Masters of Negotiated Studies: Negotiated Project 2 (20 Credit)

How can personal skill ratings in First-Person Shooter games be accurately tracked and scored to reflect player mechanical skill?

Christopher Boyce

Student Number: 19016871
Email: b0196871j@student.staffs.ac.uk

Supervisor: Shaun Reeves

Contents

Glossary

Rating System: A method used to measure and compare the skills or performance levels of players within a game based on various metrics.

Data Normalization: The process of adjusting data measurements to fit on a common scale.

Normal Distribution: A statistical distribution used to represent real-valued random variables whose distributions are not known.

True Player Performance: An ideal or theoretical value that represents the most accurate assessment of a player's ability, which rating systems aim to estimate.

Leetify: A third-party service that analyses gameplay to provide statistics and improvement suggestions.

Key Words

C++, Unreal Engine, Skill Rating System, Games Development, Mechanical Skill Rating System, AI

Section 1 : Introduction

1.1 : Aims

Investigate and develop a rating system that records and rates the mechanical skills of players and produces an easily understandable rating system that follows commonly understood rating standards.

1.2 : Objective

- Investigate current systems for individual player rating systems in popular First-Person Shooter games and learn the fundamental statistics that are tracked.
- Develop an artifact that tests a suitable population and creates data that can be recorded and evaluated.
- Produce an algorithm that will rate the population on an easily understandable system.
- Analysis of recorded data from the artifact and compare it to underlying fundamentals of other rating systems.

1.2 : Background Information

The main objective of a skill rating system is to gather the player's performance and display it in an understandable format that the user feels is fair. Due to many influences, such as dynamically changing performance and constantly differing opposition performance in games, it is not possible to perfectly calculate the rating from a single result. Rating systems need to estimate or try to get as close as possible to the “True Player Performance.”

This paper is based on the research recommendations of a previous paper into skill rating systems (Appendix A). The main implication of this paper was that the more data points you can record of a player's personal performance, the more accurately you can rate their performance in a match, thereby rating them closer to their “True Player Rating.” This paper will examine the mechanical skills of players and use their performance from a training course to score them on several statistics.

Mechanical skill in First Person Shooter (FPS) games is a primary skill. This ability to aim, track, and control weapons could arguably be the most important skill set for determining skill level and personal performance in a match. For this project, the paper focuses solely on this metric, as the artifact has no other gameplay elements, but in a majority of FPS games, more complex game mechanics and play styles can affect these statistics.

This study will investigate solutions for rating players in an academic and open-source field, as many of the current rating systems are subject to copyright or are kept secret from the gaming community, to stop them becoming susceptible to exploitation.

This paper will first investigate the current skill rating systems and the statistics they track, as well as third-party systems for games that allow players to more accurately investigate the performance of each match. It will then be followed by a section on the methodology explaining how this paper's artifact and rating systems are going to work, as well as a detailed look into the creation of bots that will be used to create the data pool. Finally, an analysis of the data will be conducted to assess the viability of the data set and the accuracy of the data to determine if the objectives are met.

Section 2 : Literature Review

2.1 : Current Systems

As discussed in Appendix A, early skill systems such as Elo (Elo, 1978), and Glicko (Glickman, 1998) utilised the win or loss in the game and the opponent's current skill rating to determine the increase and decrease in skill rating. This proved effective and can therefore be applied to any 1v1 game while taking team averages as the single data point. When first designing this system, they were not made for video games but could be adopted but the performance of the player in the game had no result of the “Rating” gain from the match. This is something that players may feel is not fair to them, for example having a close match or playing above average the player may feel like they should lose or gain more “Skill Rating” due to their performance. This was investigated by the Microsoft team and was a key pillar in TrueSkill’s 2 development with them basing one system of “Expected Contribution” and their output.

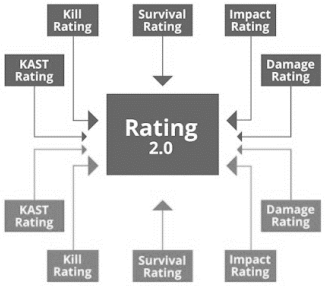
Microsoft TrueSkill was one of the first systems used in gaming that considered team performance and this gave it an increase in accuracy and therefore could balance the game depending on the team’s average performance. This then was improved with TrueSkill 2 which took user-specific statistics from the game including Kills to Death, assists, objective score, damage dealt, headshot percentage, damage taken, and further metrics (Minka, et al., 2018). These metrics were weighted and then relayed into the system. These key improvements' weightings are not known and can be changed from game to game depending on the importance of each.

