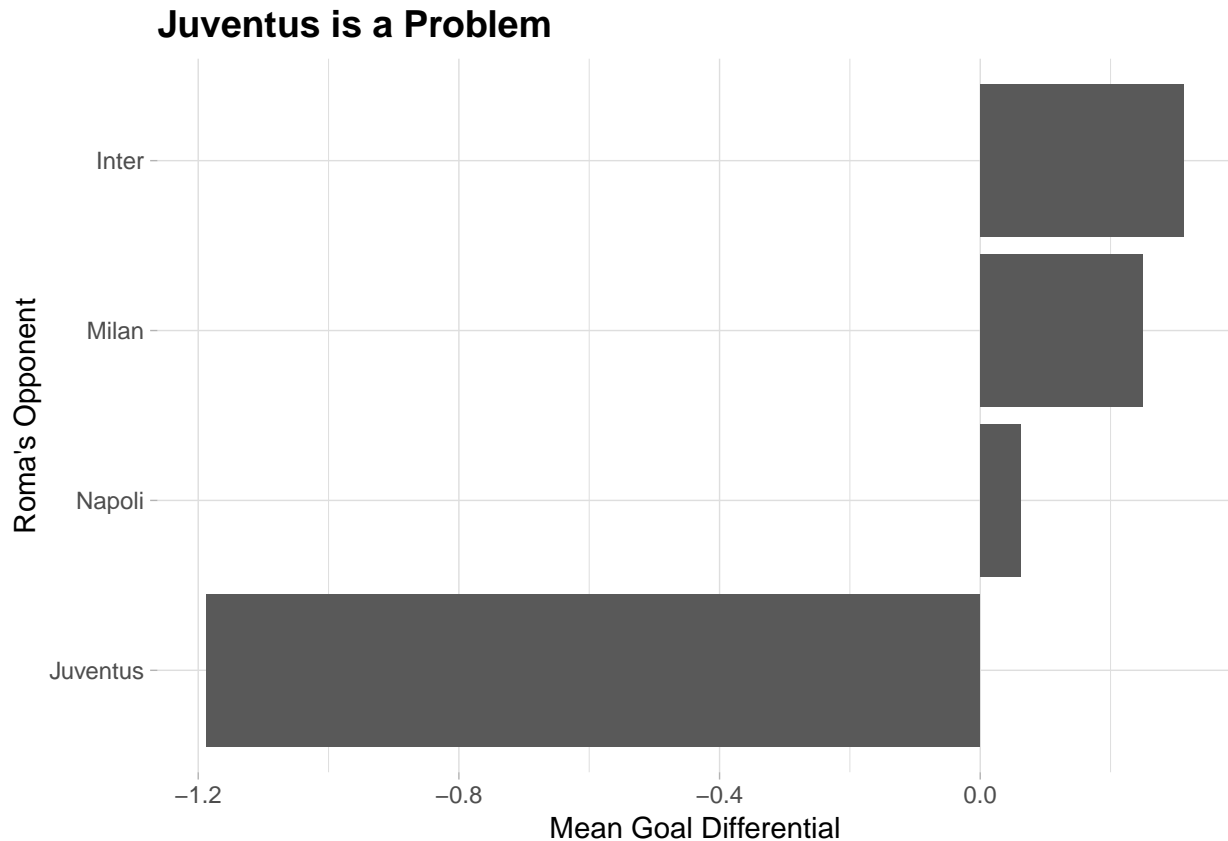
roma_goal_dif <- roma_facing_best_teams_clean %>%
  group_by(Opponent) %>%
  summarize(goal_differential = mean(goal_differential))

roma_goal_dif$Opponent <- as.factor(roma_goal_dif$Opponent)
roma_goal_dif$Opponent <- factor(roma_goal_dif$Opponent, levels = roma_goal_dif$Opponent[order(roma_goa

ggplot(roma_goal_dif, aes(x = Opponent, y = goal_differential)) +
  geom_col() + coord_flip() +
  labs(title = 'Juventus is a Problem',
    x = "Roma's Opponent",
    y = 'Mean Goal Differential') +
  theme_light() + theme(plot.title = element_text(size = 14, face = 'bold'),
    panel.border = element_blank())

```





**Interpretation & Conclusions** Juventus is the clear issue for Roma out of the top teams in the ISA. Thus, it is reasonable to explore potential strategies that could help the coach beat Juventus in the coming seasons. Before proceeding, we performed a mean goal differential analysis for Roma against all of the teams in the ISA to see if there are any mediocre or poor teams that have been an issue for Roma as well. Upon visualization, we did find another team that Roma had a negative mean goal differential against, Brescia. However, Brescia left the ISA in 2011. Thus, we moved forward in analyzing how Roma can beat Juventus.

### 0.3 How can Roma beat Juventus?

The team broke this question down into two paths of analysis. First, was to look at opposing team strategies that historically caused problems or created ease for Juventus. The team ran a dataframe containing all Juventus matches with the associated outcome for Juventus as the RHS and the opposing team's strategies as the LHS of the Apriori algorithm. The most meaningful rule found was that Roma tends to set defenceAggression to 'Press', which typically leads to a win for Juventus. This rule only had a lift of 1.1 so the team opted for the second path of analysis.

The second approach leveraged the in-game information locked away in XML format in the matches table. After parsing this data and subsetting for only Juventus matches, the team was able to visualize a pattern which indicates a potential weak spot in Juventus' defense.

Assumptions: 1. It is meaningful to measure the strength of Juventus' defense by the frequency at which goals are scored against Juventus throughout the game.

#### 0.3.1 Are there stages throughout the match where Juventus' defense is weak?

Juventus has dominated the ISA for nearly a decade. The team hypothesized that Juventus is not perfect and that there are chinks in Juventus' armour that Roma can exploit.

To get started the team extracted the in-game events from the matches table, specifically information about

when goals were scored by the home and away teams. The team then subsetted the matches to only include Juventus matches and derived an outcome column to capture whether Juventus won, lost, or tied.

Justifying the hypothesis, the team found specific time intervals in Juventus' matches where if the opponent scored, their chances to beat Juventus increased substantially. This trend was found by visualizing the frequency in which goals were scored against Juventus against the outcome of the game. This trend was then validated by the resultant association rules computed by the Apriori algorithm.

## Execution and Results:

*Citation: The following code chunk was sourced from kaggle to extract XML data from the match table:*

Njitram. "Data Munging the Incident Data." Kaggle, Kaggle, 5 Sept. 2016, link

