For office use only
T1 _______________
T2 _______________
T3 _______________
T4 _______________

Team Control Number

2021020

For office use only
F1 _______________
F2 _______________
F3 _______________
F4 _______________

2021

The International Mathematical Modeling Challenge (IM^{2}C) Summary Sheet

(Your team's summary should be included as the first page of your electronic submission.)

Muhammed Ali, Serena Williams, Michael Jordan, Roger Federer: what do all these athletes have in common? They are typically considered to be the G.O.A.T. of their respective sport- the greatest of all time. However, the term "the greatest" remains an abstract entity, fierce debates have been unable to settle the selection criteria. Some say that the greatest has the most results and records accumulated throughout the years, some say that the greatest has achieved feats unknown to mankind before, some say that the greatest has the best ability: being the strongest, fastest, and highest. To settle this debate, an objective method to determine the G.O.A.T. of any individual sport, and to an extent, any team sport is formulated.

To start with, the greatest women tennis player of 2018—Simona Halep—is identified. The model majorly considers the difference in ability of opponents and the points obtained in a match. These data are used to find the discrepancy between the points obtained and the expected performance of a player, which can compute the ability index for the player on a rolling basis. The player with the highest ability index at the end of 2018 is considered the greatest.

Then, the G.O.A.T. of men's competitive swimming (men's 100m butterfly stroke)—Michael Phelps—is determined through a two-part model. The first part narrows the scope of consideration of swimmers by selecting those with considerable amounts of achievements, through taking their rate of achieving records, and feats into account. The second part finds the G.O.A.T. by comparing the "scaled peak ability" of different swimmers, which is the ability after singling out the influence of age and technology, allowing direct comparisons across eras.

Different individual sports have different characteristics. Based on the attributes of different sports, specific changes are made to the model based on the competition style, and scoring method of a sport. For instance, tournament-style individual sports (table tennis, MMA) adapts a combination of the tennis model and the swimming G.O.A.T. model; while the swimming G.O.A.T. model can be directly applied for non-tournament-style ones (running, swimming) after collecting relevant competition data. On the other hand, subjective sports are modelled through finding a miscellaneous ability index (consisting of the actual score and the artistic component).

The major differences between team sports and individual sports lie in the contributions of players to performance and the roles of individuals. The G.O.A.T. of team sports can be determined through using the first part of the butterfly model to select teams with considerable amounts of achievements. Depending on whether the sport is tournament-styled or not, teams with the best performance are selected using similar methods for modelling individual sports. The G.O.A.T. is found through comparing the contributions (direct and indirect) of players to team performance.

Contents

1 Introduction

1.1 Introduction

What is “greatness”? In sports, athletes aim to be “Faster, Higher, Stronger”, according to the Olympics motto. Indeed, athletes have been becoming better at their respective sports throughout time. But who is the “greatest”? Is it the one with the most prizes or the one with the most world records? These questions have plagued sports fans for decades. Above all, there is the “Greatest Of All Time”—the G.O.A.T. This prestigious title has been the subject of too many heated internet debates, and everyone has a different selection criteria.

To settle this argument once and for all, we have delved deep into this problem, developed various mathematical models, and provided detailed and comprehensive analysis regarding the determination of the G.O.A.T. of different sports.

1.2 Problem Restatement

1. Develop a mathematical model for determining the greatest woman tennis player in 2018 on the basis of Grand Slam tournament results. Discuss the factors taken into consideration, and use the model to choose the greatest woman tennis player of 2018.
2. Finding the G.O.A.T. of any individual sport.
   (a) Choose one example of an individual sport (other than Women’s Tennis), and develop a mathematical model (or models) from any factors and data you find significant, measurable, and obtainable for determining the G.O.A.T. in that sport. Analyze your result.
   (b) Discuss any changes to the G.O.A.T. model from that would be required to determine the G.O.A.T. of any individual sport.
3. Discuss any changes to the G.O.A.T. models that would be required to determine the G.O.A.T. of a team sport.
4. Write a one-page letter to the Director of Top Sport describing your team’s model and your example of the G.O.A.T. for your selected individual sport.

2 Definitions and Assumptions

2.1 Definition of Important Terms in the Paper

Terms in the paper	Definition
Sport	An activity involving physical exertion and particular skills. It is usually undertaken competitively and governed by a set of rules or customs.
G.O.A.T.	The person considered the best ever to compete, perform a specific sport or activity (The Greatest of All Time).
Performance	The actual result of the athlete in a competition, which may be affected by various other factors such as technological advancements, age, and luck.

Ability	The overall performance of the athlete in all competitions took part during a specific period of time.
Strength	The physical power and innate abilities of a person.
Skill	Knowledge about a particular sport and playing techniques that are learnt and accumulated over time.
Scaled peak ability	A constant overall index of an athlete that indicates his/her maximum possible ability, without the influence of technology or age on performance.
Normalized value	The adjusted value after normalization (shifting the values from one scale to another).

2.2 Definition of Variables

Variable	**Definition**
$\overline{n}$	Mean
$\{n\}$	Normalized value of n
σ	Standard deviation
AB	Ability
SAB	Scaled peak ability

2.3 General Assumptions and Justifications

Assumption	**Justification**
Technological advancement and age affects the performance of all athletes in the same way.	This is for the ease of the construction of the model.
External factors like the temperature and humidity of the competition setting are held constant and are assumed to have no effect on the players' performance.	Since changes in these external factors are difficult to quantify, this assumption is held to simplify the model.
If a player wins by default because his/her opponent retires or is unable to play, the match is neglected.	The match result is biased and cannot fully reflect the performance of the two players, hence neglected for convenience.

3 Requirement 1

3.1 Factors taken into consideration

The athlete with the highest ability in performing the sport is considered the greatest player in a year. It is based on the performance in different matches held during the time period concerned only. Other factors like personal achievements are not considered in this model, but are detrimental to the judgement of the greatest player of all time (Section 4).
To construct the model, the performance for each one-on-match is analyzed mathematically first, and implications from each of these matches will be used to infer the players' ability in 2018. The factors taken into consideration are listed as follows:

1. Difference in abilities between opponents

In real-life tennis matches, it is almost impossible to find any two opposing players with the exactly identical strength and skill level. Therefore, a win against a much weaker opponent is incomparable to a win against a much stronger player, the same for defeats. Abilities of tennis players can be inferred from their ranks. Or from existing ability ratings, like the Elo rating.[1] Therefore, the performance of a player in the match depends on the ability of the opponent.

2. Difference between points scored by the player and the opponent in a match

The overall difference in points scored in each match is a good indication of the ability level of the two players. The larger the difference, the stronger the winner is when compared to her opponent, vice versa. Obtaining a result of "6 vs 0" (a perfect win) rather than "6 vs 4" in two consecutive sets, can display a larger ability difference between the players. In the former case, the overall difference from the winner's perspective would be +12, and +4 for the latter.

3. Number of sets played in a match

In tennis, the winner is the one who wins 2 sets first. Normally, the difference between two players is greater if it takes less sets for a player to win.

4. Performance deviations

Due to regression to the mean, it is normal to observe fluctuations in one's performance, possibly due to their varying physical conditions and mental states. Having top-notch performance does not necessarily imply being the greatest player for the year, since consistency in her performances is taken into consideration as well. The greatest player should be consistent in her performance in matches.

3.2 Additional definitions and assumptions

3.2.1 Additional Assumptions and Justifications

Assumption	Justification
Points obtained in tiebreakers are not considered.	Since the specific points for tiebreakers introduce many more possibilities and their significance is much less than the number of games won, the effect of tiebreaker points is ignored.

3.2.2 Definition of Additional Terms

Term in the paper	**Definition**
Seed	The ranking of the top 32 players.
True match point	Normalized overall difference in points scored in the match.
Forecasted win percentage	Expected probability of winning a match for a player.
Pot	The amount of match points that can be distributed in each match.
Effective matches played	Matches played within a period of time, excluding matches with default wins.

3.2.3 Definition of Additional Variables

Variable	**Definition**
AB_A	Ability of player A calculated at the end of a year.
AR_A	Ability rating of player A at an instance specified, which may change after a match is played by player A.
$AR_{A,i}$	Ability rating of player A (optionally) after tournament i, which is ordered in chronological order.
$P(A \mid A \text{ vs } B)$	Forecasted win percentage of player A when she plays against player B.
$mp_{A,B}$	True match point of a match between players A and B. Represented as an array of sets of 2 integers with each pair of numbers representing the number of games won by A and B respectively in a set. E.g. $[\{6,0\},\{6,0\}]$ means A beats B in the first and second round by 6–0. There may be 2 or 3 sets of scores, representing each set played.
$P_{A,B}$	Performance of player A when played against player B, a normalized representation of $mp_{A,B}$.
P_0	Number of match points in the pot.
P_0^A	Amount of match points in the pot contributed by player A.
N_A	The current match considered is the ${N_A}^{\text{th}}$ match player A has played within a time range, in this case, 2018.
σ_A	Ability volatility of player A calculated at the end of a year.
T_A	Total number of Grand Slam tournaments played with at least 1 valid match by player A in 2018.

3.3 Construction of model

Before any competition starts, the ability rating of each player is initialized based on their average seed in the tournaments. Players who do not have seeds for a particular tournament are assigned one seed lower than the lowest-ranking seed possible (in this case, #33). The average of each player's best 3 seeds, (or if they participated in less than 3 competitions, the average of all their seeds) is calculated. All players are ranked based on their average seeds. The player with the highest-ranked average seed will receive the highest initial ability rating and vice versa.

The mean ability rating is set to be constant at 1600. Initial ability ratings are given linearly from 1800 to 1400 points for the highest and lowest seeded players. Ratings between them are allocated linearly, so the median player will receive an ability rating of 1600. For a player A with an average seed ranking of $\frac{s}{N}$ (where s is her rank, N is the total number of players), the following equation gives her ability rating:

$$AR_A = 1400 + \frac{N-s}{N-1} \times 400$$

As there is no "draw" in tennis, $P(A)$ and $P(B)$ are complementary events. The formula for the forecasted win percentage is adapted from the Elo rating model.[7]

$$P(A \mid A \text{ vs } B) = 1 - \frac{1}{1 + (10^{(AR_A - AR_B) \div 400})}$$

$$P(B) = 1 - \frac{1}{1 + (10^{(AR_A - AR_B) \div 400})} \text{ or } P(B) = 1 - P(A)$$

The pot is the total amount of ability rating points that can be redistributed in each match. The total of ability ratings of all players is constant, but the allocation can change. When fewer matches of a player are considered previously, the uncertainty to the player's true ability rating is higher. A volatility term $\left(\frac{1}{N_A} + \frac{1}{N_B}\right)$ will be used to reflect this. At the same time, when 2 players who play against each other in a match have larger differences in ability ratings, they have a higher potential to cause a very large change in ability ratings. (For instance, when a significantly weaker player wins a stronger player, there is a larger chance for the ability rating of the winner to increase by a larger extent.) This is reflected by the absolute difference in ability ratings, $|AR_A - AR_B|$. A base pot size c_2 is added.

The size of the pot for a match between players A and B is:

$$P_0 = \left(\frac{1}{N_A + \varepsilon} + \frac{1}{N_B + \varepsilon}\right) \times (c_1 + |AR_A - AR_B|) + c_2$$

where c_1 and c_2 are constants, and in all calculations, $c_1 = 100$, $c_2 = 30$

The pot may not be contributed evenly by the two players. Instead, it is based on the expected win percentage of the players. A player with a higher forecasted win percentage contributes more to the pot, since it is expected that she has a smaller chance of losing.