This system of tracking the data is the key to this paper and is something that was emulated in “Percentile Skill” created in Appendix A which started to collect the player's data with a single entry of Kills to Deaths to either increase or decrease the skill rating gained. This showed significant improvement in finding the “True Player Skill Rating” that was expected. This conclusion kickstarted this project and the tracking of more data elements therefore potentially gathering more accurate performance metrics that could then be used in a system.

2.2 : Third Party Systems

Third party system such as HLTV Rating 2.0, Leetify or Blitz.gg and now can track these statistics using match demos and can give several forms of data points as well as giving a personal performance rating. These systems allow players in depth knowledge of the performance of each player and can summarised the game to a single statistic.

2.21 : HLTV Rating 2.0



The HLTV rating standard is used to evaluate a single match and the players' performance. This value is normally between 0.5 and 1.5, depending on the performance, with higher being better. The rating standard is used in all professional games of Counter-Strike (Valve, 1999) to determine the performance of individuals and therefore has gained merit in the community due to its ability to provide more detailed data about the game than mere Kills to Deaths. The HLTV standard uses five data points for calculation (HLTV, 2017).

KAST stands for Kills, Assists, Survival, and Trades. This is a percentage of the rounds where one of these actions occurs. This metric benefits holistic performance (Jay, 2021), not just the player with the most kills; it also benefits the team play of the player as it assesses whether they were playing for the round instead of playing the game as an individual.

Kill Rating doesn't directly relate to kills to death; instead, it looks at Kills per Round and average damage per round to gauge the direct impact they had on the round.

Survival rating examines the percentage of rounds they survive, which has a direct correlation with whether they have been playing in the later stages of the round, as well as if they have won the round and the impact they had on it.

Impact rating is a score that looks at the opening kills, multi-kills, and clutch plays per round. This score benefits rounds where certain kills are crucial, such as an entry kill that helps complete the objective of the game or a clutch that potentially swings the round in a 1 vs. X situation (Holland, 2020).

Finally, damage rating shows the total damage per round that a player has inflicted, indicating how much they individually contributed to the round and therefore were more impactful.

This system primarily focuses on individual prowess and includes some elements of team play but also comes with some caveats and bias (Sardegna, 2017). Looking at this reputable and widely used system, it is important to note key statistics that are being tracked and could be used in the system being developed. While they may not perfectly translate into a rating system, they provide a foundation for identifying what is important in a game.

2.22 : Leetify

Leetify (Leetify, 2019) is a 3rd party tracker for matches played in Counter-Strike (Valve, 1999). This software analyses the demo replay system in the game and uses an API to extract data and create statistics in the form of tables and reports. This software rates your ability in several statistics from 0 – 100 for aim, but it also creates a personal performance rating and tracks your average performance, comparing it to your previous 30 games. In the “Aim” section, it is broken down into digestible metrics (Appendix B).

The first is spotted accuracy; this metric track when you can see a player and how many shots hit the target. This can show how well the player is controlling the guns and placing their crosshair. Obviously, the higher the score, the better, as you are hitting your shots (Rogers, et al., 2024). It is important to note that they don’t count if there isn’t an enemy visible, as the player may be shooting through penetration spots or smokes to gain an advantage. This statistic helps measure the spray control as lower-skilled players are less likely to perform this correctly and, therefore, miss.

The next metric is time to damage; this metric measures the time in milliseconds from when a player can visibly see an enemy to when they first inflict damage. This indicates how quickly the player reacts and then can perform the movement to the target and hit a shot. The lower the time, the better the player is, in theory, at reacting and acting on the stimulus (Rogers, et al., 2024).

Crosshair placement is tracked using the degrees from where the enemy is first spotted to when the player first hits the enemy. This demonstrates an ability to align the crosshair with angles the enemies are likely to peek or push. This shows game sense as well as awareness; therefore, the lower the degrees' offset, the better the player thinks about the game and acts with this knowledge.

Both headshot accuracy and headshot kills are tracked to differentiate between high-skill players who aim for the head instead of the centre of mass. This shows the nuance in the player's aiming ability. The player's headshot percentage generally correlates with a higher skill bracket due to the difficulty of aiming at smaller areas as well as the ability to control the gun's recoil.

Spray accuracy is the percentage of bullets that hit the enemy target when the player fires the weapon. This metric represents the player's ability to control the recoil and demonstrates mechanical ability when aiming and firing.