```
match <- match_tbl
value_from_xpath <- function(element, xpath, to.int = F, index = 1) {
  xml_find_all(element, xpath) %>%
    {ifelse(length(.), xml_text(.[[index]]), NA)} %>%
    {ifelse(to.int, as.integer(.), .)}
}

incidents <- map_df(list('goal', 'card'), function(key) {
  match %>%
    filter_(paste0('!is.na(', key, ')')) %>%
    select_('id', key) %>%
    collect() %>%
    rename_('value' = key) %>%
    pmap_df(function(id, value) {
      xml <- read_xml(value)
      df <- xml %>% xml_find_all(paste0('/', key, '/value')) %>%
        map_df(function(n) {
          tibble_(list(
            id = ~value_from_xpath(n, './id', to.int = T),
            type = ~value_from_xpath(n, './type'),
            subtype1 = ~value_from_xpath(n, './subtype'),
            subtype2 = ~value_from_xpath(n, paste0('./', key, '_type')),
            player1 = ~value_from_xpath(n, './player1'),
            player2 = ~value_from_xpath(n, './player2'),
            team = ~value_from_xpath(n, './team'),
            lon = ~value_from_xpath(n, './coordinates/value',
                                   to.int = T, index = 1),
            lat = ~value_from_xpath(n, './coordinates/value',
                                   to.int = T, index = 2),
            elapsed = ~value_from_xpath(n, './elapsed', T),
            elapsed_plus = ~value_from_xpath(n, './elapsed_plus', T)
          ))
        })
      if (nrow(df) > 0) {
        if (length(xml_find_all(xml, paste0('/',
                                              key, '/value/', key, '_type')) > 0) {
          df %>%
            rename(tmp = subtype1) %>%
            rename(subtype1 = subtype2) %>%
            rename(subtype2 = tmp)
        }
      }
      df$game_id <- id
    })
})
```

```

    }
    return(df)
  })
})

incidents$type <- factor(incidents$type)
incidents$subtype1 <- factor(incidents$subtype1)
incidents$subtype2 <- factor(incidents$subtype2)
overtime <- incidents[incidents$elapsed > 90,]$elapsed - 90
incidents[incidents$elapsed > 90,]$elapsed_plus <- overtime
incidents[incidents$elapsed > 90,]$elapsed <- 90
incidents <- incidents %>%
  mutate(half = (elapsed %/% 46) + 1)
incidents <- incidents %>%
  mutate(half_elapsed = ifelse(half == 1, elapsed, elapsed - 45))
incidents <- incidents %>%
  mutate(elapsed_plus = ifelse(is.na(elapsed_plus), 0, elapsed_plus))
incidents <- incidents %>%
  mutate(half_elapsed = elapsed + elapsed_plus)

game_temp_home <- match %>% filter(home_team_api_id == 9885) %>%
  mutate(result = home_team_goal - away_team_goal) %>%
  select(id, home_team_api_id, away_team_api_id, home_team_goal,
         away_team_goal, result)
game_temp_away <- match %>% filter(away_team_api_id == 9885) %>%
  mutate(result = away_team_goal - home_team_goal) %>%
  select(id, home_team_api_id, away_team_api_id, home_team_goal,
         away_team_goal, result)
game_temp <- rbind(game_temp_home, game_temp_away)
game_temp$result_chr <- NA
game_temp$result_chr[which(game_temp$result == 0)] <- "draw"
game_temp$result_chr[which(game_temp$result < 0)] <- "loss"
game_temp$result_chr[which(game_temp$result > 0)] <- "win"

goal_ju_oppo <- incidents %>%
  filter(type == 'goal' & game_id %in% game_temp[, 'id'] & team != 9885)
data_plot <- merge(goal_ju_oppo, game_temp[c("id", "result_chr")],
                  by.x = "game_id", by.y = "id")

result_chr.labs <- c("Draw", "Juventus Loss", "Juventus Win")
names(result_chr.labs) <- c("draw", "loss", "win")
half.labs <- c("First Half", "Second Half")
names(half.labs) <- c(1, 2)

data_plot2 <- data_plot %>%
  group_by(half_elapsed, half, result_chr) %>%
  summarise( num_goal = n()) %>%
  mutate(goal_per_match = case_when(result_chr == "win" ~ num_goal/189,

ggplot(data_plot2, aes(x=half_elapsed, y = goal_per_match)) +
  geom_line() + facet_grid(half ~ result_chr, scales = "free_x",
    labeller = labeller(result_chr = result_chr.labs, half = half.labs)) +
  theme_minimal() + theme(plot.title = element_text(size = 14, face = 'bold'),