The amount player A contributes to the pot:

$$P_0^A = P_0 \times P(A \mid A \text{ vs } B)$$

The amount player B contributes to the pot:

$$\begin{aligned} P_0^B &= P_0 - P_0^A \\ &= P_0 \times (1 - P(A \mid A \text{ vs } B)) \\ &= P_0 \times P(B \mid A \text{ vs } B) \end{aligned}$$

As mentioned in Section 3.1, the match point can be used to reflect the difference between ability levels of two players, and in other words, how much stronger (weaker) is player A than player B. $[\{6,0\},\{6,0\}]$ indicates a perfect win, and hence assigned a $P_{A,B}$ of 1. $[\{6,0\},\{6,0\}]$ indicates a perfect loss, hence assigned $P_{A,B}$ of 0. As there is a finite number of possible match points in a tennis match, all match points are on a spectrum from a perfect win by player A (i.e. a perfect loss by player B) to a perfect loss by player A (i.e. a perfect win by player B) with even spacing. Other possible match points are rated by the number of sets played, then by the difference in the number of games won by player A and player B.

For example, in the perspective of Player A:

$$\begin{aligned} [\{6,0\},\{6,0\}] &> [\{6,3\},\{6,0\}] > [\{6,3\},\{3,6\},\{6,0\}] \\ &> [\{6,3\},\{3,6\},\{0,6\}] > [\{3,6\},\{0,6\}] > [\{0,6\},\{0,6\}] \end{aligned}$$

There are 1470 possible numbers of $mp_{A,B}$. After evaluating them with the algorithm described above, there are 45 different ranks of match points. (Refer to Section 10). A normalized performance $P_{A,B}$ is assigned. A particular match point is ranked $\frac{a}{45}$ in all possible match points, where the 1$^{\text{st}}$ match point is a perfect win by player A. The performance is given by the equation:

$$P_{A,B} = \frac{45 - a + 1 - 1}{44} = \frac{45 - a}{44}$$

Note that $P_{B,A} = 1 - P_{A,B}$, so the match point for player B is $\frac{(45-(45-a))-1}{44} = \frac{a-1}{44}$.

The pot will be redistributed to the two players, and the distribution is proportional to their performances. The ability ratings of players A and B is updated as follows:

$$\begin{aligned} AR_{A(\text{updated})} &= AR_A - P_0^A + P_{A,B} \times P_0 \\ &= AR_A - P_0 \times P(A \mid A \text{ vs } B) + P_{A,B} \times P_0 \\ &= AR_A + P_0 \times (P_{A,B} - P(A \mid A \text{ vs } B)) \end{aligned}$$

When the performance of player A is equal to their expected chance of winning the match, her actual performance is equal to her expected performance, so the current ability rating of player A is accurate and thus has no tendency for change.

$$\begin{aligned} P_{A,B} &= P(A \mid A \text{ vs } B) \\ P_{A,B} - P(A \mid A \text{ vs } B) &= 0 \\ AR_{A(\text{updated})} &= AR_A + P_0 \times 0 \\ AR_{A(\text{updated})} &= AR_A \end{aligned}$$

When the performance of player A is higher than their expected chance of winning the match, the current ability rating of player A is underestimated. The ability rating should be increased.

$$\begin{aligned}
P_{A,B} &> P(A \mid A \text{ vs } B) \\
P_{A,B} - P(A \mid A \text{ vs } B) &> 0 \\
AR_{A(\text{updated})} &= AR_A + P_0 \times (P_{A,B} - P(A \mid A \text{ vs } B)) \\
AR_{A(\text{updated})} &> AR_A
\end{aligned}$$

Conversely, when the performance of player A is lower than their expected chance of winning the match, the current ability of player A is overestimated. The ability rating should be decreased.

As the pot is contributed unevenly, a player with a lower ability rating gains more when they fully win against a player with a higher ability than when the player with a higher ability fully wins. This can show that beating a great player has a larger significance than beating an average player.

The ability rating is updated throughout the competitions played in the year chronologically. (Australian Open (15/1) → French Open (27/5) → Wimbledon Championships (2/7) → US Open (27/8)) Within competitions, the ability ratings are updated in the order matches are played. (Fourth round → Quarterfinals → Semifinals → Finals) The ability volatility of the player is the standard deviation of their ability ratings after each tournament they played. Terms for tournaments where the player has not played are skipped and the denominator is reduced accordingly. For players who only joined 1 tournament, the ability volatility is not calculated.

$$\sigma_A = \sqrt{\sum_{i=1}^{T_A} \frac{\left(AR_{A,i} - \overline{AR_A}\right)^2}{T_A}}$$

The final ability is given by:

$$AB_A \pm \sigma_A$$

Where the highest possible ability is given by:

$$AB_A + \sigma_A$$

3.4 Implementation and evaluation of model

Rank	Player	Ability (± ability volatility)
1	Simona Halep	1645.87 ± 27.17
2	Angelique Kerber	1630.92 ± 34.72
3	Sloane Stephens	1595.12 ± 19.95

It is found Simona Halep has both the highest ability (1645.87) and the highest possible ability (1673.04) when taking volatility into account and she is the G.O.A.T. of women's tennis in 2018. Detailed results are listed in Section 10.

The model is very sensitive to fluctuations in performance. This results in consistently well-performing athletes ranked at the top. When given more tournament or match data, our model can easily give more accurate and less volatile results.

When compared against the official Elo rankings of players at the end of 2018,[6] the ranking from our model has some slight deviations. The most significant difference is Serena Williams. She was given an ability of 1541.08±14.14, ranking 7th out of the players who joined at least 2 tournaments. However, the Elo rankings considers her one of the top 3 women tennis players consistently. This difference can be explained by the fact that our model does not take prior results that are not represented in seeds into account, and that only 4 tournaments are considered.

A more advanced version of this model is to use the Elo ratings of players to project win probability instead of using seed rankings. The computation of Elo rating considers all competitions that the player has joined since debut, which can be a more accurate reflection of the level of ability of the player, and hence increase the accuracy of calculating the forecasted win percentage. In this case, the most updated weekly Elo ratings of players before the competition are used. (e.g. If the Fourth round of Australian Open is held on 15/1, Elo ratings updated as of 8/1 are used).

4 Requirement 2

Before finding the G.O.A.T. of a sport, some key differences between different types of sports would be highlighted first.

1. Competing methods

There are two types of sports: the first is conducted through "one-on-one matches", in which a player/team competes against another player/team, and a sole victor is declared ("tournament-style sports"). Some examples would be fencing, tennis. The other type of sport is conducted through comparison against an inanimate standard: individual/team results are recorded and compared with all other competitors, the one with the best result is declared the victor ("Non-tournament-style sports").

2. Scoring types

There are two scoring types: For non-tournament-style sports, there are two scoring categories: subjective type and objective type. The subjective type results are compiled through judges scoring a competitor, for example gymnastics, figure skating and diving. The objective type results are compiled by recording a measurement in standard conditions, for example competitive swimming, track, and alpine skiing.

4.1 Finding the G.O.A.T. of one chosen sport

4.1.1 Background of the chosen sport

The chosen individual sport is competitive swimming (men's 100m butterfly stroke). It is an active, non-tournament-style and subjective sport. The ability of a swimmer is reflected by the time taken to swim a certain distance in a specific stroke under standardised conditions. Our model will determine the G.O.A.T. for men, and it will be selected from swimmers who are active in the sport from 1968 to 2019. Since the nature of swimming (non-tournament-style) is different from that of tennis (tournament-style), the tennis model cannot be directly applied.

The G.O.A.T. is the swimmer with the best swim-time, after singling out the effects of technology and age.

Determining the G.O.A.T. is different from determining the greatest player in a specific year. Besides having a high level of ability in the sport, he should have sufficient influence as well. If an athlete is highly competent in the sport, but is not widely recognized and known in the public, he would not be well-qualified as the G.O.A.T.

4.1.2 Model 2 Part 1

The scope of great swimmers has to be narrowed first. For convenience of construction of the model, the sportsmen preliminarily considered are those who have participated in the event of men's 100m butterfly stroke in either the World Swimming Championships or the Olympic Games from 1968 to 2019 while ranking 1–7 in the finals.

4.1.2.1 Factors Taken into Consideration

1. Prizes

Prizes obtained in global competitions (e.g. World Swimming Championship, Swimming World Cup) and international competitions (e.g. European Open, Asian Swimming Championships) are considered, as a swimmer with international standing must be one of the greatest of their nation. Competitions with a larger scale tend to attract stronger players, which increases their difficulty and level of prestigiousness.

2. Records

World records broken by the athlete are considered, since they can, to a certain extent, reflect his ability level (he is/was the best in the world for a sports event).

4.1.2.2 Construction of Model

(i) Definition of Additional Variables

Variable	Definition
Co_{glob}	Competition scale constant for global competitions
$Pr_{\text{scale}}(\text{gol})$	Prize scale constant for gold medals / first place prizes
$\#Pr_{\text{gol,glob}}$	Number of gold medals / first place prizes an individual has obtained in global competitions
$\#Co_{\text{glob}}$	Number of global competition events an individual has participated in
$Pr(\text{glob})$	Prize factor of global competitions of an individual
Pr	Prize index of an individual
D	Duration of career of an individual
$\#Re$	Number of world records broken by an individual
ΔRe	Rate of world records broken by an individual

Re	Record index
W	Weighing of prize index
th	The threshold of the z-score for selecting an athlete
Ach	Achievement index of an individual

(ii) Additional Assumptions and Justifications

Assumption	**Justification**
Feats are not considered in this model.	Since feats are hard to be quantified in an objective manner, they are not considered for convenience. However, for subjective-scoring sports, feats could be an extra area of concern. This will be discussed in greater detail in Section 4.2.
Only achievements from the World Swimming Championships and the Olympic Games are considered.	Most good swimmers have participated in these most prestigious competitions. This is also to standardise all data as pool conditions are identical at these venues and shall minimally affect swimmers' performances.
Only achievements from 100m butterfly in standard 50m swimming pools are considered.	This is to standardise all data to swimmers swimming 2 50m lap and turning once, which takes time.

(iii) Construction of Model

For normalization of values within the range of $0.1x$ to x, where x is a positive arbitrary constant:

$$\{Pr\} = \frac{Pr - Pr_{\min}}{Pr_{\max} - Pr_{\min}} \times 0.9x + 0.1x$$

The values of the three medals—gold, silver and bronze—are quantified based on their rank. The higher the rank, the more prestigious the prize is.

Constant	$\mathbf{Co_{glob}}$	$\mathbf{Co_{int}}$	$\mathbf{Pr_{scale}(gol)}$	$\mathbf{Pr_{scale}(sil)}$	$\mathbf{Pr_{scale}(bron)}$
Value	1.5	1.25	3	2	1

The prize index of different scales of competition is the weighted sum of the three levels of attainments (gold, silver, bronze). As we are currently only considering global competitions, $Pr(\text{intl}) = 0$. Prize index is found as follows:

$$Pr(\text{glob}) = \frac{\#Pr_{\text{gol,glob}} \times Pr_{\text{scale}}(\text{gol}) + \#Pr_{\text{sil,glob}} \times Pr_{\text{scale}}(\text{sil}) + \#Pr_{\text{bron,glob}} \times Pr_{\text{scale}}(\text{bron})}{\#Co_{\text{glob}}}$$

$$Pr(\text{intl}) = \frac{\#Pr_{\text{gol,intl}} \times Pr_{\text{scale}}(\text{gol}) + \#Pr\text{sil, intl} \times Pr_{\text{scale}}(\text{sil}) + \#Pr_{\text{bron,intl}} \times Pr_{\text{scale}}(\text{bron})}{\#Co_{\text{intl}}}$$

This prize index of an individual is then found through the below formula:

$$Pr = \frac{Pr(\text{glob})}{Co_{\text{glob}}} + \frac{Pr(\text{intl})}{Co_{\text{intl}}}$$

Secondly, the records of the swimmers are quantified through finding the rate of world records broken. The higher the rate, the better the swimmer is.

$$\Delta Re = \frac{\#Re}{D}$$

The achievement index can be found by the weighted sum of the prize index and the record index. The weights are assigned by considering the relative importance of each to "greatness".

$$Ach = W \cdot Pr + (1 - W)Re$$

* W is taken as 1 in this case, it is adjustable according to the nature of other sports.

