Cross Country Course Standardization

Arianna DeBoer and Annie Wicker

November 2024

1 Project Description, Background, and Motivation

Cross country courses can vary significantly in their difficulty based on factors like terrain (flat or hilly), surface (grass or dirt), and altitude. Additionally, the weather on the given day can influence course times. Comparing runners' times between different courses can lead to misleading conclusions about capabilities and future race results. The Track and Field Results Reporting System (TFRRS) is a reputable platform that publishes race results for cross country races across the country. However, it does not have a way to compare results from different courses.

The goal of this project is to create an automated workflow that collects data from TFRRS to standardize runners' times across cross country courses. The goal is to be able to compare race times on courses that may have differed in terrain, elevation, and other factors. Though our analysis will not directly measure the effect of weather fluctuations or specific terrains, it will provide an algorithm that compares runners' times across different courses and different days to predict how they will do on a future course. This will prove a useful tool for collegiate coaches to allow for more accurate comparison of runners' performances.

2 Data Description

The Track and Field Results Reporting System (TFRRS) is a platform that compiles data from college cross country and track and field competitions. This website obtains data from individual timing companies and compiles it into an organized and uniform format. Each meet has its own webpage which contains results from every race run at that meet. For example, the page for 2024 Panorama Farms Invitational has results from both the women's six kilometer race and the men's eight kilometer race. The sample page described can be found through this link. Our algorithm can scrape pages like this, with one women's race and one men's race, to extract the results from the specified gender. Races that include more than one men's or women's race are not usable, however. This includes meets which have multiple races for each gender.

Each race result page on TFRRS contains the name of the race, the date it was run, and the distance of both the men's and women's courses. The page also contains information about each person who ran in the race.

The column 'name', lists the runner's name in the format 'First Last', and is consistent across all races this person has run.

The column 'year' lists the runner's eligibility for this season in the format 'GR-#', where 'FR-1' stands for first year of eligibility, 'SO-2' second, 'JR-3' third, and 'SR-4' fourth. This eligibility is not based on the runner's year of school, but rather on how many seasons of cross country they have run. For example, a sophomore who didn't race during their freshman year would still be listed as 'FR-1' in TFRRS. There is the potential for a person to run a race unaffiliated with a school; in this case, their 'year' would appear as '?'.

The column 'team' displays the team a person is competing for. Names of teams are also standardized across meets; for example, NC State will never appear as 'North Carolina State' or anything else other than 'NC State'. Those racing independent of a school will be listed as 'Unattached' in this column.

The column 'time' shows the official time each person ran in this race. TFRRS lists time in the format '##:##.#', where the numbers to the left of the colon represent minutes and those to the right represent seconds. Results are listed in order of time and place.

Columns listing the average pace of each runner and the number of points they earned for their team are also present in every TFRRS results page, but we are omitting these from our database since they do not provide any additional insight to our analysis.

The scraped data has been compiled into pandas dataframes. By providing the URL of a race results page from TFRRS, a user can obtain compiled results for men's or women's races. The dataframe contains the data described above: the place each runner finished (in that specific race), their name (First Last), year, college, average mile pace, total race time, score (the runner's place as a point value), the name of the meet, and the date this meet was run. These data are recorded for every runner who ran the race whose URL is entered. The script to obtain the scraped data can be found in this Github repository.

Once the data has been scraped, it is loaded into a relational database which has been set up in fourth normal form. This database has three separate tables: tRace, tRunner, and tRaceResult. tRace includes an ID, listed as race_id, for each race, as well as the name of the race, the distance, and the date it was run. tRace is connected to tRaceResult by the race ID. tRunner includes an ID, listed as runner_id, for each runner, as well as their name, eligibility (ex. JR-3), and the school they run for. tRaceResult documents each specific race result,

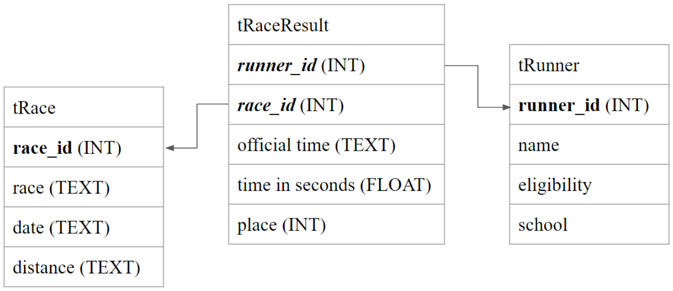

Figure 1: Database Entity Relationship Diagram

and is connected to tRunner by runner_id and to tRace by race_id. tRaceResult also includes the time and place each runner got at each race.

3 Progress and Next Steps

We have finished scraping and loading the data into a relational database, and are now working on writing functions which allow for analyses. Our first project was to create a function that compares two courses, specified by the user. This function finds all the runners who ran in both races, and then calculates the average difference in times between the two races for this group of runners. This function will work well when comparing two courses on which multiple runners have raced to see how much faster one course was. For example, a difference of negative 30 seconds between two races would mean, on average, runners ran 30 seconds on the first course than the second, indicating that the second course might be more difficult for some reason.

The version we have created works, but it requires the user to input the race ID's rather than the race names. This reduces the need to make sure a name is typed exactly as listed in TFRRS, but it also forces the user to find the race ID's of the courses they want to compare before they run this function. Because of this, we have also written a function which allows the user to input a fraction of a race name in order to pull up the race ID's for any races with this fragment in the name.

After finishing course comparisons between two courses, we hope to come up with a more generalized network that allows for comparison between two runners who have never raced against each other. This process will take much longer, and will involve drafting multiple methods of running this comparison before we decide what the best method of comparison is.

4 Group Work

Annie wrote the web-scraping script to get the data from TFRRS and compiled it into dataframes. Arianna created the database and loaded the scraped data into it. Both of us are working on writing functions that will allow us to run various analyses on cross country courses.