Medal Expectancy Project White Paper

Problem Statement

- How much can we expect the Monte Carlo output to change between runs due to the inherent variance of the simulation?
- What is the threshold that indicates a meaningful change in ME vs model variance – 0.5%? 1%? 5%?
- What is the optimal number of simulations to run during our Monte Carlo process to converge to the "true" medal expectancy?
- Which athletes experience the largest variance in ME from run to run?

Process

I started by making observations on individual sports, running sets of the pre-existing Monte Carlo simulations in order to identify the variance of each set of simulations, as well as the change in medal expectancy from run to run. Initially, I ran 10 iterations of 100, 1000, 3500, 5000, 7500, 10,000, 20,000, and 50,000 repetitions of the Monte Carlo simulations, and then, as I began to see a pattern in the results, narrowed it to 5 simulations of 100, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 5000, and 7500. Within each set of simulations, I calculated the standard deviation of the ME for the Monte Carlo process, as well as the runtime for each set of simulations. I also calculated the elbow of the standard deviation curve in order to mathematically identify the precise number of simulations at which the standard deviation began to level. For each sport, I compiled visualizations to represent these observations.

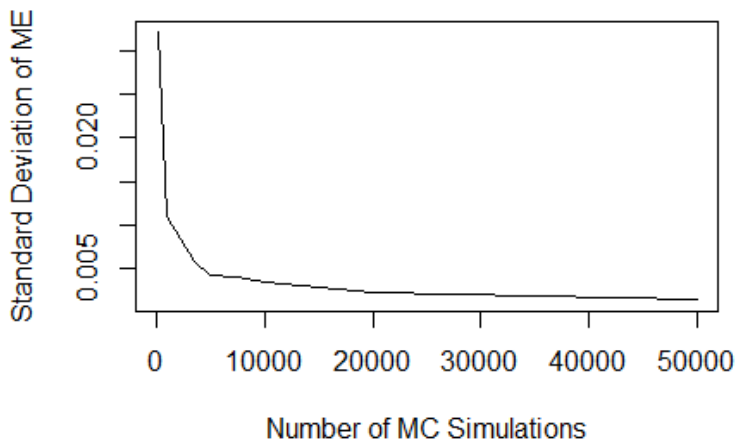
Observations

In general, there is, as one would expect, a decrease in variance as the number of simulations increases. However, at a given point, that curve levels out and becomes insignificant, as the graph below indicates. The find_curve_elbow function from the R pathviewr package, finds the "elbow" in bivariate data, enabling us to mathematically identify this point.

The theory of an elbow in a curve comes from machine learning and clustering, where Euclidean distance is used to calculate the distance between points to a centroid, thus creating clusters and identifying convergence in the data. Once plotted, these distances form an elbow, similar to the graph below. As the number of clusters increases, the Within-Cluster Sum of Square value decreases. At some point, the graph changes rapidly and creates that elbow shape. The K value corresponding to the "point" of the elbow is the optimal K-value, or number of clusters. For application to this project, I realized that the same method of identifying the point in the curve that changes rapidly could be used to identify where the reduction in variance becomes less significant as the number of simulations are increased.

Sport A: Standard Deviation of
Medal Expectancy for the 10 Athletes with the Highest

Medal Expectancy vs. Number of Monte Carlo Simulations



When calculating the elbow of the curve for each sport, I compared the average standard deviation across five runs per each number of simulations with the number of simulations.