The standard score ("z-score") of the achievement index of the swimmer is calculated, in order to determine the scope of swimmers considered in part 2 of the model.

$$z_{Ach} = \frac{Ach - \overline{Ach}}{\sigma_{Ach}}$$

Those with z_{Ach} above the threshold th are included in the pool of swimmers in part 2 of the model.

* Here, th is taken as 1.

4.1.3 Model 2 Part 2

After selecting some swimmers with a considerable amount of achievement, the G.O.A.T. is determined among the pool using part 2 of the model. It can be used to differentiate between swimmers with similar levels of achievement.

The data used in this model are from the finals of Olympic Games and FINA World Championships from 1968 to 2019, for men's 100m butterfly stroke, and with competitors aged 17–32.[8] Only swimmers with ranks 7 or above are considered. The choice of data selection is justified below:

1. 1968 is the year when the Olympic Games included the event of men's 100m butterfly stroke. The wide scope ensures that top-achieving swimmers from different eras are considered.

2. Including data from international and national competitions would be redundant, since most, if not all, top world-class swimmers have participated in at least one of the two considered competitions, since the Olympic Games and World Championships are of high prestige. Since data from the finals round is counted, there would be less discrepancy between swimmers' performance, since the finalists are those with high ability. We also excluded those with rank 8 to reduce errors caused by unpredictable factors.

4.1.3.1 Additional Assumptions and Justifications

Assumption	Justification
The swimmer's overall performance across his career is not considered.	The G.O.A.T. is the swimmer with the best swim-time. If the overall performance is taken into consideration, then the model is biased towards players with stable but not the best results.

4.1.3.2 Factors taken into consideration

1. Ability VS Scaled peak ability

The abilities of athletes are greatly affected by changes in technological level and their ages. Therefore, it is difficult to directly compare their raw abilities, since it would cause bias in measurement. This is known as the problem of bridging era gaps.[4] As mentioned in Section 2.1, the scaled peak ability of an athlete indicates how well they would perform if they were all hypothetically born in the same year and of the same age, which is a constant value. Since all athletes are compared under constant conditions free from the effects of technology and age, those with higher scaled peak abilities should be more capable / "better" than those with lower values. It would be a fairer measure than to directly compare their abilities. Hence, the G.O.A.T. would be the athlete with the highest scaled peak ability.

4.1.3.3 Rationale of the Model

As mentioned above, apart from the swimmer himself, there are 2 major factors that affect one's ability. In this model, speed is the indication of a swimmer's ability.

1. Technological factor

The pace of revolution of technology has seen a significant increasing trend over the years. Due to technological advancements, athletes may have improved training infrastructure and enjoy better-designed, more nutritional diets. Through analyzing the data, a decreasing trend in swim time against year is observed, so the fact that technology positively impacts performance can be further justified. Hence, it is difficult to compare a swimmer who is 20 in 1990 with another swimmer who is of the same age in 2020, due to the technological factor. (Refer to Section 10 Appendix Figure A1: Speed of swimmers against year (unscaled))

2. Age factor

The graph below is obtained through analyzing the set of data. The performance of a swimmer is under improvement with increasing age, but this trend reverses when the age meets the threshold value of 25. The performance deteriorates with increasing age after 25, which might be due to worsened physical strength. So, it is hard to directly compare a 35 year old swimmer with a 18 year old one, in the year of 2019, due to the age factor. (Refer to Section 10 Appendix Figure B2:

Speed of swimmers against age (scaled with technological factor)) Note that speed of swimming (v) is found by dividing the distance by time taken (t), i.e. $v = \frac{100}{t}$.

4.1.3.4 Construction of Model 2 Part 2

(i) Definition of Additional Variables

Variable	Definition
T	Technological factor
A	Age factor
St	Strength
Sk	Skill
v	Swimming speed
x	Age
y	Year

(ii) Construction

From Figure A1, it can be seen that the technological factor can be approximated by a linear function. The technological factor is given by $T = f(y) = ay + b$. We may find the coefficients using the least-square method. The data used are the raw swimming speeds of athletes with ranks 7 or above between 1968 and 2019 in the final event of the World Swimming Championships and the Olympic Games.

$$
\begin{aligned}
S(a,b) &= \sum_{i=1}^{m} \left|T_i - (ay_i + b)\right|^2 \\
\frac{\mathrm{d}S}{\mathrm{d}a} &= \sum_{i=1}^{m} \left(-2y_i\left(T_i - ay_i - b\right)\right) = 0 \\
\frac{\mathrm{d}S}{\mathrm{d}b} &= \sum_{i=1}^{m} \left(-2\left(T_i - ay_i - b\right)\right) = 0 \\
a\left(\sum_{i=1}^{m} y_i^2\right) + b\left(\sum_{i=1}^{m} y_i\right) &= \sum_{i=1}^{m} y_iT_i \qquad (1) \\
a\left(\sum_{i=1}^{m} y_i\right) + b\left(\sum_{i=1}^{m} 1\right) &= \sum_{i=1}^{m} T_i \qquad (2) \\
(1) : 852887511a + 427209b &= 805115.1485608638 \qquad (3) \\
(2) : 427209a + 214b &= 403.1989143981613 \qquad (4) \\
a &= 0.004273722678214865, b = -6.647541482426789 \\
\therefore T = f(y) &= 0.00427y - 6.65 \text{ (corr to 3 sig fig)}
\end{aligned}
$$

The R^2 value for the technological factor is 0.873891715613625, which shows that the year significantly affects a swimmer's performance.

The raw swimming speed scaled with the technological factor is found as follows:

$$v' = \frac{v}{T}$$

There is not much correlation between the raw swimming speed and age. Analyzing the data collected, most of the older swimmers participated in the swimming competitions at the later years (i.e. the most recent 10–15 years), and since it is known that swimmers from recent years have better performance due to technological advancements, it causes bias in modelling. Therefore, the speed scaled with the technological factor is considered in this case. The age factor is given by $A = g(x) = ax^2 + bx + c$. To find the constants a, b, and c, the least square method is used again. The data used are the speeds scaled with the technological factor (v') and the age of participating in the competition.

The calculations are similar to that of the technological factor and can be found in the appendix

$$\therefore A = g(x) = -0.000144x^2 + 0.00634x + 0.932(\text{corr to 3 sig fig})$$

The R^2 value for the age factor is 0.02952031171685565. The effects of the age factor on performance may not be as significant as the technological factor, which can be explained by the fact that athletes of the same age have largely varying performances.

After finding the technological factor and age factor, the scaled peak ability can be found by:

$$SAB = v'' = \frac{v'}{g(x)}$$

The G.O.A.T. is the swimmer with the highest scaled peak ability among the chosen ones, i.e. the best swim-time after removing the effects of technology and age.

(iii) Additional Note

As shown from Figure B2, the speed against the age curve is an inverted U-shaped shape, with the local maximum at 24. Since speed is a measure of a swimmer's skill and strength, the ability of a swimmer follows the same trend too, with a quadratic function as the best fit.

Ability is a miscellaneous measure of a swimmer's strength and skill. The strength against age curve has a similar quadratic function as the ability against age curve. There is a consistent increase in a swimmer's strength, since his body is still under growth. After that, his body functions start deteriorating, hence a decrease in his strength. The function of strength against age is given as $St(x) = ax^2 + bx + c$, where a, b, c are arbitrary constants.

Skill refers to the swimmer's knowledge about the sport and its techniques, and it will be accumulated over time. Therefore, skill increases with age, despite a gradual decrease in strength after 24. It should be noted that skill increases at a decreasing rate with age: in the first few years that the swimmer starts swimming, he transitions from knowing nothing about the sport to learning different swimming techniques, so skill increases at a faster rate; after 30 years of swimming, there is smaller room for consistent improvement, since the swimmer is already knowledgeable in the sport, so skill increase at a slower rate. It follows the law of diminishing marginal returns, and its curve can be taken reference from the learning curve.[5] The function of skill against age is logarithmic, given by the equation $Sk(x) = d \cdot \log(x)$, where d is an arbitrary constant.

Therefore, the function of ability against age can be expressed as $Ab(x) = ax^2 + bx + c + d \cdot \log(x)$, where a, b, c, d are arbitrary constants.

4.1.4 Analysis of Model 2 Part 2

Swimmer	Scaled Peak Ability (ms^{-1})	Year of Competition	Actual Speed (ms^{-1})	Age (during competition)
Michael Phelps	1.034631845	2009	2.007226014	24
Mark Spitz	1.033572397	1972	1.842638659	22
Ian Crocker	1.031239858	2005	1.984126984	22
Michael Gross	1.027753302	1984	1.883948757	20
Pablo Morales	1.025599003	1984	1.878639875	19

As observed from the results, the G.O.A.T. of men's 100m butterfly stroke is Michael Phelps. It is worth noting that 3 out of the top 5 swimmers are from more previous years (1970s and 1980s), even though their actual swim time might not be the best. This shows that technology indeed influences a swimmer's raw performance. Also, the top 5 swimmers are quite young during the competition (19 to 24), which can be explained by the fact that a swimmer's physical condition is peak at younger ages. The results are consistent with the presumptions that technology and age both have impacts on a swimmer's performance.

The results of all swimmers selected in part 1 will be included in Section 10.

4.2 Application to all individual sports

Given that there are multiple types of sports, our G.O.A.T. models should be adjusted to accommodate the specific requirements of each sport. It should be noted that before further modelling, athletes with a considerable amount of achievements should be shortlisted first with Model 1.

1. Tournament-style sports

For tournament-style sports like table tennis, fencing and MMA, there is no concrete standard to measure an individual's ability against, comparison must be done between opponents to find the difference in ability. Therefore, Model 1 can be modified to find the G.O.A.T. Since rankings of players may not be fully accurate in reflecting the projected win probability, existing scoring indexes like the Elo rating of different players can be used instead, as mentioned in Section 3.4. Since these indexes can reflect the relative abilities of players against other athletes of their time, the problem of discrepancy in ability between players from different eras is not as significant as non-tournament style sports. These indexes will not be subject to change under the influence of technology, so they can be used for direct comparison.

By plotting the scoring indexes of different players against the time since their first debut, graphical analysis can be used to identify the G.O.A.T. The curve of his scoring index should be less fluctuating and he should have one of the highest sets of scoring indexes across his career.

The time interval for which scoring indexes are recorded is a concern too. If the time interval is small (e.g. 1 month, 3 months), it would be more specific but less accurate, since a player's performance in the short term is prone to random fluctuations, but it should be a consistently increasing trend in the long run instead. Therefore, a time interval of 1 year is considered in this case.

Referring to Figure C in Section 10, it can be seen that Player 1 has the highest sets of Elo ratings and has a rather consistent performance, so he is chosen as the G.O.A.T. A similar method can be applied for a larger athlete pool size for different sports.

2. Non-tournament style sports

For non-tournament style sports like swimming and running, the G.O.A.T. can be found with a modified version of Model 2. The effects of age and technology on the ability of players differ for each sport, so Model 2 cannot be directly applied. One way is to collect data of athletes from global contests throughout a long period of time (e.g. 1970 to 2020), then use the same method to model the technological factor and the age factor to find the scaled peak ability.