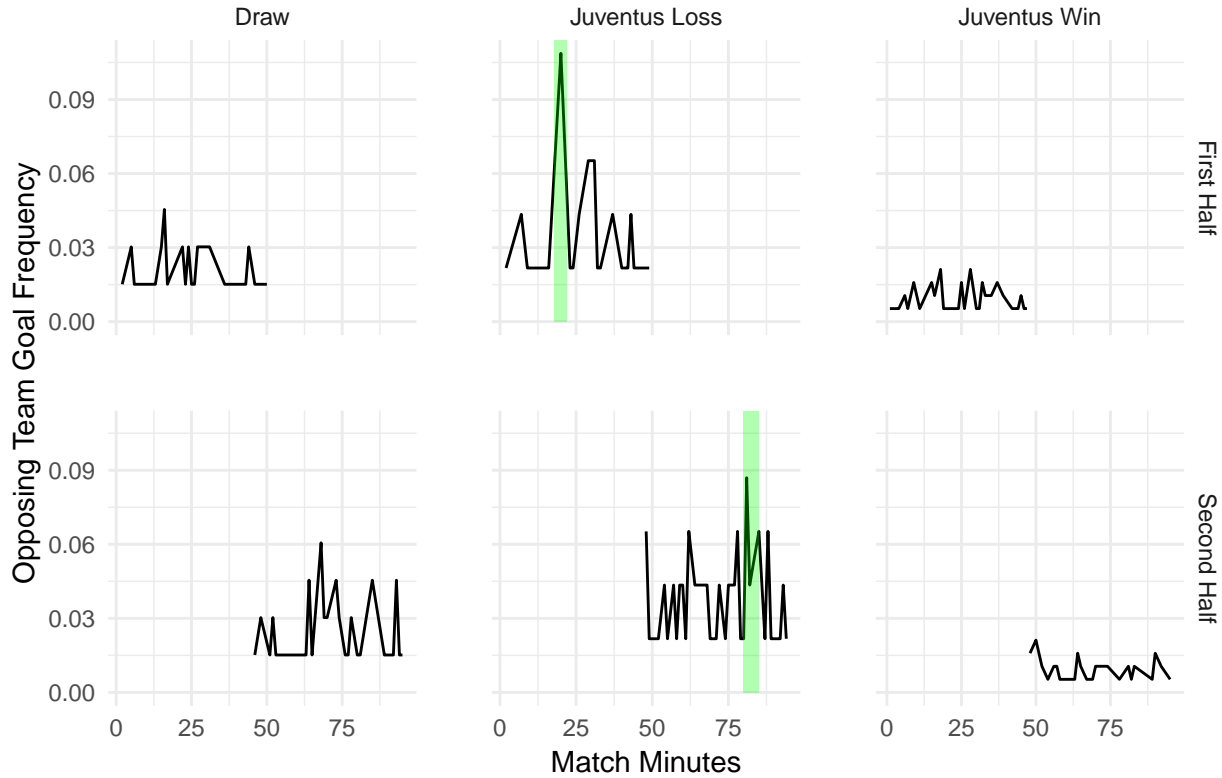
```

```

        panel.spacing = unit(1, 'cm')) +
labs(title = "Juventus' Defense is Weak at Certain Stages of The Match",
     x = "Match Minutes", y = "Opposing Team Goal Frequency") +
geom_rect(data = data.frame(result_chr = "loss", half = 1),
        aes(xmin = 18, xmax = 22, ymin = 0, ymax = Inf),
        alpha = 0.3, fill="green", inherit.aes = FALSE) +
geom_rect(data = data.frame(result_chr = "loss", half = 2),
        aes(xmin = 80, xmax = 85, ymin = 0, ymax = Inf),
        alpha = 0.3, fill="green", inherit.aes = FALSE)

```

## Juventus' Defense is Weak at Certain Stages of The Match



**Interpretation & Conclusions** Figure 5 shows the frequency at which goals are scored during each minute of a match for all of the games Juventus won, lost, and tied. Most notably, when Juventus' opponents scored during the 18-22 and 81-85 minute marks of the match, Juventus was unable to recover. As expected, the frequency of goals scored against Juventus was higher throughout the matches they lost when compared against the frequency of goals scored against them when they won, but the frequency was especially high at these two time intervals of the match. Thus, the team concluded that to beat Juventus, Roma should creatively attack Juventus during these time intervals. The team conjectured that this pattern is either due to Juventus' defense being weakest at these times or that a goal at these times is a severe blow to Juventus' morale.

### 0.3.2 Can we validate Juventus' defensive weaknesses using association rules?

It seems that there are pockets of opportunity where opposing teams can score against Juventus and consequentially win. In order to verify this hypothesis, the team used the Apriori algorithm to analyze the lift, confidence, and support of the timing intervals when goals were scored against Juventus as the LHS and the outcome of the match for Juventus as the RHS. All matches were bucketed into the following time intervals. First half is bucketed as so: before 18 mins, 18-22 mins and after 22 mins. Similarly, the second

half was bucketed as before 81 mins, 81-85 mins and after 85 mins. Each match was treated as a transaction. The team then ran the algorithm and assessed whether the 18-22 and 81-85 minute buckets resulted in a Juventus loss above and beyond random chance. This allowed us to verify the hypothesis that Juventus can be beaten, if attacked at certain times in the match.

## Execution and Results:

*Prep the dataset and create columns for opposing team id, goals scored by Juventus, goals scored by the opposition and the outcome of the match*

```
Italy_Serie_A_matches <- match_tbl %>%
  filter(league_id == 10257)

juventas_matches <- Italy_Serie_A_matches %>%
  select(-country_id, -league_id, -stage) %>%
  filter(home_team_api_id == 9885 | away_team_api_id == 9885)

juventas_matches['Opponent'] <- ifelse(juventas_matches$home_team_api_id != 9885,
  juventas_matches$home_team_api_id,
  ifelse(juventas_matches$away_team_api_id != 9885,
    juventas_matches$away_team_api_id, NA))

juventas_matches['goals'] <- ifelse(juventas_matches$home_team_api_id == 9885,
  juventas_matches$home_team_goal,
  ifelse(juventas_matches$away_team_api_id == 9885,
    juventas_matches$away_team_goal, NA))

juventas_matches['opponent_goals'] <- ifelse(juventas_matches$home_team_api_id != 9885,
  juventas_matches$home_team_goal,
  ifelse(juventas_matches$away_team_api_id != 9885,
    juventas_matches$away_team_goal, NA))

juventas_matches['juv_outcome'] <- ifelse(juventas_matches$goals -
  juventas_matches$opponent_goals > 0, 'Win',
  ifelse(juventas_matches$goals -
    juventas_matches$opponent_goals < 0, 'Loss', 'Tie'))

juventas_matches <- juventas_matches %>%
  select(juv_outcome, goals, opponent_goals, Opponent, id:match_api_id)

juv_opponents_apis <- unique(juventas_matches$Opponent)
juv_opponents_team_atts <- team_atts_tbl %>%
  filter(team_api_id %in% juv_opponents_apis)

juv_opponents_team_atts <- juv_opponents_team_atts %>%
  select(team_api_id, date, ends_with('Class'))

juv_opponents_team_atts$date <- substr(juv_opponents_team_atts$date, start = 1, stop = 4)
juventas_matches$season <- substr(juventas_matches$season, start = 1, stop = 4)
juv_opponents_team_atts <- juv_opponents_team_atts %>%
  rename(season = date)
juventas_matches <- juventas_matches %>%
  rename(team_api_id = Opponent)

juv_matches_with_team_atts <- merge(juventas_matches, juv_opponents_team_atts,
  by = c('team_api_id', 'season'))
```