Counter-strafing is a unique part of Counter-Strike (Valve, 1999) with the ability to counter inaccuracies if moving in certain patterns. This movement pattern is when the player hits the opposite key to which they're moving before firing; this allows for a precise bullet. With games featuring movement inaccuracy, it would be important to measure how often the player moves and shoots, as typically lower-skilled players have not learned this advanced movement technique or cannot utilize it effectively.

Finally, the overall accuracy is tracked; this shows how many bullets that are fired hit the enemy, demonstrating the player's definitive ability to hit the target. The accuracy is important to track because it is an overall statistic of all bullets hit. Accuracy can vary greatly depending on the playstyle of the player; for example, a high-level player might spray through smokes, whereas a player in a supportive position or sniping role may have better accuracy but fire fewer shots. This is important to track because all other statistics can be divided by the accuracy, thereby normalizing the data between players.

Section 3 : Methodology

3.1 : Gun systems

This artifact is going to develop a first-person shooter game that will test the players' mechanical abilities by running a training course with targets that mimic other players in an FPS game. This first-person game will use gun recoil systems similar to many popular competitive first-person shooter games such as VALORANT (Riot Games, 2020), Counter-Strike (Valve, 1999), and Tom Clancy's Rainbow Six Siege (Ubisoft, 2015). This recoil system will push each bullet out of the gun upwards, and the player will have to compensate by pulling down to the left and right. These patterns will be consistent every time the player shoots, therefore learning these and being able to mechanically control them with the input of the mouse will be an advantage.

The targets that the player will be shooting at will have different hitboxes depending on the body position, with headshots dealing the highest damage, body shots dealing a medium amount of damage, and arms and legs dealing the least amount of damage; this emulates many first-person shooter games. The player will also have inaccuracies while running and walking to emulate first-person games; when the player stops, the guns will instantly become accurate, therefore teaching the player a core movement mechanic of the game.

3.2 : AI Systems

Due to the large dataset needed for this experiment, AI bots are going to be created to complete the course. These AI bots will have several variables to try and emulate human reaction times and aiming accuracy. A total of 100 unique bots will range across five main performance bands: terrible, bad, average, good, and great. The bots' statistics will be normally distributed; these include a reaction time from when they see an enemy to when they initiate shooting and a separate reaction time from when they see an enemy to when they stop walking. To control recoil, there will be a curve that represents their ability to control recoil, and another curve that represents the variability in their shots. Each bot will have a headshot percentage to calculate how often they aim for the head. Additionally, they will have an integer that

represents the number of bullets they will miss before resetting their aim, and a float representing the speed at which they recalibrate their aim.

3.3 : Data Collection

Several metrics are tracked for each bot in the study. The first metric is total accuracy, calculated by dividing the total shots hit by the total shots fired.

The next metric is headshot percentage, calculated by dividing the shots that hit the head by the total shots hit. Additionally, the percentage of time each bot is moving while firing is monitored, which involves recording the bot's movement speed during shooting activities.

Another significant metric is the average time from sighting to kill. Each time an enemy is spotted, the time between spotting the enemy and killing that enemy is tracked and then averaged throughout the entire session. Similarly, the time to damage measures the time from when the bot first sees an enemy to when it first inflicts damage; these times are averaged at the end of the course. Finally, the time of how long it takes to inflict lethal damage on an enemy; these times are also averaged at the end.

All these data points are then normalized to a value between zero and one and then added to calculate a total score. This total score is then passed to the scoreboard system, which then uses a Gaussian curve calculation that will get the mean, variance, Z-Score, and standard deviation and recalculate all bots' scores, making the system flexible depending on the pool of data. This will then calculate the percentile that the bot is placed in, and finally, this score will be linearly interpolated between -10 and 10 to give a value that represents the mechanical skill.

Section 4 : Analysis

4.1 : First Test Analysis

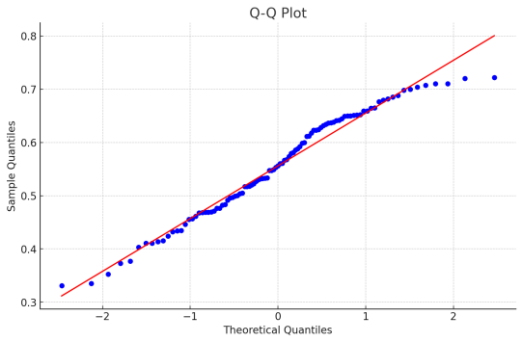


Figure 1 : First Run Q-Q Plot

Quantile-to-quantile plot graphs show if a dataset follows a particular distribution; in this case, it is used to help how the normal distribution. The theoretical perfect normal distribution is shown using the red line, and the blue dots represent the actual quantities from the sample data. This means the higher the deviation from the red line, the more the data isn't normally distributed. The data from the first run have an extreme left deviation, showing that there are fewer data points at the lower end of the scale, and there is an extreme right deviation showing more data points on the right than there would be in a normal distribution. Overall, the data shows similarities to a normal distribution but has clear signs of deviation.