Another method is to make use of the data from swimming in Model 2, and adjust the technological factor and age factor accordingly based on online research and speculation of how technology and age affect the performance of players for that particular sport. It is a more convenient way to find the G.O.A.T., but it might be less accurate than the previous method.

3. Subjective scoring sports

Subjective scoring sports include gymnastics, diving and figure skating, where the performance of the athlete is based on subjectivity of the judges. Also, the standards for judging performance are relatively more prone to changes than conventional sports like running. There is rapid evolution of techniques over time too. As more difficult techniques are invented as time progresses, modern athletes of subjective scoring sports tend to have better skills than past athletes. Therefore, it is hard to model their abilities objectively.

One way to model is to first select competent athletes with Model 2 Part 1, combined with the considerations for the feats of players (for example, inventing new techniques like the V-style in ski jumping, achieving unprecedented results like Nadia Comaneci's perfect 10 in gymnastics, etc.) Then, consider the actual score and the artistic component of the performance. To a certain extent, the actual score given by the judges can reflect the ability of the athlete. The artistic component demonstrates an individual's understanding of their performance (e.g. choreography, aesthetics of the performance). An overall index can be calculated from the weighted sum of the actual score and the artistic component, then further modelled through the method of Model 2.

5 Requirement 3

5.1 Additional assumptions and justifications

Assumption	Justification
Team sports with frequently changing team compositions are not considered.	It is hard to find the overall performance of the team if the team composition varies, and since the achievements by the team are not obtained by the same members, it might under/overstate the level of ability of the team. This leads to high inaccuracies in finding the G.O.A.T.

5.2 Analysis of team sports

There are some notable differences between individual sports and team sports.

1. Number of participating individuals and their contributions to team performance

Given that there are > 1 individuals per team, it is harder to determine who is the greatest team sport player based on the overall results of the team. The final result is the sum of work between individuals, but each individual contributes to a different extent. The contribution of each individual can be split into two components: direct contribution and indirect contribution.

Direct contribution can be defined by affecting the team's final results in a match directly, for example, by shooting the hoop in basketball, successfully spiking in volleyball, or directly scoring a point in pairs badminton.

Indirect contribution can be defined by assisting another team member in having direct contribution to the team, for example by passing the ball to the shooter in basketball (also known as assists) or passing the ball in international football.

In cases where the contributions of each individual team performance is equal, such as synchronised swimming or team competitive archery events, their contributions will be determined by their individual performance (how they perform as an individual in the competition). Usually, a scoring system with the same type as the sport is used. For instance, in team competitive archery, the objective score achieved by each of the individuals can be compared.

2. Role of individuals

Team sports can be split into two types: role sports and non-role sports. In role sports, different individuals have different roles. Role sports may include volleyball, American football, netball and pairs skating. In football, some players take up offensive roles (e.g. forward), some take up defensive roles (e.g. goalkeeper), and some take up both offensive and defensive roles simultaneously depending on the situation (e.g. midfielder). In non-role sports, different individuals have identical/similar roles, like dual badminton.

In role sports, as different roles have different duties in the competition, they would contribute in different ways, so the contribution of each specific individual would have to be considered.

5.3 Adjustments to G.O.A.T. model

There are 3 types of team sports to be considered. Teams with considerable amounts of achievements are selected through Model 2 part 1 first.

1. Tournament-style team sports (e.g. basketball, volleyball)

Making use of the tennis model in Section 3, teams are equivalent to the tennis players in the model, and the abilities of the teams are calculated either through their ranks (in this case, the tennis model can be directly applied), or through existing ranking indicators (e.g. Elo ratings for men football). Teams with abilities above a certain threshold will be considered.

Then, the contributions of individual players to the team performance are considered. This is found through calculating the weighted average of a player's direct and indirect contributions in the finals and semifinals of global and international competitions across different years. The graphical analysis method used in tournament-style individual sports (Section 4.2) can be applied to find the G.O.A.T. through comparing the overall contributions of different players against the year since the team's debut.

2. Non-tournament-style team sports (e.g. dragonboat)

Through finding data from past competitions, correlation between the team's ability, strength and skill, and the average age of the team members can be found. Using a similar method as Model 2, the scaled peak ability of the team is determined. Teams with scaled peak abilities above a certain threshold are selected for further analysis.

Similar to tournament-style team sports, the contributions of individual players to team performance is calculated in the same way. The G.O.A.T. is the one with the highest overall contribution.

3. Objective scoring team sports (e.g. swimming relay)

The method is similar to finding the G.O.A.T. of objective individual sports, except that individual performances are synthesised from the team performance, and are compared on an individual basis. For instance for freestyle swimming relay, the time taken for each member to swim the distance is compared against other team members and other teams, swimmers with a shorter time contribute more.

4. Subjective scoring team sports (e.g. synchronized swimming)

As mentioned in Section 4.2, it is difficult to model the ability of players of subjective scoring sports, since subjective scoring is involved. One way is to use the same modelling method for subjective scoring individual sports, to find the top-performing teams first. The athlete who has less errors in performance and performs more techniques with high levels of difficulty is considered the G.O.A.T.

6 Strengths, Weaknesses and Improvements of Model

6.1 Strengths of Model

1. Ease of implementation

After inputting correct data, values like the ability of tennis players in a given year, and the scaled peak abilities of swimmers, can be effectively predicted through the aid of computer programs and graphical analysis.

2. Generality

Anyone can collect the information on sportsmen with ease. There is no limitation on the specific number of individuals to be included in the models, and they can be customized easily based on the characteristics and nature of different sports, to predict the G.O.A.T. of both team and individual sports.

3. Comprehensiveness

Many factors are taken into account for the construction of models. For model 1, the players' abilities are not determined solely through the number of wins and losses, but are also based on the skill levels of opponents, overall difference in points obtained and fluctuations in performance. They can better simulate real life situations and provide accurate results.

Model 2 considers both the achievements and the abilities of the athletes. For part 2 of the model, it singles out the effect of technology and age on the performances of different swimmers, which allows unbiased analysis of their true abilities, avoiding the problem of the bridging era gaps.

6.2 Limitations of Model

1. Large amount of data required

For Model 2 part 2, a large amount of data for athletes across a large year span is required to model the technological factor and age factor. Insufficient data may lead to inaccuracy in determining the G.O.A.T.

2. Considerable number of assumptions made

The assumptions are made for the ease of calculation. However, they may potentially have significant impacts on the performance of the athletes in real life. (For instance, the changes in external conditions of the competition setting may affect performance). This might lead to slight inaccuracies in the results predicted by the models.

3. Incomprehensiveness in quantifying variables in the model

For model 2 part 1, achievements attained by athletes may not be objectively quantified enough, since it is hard to determine the level of attainment through assigning arbitrary weights to awards obtained in different competitions. The level of ease of obtaining the same award may differ for different years as well. However, despite the flaws of the model, it can still serve as a rough indication of the influence and ability of athletes.

6.3 Improvements of Model

1. Model 1

As mentioned in Section 3.4, an advanced version of the model can be constructed by using the Elo rankings of tennis players instead of the rank of their average seeds, which can increase the accuracy. Leniency may be considered for players with specific conditions (e.g. pregnancy, injury).

2. Model 2 Part 1

More factors of consideration may be included, like the number of feats, duration of world records held, and number of personal records broken. This improves the comprehensiveness of the model.

Also, since the value placed on different competitions may be different across years (e.g. Olympic games may be more important than World Championships in the 1970s), the achievement index can be adjusted based on the relative importance of different competitions in different time periods, through online research.

3. Model 2 Part 2

The scope of consideration can be extended to more international and global competitions, like the FINA Swimming World Cup, Asian Games or Commonwealth Games, which could possibly raise the accuracy in modelling the technological and age factors with a larger data set.

7 Conclusion

“Greatness” in sports has always been an abstract concept. To define the G.O.A.T. of different sports, we have developed various mathematical models to achieve our goal.

For requirement 1, based on differences in abilities of players in the match, overall difference in points scored, and fluctuations in performance, the greatest female tennis player in 2018 is found to be Simona Halep through calculating the ability ratings of the players on a running basis.

For requirement 2a, a two-part model is developed to choose the G.O.A.T. of men's competitive swimming based on their achievements first, then through comparing their scaled peak abilities (the ability of swimmers free from the effects of technology and age). Achievements include an individual's prizes, records and feats.

For requirement 2b and 3, adjustments to the existing models are made accordingly based on the characteristics of different sports. Players are selected through Model 2 Part 1 first. Tournament-style sports would be modelled through a modified version of Model 1, while non-tournament-style sports are modelled through Model 2 Part 2. For team sports, the G.O.A.T. would be first determined by the ability of the team they belong to. Then, based on the different characteristics of each sport, the model will be adjusted to determine the greatest teams of a team sport, using similar methods for individual sports. The G.O.A.T. would be the one with the largest contribution to team performance.

8 Requirement 4: Letter to the Director

Dear Sir/Madam,

Our team has evaluated the definition of "greatness" in sports, and constructed models to find the G.O.A.T. of men's competitive swimming (100m butterfly stroke), which is then extended to other individual and all team sports.

Our model for finding the G.O.A.T. of any individual sport is split into 2 parts: achievement and ability. The first part, achievement, accounts for the number of prizes obtained from competitions of different scales and the average number of world records achieved per year in their career. This prevents bias towards athletes who have short careers, since the cumulative number of achievements may not be the best measure of influence. The purpose of this is to narrow the scope of consideration to swimmers with adequate influence and considerable amounts of achievement.

The second part of the model considers the scaled peak ability (the real ability after singling out the effects of technology and age) of the swimmers. Take men's competitive swimming (men's 100m butterfly stroke) as an example. After narrowing down the list of swimmers considered, the second part of the model is used. Since it is found that both technology (the overall swim speed increases with time) and age (younger swimmers tend to perform better than older ones) have great influence on a swimmer's performance, their impacts are eliminated through scaling a swimmer's ability at their prime to allow direct comparison between swimmers from different eras and age groups. The G.O.A.T. is found to be Michael Phelps, with Mark Spitz coming as a close second.

Our models are customizable according to the characteristics and competing methods of different individual sports. The first part of the model is still used to determine the best sportsmen.

1. **Non-tournament style sports** (e.g. swimming and track)
For these sports, individuals compete against an inanimate standard. The second part of the model can be modified by entering different data from competitions throughout a large year span.

2. **Tournament style sports** (e.g. tennis, MMA)
Individuals compete through "one-on-one" matches for tournament style sports. In this case, their ranks can be used to project their winning probabilities in different matches, which is then utilized for calculating the ability indexes of different players. The ability indexes for players from different eras and age groups can be directly compared to find the G.O.A.T.

3. **Subjective scoring sports** (e.g. dancing, gymnastics)
These are mostly non-tournament style sports, where results are determined by the opinions of judges. Since these sports usually have an artistic nature, the actual score obtained and the artistic component (e.g. aesthetic of the performance, choreography, etc.) can be used to calculate the overall ability and hence for the determination of the G.O.A.T.

Our model for individual sports can be applied to team sports too, albeit with a few changes. Team sports can be additionally classified into role sports and non-role sports, and the contributions of individual players are usually varied. Part 1 of the original model can be used to determine the best teams in a team sport. Then, based on the aforementioned characteristics for both individual and team sports, the second part of the model could be altered to determine the degree of contribution of each individual of a team. The individual who contributes the most is the greatest.

We sincerely hope that our findings are useful to you, and can help settle the argument of the G.O.A.T. If you have any inquiries, please feel free to contact us again.