```

get_team_incidents_table <- function(match, incidents, team_id){
  match_team_home <- subset(match, home_team_api_id == team_id, select=c("id","match_api_id","home_team_
  match_team_away <- subset(match, away_team_api_id== team_id, select=c("id","match_api_id","away_team_
  names(match_team_home) <- c("match_id","match_api_id",
    "main_team_api_id", "oppo_team_api_id")
  names(match_team_away) <- c("match_id","match_api_id",
    "main_team_api_id", "oppo_team_api_id")
  match_team_away <- match_team_away[c("match_id","match_api_id",
    "main_team_api_id", "oppo_team_api_id")]
  match_team_all <- rbind(match_team_home, match_team_away)
  incidents <- incidents %>% mutate(timeperiod =
    case_when(elapsed<18 ~'FirstHalfStart', elapsed<=22 ~'FirstHalfMid',
      elapsed<=45 ~'FirstHalfEnd', elapsed<=80 ~'SecHalfStart',
      elapsed<=85 ~'SecHalfMid', elapsed<=90 ~'SecHalfEnd'))
  incidents$timeperiod[which(incidents$timeperiod == "FirstHalfLast5" &
    incidents$elapsed_plus > 0)] <- 'FirstHalfLast5_OT'
  incidents$timeperiod[which(incidents$timeperiod == "SecHalfLast5" &
    incidents$elapsed_plus > 0)] <- 'SecHalfLast5_OT'
  incidents_temp <- incidents %>%
    filter(game_id %in% match_team_all[, 'match_id']) %>%
    group_by(game_id,type,team,timeperiod) %>%
    summarise(count = n()) %>%
    unite(key,type,timeperiod,sep = '_') %>%
    spread(key, count)

  match_team_all_1 <- merge(match_team_all, incidents_temp,
    by.x=c("match_id","main_team_api_id"),
    by.y = c("game_id","team"))
  colnames(match_team_all_1)[5:16] <- paste(colnames(match_team_all_1[,c(5:16)]),
    "MainTeam",sep = "_")
  match_team_all_2 <- merge(match_team_all_1, incidents_temp,
    by.x=c("match_id","oppo_team_api_id"),
    by.y = c("game_id","team"))
  colnames(match_team_all_2)[17:28] <- paste(colnames(match_team_all_2[,c(17:28)]),
    "OppoTeam",sep = "_")

  return(match_team_all_2)
}

incidents_ju <- get_team_incidents_table(match, incidents, team_id = 9885)

```

*Include the goal timing information into the matches outcome data and select the relevant columns for analysis*

```

juv_matches_with_team_atts <- merge(juv_matches_with_team_atts, incidents_ju,
  by.x='id', by.y = 'match_id')

juv_arules <- juv_matches_with_team_atts %>%
  select(juv_outcome, goal_FirstHalfEnd_MainTeam : goal_SecHalfStart_MainTeam,
    goal_FirstHalfEnd_OppoTeam : goal_SecHalfStart_OppoTeam)

```

*Convert numerical data into categories and transform the dataset into transactions to prep for apriori algorithm*

```

juv_arules$juv_outcome <- as.factor(juv_arules$juv_outcome)
juv_arules$goal_FirstHalfEnd_MainTeam <- as.factor(juv_arules$goal_FirstHalfEnd_MainTeam)
juv_arules$goal_FirstHalfEnd_OppoTeam <- as.factor(juv_arules$goal_FirstHalfEnd_OppoTeam)
juv_arules$goal_FirstHalfMid_MainTeam <- as.factor(juv_arules$goal_FirstHalfMid_MainTeam)
juv_arules$goal_FirstHalfMid_OppoTeam <- as.factor(juv_arules$goal_FirstHalfMid_OppoTeam)
juv_arules$goal_FirstHalfStart_MainTeam <- as.factor(juv_arules$goal_FirstHalfStart_MainTeam)
juv_arules$goal_FirstHalfStart_OppoTeam <- as.factor(juv_arules$goal_FirstHalfStart_OppoTeam)
juv_arules$goal_SecHalfEnd_MainTeam <- as.factor(juv_arules$goal_SecHalfEnd_MainTeam)
juv_arules$goal_SecHalfEnd_OppoTeam <- as.factor(juv_arules$goal_SecHalfEnd_OppoTeam)
juv_arules$goal_SecHalfMid_MainTeam <- as.factor(juv_arules$goal_SecHalfMid_MainTeam)
juv_arules$goal_SecHalfMid_OppoTeam <- as.factor(juv_arules$goal_SecHalfMid_OppoTeam)
juv_arules$goal_SecHalfStart_MainTeam <- as.factor(juv_arules$goal_SecHalfStart_MainTeam)
juv_arules$goal_SecHalfStart_OppoTeam <- as.factor(juv_arules$goal_SecHalfStart_OppoTeam)

juv_trans = as(juv_arules, 'transactions')

```

Generate rules. Since Juventus does not lose often, the support and confidence was set to a low threshold. The goal was to check whether goals scored during the 18-22 minute mark (variable name = goal\_SecHalfMid\_OppoTeam) and 81-85 minute mark (variable name = goal\_FirstHalfMid\_OppoTeam) had a high lift with the outcome as a Juventus loss