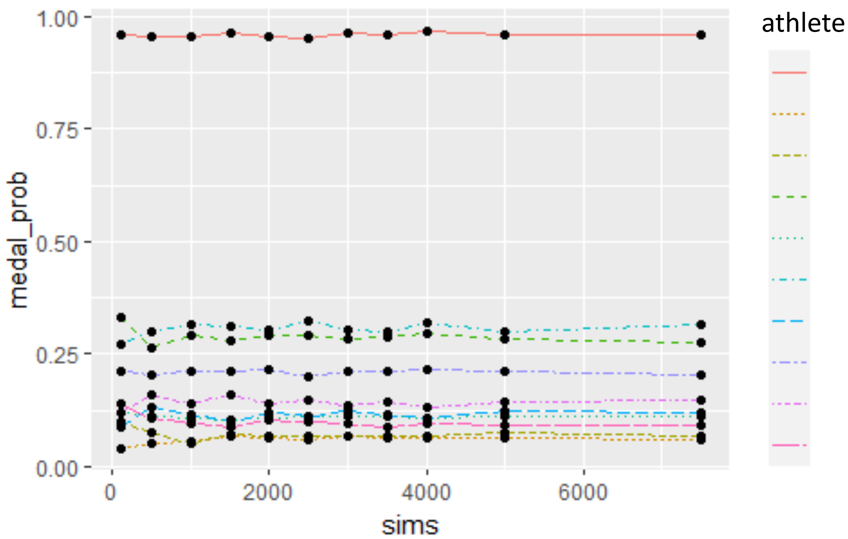
Sport		Optimal Number of Simulations as identified by the Elbow Curve	Resulting Standard Deviation of Athlete Medal Expectancy
Sport 1	MC	1500	0.0024
Sport 2	MC	1000	0.0025
Sport 3	MC	3500	0.0014
Sport 4	MC	2000	0.0015
Sport 5	MC	1000	0.0020
Sport 6	H2H	500	0.0022
Sport 7	H2H	1500	0.0018
Sport 8	H2H	1000	0.0042
Sport 9	H2H	1000	0.0015

Across the five observed multi-competitor sports, the elbow curve ranged from 1000 to 3500, with 1000 being the most frequent result. The standard deviation ranged from 0.0014 to 0.0025, with more simulations producing predictably lower variance in the probability of an athlete's medal expectancy. The precision of this result could be improved by testing more variations in number of simulations, however, from the sports and given number of simulations evaluated, the average elbow of the curve is approximately 1,800.

H2H sports showed a bit more variance between the sports, with the elbow of the curve at 500 for Sport 6 and 1,500 for Sport 7.

Worth considering is that a variance of 0.0025, when applied to an athlete's medal probability is less than a percentage increase or decrease in medal probability. The chart below helps to visualize this for the ten athletes with the highest probability of medal expectancy (and therefore, the ten athletes most affected by variance).

Sport B:
Number of Monte Carlo Simulations vs. Medal Probability for the 10 Athletes with the Highest Medal Expectancy



When testing the elbow curve for just the ten athletes with the highest medal expectancy, the results were similar, though the average standard deviation was slightly higher.

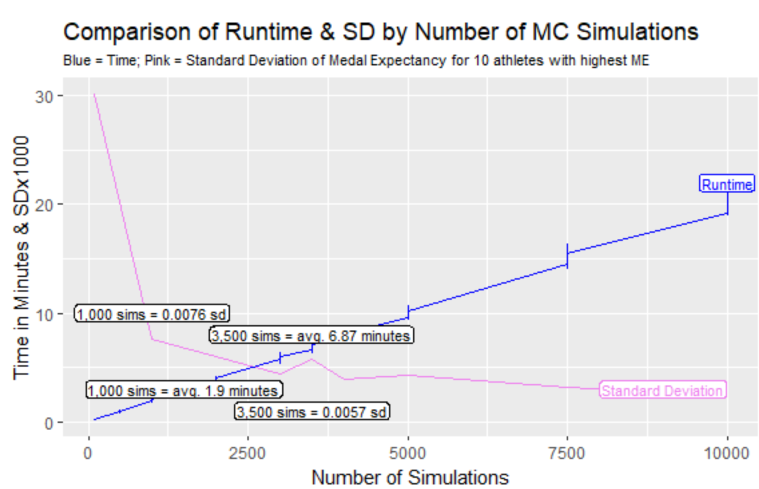
Sport		**Optimal Number of Simulations as identified by the Elbow Curve**	**Resulting Standard Deviation of Athlete Medal Expectancy**
Sport 1	MC	1500	0.0075
Sport 2	MC	1000	0.0094
Sport 3	MC	1000	0.0097
Sport 4	H2H	1000	0.011
Sport 5	H2H	2000	0.007
Sport 6	H2H	1000	0.007
Sport 7	H2H	1000	0.006

Notably, even though the elbow of the curve is similar, the resulting standard deviation is, on average, 3- to 4-times higher when considering just the ten athletes with the highest medal expectancy. For Men's Sport 4, at 1,000 simulations, the expected variance is just over .1% in the probability of medal expectancy for these athletes.

The runtime of the Monte Carlo process proved to increase linearly as the number of simulations increased. The exact runtime varied between sports, likely as a result of the number of teams, athletes, and competitions involved in any given sport.