Yours faithfully,
IMMC2021020

9 References

References

[1] : Elo rating system. (2021, March 12). Retrieved March 18, 2021, from https://en.wikipedia.org/wiki/Elo_rating_system#:~:text=The%20Elo%20rating%20system%20is%20a%20method%20for,system%20over%20the%20previously%20used%20Harkness%20system

[2] : Ross, J. (n.d.). ACSM Metabolic Calculations. Retrieved March 18, 2021, from https://summitmd.com/pdf/pdf/090626_aps09_970.pdf

[3] : Koutlianos, N., Dimitros, E., Metaxas, T., Cansiz, M., Deligiannis, A., & Kouidi, E. (2013, April). Indirect estimation of VO2max in athletes By Acsm's EQUATION: Valid or not? Retrieved March 18, 2021, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3743617/

[4] : Berry, S. M., Reese, C. S., & Larkey, P. D. (1999, September). Bridging Different Eras in Sports. Retrieved March 19, 2021, from https://www.vanderbilt.edu/psychological_sciences/graduate/programs/quantitative-methods/quantitative-content/berry_reese_larkey_1999.pdf

[5] : Learning curve. (2021, March 17). Retrieved March 19, 2021, from https://en.wikipedia.org/wiki/Learning_curve

[6] : United, O. (1970, January 01). Women's ELO RANKINGS 05/11/2018. Retrieved March 20, 2021, from https://tenniseloranking.blogspot.com/2018/11/womens-elo-rankings-05112018.html

[7] : Jeff, A. (2019, December 03). An introduction to tennis elo. Retrieved March 20, 2021, from http://www.tennisabstract.com/blog/2019/12/03/an-introduction-to-tennis-elo/

[8] : Official fina website. (n.d.). Retrieved March 20, 2021, from https://www.fina.org/results?year=2021&month=latest&disciplines=

10 Appendix

10.1 Additional Data, Results and Programs for Section 3

Rank	Player	Ability (± ability volatility)
1	Simona Halep	1645.87 ± 27.17
2	Angelique Kerber	1630.92 ± 34.72
3	Sloane Stephens	1595.12 ± 19.95
4	Garbine Muguruza	1591.8
5	Caroline Wozniacki	1580.54 ± 24.59
6	Naomi Osaka	1580.47 ± 31.67
7	Jelena Ostapenko	1561.02
8	Madison Keys	1549.79 ± 13.42
9	Serena Williams	1541.08 ± 14.14
10	Elina Svitolina	1540.18 ± 8.4
11	Anastasija Sevastova	1531.53
12	Julia Gorges	1523.01
13	Elise Mertens	1515.58 ± 16.46
14	Kiki Bertens	1508.85
15	Daria Kasatkina	1505.82 ± 10.36

Table A: Top 15 women tennis players based on Model 1

Codes

Find all possible match points of a tennis match

```python
one_match_outcomes = [[7, 6], [7, 5], [6, 4], [6, 3], [6, 2], [6, 1], [6, 0], [0, 6], [1, 6], [2, 6], [3, 6], [4, 6], [5, 7], [6, 7]]
# win: i = 0 - 6 (inclusive)
# lose: i = 7 - 13 (inclusive)
all_outcomes = []
# permute
for i in range(len(one_match_outcomes)):
  if (i <= 6):
    win1 = 1 # player 1 wins
  else:
    win1 = 2
  for j in range(len(one_match_outcomes)):
    temp = [one_match_outcomes[i]]
    temp.append(one_match_outcomes[j])
    if (j <= 6):
      win2 = 1
    else:
      win2 = 2
    if (win1 != win2):
      # needs set 3
      for k in range(len(one_match_outcomes)):
        temp = [one_match_outcomes[i]]
        temp.append(one_match_outcomes[j])
        temp.append(one_match_outcomes[k])
        all_outcomes.append(temp)
    else:
      all_outcomes.append(temp)
print(all_outcomes)
print(len(all_outcomes))
# expected number of outcomes = 14 * 14 + 14 * 7 * 14 - 14 * 7 = 1470
```

Calculate and rank match ratings

```
rating_to_match_points={}

def rating_for_match_points(match1):
  # below are intermediate variables to evaluate the rating that is used to rank match points
  # first see if it is a win or a lose
  rating = 0
  number_sets_won = 0
  for i in match1:
    # looping for each set
    if (i[0] > i[1]):
      number_sets_won = number_sets_won + 1 # this set is won by player A
    # note that to win the match, 2 sets must be won
  won = 0
  if (number_sets_won == 2):
    won = 1 # won
  else:
    won = -1 # lost
  # scores for matches won will be positive, scores for matches lost will be negative, but the magnitude of the rating is calculated with the same way
  # then consider the number of sets won, the magnitude of the rating is higher when only 2 sets are played
  if (len(match1) == 2):
    rating = rating + won * 100 # if the game is lost, losing within 2 sets is 'worse' for player A so the rating is lower (because this will evaluate to -100)
    # 100 is a constant that will be larger than the difference in points scored to ensure that number of sets played is a more important factor
  # finally calculate the difference in points scored
  # there is no need to take average of difference in point scored because games with 2 and 3 sets will always be separate from each other from the constant added above
  point_difference = 0
  for i in match1:
    # looping each set played
    point_difference = point_difference + i[0] - i[1]
    # note that if the game is won, point_difference will be a positive number and vice versa
    # the larger the magnitude of point_difference, the larger the magnitude of the rating
  rating = rating + point_difference
  return rating

for i in all_outcomes:
  temp = rating_for_match_points(i)
  if (temp in rating_to_match_points):
    rating_to_match_points[temp].append(i)
  else:
    rating_to_match_points[temp] = [i]

key = rating_to_match_points.keys()
key = sorted(key, reverse=True) # ratings ("keys" to the dict storing possible match ratings) are sorted from "perfect win by A" to "perfect loss by A"
# for i in key:
  # print("key =", i)
  # print(rating_to_match_points[i])
print("number of possible match ratings", len(key))
print("all keys", key)
```

Define utility functions

```
from math import sqrt
def sd(a):
  # a is a list of floats/ints
  if (len(a) == 1):
    return None
  temp = 0
  mean = sum(a) / len(a)
  for i in a:
    temp += (i - mean) * (i - mean)
  temp /= len(a)
  temp = sqrt(temp)
  return temp
```

```python

def normalize(a, N, reverse):
  # a: order
  # N: number of items
  if reverse == True:
    # we want to reverse the order, e.g.\ rank 1 should be the "highest ranking"
    return ((N - a + 1) - 1) / (N - 1) # reverse, then normalize
  else:
    return ((a - 1) / (N - 1)) # just normalize

print(normalize(key.index(-112) + 1, 45, 1)) # try it out!
```

Defining matches and players

```python
class match:
  def __init__(self, a, b, match_points):
    self.player_a = a # this is an index!
    self.player_b = b
    self.match_points = match_points # key of the score
    self.performance = normalize (key.index(rating_for_match_points(match_points)) + 1, 45, True) # performance for player A, note that key.index returns a 0-indexed rank
    self.log = {} # will put in a log dict when it is run
  def describe(self):
    print("#############")
    print(self.player_a, "vs", self.player_b)
    print(self.match_points, self.performance)
    print(self.log)
    print("#############")
class player:
  def __init__(self, name, avg_seed):
    self.name = name
    self.ability_rating = 0 # init this later
    self.avg_seed = avg_seed
    self.matches_played = 0
    self.sd = 0
    self.ratings = [] # stores old ability ratings after each grand slam tournament
    self.tournaments = [] # stores the names of tournaments played

def win_probability(ability_rating1, ability_rating2):
  # based on Elo
  return 1 - (1 / (1 + 10 ** ((ability_rating1 - ability_rating2) / 400)))
```

Import data and create a reset function

```python
all_players = {}
def reset_players():
  global all_players
  all_players = {}
  all_players["Simona Halep"] = player("Simona Halep", 1)
  all_players["Naomi Osaka"] = player("Naomi Osaka", 20)
  all_players["Barbora Strycova"] = player("Barbora Strycova", 23)
  all_players["Karolina Pliskova"] = player("Karolina Pliskova", 7)
  all_players["Hsieh Su Wei"] = player("Hsieh Su Wei", 33)
# [35 lines skipped]

  print(len(all_players))
  temp2 = set()
  for i in all_players:
    temp2.add(all_players[i].avg_seed) # set is used to count distinct items
  temp2 = list(temp2)
  for i in all_players:
    all_players[i].ability_rating = 1400 + normalize(temp2.index(all_players[i].avg_seed) + 1, len(temp2), True) * 200
    # edit init value here
    # print(all_players[i].name, all_players[i].ability_rating)
  print("players reset")

reset_players()
```

Import match data (by chronological order)

```
australian_open = []
french_open = []
wimbledon_championships = []
us_open = []

australian_open.append(match("Simona Halep", "Naomi Osaka", [[0, 6], [6, 4], [3, 6]]))
australian_open.append(match("Hsieh Su Wei", "Angelique Kerber", [[6, 4], [5, 7], [2, 6]]))
australian_open.append(match("Madison Keys", "Caroline Garcia", [[6, 3], [6, 2]]))
australian_open.append(match("Barbora Strycova", "Karolina Pliskova", [[7, 6], [3, 6], [2, 6]]))
australian_open.append(match("Petra Martic", "Elise Mertens", [[6, 7], [5, 7]]))
australian_open.append(match("Denisa Allertova", "Elina Svitolina", [[3, 6], [0, 6]]))
australian_open.append(match("Anett Kontaveit", "Carla Suarez Navarro", [[6, 4], [4, 6], [6, 8]]))
australian_open.append(match("Magdalena Rybarikova", "Caroline Wozniacki", [[3, 6], [0, 6]]))
australian_open.append(match("Simona Halep", "Karolina Pliskova", [[6, 3], [6, 2]]))
australian_open.append(match("Angelique Kerber", "Madison Keys", [[6, 1], [6, 2]]))
```

```python
# [5 lines skipped]

french_open.append(match("Simona Halep", "Elise Mertens", [[6, 2], [6, 1]]))
# [12 lines skipped] (some matches are invalid)

wimbledon_championships.append(match("Hsieh Su Wei", "Dominika Cibulkova", [[4, 6], [1, 6]]))
# [14 lines skipped]

us_open.append(match("Kaia Kanepi", "Serena Williams", [[0, 6], [6, 4], [3, 6]]))
# [14 lines skipped]

print(len(australian_open), len(french_open), len(wimbledon_championships), len(us_open))
# 15 13 15 15
```

Define how a match is played

```python
def play_match(match1, tournament_name):
  # match1 is a match object
  global all_players
  player1 = all_players[match1.player_a] # this makes accessing variables easier, but variables should NOT be edited using this!
  player2 = all_players[match1.player_b]
  pot = (1 / (player1.matches_played + 5) + 1 / (player2.matches_played + 5) ) * (abs(player1.ability_rating - player2.ability_rating) + 100) + 30
  win_probability1 = win_probability(player1.ability_rating, player2.ability_rating)
  # print("before")
  # print("A:", all_players[match1.player_a].ability_rating)
  # print("B:", all_players[match1.player_b].ability_rating)
  # print("performance:", match1.performance, "win_probability1", win_probability1)
  match1.log["A rating before"] = all_players[match1.player_a].ability_rating
  match1.log["B rating before"] = all_players[match1.player_b].ability_rating
  all_players[match1.player_a].ability_rating = player1.ability_rating + pot * (match1.performance - win_probability1) # update A
  all_players[match1.player_b].ability_rating = player2.ability_rating + pot * (win_probability1 - match1.performance) # update B
  all_players[match1.player_a].matches_played += 1
  all_players[match1.player_b].matches_played += 1
  if (len(all_players[match1.player_a].tournaments) == 0 or all_players[match1.player_a].tournaments[-1] != tournament_name):
    all_players[match1.player_a].tournaments.append(tournament_name)
  if (len(all_players[match1.player_b].tournaments) == 0 or all_players[match1.player_b].tournaments[-1] != tournament_name):
    all_players[match1.player_b].tournaments.append(tournament_name)
  # print("after")
  # print("A:", all_players[match1.player_a].ability_rating)
  # print("B:", all_players[match1.player_b].ability_rating)
  match1.log["A rating after"] = all_players[match1.player_a].ability_rating
  match1.log["B rating after"] = all_players[match1.player_b].ability_rating
  match1.log["pot"] = pot
  match1.describe()
```