```

juv_rules <- apriori(juv_trans, parameter = list(supp = 0.01, conf = 0.01))
juv_rules1 <- juv_rules %>%
  subset(subset = (lhs %pin% 'goal_SecHalfMid_OppoTeam' | lhs
                    %pin% 'goal_FirstHalfMid_OppoTeam')) %>%
  subset(subset = rhs %pin% 'Loss') %>%
  subset(subset = lift > 1.05)

juv_rules_df1 <- as(juv_rules1, 'data.frame') %>%
  arrange(-lift)

juv_rules_df <- as(juv_rules_df1, 'data.frame')
kable(juv_rules_df)

```

rules	support	confidence	lift	count
{goal_SecHalfMid_OppoTeam=1} => {juv_outcome=Loss}	0.0181818	0.3333333	3.055556	4
{goal_FirstHalfMid_OppoTeam=1} => {juv_outcome=Loss}	0.0136364	0.2307692	2.115385	3

**Interpretation & Conclusions** After ingesting the rules above, it can be concluded that when the opposing team scores a goal during the 18-22 and 81-85 minute marks of the match there is a considerable increase in the chances of Juventus losing the match. Thus, validating our hypothesis that there are pockets of opportunity where opposing teams can score against Juventus and consequentially win.

Looking beyond just in-game strategies on how to beat Juventus, the team explored patterns in ISA player attributes that above and beyond random chance led to success of their respective teams.

## 0.4 What player attributes are most indicative of success in the ISA?

The objective of this analysis is to provide a source of reference for the coach and subordinate scouts when searching for players. This resource will define the key attributes high performing players in the ISA on average have and compare this skillset to the skillset of Roma's current scouting practices. We hope this resource will serve as a guide throughout Roma's scouting efforts and help identify players that will have a substantial impact on Roma's performance when they join the team. Assumptions: 1. It does not matter

whether the team is home or away 2. The coach's acceptable level of error is 0.2 when conducting statistical tests

## Execution and Results:

*Grab only ISA matches & drop betting providers/detailed match events/player X, Y coordinates*

```
Italy_Serie_A_matches <- match_tbl %>%  
  filter(league_id == 10257) %>%  
  select(id:away_team_goal, home_player_1:away_player_11)
```

*Dropping irrelevant cols & creating goal differential*

```
Italy_Serie_A_matches <- Italy_Serie_A_matches %>%  
  select(-country_id, -league_id, -stage) %>%  
  mutate(goal_diff = home_team_goal - away_team_goal)
```

*Wrangle league table for association rules*

```
ISA_home_matches <- Italy_Serie_A_matches %>%  
  select(id:home_team_api_id, home_team_goal:home_player_11, goal_diff) %>%  
  mutate(outcome = ifelse(goal_diff > 0, 'won',  
                           ifelse(goal_diff < 0, 'loss', 'tie'))) %>%  
  rename(goals = home_team_goal, opponent_goals = away_team_goal) %>%  
  select(outcome, goal_diff, goals, opponent_goals,  
         id:home_team_api_id, home_player_1:home_player_11)  
  
ISA_away_matches <- Italy_Serie_A_matches %>%  
  select(id:match_api_id, away_team_api_id:away_team_goal, away_player_1:goal_diff) %>%  
  mutate(outcome = ifelse(goal_diff < 0, 'won',  
                           ifelse(goal_diff > 0, 'loss', 'tie'))) %>%  
  rename(goals = away_team_goal, opponent_goals = home_team_goal) %>%  
  select(outcome, goal_diff, goals, opponent_goals,  
         id:away_team_api_id, away_player_1:away_player_11)  
  
col_names = c('outcome', 'goal_diff', 'goals', 'opponent_goals',  
              'match_id', 'season', 'date', 'match_api_id',  
              'team_api_id', 'player_1', 'player_2', 'player_3',  
              'player_4', 'player_5', 'player_6', 'player_7',  
              'player_8', 'player_9', 'player_10', 'player_11')  
  
names ISA_home_matches <- col_names  
names ISA_away_matches <- col_names
```

```
ISA_matches <- rbind ISA_home_matches, ISA_away_matches)
```

*Grab player attributes*

```
# ISA player attributes  
ISA_players_vec <- unique(c(ISA_matches$player_1, ISA_matches$player_2,  
                           ISA_matches$player_3, ISA_matches$player_4,  
                           ISA_matches$player_5, ISA_matches$player_6,  
                           ISA_matches$player_7, ISA_matches$player_8,  
                           ISA_matches$player_9, ISA_matches$player_10,  
                           ISA_matches$player_11))  
  
ISA_players <- player_tbl %>%  
  filter(player_api_id %in% ISA_players_vec)
```



```
ISA_players <- merge(ISA_players, player_atts_tbl, by = 'player_api_id')
```

```
ISA_players <- ISA_players %>%  
  select(player_api_id, player_name, birthday, height,  
         weight, date, overall_rating:gk_reflexes) %>%  
  arrange(player_api_id, date)
```

```
isa_players_arules <- ISA_matches %>%  
  select(outcome, player_1:player_11)
```

*Check for missing values - lost 274 records*

```
miss_var_summary(isa_players_arules)
```

```
## # A tibble: 12 x 3  
##   variable  n_miss pct_miss  
##   <chr>      <int>   <dbl>  
## 1 player_11     89    1.47  
## 2 player_10     68    1.13  
## 3 player_9      30    0.497  
## 4 player_2      19    0.315  
## 5 player_7      19    0.315  
## 6 player_5      15    0.249  
## 7 player_8      13    0.215  
## 8 player_4      12    0.199  
## 9 player_3       7    0.116  
## 10 player_6      7    0.116  
## 11 outcome       0     0  
## 12 player_1       0     0
```

```
isa_players_arules <- isa_players_arules[complete.cases(isa_players_arules), ]
```

```
isa_players_arules$outcome <- as.factor(isa_players_arules$outcome)  
isa_players_arules$player_1 <- as.factor(isa_players_arules$player_1)  
isa_players_arules$player_2 <- as.factor(isa_players_arules$player_2)  
isa_players_arules$player_3 <- as.factor(isa_players_arules$player_3)  
isa_players_arules$player_4 <- as.factor(isa_players_arules$player_4)  
isa_players_arules$player_5 <- as.factor(isa_players_arules$player_5)  
isa_players_arules$player_6 <- as.factor(isa_players_arules$player_6)  
isa_players_arules$player_7 <- as.factor(isa_players_arules$player_7)  
isa_players_arules$player_8 <- as.factor(isa_players_arules$player_8)  
isa_players_arules$player_9 <- as.factor(isa_players_arules$player_9)  
isa_players_arules$player_10 <- as.factor(isa_players_arules$player_10)  
isa_players_arules$player_11 <- as.factor(isa_players_arules$player_11)
```