Sport	**Average Runtime (in minutes) for 1000 simulations**
Sport 1	1.6
Sport 2	1.89
Sport 3	7.2
Sport 4	1.94
Sport 5	2.16
Sport 6	4.97
Sport 7	5.94
Sport 8	6.91
Sport 9	7.89

When comparing runtime with the average standard deviation for a given number of simulations, and considering that there are currently 950 sports to be considered, it is evident that the time cost of running more simulations than necessary becomes significant. In the graph below, which tracks runtime and standard deviation for Sport C, the minimal gains in accuracy (as defined here as a decrease in expected variance) between 1000 and 3500 simulations are noted.



Finally, in order to specify the improvement in accuracy as the number of simulations increases, as well as the cost of the increased runtime, I created the following tables for several of the sports. The first column lists the number of simulations, followed by the average standard deviation in medal probability and average runtime. The last two columns indicate the change from one row to the next in runtime and standard deviation as a percentage.

Sport C:
Change in Standard Deviation & Runtime

sd_graph.sims	sd_graph.sd	time2.avg	time_pct_change	sd_pct_change
100	0.0058444568	0.1983950	NA	NA
500	0.0028684654	0.9740938	390.98704	-50.919897
1000	0.0020267200	1.9444062	99.61181	-29.344798
2000	0.0014595839	3.9157528	101.38554	-27.982956
3000	0.0012166314	5.8809927	50.18805	-16.645326
3500	0.0011210153	6.8726664	16.86235	-7.859087
4000	0.0010240975	7.9333112	15.43280	-8.645535
5000	0.0009550440	9.9758752	25.74668	-6.742863
7500	0.0007518639	15.1139947	51.50545	-21.274426
10000	0.0006438269	20.1983019	33.63973	-14.369225

Notably, increasing the number of simulations from 1000 to 2000 gives a 27% improvement in standard deviation, while increasing the runtime by only about 2 minutes (which is admittedly 101%).

Sport D:
Change in Standard Deviation & Runtime

sd_graph.sims	sd_graph.sd	time2.avg	time_pct_change	sd_pct_change
100	0.0060554676	0.2167677	NA	NA
500	0.0030185963	1.0733117	395.14376	-50.150896
1000	0.0020379989	2.1613585	101.37287	-32.485212
1500	0.0017384412	3.2303307	49.45835	-14.698622
2000	0.0014176685	4.3408268	34.37716	-18.451739
2500	0.0013752584	5.4194689	24.84877	-2.991544
3000	0.0011502796	6.6061886	21.89734	-16.359014
3500	0.0010167828	7.6724277	16.14000	-11.605598
4000	0.0010923796	8.8056963	14.77066	7.434899
5000	0.0008793976	11.1013015	26.06955	-19.497068
7500	0.0008209096	16.6146722	49.66418	-6.650917

For Sport D, the results are similar. Increasing the number of simulations from 1000 to 2000 is a 35% improvement in standard deviation, with an average cost of just over 2 minutes.

Sport E:
Change in Standard Deviation & Runtime

sd_graph.sims	sd_graph.sd	time2.avg	time_pct_change	sd_pct_change
100	0.0082403332	0.1857524	NA	NA
500	0.0038861918	0.9381796	405.06988	-52.839385
1000	0.0025467534	1.8843335	100.84998	-34.466607
1500	0.0020716951	2.8133667	49.30301	-18.653484
2000	0.0017479230	3.8257107	35.98336	-15.628368
2500	0.0017137904	4.7565345	24.33074	-1.952753
3000	0.0014930984	5.7234022	20.32714	-12.877418
3500	0.0013130698	6.6401365	16.01730	-12.057384
4000	0.0013905168	7.4620218	12.37754	5.898158
5000	0.0012014582	9.4089902	26.09170	-13.596283
7500	0.0009868236	14.2128553	51.05612	-17.864506

For Sport E, there is a 37% improvement in the standard deviation between 1000 and 2000 simulations, with a time cost of approximately 2 minutes.

