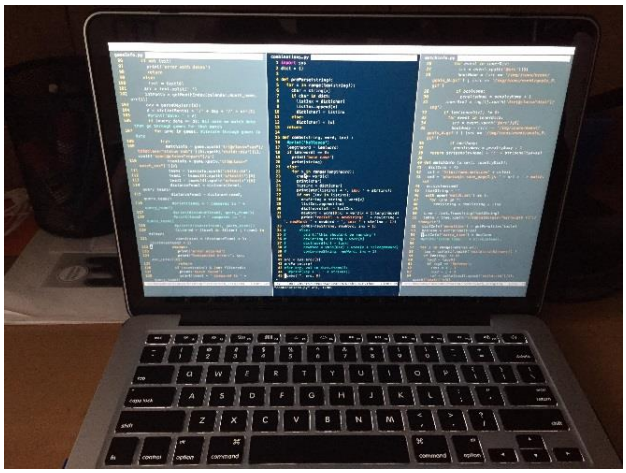

Betting and Match Fixing in Professional Sports
Deepan Saravanan (EAS 2018)
Advisor: Eugen Dimant

My PURM research project for the 2016 summer involved aggregating multiple volumes of sports betting data in a single, well-formatted CSV file, for future research purposes. More specifically, my duties were to find the correct bets for each of the games in both the UEFA Champions League and UEFA Europa League from the years 2007 to 2014. Each season consisted of at least of 150-200 matches, thus culminating with more than 2500 games to find betting information for. Moreover, after this large task had been finished, I also built several web-crawlers to find match statistics for each of the games.

As you can imagine, a lot of the problems I ran into with this project were mainly because of the sheer volume of data that needed to be analyzed. Each game for both leagues consisted of around 2000 - 7000 bets placed. The larger games with more at stake easily surpassed this amount, as the betting world often surges in activity when the teams involved are more renown. For each game, I had to track the correct bet file (which were organized by weeks) and filter for the correct betting information. Interestingly, I had severely underestimated the time my scripts would take for this filtering to occur, with the first suite of script built taking about twenty minutes for each game! Only later did I realize that the time delay made perfect sense; Just a week of betting data came to around two to three gigabytes of size, and since Python is innately slow at manual iteration, I decided build a subroutine that enabled me to filter bets in STATA. This resulted in a staggering 97% reduction in time, significantly lowering the time for parsing the game for both leagues.

After I had accomplished this concatenation of betting data for each game in the dataset, I moved on to scrape the web for red/yellow card, penalty kick, referee, and referee nationality information for each of the games. Scraping for data presented its own problems, as the dataset I was working with had team names in German, and the UEFA website in English. My first thoughts were to screen the UEFA website and build an index of German name to English counterpart. However, I noticed that most names were very similar, often missing 1-2 characters or have 1-2 characters extra (i.e. AS Roma vs AS Rom, Torino vs [Juventus] Turin). Using the similarity to my advantage, I manipulated the edit-distance algorithm to compare team names instead of checking for direct equality.

Overall, this project was a very positive experience, and through it I have learned a lot about working with large datasets and the importance of optimization. The project was also a great introduction to academia, and has certainly spurred my interest in the field.