```
isa_players_trans = as(isa_players_arules, 'transactions')
```

*Implement Apriori. Fix rhs to outcome of match and set max length of itemset to 2 because we only want to look at the impact of each specific player in the ISA. Tune support and confidence to get a digestable number of results that are meaningful*

```
isa_player_rules <- apriori(isa_players_trans,  
                           parameter = list(supp = 0.005, conf = 0.5, maxlen = 2))
```

```
outcome_isa_players_rules <- isa_player_rules %>%  
  subset(subset = rhs %pin% 'outcome') %>%  
  subset(subset = lift > 1.8)
```

```
outcome_isa_players_rules <- sort(outcome_isa_players_rules, by = 'lift')
inspect(outcome_isa_players_rules)
```

```
##      lhs                rhs      support    confidence lift
## [1] {player_9=111862} => {outcome=won} 0.008333333 0.8000000 2.146251
## [2] {player_2=30902}  => {outcome=won} 0.011458333 0.7674419 2.058903
## [3] {player_2=96616}  => {outcome=won} 0.005034722 0.7250000 1.945040
## [4] {player_8=248453} => {outcome=won} 0.009548611 0.7236842 1.941510
## [5] {player_6=49939}  => {outcome=won} 0.010416667 0.7228916 1.939383
## [6] {player_5=30474}  => {outcome=won} 0.011979167 0.7187500 1.928272
## [7] {player_10=35724} => {outcome=won} 0.005902778 0.6938776 1.861544
## [8] {player_7=41890}  => {outcome=won} 0.006597222 0.6909091 1.853580
## [9] {player_3=24235}  => {outcome=won} 0.019097222 0.6790123 1.821663
## [10] {player_7=30731} => {outcome=won} 0.018576389 0.6772152 1.816842
## [11] {player_1=181920} => {outcome=loss} 0.005381944 0.6595745 1.811707
##      count
## [1]    48
## [2]    66
## [3]    29
## [4]    55
## [5]    60
## [6]    69
## [7]    34
## [8]    38
## [9]   110
## [10]  107
## [11]   31
```

Create dataframe of best players in the ISA based on rules generated by apriori

```
best_isa_players_df <- as(outcome_isa_players_rules, 'data.frame')
best_isa_players_df['player_api_id'] = c(111862, 30902, 96616, 248453,
                                         49939, 30474, 35724, 41890, 24235, 30731, 181920)

best_isa_players_df <- best_isa_players_df %>%
  select(player_api_id, support:count)

best_isa_players_atts <- ISA_players %>%
  filter(player_api_id %in% best_isa_players_df$player_api_id)
```

Examine and drop missing values

```
miss_var_summary(best_isa_players_atts)
```

```
## # A tibble: 44 x 3
##   variable      n_miss pct_miss
##   <chr>        <int>    <dbl>
## 1 overall_rating      1    0.220
## 2 potential           1    0.220
## 3 preferred_foot      1    0.220
## 4 attacking_work_rate  1    0.220
## 5 defensive_work_rate  1    0.220
## 6 crossing            1    0.220
## 7 finishing           1    0.220
## 8 heading_accuracy     1    0.220
```

```
## 9 short_passing          1    0.220
## 10 volleys               1    0.220
## # ... with 34 more rows
```

```
best_isa_players <- best_isa_players_atts[complete.cases(best_isa_players_atts), ]
```

*Transform data into a feature vector of group means of the best player attributes*

```
best_isa_players_bg <- best_isa_players %>% # had to drop attacking & defensive work rate cols
  select(player_api_id:weight, preferred_foot:defensive_work_rate) %>%
  group_by(player_api_id) %>%
  summarise(player_name = first(player_name), bday = min(birthday),
            preferred_foot = first(preferred_foot),
            height = mean(height), weight = mean(weight))
```

```
best_isa_players_avgs <- best_isa_players %>%
  select(player_api_id, overall_rating, potential, crossing:sliding_tackle) %>%
  group_by(player_api_id) %>%
  summarise_all(mean)
```

```
best_isa_players_grouped <- merge(best_isa_players_bg, best_isa_players_avgs,
                                by = 'player_api_id')
```

```
best_skills_avgs <- best_isa_players_grouped %>%
  summarise_all(mean) %>%
  select(overall_rating:sliding_tackle)
```

*Create another feature vector of group means but this time for all ISA players*

```
miss_var_summary(ISA_players)
```

```
## # A tibble: 44 x 3
##   variable      n_miss pct_miss
##   <chr>         <int>   <dbl>
## 1 attacking_work_rate    382    1.18
## 2 volleys               330    1.02
## 3 curve                 330    1.02
## 4 agility               330    1.02
## 5 balance               330    1.02
## 6 jumping               330    1.02
## 7 vision                330    1.02
## 8 sliding_tackle        330    1.02
## 9 overall_rating        109    0.338
## 10 potential            109    0.338
## # ... with 34 more rows
```

```
isa_players <- ISA_players[complete.cases(ISA_players), ]
```

```
isa_players_bg <- isa_players %>%
  select(player_api_id:weight, preferred_foot:defensive_work_rate) %>%
  group_by(player_api_id) %>%
  summarise(player_name = first(player_name), bday = min(birthday),
            preferred_foot = first(preferred_foot),
            height = mean(height), weight = mean(weight))
```

```
isa_players_avgs <- isa_players %>%
  select(player_api_id, overall_rating, potential, crossing:sliding_tackle) %>%
  group_by(player_api_id) %>%
```

```

summarise_all(mean)

isa_players_grouped <- merge(isa_players_bg, isa_players_avgs, by = 'player_api_id')

total_skills_avgs <- isa_players_grouped %>%
  summarise_all(mean) %>%
  select(overall_rating:sliding_tackle)

```

*Combine feature vectors and prep for plotting*

```

skills_difs <- best_skills_avgs - total_skills_avgs
skills_difs <- skills_difs %>%
  gather(key = 'attributes', value = 'rating') %>%
  arrange(desc(rating))

skills_difs$attributes <- factor(skills_difs$attributes,
  levels = skills_difs$attributes[order(-skills_difs$rating)])

```

*Perform t-tests on attributes to identify discrepancies between best players and all ISA players*