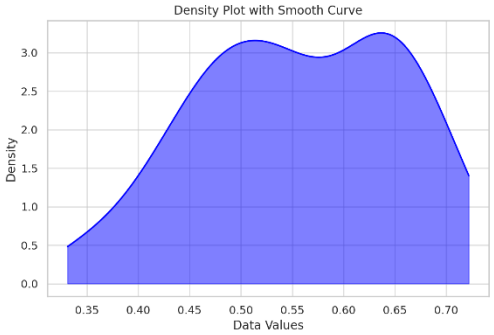


Figure 2 : First Run Histogram

This is a histogram connected with curves. This helps to see if the data is normalized, with the mass of data being in the middle and tails at each end. The data shows a peak in the middle, suggesting that the most frequent data is concentrated around the middle but has two main peaks. The data shows a spread from 0.35 to 0.7, with the bulk of the data entering at 0.5 to 0.65. The curve being smooth on each side helps show that the data follows normalized patterns but isn't directly normally distributed. There isn't a high skewness to the data; this shows the median and mean are close together, which is a characteristic of a normal distribution.

4.2 : Second Test Analysis

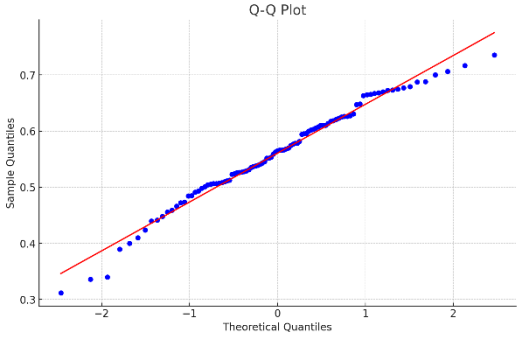


Figure 3 : Second Run Q-Q Plot

The second Quantile-to-quantile plot shows a closer alignment with the theoretical line. This suggests that the data is more likely to follow the normal distribution. The tails do not deviate as much from the previous Q-Q plot, therefore showing potentially less skewness or fewer outliers. There is a clear set of bands with a jump between each one; this could be caused

by the different recoil curve lines, each having a slight jump in performance, therefore causing these jumps.

The data this time only has one peak showing that the data is unimodal. The data has a slight skewness being above 0.5 showing the mean of the data will be on the higher of the middle. The data has a clear middle and curve flowing downwards to show good normality of the data. The spread this time falls between 0.45 and 0.6 slightly differing from the first test but the symmetric shape of the curves shows strong results.

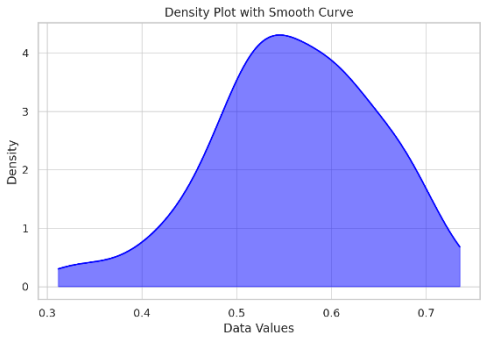


Figure 4 : Second Run Histogram

4.3 : Third Test Analysis

This Quantile-to-Quantile plot similarly shows a general trend that is close to the expected, but this graph has the biggest deviation of the three graphs. The graph also has a heavy right tail and a smaller but noticeable left tail, showing the data can be skewed. There is a high central tendency that helps prove the data follows a normal curve.

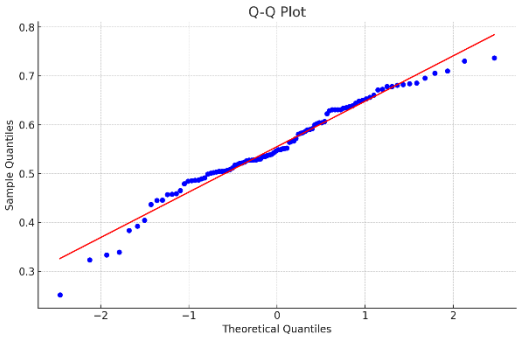


Figure 5 : Third Run Q-Q Plot

The shape of the density plot histogram shows a non-symmetric curve with a sharp peak and an “S” shaped gradual decline on the right side. This data is shown to be the least normally distributed of the three datasets, with the sharpest peak and high skewness.