Run model

```
reset_players()
for i in range(len(australian_open)):
  play_match(australian_open[i], "australian open")

for i in all_players:
  if (len(all_players[i].tournaments) != 0 and all_players[i].tournaments[-1] ==
   ↪  "australian open"):
    all_players[i].ratings.append(all_players[i].ability_rating)

for i in range(len(french_open)):
  play_match(french_open[i], "french open")

for i in all_players:
  if (len(all_players[i].tournaments) != 0 and all_players[i].tournaments[-1] ==
   ↪  "french open"):
    all_players[i].ratings.append(all_players[i].ability_rating)

for i in range(len(wimbledon_championships)):
  play_match(wimbledon_championships[i], "wimbledon championships")

for i in all_players:
  if (len(all_players[i].tournaments) != 0 and all_players[i].tournaments[-1] ==
   ↪  "wimbledon championships"):
    all_players[i].ratings.append(all_players[i].ability_rating)

for i in range(len(us_open)):
  play_match(us_open[i], "us open")

for i in all_players:
  if (len(all_players[i].tournaments) != 0 and all_players[i].tournaments[-1] ==
   ↪  "us open"):
    all_players[i].ratings.append(all_players[i].ability_rating)
  print(i)
  print(all_players[i].ratings)
```

Output results for all players (ranked)

```
temp3 = sorted (list(range(len(all_players))), key = lambda x:
 ↪  all_players[list(all_players.keys())[x]].ability_rating, reverse=True)
print(temp3)
for j in temp3:
  i = list(all_players.keys())[j]
  all_players[i].sd = sd(all_players[i].ratings)
  if (all_players[i].sd == None):
    print(i.ljust(25), round(all_players[i].ability_rating, 2))
  else:
    print(i.ljust(25), round(all_players[i].ability_rating, 2), "+/-",
     ↪  round(all_players[i].sd, 2))
```

10.2 Additional Data, Results and Programs for Section 4

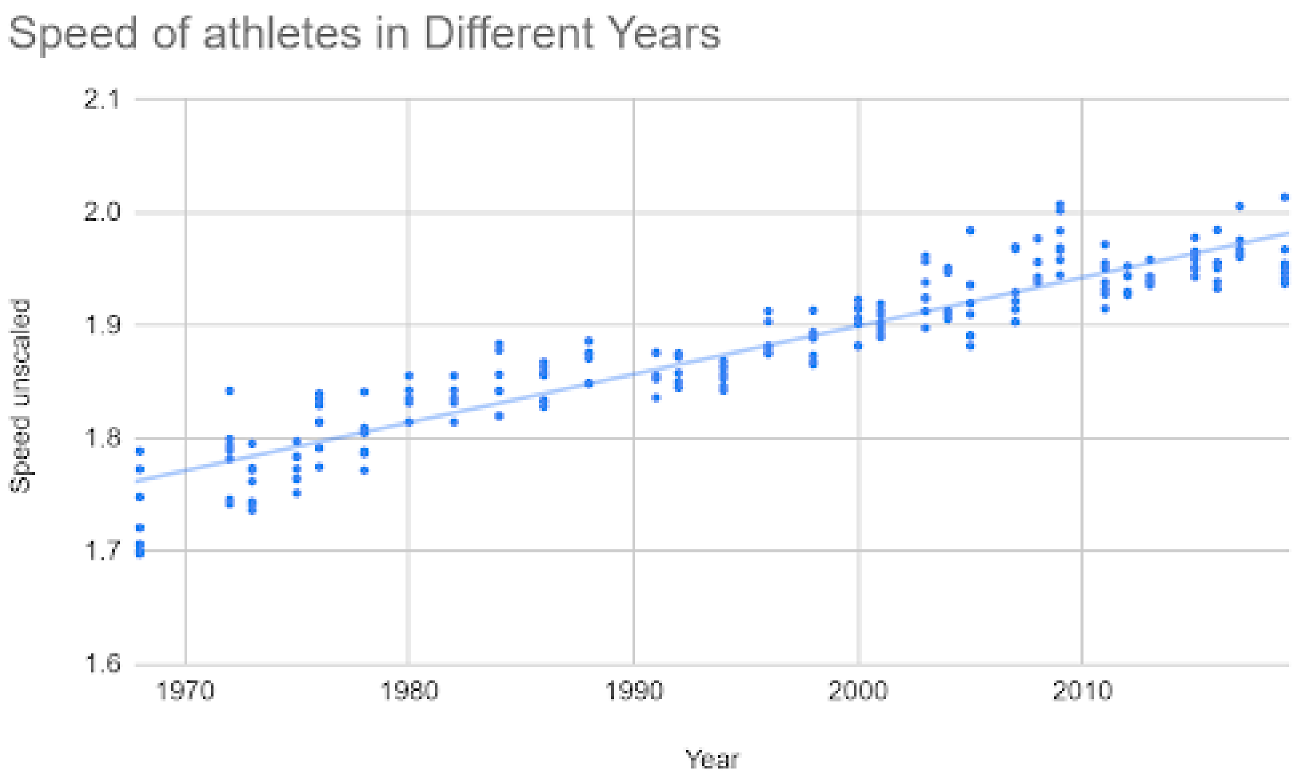

Figure A1: Speed of swimmers against year (unscaled)

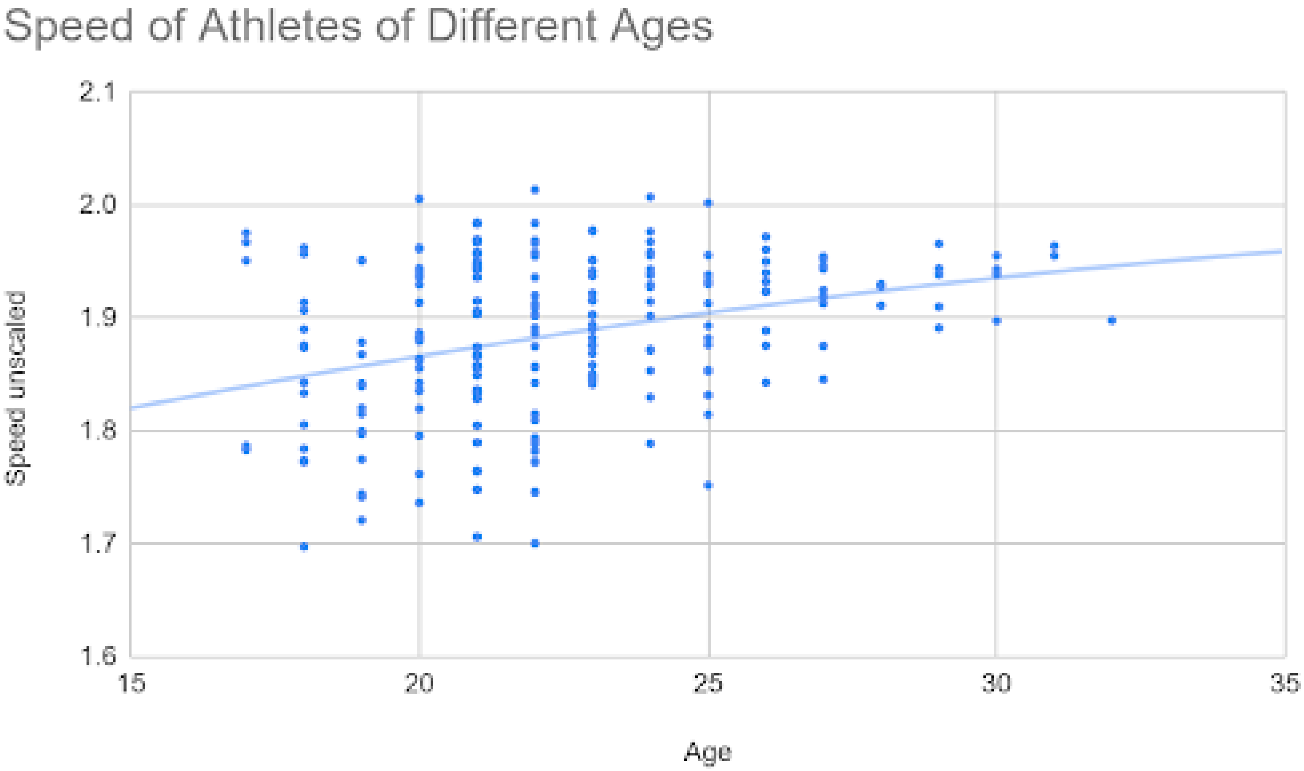

Figure B1: Speed of swimmers against age (unscaled)

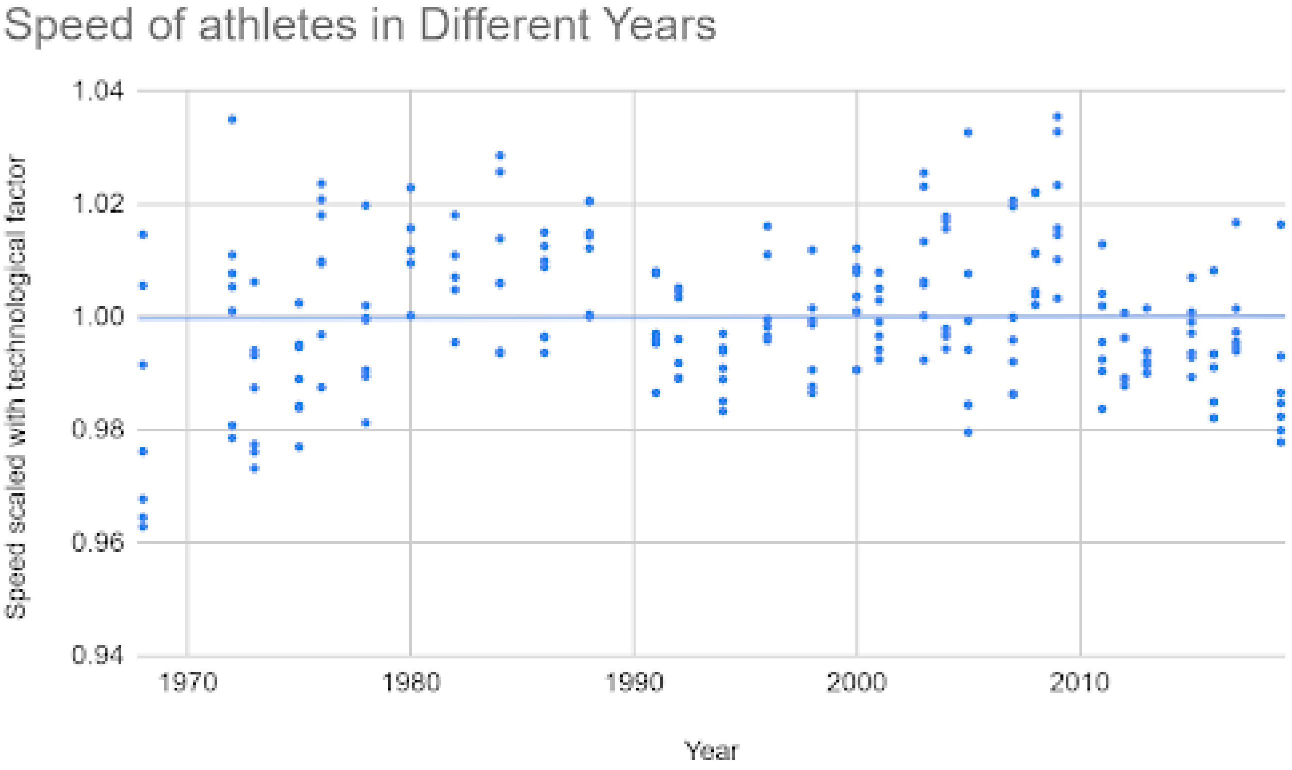

Figure A2: Speed of swimmers against year (scaled with technological factor)