```

t.test(isa_players_avgs$sliding_tackle, best_isa_players_avgs$sliding_tackle)
t.test(isa_players_avgs$standing_tackle, best_isa_players_avgs$standing_tackle)
t.test(isa_players_avgs$interceptions, best_isa_players_avgs$interceptions)
t.test(isa_players_avgs$marking, best_isa_players_avgs$marking)
t.test(isa_players_avgs$long_passing, best_isa_players_avgs$long_passing)
t.test(isa_players_avgs$vision, best_isa_players_avgs$vision)
t.test(isa_players_avgs$volleys, best_isa_players_avgs$volleys)
t.test(isa_players_avgs$reactions, best_isa_players_avgs$reactions)
t.test(isa_players_avgs$curve, best_isa_players_avgs$curve)
t.test(isa_players_avgs$short_passing, best_isa_players_avgs$short_passing)
t.test(isa_players_avgs$long_shots, best_isa_players_avgs$long_shots)
t.test(isa_players_avgs$aggression, best_isa_players_avgs$aggression)
t.test(isa_players_avgs$overall_rating, best_isa_players_avgs$overall_rating)
t.test(isa_players_avgs$ball_control, best_isa_players_avgs$ball_control)
t.test(isa_players_avgs$dribbling, best_isa_players_avgs$dribbling)
t.test(isa_players_avgs$free_kick_accuracy, best_isa_players_avgs$free_kick_accuracy)
t.test(isa_players_avgs$potential, best_isa_players_avgs$potential)
t.test(isa_players_avgs$stamina, best_isa_players_avgs$stamina)
t.test(isa_players_avgs$penalties, best_isa_players_avgs$penalties)
t.test(isa_players_avgs$crossing, best_isa_players_avgs$crossing)
t.test(isa_players_avgs$shot_power, best_isa_players_avgs$shot_power)
t.test(isa_players_avgs$strength, best_isa_players_avgs$strength)
t.test(isa_players_avgs$jumping, best_isa_players_avgs$jumping)
t.test(isa_players_avgs$heading_accuracy, best_isa_players_avgs$heading_accuracy)
# not significant
t.test(isa_players_avgs$agility, best_isa_players_avgs$agility)
t.test(isa_players_avgs$finishing, best_isa_players_avgs$finishing)
# not significant
t.test(isa_players_avgs$positioning, best_isa_players_avgs$positioning)
# not significant
t.test(isa_players_avgs$sprint_speed, best_isa_players_avgs$sprint_speed)
# not significant
t.test(isa_players_avgs$acceleration, best_isa_players_avgs$acceleration)
# not significant
t.test(isa_players_avgs$balance, best_isa_players_avgs$balance)

```

```

# not significant

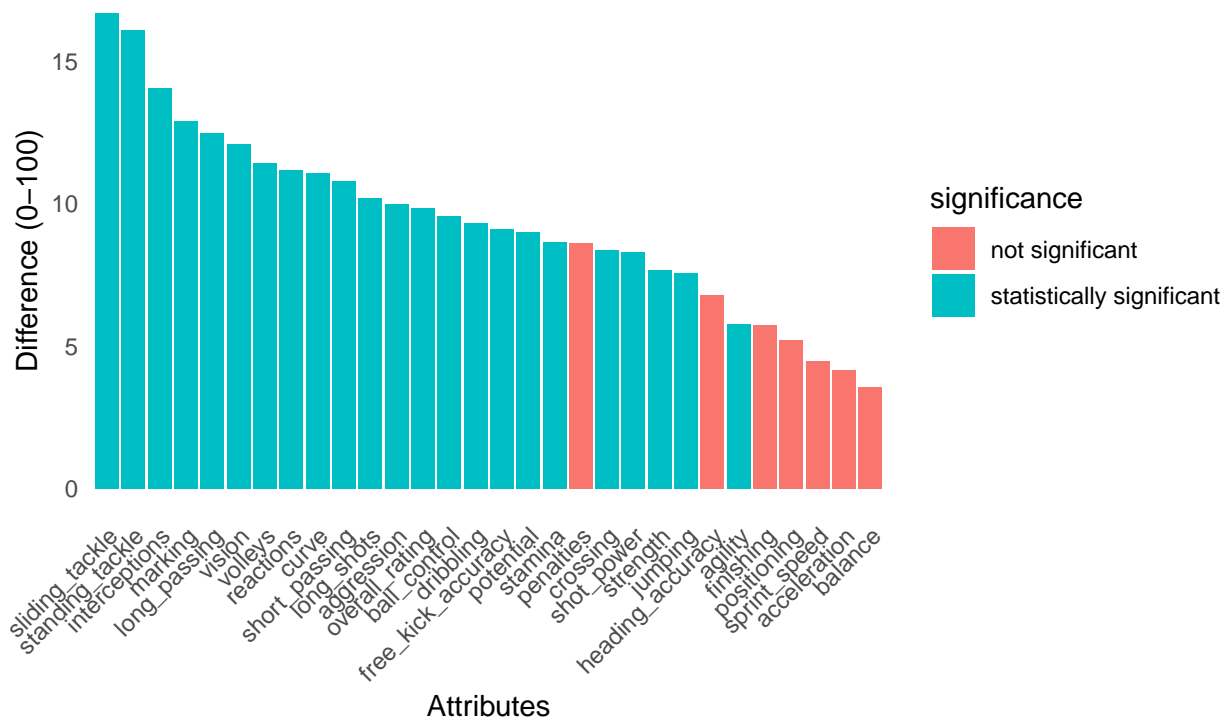
skills_difs['significance'] = c('statistically significant', 'statistically significant',
                                'statistically significant', 'statistically significant',
                                'statistically significant', 'statistically significant',
                                'statistically significant', 'statistically significant',
                                'statistically significant', 'statistically significant',
                                'statistically significant', 'statistically significant',
                                'statistically significant', 'statistically significant',
                                'not significant', 'statistically significant',
                                'statistically significant', 'statistically significant',
                                'statistically significant', 'not significant',
                                'statistically significant', 'not significant',
                                'not significant', 'not significant',
                                'not significant', 'not significant' )

