**Introduction to the Problem**

We have limited ability to accurately forecast the outcomes of soccer matches. Paradoxically, the unpredictability of soccer matches makes the business of betting possible. Gamblers try their luck at predicting outcomes, and betting houses set odds that are profitable.

Fortunately, a lot of data is available for soccer matches, from goals and penalties per match to individual player attributes. Part of the problem will be to determine which of these data is significant, and which is irrelevant.

**Introduction to the Data; Fields**

The data originally comes as an SQLite database with eight tables, which were converted into eight separate data frames:

Tables

* country, league, match, player, playerattributes, teams, teamattributes, and sqlitesequence

We will begin by focusing on match, playerattributes, and teamattribues. The other data frames contain useful information, such as which teams are grouped together in leagues. However, they are small and require little analysis.

The focus data frames are larger and require some restructuring.

The most important of these data frames is match. It contains individual match information, such as number of goals scored, fouls committed, and which teams progressed in stages.

match

The variables in match can be divided into four categories:

(1) “Base Data”: The majority of our analysis will probably come from this data frame. It includes variables such as “home\_team\_goals” and “match\_api\_id.” Because of this, I created data frame that dropped all variables except for those in this category - 'matchMB'.

Variables include: country\_id, league\_id, season, stage, date, match\_api\_id, home\_team\_api\_id, away\_team\_api\_id, home\_team\_goal, and away\_team\_goal

These variables will allow us to create additional fields to describe whether a team won and how many points they received for the win/draw/loss. They will also allow to create a summary data frame that includes a team’s stats for the end of each stage. This summary data will be useful for plotting and for regression analysis.

(2) “Player Data”: This data shows the position on of first, second, and third string players. The variables are coordinates on the plane of a soccer field.

(3) “XML data”: These variables possess details of shots and gameplay. Because the number of shots and plays can vary from game to game, it is stored in nested data frames and is not tidy. We will need to determine which format it is written in, and then download and learn any necessary packages to extract the desired data.

Variables include: goal, shoton, shotoff, foulcommit, card, cross, corner, possession

(4) Betting house odds: These numbers show the odds placed on games by various betting houses. Because we want to find probabilities of wins and stage progression ourselves, we won’t be using these numbers in our analysis. Depending on where our analysis takes us, we may end up using these number for comparison.

playerattributes

The fields in player attributes put values on various players skill. This data frame contains 43 variables such as preferred\_foot, ball\_control, sprint\_speed, and overall\_rating. These attributes were pulled from EA Sports, and the data author does not give details where from EASports these data came from, or how EA Sports made the calculations.

teamattributes

The fields for team attributes put values on various team skills. This data frame contains 26 variables. Example include chanceCreationPassing, chanceCreationShooting, and defencePressure. These are also pulled from EA Sports. We do not know how the data were calculated, nor whether EA Sports is a primary or non-primary source.

Limitations

This is a relatively involved data set. It has 198 variables between 8 data frames, which does not include calculated fields or data within nested XML data frames. The nested XML data frames contain rich information such as ball possession by team, goal types, and players who scored.

While any data set’s limitations can be defined by the data it does not have, this data set is a case where the limitations are better ascribed to the analyst’s creativity and by data missing within the data set itself. For example, the match data frame has ~25,000 rows. However, only 8,125 of those rows contain complete XML nested data. The other rows are missing data on potentially important soccer events, such as crosses, corner kicks, and length of ball possession. Generally speaking, the earlier seasons in this data set do not contain the nested XML data. This limits how far back we can go in our analysis, which may limit seeing any higher level trends.

This data set also doesn’t contain information on league rules or other external events, such as league policy changes or weather during the soccer matches.

Cleaning and Wrangling

This data set is intended to thwart practitioners of R. It is stored in SQLite format, and many interesting variables are stored in nested XML data frames within the match table. This required using a combination of R packages - RSQLite, DBI, XML, dplyr, and magrittr.

The initial transformation involved reading the SQLite data into R and creating a variable for each table. This data was still in SQLite format. R plays nicer with csv files, so I proceeded to write each table to a csv file stored in my project folder. I then created a variable for each csv file and read it back into R. This allowed me to move forward with base R and more common R packages, such as dplyr and magrittr.