Sport F:
Change in Standard Deviation & Runtime

sd_graph.sims	sd_graph.sd	time2.avg	time_pct_change	sd_pct_change
100	0.0073196827	0.5988679	NA	NA
500	0.0035349586	3.0157997	403.58346	-51.706123
1000	0.0023601079	5.9437400	97.08670	-33.235204
1500	0.0018391132	8.7449936	47.12948	-22.075036
2000	0.0014023720	11.5294436	31.84050	-23.747380
2500	0.0012872031	14.1239504	22.50331	-8.212433
3000	0.0015222140	17.0811795	20.93769	18.257481
3500	0.0012352963	20.1414503	17.91604	-18.848708
4000	0.0012157552	23.3118592	15.74072	-1.581902
5000	0.0009125995	29.2904631	25.64619	-24.935584
7500	0.0006227363	43.2869914	47.78528	-31.762366

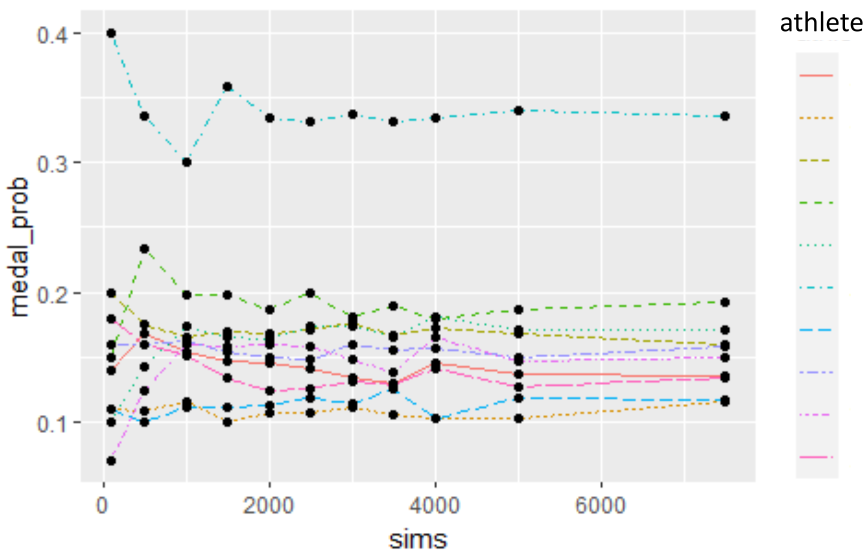
Sport F is even more significant: a 45% improvement in standard deviation between 1000 and 2000 simulations. The time required by the H2H Monte Carlo process is greater than that of the MC process, so though the time cost still increases linearly, it is greater, at just over 5 minutes.

Conclusions

Mathematically, the optimal number of simulations, as determined by the find_elbow_curve function, varies somewhat between sports. Given that the average elbow of the curve is approximately 1,800, and that the improvement in expected variance ranges from 27% to 35% between 1,000 and 2,000 simulations, my recommendation would be to run each sport's Monte Carlo simulation 2,000 times to optimize the results. The caveat to this is that the approximate runtime per sport is between 3-4 minutes, which, for 950 sports is approximately 63 hours: close to double the current estimated runtime (at 1,000 simulations).

If the time cost is considered reasonable for the resulting improvement in variance, the expected deviation in a given athlete's medal expectancy probability is, on average: 0.0015, which translates to a 0.015% increase or decrease in probability. By comparison, for 1000 simulations, the expected deviation is 0.0021, or a 0.22% increase or decrease in probability. The greatest variance in medal expectancy will be evident in athletes with the highest medal expectancy, in which case the shift from 1,000 to 2,000 simulations will make the greatest improvement in accuracy.

Sport G:
Number of Monte Carlo Simulations vs. Medal Probability for the 10 Athletes with the Highest Medal Expectancy



Here, it is evident that, for the athletes with the highest medal expectancy, the probability of medal expectancy can shift as much as 10%, depending on the number of simulations. Notably, at 2,000 simulations, the variance stabilizes significantly. (Note: this particular graph, of Sport G, showed the greatest variance in results of the sports examined: at 1,000 simulations, the average standard deviation is 0.0094; at 2,000 simulations, it decreases to 0.0066).

For all athletes, including those with the highest medal expectancy, assuming the Monte Carlo process is run 2,000 times, a .5 to 1% change in the probability of an athlete's medal expectancy can be considered significant.