ggplot(skills_difs, aes(x = attributes, y = rating, fill = significance)) +
  geom_col() + theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid = element_blank(),
        axis.ticks.x = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(size = 18, face = 'bold'),
        plot.subtitle = element_text(size = 12)) +
  labs(title = 'The Uncommon Skills Best Players Have',
       subtitle = 'best ISA players avg ratings - all ISA players avg ratings',
       y = 'Difference (0-100)',
       x = 'Attributes')

```

# The Uncommon Skills Best Players Have

best ISA players avg ratings – all ISA players avg ratings



**Interpretation & Conclusions** There are certain characteristics that are in fact indicative of wins in the ISA. To make this insight actionable we compared the best player skillset to Roma's player skillset, hypothesizing that there would be similar discrepancies as shown in the graph above. If so, these discrepancies could help inform scouting decisions going forward.

## 0.4.1 How does Roma's player skillset compare to the skillset of the best players in the ISA?

If the hypothesis is true - Roma's skillset is not aligned with the best player skillset - then this would shed light on where Roma needs to adjust their scouting practices.

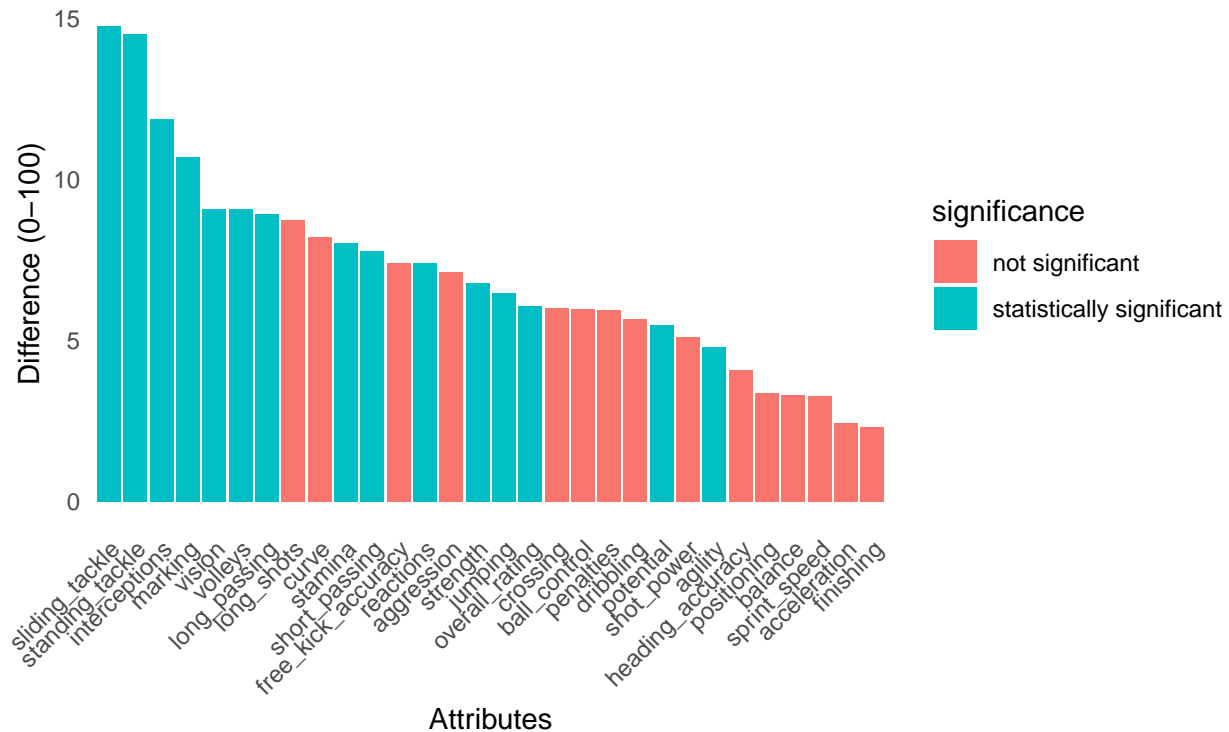
### Execution and Results:

The exact same steps were used to create the following visualization. Instead of comparing the best player skillset to all players in the ISA, we only compared the best player skillset against Roma's player skillset this time. Thus, the source code has not been included but is running in the background.

```
ggplot(roma_skills_difs, aes(x = attributes, y = rating, fill = significance)) +
  geom_col() + theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        panel.grid = element_blank(),
        axis.ticks.x = element_blank(),
        panel.background = element_blank(),
        plot.title = element_text(size = 18, face = 'bold'),
        plot.subtitle = element_text(size = 12)) +
  labs(title = 'The Most Important Attributes for Scouting',
       subtitle = 'Best ISA players avg ratings - Roma players avg ratings',
       y = 'Difference (0-100)',
       x = 'Attributes')
```

# The Most Important Attributes for Scouting

Best ISA players avg ratings – Roma players avg ratings



**Interpretation & Conclusions** The resultant discrepancies in attributes between Roma players and the best players that are statistically significant have been identified in blue. Roma’s coach and the subordinate scouts should adjust their scouting practices accordingly based upon these attributes.

## 0.5 Final Recommendations

The team has two recommendations for the coach of AS Roma:

1. When facing Juventus, save one of Roma’s best goal scoring players until the 18th minute, insert this player into the lineup and have him play until halftime. After halftime, reserve this player until the 80th minute of the match and play him until the end of the game. This strategy will exploit the proven weak points in Juventus’ defense that occur during the 18-22 and 81-85 minute marks of the match. Additionally, the coach should formulate a set play for these stages of the game.

It is important to note that the coach must determine whether the benefits of reserving a certain player’s stamina until the optimal times to score against Juventus outweigh the costs of not having him in for the entire game.

2. Roma should adjust their scouting practices according to the “The Most Important Attributes for Scouting” plot. Specifically, if an attribute is colored in blue, that attribute should be targeted during Roma’s scouting efforts. This will ensure Roma is accumulating talent that will translate into future success.

However, there is no guarantee that the best skillset historically will provide the optimal results in the future. There could be new strategy innovations by coaches or new skillsets brought into the league by young phenoms that shift the dynamics of gameplay. The coach must be cognizant of this and continue to innovate novel strategies.