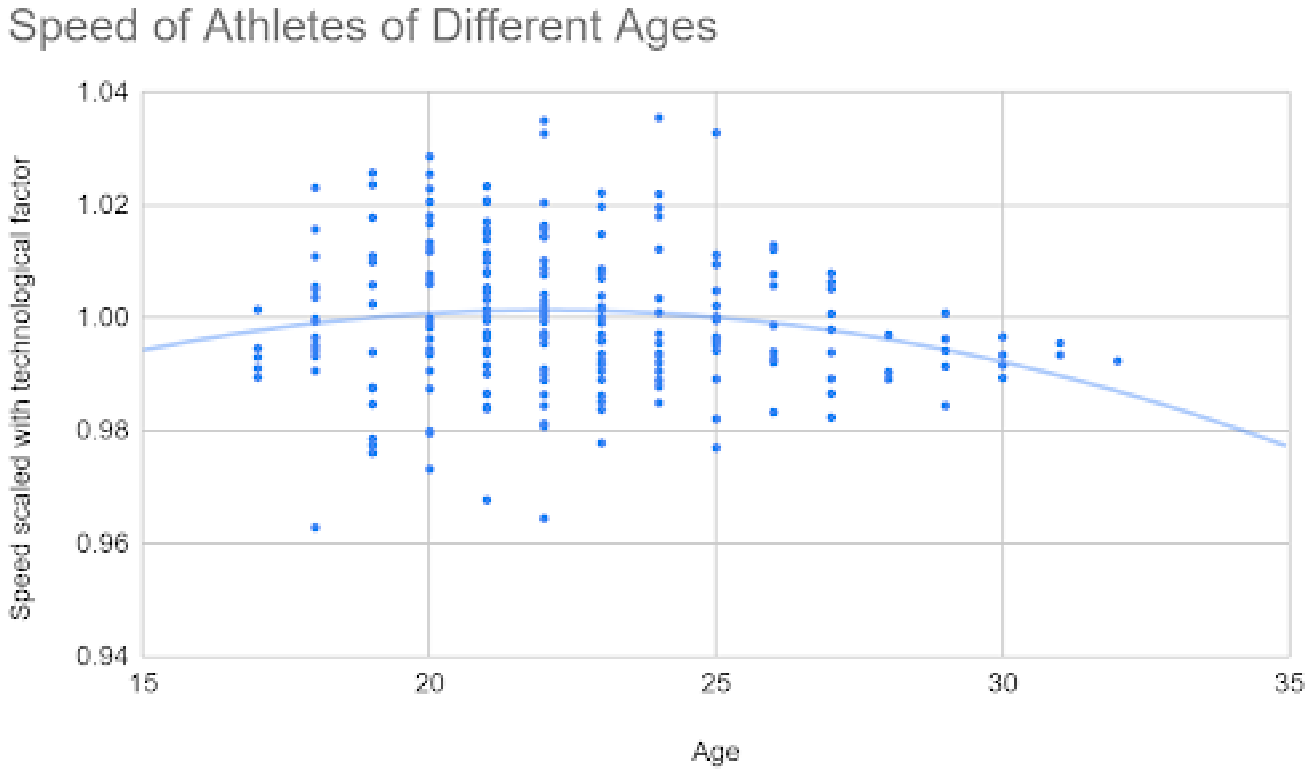

Figure B2: Speed of swimmers against age (scaled with technological factor)

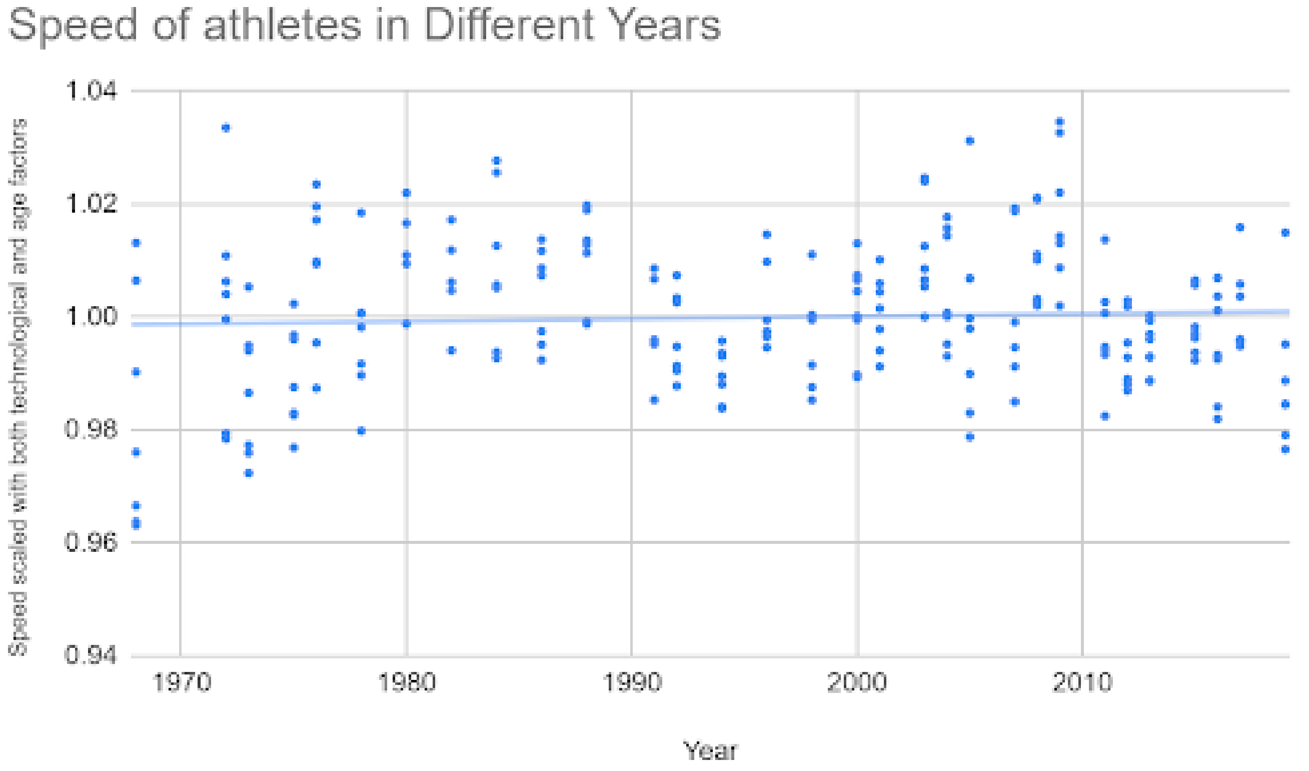

Figure A3: Speed of swimmers against year (scaled with both technological and age factors)

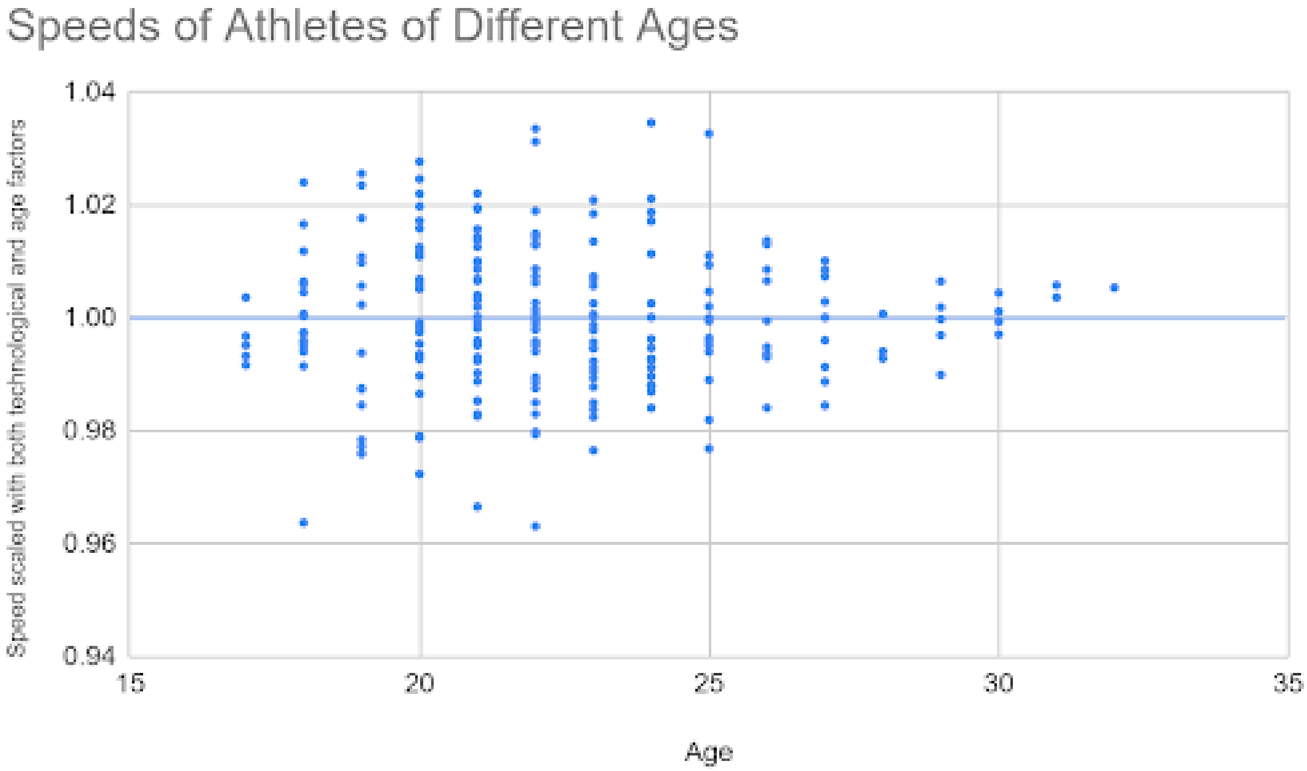

Figure B3: Speed of swimmers against age (scaled with both technological and age factors)