To test this, I performed a hypothesis test using the mean standard deviation of five runs of 2,000 simulations for 32 sports (both H2H & MC). The null hypothesis was that the mean of the standard deviations is 0.005 or greater; the alternative hypothesis is that the mean of the standard deviation is less than 0.005. The resulting p-value is 1.939e-06, which is less than the level of significance of 0.05. Therefore, we reject the null hypothesis and conclude that a standard deviation of 0.005 or greater can be considered significant.

```
        One Sample t-test

data:  sport_data_comp$sd.all
t = -14.715, df = 31, p-value = 7.823e-16
alternative hypothesis: true mean is less than 0.005
95 percent confidence interval:
        -Inf 0.002304907
sample estimates:
  mean of x
0.001953939
```

If a given athlete has an average medal expectancy probability of 1.0 (or 100%), and there is 0.005 variation from that, that athlete's probability of getting a medal at the next Olympics could vary from .95 to 1.05. Because the variance affects athletes with a higher medal probability more than those with a lower medal expectancy, it is important to realize that the athletes with the highest medal probability are also the athletes where we can expect to see a variance closer to .5% between runs. To illustrate this, I performed the same hypothesis test, but with just the data from the 10 athletes with the highest medal expectancy per sport.

```
        One Sample t-test

data:  sport_data_comp$sd.top10
t = 9.7866, df = 31, p-value = 1
alternative hypothesis: true mean is less than 0.005
95 percent confidence interval:
        -Inf 0.007986178
sample estimates:
  mean of x
0.007545222
```

For the athletes with the highest medal expectancy, the set-up of the hypothesis test was the same: the null hypothesis was that the mean of the standard deviations is 0.005 or greater; the alternative hypothesis is that the mean of the standard deviation is less than 0.005. The resulting p-value is 1, which is greater than the level of significance of 0.05. Therefore, we fail to reject the null hypothesis, which does not automatically mean that the null hypothesis is true, only that the data does not prove it to be false. This provides some indirect confirmation of the theory that the variance will be greater among these athletes. Changing the value of the hypothesized mean to 0.01 results in a p-value of 6.248e-11, which would mean rejecting the null hypothesis, and concluding that, for the athletes with the highest medal expectancy, a standard deviation of 0.01 or greater can be considered significant. If a given athlete has an average medal expectancy probability of 1.0 (or 100%), and there is 0.01 variation from that, that athlete's probability of getting a medal at the next Olympics could vary from .99 to 1.01—which is a 1% change.

```
        One Sample t-test

data:  sport_data_comp$sd.top10
t = -9.4389, df = 31, p-value = 6.248e-11
alternative hypothesis: true mean is less than 0.01
95 percent confidence interval:
        -Inf 0.007986178
sample estimates:
  mean of x
0.007545222
```

For the athletes with the highest medal expectancy, a 1% or greater change in medal expectancy can be considered significant.

Out of curiosity, from the 32 sports that I evaluated, I selected the athlete with the highest standard deviation across five runs of 2000 simulations in each sport, then sorted based on that (the sd_highest_sd column below). The _highest_sd suffix on the last four columns indicates that these are columns relating to that athlete (within the sport) that was identified as having the highest standard deviation. The .all and .top10 suffixes indicate the standard deviation and average medal probability for all athletes within the sport and the 10 athletes with the highest medal expectancy respectively.

The athletes in the athlete_team_highest_sd column are the athletes in those sports who have the most variance, and their corresponding average medal probability across the five runs of 2000 simulations is the avg_medal_prob_highest_sd column on the far right. Here, it is notable that the highest variance does not seem to correspond to the highest medal probability as predicted.

10 Athletes with the Highest Standard Deviation

sport	sims	sd.all	avg_medal_prob.all	sd.top10	avg_medal_prob.top10	athlete_team_highest_sd	country_highest_sd	sd_highest_sd	avg_medal_prob_highest_sd
Sport 1	2000	0.0009901050	0.019480519	0.008430037	0.23983	Athlete / Team 1	CAN	0.019049278	0.019480519
Sport 2	2000	0.0022475097	0.038961039	0.009555360	0.24632	Athlete / Team 2	GER	0.018045082	0.038961039
Sport 3	2000	0.0037958700	0.130434783	0.008396800	0.29924	Athlete / Team 3	CRO	0.017732033	0.130434783
Sport 4	2000	0.0016189759	0.043478261	0.009331968	0.29098	Athlete / Team 4	ESP	0.017592612	0.043478261
Sport 5	2000	0.0025339413	0.090909091	0.007776539	0.29079	Athlete / Team 5	GBR	0.016517415	0.090909091
Sport 6	2000	0.0052313536	0.114285714	0.009994469	0.29296	Athlete / Team 6	CUB	0.016507574	0.114285714
Sport 7	2000	0.0006190707	0.014218009	0.007742303	0.27915	Athlete / Team 7	USA	0.016328656	0.014218009
Sport 8	2000	0.0011057652	0.022058824	0.008108378	0.24185	Athlete / Team 8	POL	0.015809016	0.022058824
Sport 9	2000	0.0005592077	0.013761468	0.007414618	0.28237	Athlete / Team 9	CHN	0.015610093	0.013761468
Sport 10	2000	0.0012541462	0.020000000	0.010359893	0.21369	Athlete / Team 10	ESP	0.015182226	0.020000000