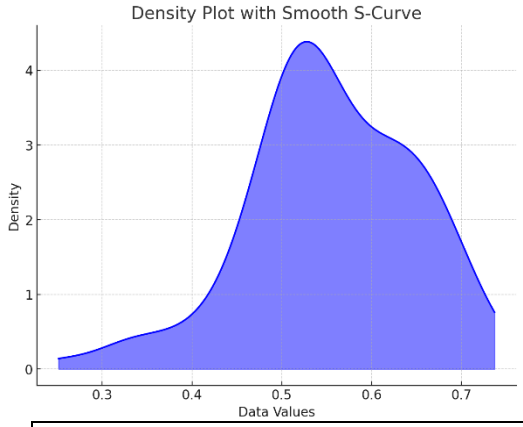


Figure 6 : Third Run Histogram

4.4 : Final Rating in Game Rating

This final graph shows the in-game rating assigned to each bot; the final rating of each bot is represented by a point. As observed, exactly 50 percent of bots are above average, which is indicative of the system rating the bots based on a mean and adjusting when new data is added. This could also be evidence of a zero-sum game, which, when combined, results in a total of -0.516, signifying a well-designed system.

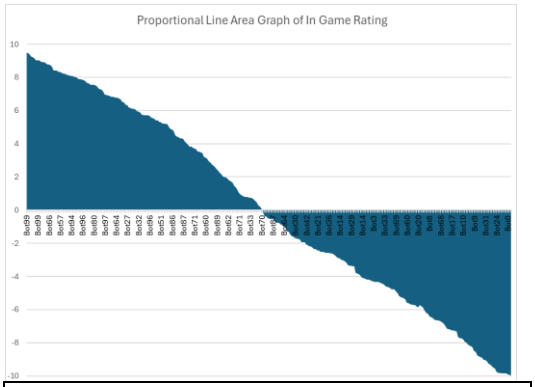


Figure 7 : Final Rating Line Area Graph

Section 5 : Results

5.1 : Data Results

Using the Quantile-to-Quantile graphs and the curve histograms, it shows a close-to-normal distribution two-thirds of the time, which indicates that the system is starting to work. The issues arise with the shape and skewness of the curves, which show a general skew to the right, indicating that more bots are classed as higher-skilled. This is highly likely due to the way the bots are programmed, and the main contributor to this is the recoil pattern negation solution, which is shown in Figure 3 where there is a clear jump between sections of bots. This is consistent due to the five levels of recoil control implemented into the solution. Also, because each run only considers one run from the bot, a fluke that can cause overperformance or underperformance for a bot can influence the data. Despite these downsides, it is still possible to conclude that the data is closely normalized and is representative of a real population of players.

The score values are fed to the rating system when in-game, it perfectly splits the population in the middle creating an exact average (Figure 7), therefore assigning the correct rating to the mechanical abilities of a bot. This shows a successful implementation that can rate mechanical ability against live data changes. If this were implemented into a live game with human test subjects, it would be able to reproduce these results, giving each one a percentile score and finally a rating. Visually inspecting the raw data also gives credence to the data as many of the higher skill or “Pro” level bots are at the top of the data and the spread throughout is consistent with what is to be expected. This is also seen in Figure 7 with bot 99 being the top performer and bot 0 being at the bottom. This was implemented into the design to make the data easier to analyse.

Section 6 : Conclusion

This study has systematically explored the development of a novel skill rating system for First-Person Shooter (FPS) games, focusing specifically on mechanical skills. The findings confirm by recording statistically important metrics such as headshot percentage, reaction times and accuracy it is possible to create a nuanced and accurate assessment of the players mechanical skill.

However, the study is not without limitations. The reliance on AI-generated data, while useful in controlled testing environments, may not perfectly mimic real-world player behaviour. Future research should aim to validate these findings with human participants or already collected data set.

In conclusion, this research contributes to a better understanding of skill assessment in FPS games and lays the groundwork for more sophisticated and fairer, skill rating system.

Bibliography

Elo, A., 1978. *Elo: The Rating of Chessplayers, Past and Present*. 1st Edition ed. New York: Arco Publishing.