Calculations for Finding the Age Factor in Model 2 Part 2

$$S(a,b,c) = \sum_{i=1}^{m} \left|A_i - \left(ax_i^2 + bx_i + c\right)\right|^2$$

$$\frac{\mathrm{d}S}{\mathrm{d}a} = \sum_{i=1}^{m} -2x_i^2 \left(A_i - ax_i^2 - bx_i - c\right) = 0$$

$$\frac{\mathrm{d}S}{\mathrm{d}b} = \sum_{i=1}^{m} -2x_i(A_i - ax_i^2 - bx_i - c) = 0$$

$$\frac{\mathrm{d}S}{\mathrm{d}c} = \sum_{i=1}^{m} -2(Ai - axi2 - bxi - c) = 0$$

$$a\left(\sum_{i=1}^{m} x_i^2\right) + b\left(\sum_{i=1}^{m} x_i\right) + c\left(\sum_{i=1}^{m} 1\right) = \sum_{i=1}^{m} A_i \tag{1}$$

$$a\left(\sum_{i=1}^{m} x_i^3\right) + b\left(\sum_{i=1}^{m} x_i^2\right) + c\left(\sum_{i=1}^{m} x_i\right) = \sum_{i=1}^{m} x_i A_i \tag{2}$$

$$a\left(\sum_{i=1}^{m} x_i^4\right) + b\left(\sum_{i=1}^{m} x_i^3\right) + c\left(\sum_{i=1}^{m} x_i^2\right) = \sum_{i=1}^{m} x_i^2 A_i^2 \tag{3}$$

$$(1) : 60587354a + 2548390b + 109406c = 109359.91092040496 \tag{4}$$

$$(2) : 2548390a + 109406b + 4792c = 4791.115245163256 \tag{5}$$

$$(3) : 109406a + 4792b + 214c = 213.99914543249392 \tag{6}$$

$$a = -0.0001440063069586, b = 0.006340210104413, c = 0.931644918838$$

Codes

Settings

```python
# Number of entries to input
N = 831

# List of entries
entries = []

# Lowest rank considered
min_rank = 7

# Range of ages considered
min_age = 15
max_age = 35

# Range of years considered
min_year = 1968
max_year = 2019

# Names of competitions
competitions = ["Others", "FINA World Championships", "Olympic Games", "FINA Swimming World Cup", "FINA World Championships (25m)"]

# Indices of competitions considered
competitions_used = [1, 2]

# Coefficients of technological factor
T_coefficient_a = 0.004273722678214865
T_coefficient_b = -6.647541482426789

# Coefficients of age factor
A_coefficient_a = -0.0001440063069586
A_coefficient_b = 0.006340210104413
A_coefficient_c = 0.931644918838
```

Define functions

```python
# Calculates the technological factor of a year
def technological_factor(year):
  global T_coefficient_a, T_coefficient_b
  return T_coefficient_a * year + T_coefficient_b

# Calculates the age factor of an age
def age_factor(age):
  global A_coefficient_a, A_coefficient_b, A_coefficient_c, A_coefficient_d
  return A_coefficient_a * (age ** 2) + A_coefficient_b * age + A_coefficient_c

# Calculates the R squared value
def R_squared(y, f):
  n = len(y)
  y_mean = sum(y) / n
  S_y = sum((y[i] - y_mean) ** 2 for i in range(n))
  S_e = sum((y[i] - f[i]) ** 2 for i in range(n))
  return 1 - S_e / S_y
```

Define entry

```
class Entry():
  # Initialization
  def __init__(self, rank, name, age, year, time, competition_id):
    self.rank = int(rank)
    self.name = name
    self.age = int(age)
    self.year = int(year)
    self.time = float(time)
    self.speed = 100 / self.time
    self.competition_id = int(competition_id)

  # Describes self
  def describe(self):
    return(" ".join(list(map(str, [self.rank, self.name, self.age, self.year,
     ↪  self.time, self.speed, competitions[self.competition_id]]))))

  # Returns speed scaled with technological factor
  def speed_scaled_with_T(self):
    return self.speed / technological_factor(self.year)

  # Returns speed scaled with age factor
  def speed_scaled_with_A(self):
    return self.speed / age_factor(self.age)

  # Returns speed scaled with both the age factor and the technological factor
  def speed_scaled_with_AT(self):
    return self.speed / technological_factor(self.year) / age_factor(self.age)

  # Checks if this entry is within the desired range
  def validate(self):
    if self.rank > min_rank:
      return False
    if self.age < min_age or self.age > max_age:
      return False
    if self.year < min_year or self.year > max_year:
      return False
    if self.competition_id not in competitions_used:
      return False
    return True
```

Input entries

```
input_list = input().split()
entries = []
for i in range(N):
  rank, name, age, time, year, competition_id = input_list[i * 6 : (i + 1) * 6]
  entry = Entry(rank, name, age, year, time, competition_id)
  if entry.validate():
    entries.append(entry)
```

Finding systems of equations using the least square method

```
# Find an equation for T from dS/da = 0
def equation_for_T_from_da():
  values = { 'a': 0, 'b': 0, 'sum': 0 }
  for entry in entries:
    values['a'] += entry.year ** 2
    values['b'] += entry.year
    values['sum'] += entry.year * entry.speed
  return values

# Find an equation for T from dS/db = 0
def equation_for_T_from_db():
  values = { 'a': 0, 'b': 0, 'sum': 0 }
  for entry in entries:
    values['a'] += entry.year
    values['b'] += 1
    values['sum'] += entry.speed
  return values

# Find an equation for A from dS/da = 0
def equation_for_A_from_da():
  values = { 'a': 0, 'b': 0, 'c': 0, 'sum': 0 }
  for entry in entries:
    values['a'] += entry.age ** 4
    values['b'] += entry.age ** 3
    values['c'] += entry.age ** 2
    values['sum'] += entry.age ** 2 * entry.speed_scaled_with_T()
  return values

# Find an equation for A from dS/db = 0
def equation_for_A_from_db():
  values = { 'a': 0, 'b': 0, 'c': 0, 'sum': 0 }
  for entry in entries:
    values['a'] += entry.age ** 3
    values['b'] += entry.age ** 2
    values['c'] += entry.age
    values['sum'] += entry.age * entry.speed_scaled_with_T()
  return values

# Find an equation for A from dS/dc = 0
def equation_for_A_from_dc():
  values = { 'a': 0, 'b': 0, 'c': 0, 'sum': 0 }
  for entry in entries:
    values['a'] += entry.age ** 2
    values['b'] += entry.age ** 1
    values['c'] += 1
    values['sum'] += entry.speed_scaled_with_T()
  return values

# Output coefficients for system of equations
print("Technological factor:")
equation = equation_for_T_from_da()
print(equation['a'], 'a', '+', equation['b'], 'b', '=', equation['sum'])
equation = equation_for_T_from_db()
print(equation['a'], 'a', '+', equation['b'], 'b', '=', equation['sum'])
print()
print("Age factor:")
equation = equation_for_A_from_da()
print(equation['a'], 'a', '+', equation['b'], 'b', '+', equation['c'], 'c', '=',
↪   equation['sum'])
equation = equation_for_A_from_db()
print(equation['a'], 'a', '+', equation['b'], 'b', '+', equation['c'], 'c', '=',
↪   equation['sum'])
equation = equation_for_A_from_dc()
print(equation['a'], 'a', '+', equation['b'], 'b', '+', equation['c'], 'c', '=',
↪   equation['sum'])
```

Find R^2 values

```
# R squared value for technological factor
def T_R_squared(a, b):
  y, f = [], []
  for entry in entries:
    y.append(entry.speed)
    f.append(a * entry.year + b)
  return R_squared(y, f)

# R squared value for age factor
def A_R_squared(a, b, c):
  y, f = [], []
  for entry in entries:
    y.append(entry.speed_scaled_with_T())
    f.append(a * entry.age ** 2 + b * entry.age + c)
  return R_squared(y, f)

```

```
print("R squared value for technological factor:", T_R_squared(T_coefficient_a, T_coefficient_b)) # Result: 0.873891715613625
print("R squared value for age factor:", A_R_squared(A_coefficient_a, A_coefficient_b, A_coefficient_c)) # Result: 0.029520311716855565
```

Name	Gold	Silver	Bronze	Z_{Ach}	Scaled Speed
MichaelPHELPS	5	3	0	6.130854451	1.034631845
MarkSPITZ	1	1	0	1.04519471	1.033572397
IanCROCKER	2	3	0	3.270170847	1.031239858
MichaelGROSS	1	3	0	2.316609645	1.027753302
PabloMORALES	2	1	0	1.998755912	1.025599003
MattGRIBBLE	2	0	0	1.363048444	1.022032759
AnthonyNESTY	2	0	1	1.680902178	1.019786253
JoeBOTTOM	1	2	0	1.680902178	1.019572076
CaelebDRESSEL	2	0	0	1.363048444	1.015899184
LarsFROLANDER	2	2	0	2.634463379	1.01306605
MichaelKLIM	1	1	0	1.04519471	1.011050406
BruceROBERTSON	1	1	0	1.04519471	1.01089498
JosephSCHOOLING	1	0	2	1.04519471	1.006958
LaszloCSEH	0	3	0	1.363048444	1.006544012
ChadLE CLOS	2	1	1	2.316609645	1.005748155
RafalSZUKALA	1	1	0	1.04519471	1.003376296
GregJAGENBURG	1	1	0	1.04519471	1.002403712

Table B: Top 17 swimmers (for men's 100m butterfly)

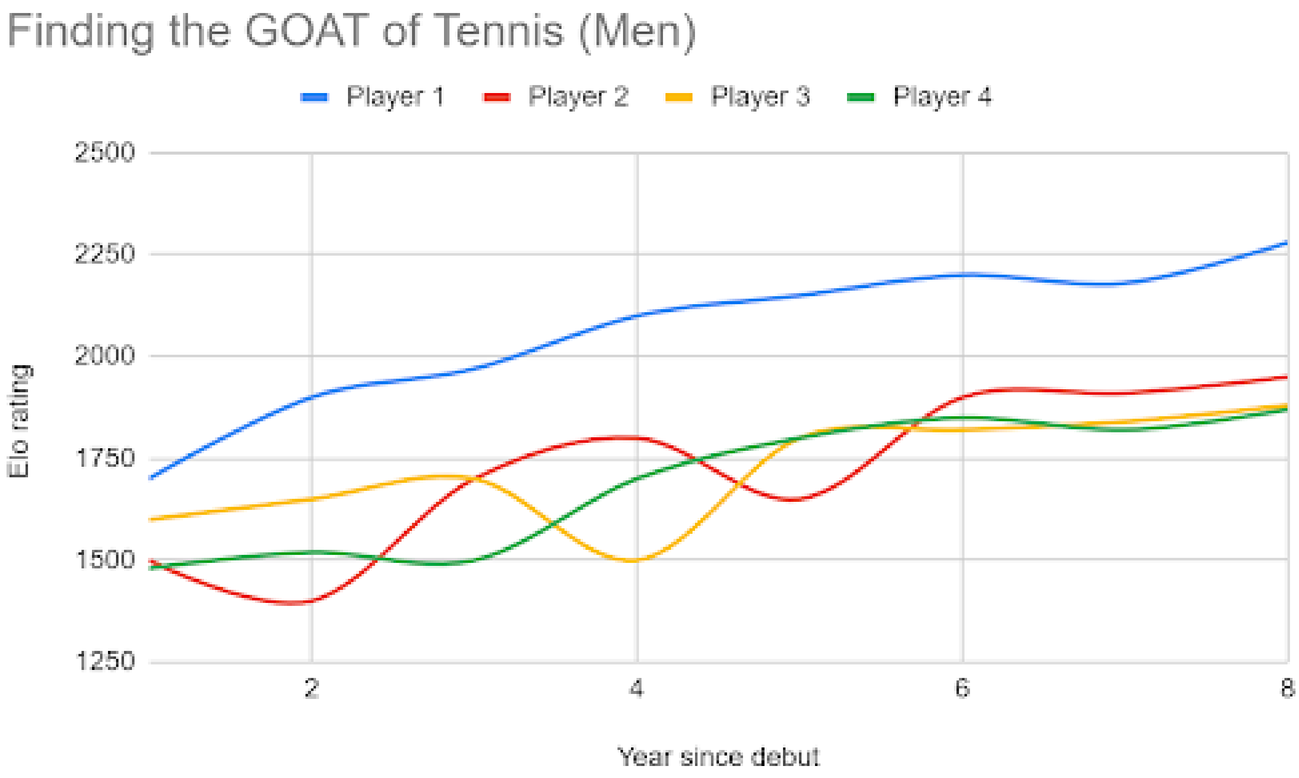

Figure C: Illustration for graphical analysis of G.O.A.T.