By comparison, I also sorted the sports by their average standard deviation—across the athletes with the highest medal probability (the sd.top10 column below). For the ten sports with the highest variance among those athletes with the highest medal probability, those ten athletes seem to have an average medal probability of ~.2 or higher (avg_medal_prob.top10).

10 Sports with the highest Standard Deviation among athletes with the highest Medal Expectancy

sport	sims	sd.all	avg_medal_prob.all	sd.top10	avg_medal_prob.top10	athlete_team_highest_sd	country_highest_sd	sd_highest_sd	avg_medal_prob_highest_sd
Sport 1	2000	0.0012541462	0.020000000	0.010359893	0.21369	Athlete / Team 1	ESP	0.015182226	0.020000000
Sport 2	2000	0.0052313536	0.114285714	0.009994469	0.29296	Athlete / Team 2	CUB	0.016507574	0.114285714
Sport 3	2000	0.0022475097	0.038961039	0.009555360	0.24632	Athlete / Team 3	GER	0.018045082	0.038961039
Sport 4	2000	0.0016189759	0.043478261	0.009331968	0.29098	Athlete / Team 4	ESP	0.017592612	0.043478261
Sport 5	2000	0.0017648159	0.052631579	0.009038772	0.29606	Athlete / Team 5	ARG	0.014855134	0.052631579
Sport 6	2000	0.0014361557	0.015625000	0.008830693	0.19214	Athlete / Team 6	LTU	0.014332655	0.015625000
Sport 7	2000	0.0007832864	0.010169492	0.008752627	0.21969	Athlete / Team 7	ESP	0.014791890	0.010169492
Sport 8	2000	0.0028623662	0.088235294	0.008572936	0.28822	Athlete / Team 8	ARG	0.014549055	0.088235294
Sport 9	2000	0.0009901050	0.019480519	0.008430037	0.23983	Athlete / Team 9	CAN	0.019049278	0.019480519
Sport 10	2000	0.0037958700	0.130434783	0.008396800	0.29924	Athlete / Team 10	CRO	0.017732033	0.130434783

Some of the sports with the highest variance among these ten athletes are the same as those with the individual athletes with the highest variance, but it is not the exact same list. Some sports do overlap, lending the question of whether there are sport-specific factors that could potentially influence the variance in medal expectancy predictions. Without extensive knowledge of each sport and the number of competitions and/or rankings used to compile the medal expectancy predictions, I do not have any well-founded conclusions, but could speculate that the frequency and variance of the data itself may play a role in the variance in the medal expectancy predictions.

Project Files:

D:\MedalExpectancy-Combined\Monte Carlo Analysis - WIST Fellow 2023

- Medal Expectancy Project White Paper – report of the project
- Medal Expectancy Project Presentation – PowerPoint presentation summarizing the project
- SportData – compilation of 32 sports at 2000 simulations
- Compiled Sport Data for Analysis – csv files with the resultdata from the simulations for 32 sports
- Project R Files – the R script files used for analysis
    - H2HMonteCarloWithoutRoundRobin – AbiWC.R – for running loops of Monte Carlo process for a given H2H sport without Round Robin
    - H2HMonteCarloWithRoundRobin – AbiWC.R – for running loops of Monte Carlo process for a given H2H sport with Round Robin
    - MCMonteCarlo_Abi_WC.R – for running loops of Monte Carlo process for multi-competitor sports
    - SimulationAnalysis.R – the code I used for comparing various sets of simulations, runtime, and standard deviation
    - sd_test.R – the code for compiling data from various Monte Carlo sport simulations into one data frame and performing a hypothesis test on the expected variance with that number of simulations