Glickman, M. E., 1998. The Glicko system. *Applied Statistics,* Volume Volume 48, pp. 377-394.

HLTV, 2017. *INTRODUCING RATING 2.0*. [Online]
Available at: https://www.hltv.org/news/20695/introducing-rating-20
[Accessed 09 05 2023].

Holland, P., 2020. *csgo-impact-rating*. [Online]
Available at: https://github.com/phil-holland/csgo-impact-rating
[Accessed 18 04 2024].

Jay, M., 2021. EVALUATING PLAYER IMPACT THROUGH KAST. p. 3.

Leetify, 2019. *Leetify Dashboard*. [Online]
Available at: https://leetify.com/app
[Accessed 18 04 2024].

Minka, T., Cleven, R. & Zaykov, Y., 2018. *TrueSkill 2: An improved Bayesian skill rating system*. [Online]
Available at: https://www.microsoft.com/en-us/research/uploads/prod/2018/03/trueskill2.pdf
[Accessed 11 01 2023].

Riot Games, 2020. *Valorant*. [Online]
Available at: https://playvalorant.com/en-gb/
[Accessed 19 04 2024].

Rogers, E. J. et al., 2024. *KovaaK's aim trainer as a reliable metrics platform for assessing shooting proficiency in esports players: a pilot study*. [Online]
Available at: https://www.frontiersin.org/articles/10.3389/fspor.2024.1309991/full
[Accessed 19 04 2024].

Sardegna, C., 2017. *Exploring Problems with Counter-Strike Rating Systems*. [Online]
Available at: https://chrissardegna.com/blog/problems-with-csgo-rating-systems/
[Accessed 09 05 2023].

Ubisoft, 2015. *Tom Clancy Rainbow Six Siege*. [Online]
Available at: https://www.ubisoft.com/en-gb/game/rainbow-six/siege
[Accessed 19 04 2024].

Valve, 1999. *Counter Strike*. [Online]
Available at: https://www.counter-strike.net/
[Accessed 18 04 2024].

Appendices

Appendix A : Skill Rating Systems Paper

Appendix B : Screenshots of Leetify

My Team WIN	Rating	Personal Performance	HLTV Rating	K/D	ADR	Aim	Utility
BIRZHAN	+15.68 ★	+8.12 ★	1.88 ★	2.13 ★	140 ★	92 ★	48
Dys	+7.20	+4.53	1.62	1.77	109	83	51
Fast Lose	-4.29	-5.04	1.02	0.95	77	50	63
Jake	-5.26	-5.25	0.70	0.71	66	30	49
angetsu	-10.64	-10.14	0.49	0.38	33	70	46

Enemy Team LOSS	Rating	Personal Performance	HLTV Rating	K/D	ADR	Aim	Utility
kaspeR	+3.58	+3.35	1.09	1.06	85	52	69 ★
★aimStar	+2.71	+0.86	1.27	1.06	85	71	45
kroq	-0.78	-0.26	0.88	0.85	78	91	48
Meister	-1.69	-2.05	0.97	0.95	92	57	44
Rolnik Sztachetka	-7.13	-5.06	0.42	0.37	36	50	58

My Team WIN	Spotted Accuracy	Time to Damage	Cross. Placement	Head Accuracy	HS Kill %	Spray Accuracy	Counter-Strafing	Accuracy (All)
angetsu	32%	844ms	7.63°	29%	83%	44%	100%	16%
BIRZHAN	55%	570ms	4.96°	30%	59%	50%	86%	26%
Dys	45%	461ms	7.21°	23%	61%	60%	70%	16%
Fast Lose	29%	641ms	7.47°	12%	28%	36%	78%	10%
Jake	19%	500ms	10.67°	22%	67%	10%	61%	17%

Enemy Team LOSS	Spotted Accuracy	Time to Damage	Cross. Placement	Head Accuracy	HS Kill %	Spray Accuracy	Counter-Strafing	Accuracy (All)
kaspeR	35%	813ms	12.13°	18%	53%	48%	87%	13%
kroq	50%	492ms	11.08°	21%	65%	80%	92%	18%
Meister	30%	484ms	8.73°	21%	47%	15%	82%	18%
Rolnik Sztachetka	26%	672ms	6.86°	31%	100%	17%	73%	13%
★aimStar	42%	516ms	10.64°	17%	42%	40%	84%	20%

Appendix C : Raw Data from calculation of Q-Q plots, histograms