At this point, it was easy to select the non-XML data with dplyr to create various plots and simple regressions. Converting the XML data into a data frame, and then extracting the desired data required the following process:

1. Use dplyr to select the desired columns
2. library(XML)
3. Remove all rows with NA or incomplete data:

# removes all rows in matchAD1$possession with NA

test <- test[complete.cases(test$possession),]

# remove all rows where test$possession contains only "<possession />"

test <- test[!(test$possession == "<possession />"),]

# removeall rows where test$possession contains only "<possession />"

test <- test[!(test$card == "<card />"),] # removes 40 more rows

1. Create a for loop that extracts the necessary data into a new variable. The for loop contains elements of the dplyr package and the XML package.
   1. All XML columns did not convert into a clean data frame. Many data values were in the wrong column. I mitigated this within the for loop for all variables except “possession.”
   2. The “possession” variable also occasionally had incorrect values. Also, the output of the for loop contained excess characters that needed to be removed. I mitigated both of these issues outside of the for loop using substr() and magritrr().

These for loops took about 12 hours to run.

Note: In the early stages, I also relied heavily on plyr while wrangling the XML data, because much of the helpful online documentation provided plyr examples. I later found and implemented dplyr equivalents to prevent running both packages simultaneously.

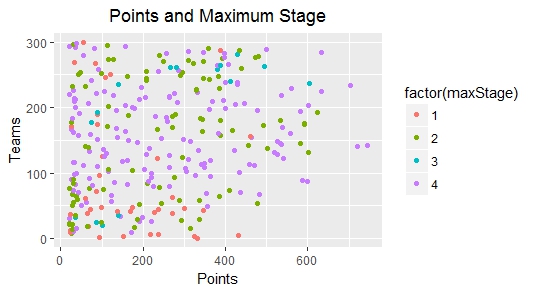
Preliminary exploration

Before I unpacked the XML data, I created various plots to see if teams with more points or goals proceeded to further stages.

ggplot(data = statsMB3, aes(spoints, statID, col = factor(maxStage))) + geom\_point() +

labs(title = "Points and Maximum Stage") + theme(plot.title = element\_text(hjust = 0.5)) +

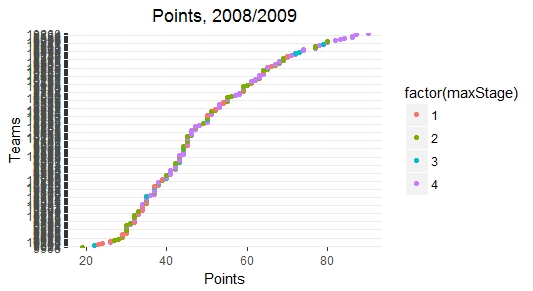
labs(x = "Points", y = "Teams")



This plot suggests a very loose association between points accumulated and stages progressed.

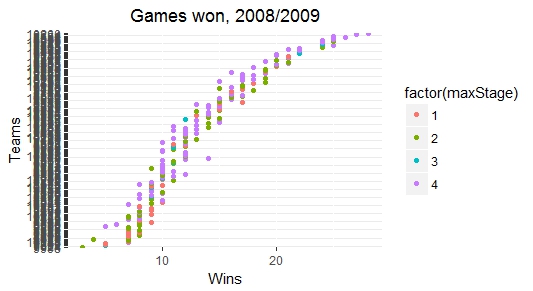
I also made plots with the following changes: (1) plot goal differential instead of points (2) plot single seasons to reduce clutter (3) create a step-wise plot.

Here is an example of teams and stages ordered by points during the 2008/2009 season:

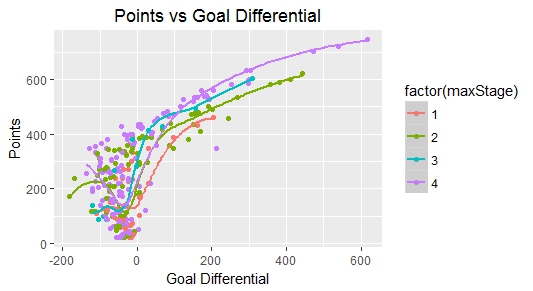


Again, it is difficult to see, but the lower end seems to have more teams that only made it to stage one, and the higher end has stage 3 and 4 teams.

Games won and stages progressed, 2008/2009:



This chart shows teams’ goal differential, points acquired, and stage progressed to:



Approach

The approach hasn’t necessarily changed. However, the analysis was limited until I could successfully unpack the XML data. The above plots suggest a very loose association between goals scored, points acquired, and stage progression. The XML data may shed further light on this. However, it might be appropriate to limit this initial analysis to identifying factors that contribute to a win. This will mean that our analysis is divided into these general concepts:

1. Significance of points, wins, and goals on stage progression
2. Significance of match factors on goals or wins

I originally wanted to include a third analysis:

1. Significance of player stats on match factors

Given the routes available within (1) and (2), it may be appropriate to contribute a subsequent study to (3) at a later date.