Investigating the Construct of Luck and its Impact on Performance in Professional Apex Legends

Staffordshire University

Word Count - 5593

Callum Clark

May 2024

Abstract

Apex Legends is a game that, on a casual level, can be defined by the luck of the items that you receive and where the zone closing in on you ends up. However, this paper shows that there is a genuine correlation between the luck which professional players have and their performances in a competitive setting. This report explores the different aspects which lead to luck being such a defining factor in the genre, seeking a better understanding of why random elements have such major impact on a game which most tout to be the esports title which takes the most skill.

Ethics

The ethics for this project were cleared by Dr Ying-Ying Law of Staffordshire University in November of 2023, with proof of such and additional remarks from James Foster found in Appendix A

Contents

Chapter 1

Introduction and Literature Review

1.1 Introduction to Apex Legends and the Battle Royale genre

Apex Legends is a Battle Royale game produced by Respawn Entertainment and published in 2019 by EA (Electronic Arts, 2019). The Battle Royale genre provides a spin on the traditional FPS genre, the concept of these titles being to scavenge for weapons and fight to the death in arenas that decrease in size over time.

A defining feature of Battle Royale FPS games is the gradual reduction of the playable area, commonly referred to as 'the circle'. Throughout a match of Apex Legends, the playable map is restricted over time to force players into confined situations which test their awareness, game sense and individual skills to be the final player or team standing in the game (Choi and Kim, 2018). Another defining factor of the Battle Royale FPS genre is the aspect of randomised loot and item spawns. An RNG algorithm determines this and determines the types of weapons, armour, and other items that spawn in specific areas of the map with each game. The RNG algorithm attempts to produce a unique game state for every player every time they play the game.

This paper explores the interplay between luck and skill in the Battle Royale genre, using Apex Legends as a case study. This paper will analyse the mechanics of loot distribution and circle generation through statistical analysis, examining how these random elements can potentially affect individual player performance and team strategies within a competitive setting.

1.2 Definition of Metrics

Apex Legends is a team game, and as such there is a need for our metrics to understand more than the individual player performance; as much as it is important. For this reason, I intend to use damage output as the key metric in understanding performance both on an individual as well as a team level.

First, we need to be able to define the concept of 'luck'. The philosopher Nicholas Rescher (2021) proposed that luck in an uncertain situation is measured by the difference between yield and expectation; or through the concept that skill enhances expectation and reduces luck in a game. Michael J. Mauboussin's book 'The Success Equation' (2012), we gain further insight into the role of luck in life. The book establishes a spectrum of pure luck vs pure skill (using the lottery and chess as examples) in order to plot decisions that we make in life in regards to the prevalence of luck vs skill. In most cases, there is a combination of all in everyday life (Mauboussin, 2012). In Apex Legends, there is a strong middle-ground between Luck and Skill as a player's ability, as well as the random game state of a game of Apex Legends, means that there will be facets of both in average gameplay due to the nature of random loot drops as well as circle generation.

Along with this, the definition of 'randomness' should be explored; randomness is the measure of uncertainty of an outcome. In statistics, a random variable is an assignment of a numerical value to each possible outcome of an event space (Simon de Laplace, 1812) (Kolmogorov, 1950). Ramsey Theory defines that pure randomness is impossible (Graham and Butler, 2015); with Theodore Motzkin arguing that 'while disorder is more probable in general, complete disorder is impossible' (Hans Jürgen Prömel, 2005). On the contrary, however, work in the world of quantum physics has shown that on the quantum level, there are instances of true randomness (Marletto, 2016). An example of such is the unpredictable nature of photon behaviour as seen from a double-slit experiment (Marletto, 2016) which demonstrates that there is true randomness on a quantum level. This is supported by further work from Drahi et al. (2020).

1.2.1 Skill vs Luck

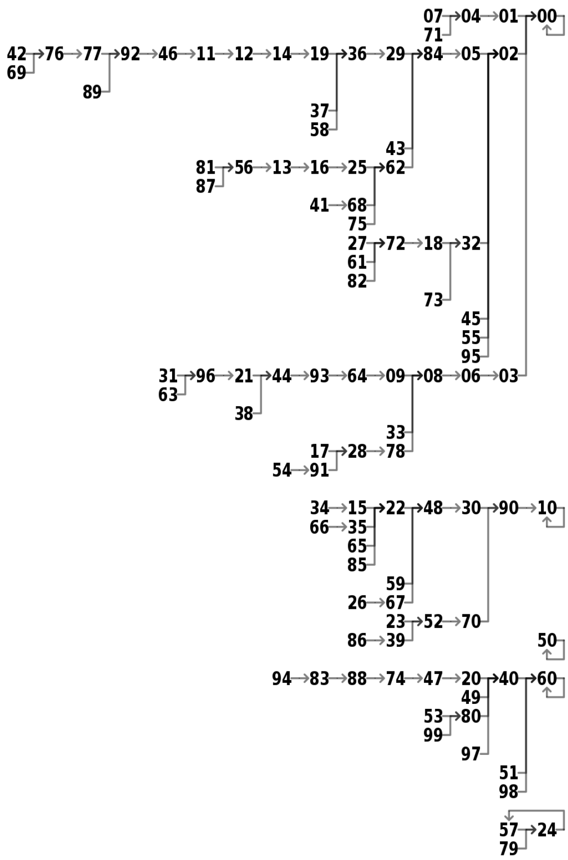
Following on from our definition of luck, 'The Success Equation' provides a great basis in order to better understand the situations in which skill and luck impact a situational outcome. Mauboussin argues that luck is not just pure chance. It is influenced by the environment in which we make our decisions and the external factors which influence this. A key example is the preparedness of your competition influencing your 'luck' by giving you further opportunities to be 'lucky' and take advantage of a random situation that positively impacts you. It's not to say that you will always win if your opponent is unprepared, but you are far more likely to be able to take advantage of such situations when an opponent is not prepared. Another impact is informed decision-making; whilst you cannot predict an individual random outcome, you can use an understanding of the odds in order to put yourself in the best situation to capitalise on a specific outcome. An example could be placing a high bet on the flop in Poker because of an elevated chance to win the hand based on the first 3 cards flipped. Whilst you don't yet know the cards on Fourth Street and the River, you can inform your decisions based on only the information you have available to you.

1.3 Psudeorandom Number Generation

A pseudorandom number generator (PRNG) is an algorithm for generating a sequence of numbers that approximate a sequence of random numbers. PRNG algorithms are never truly random due to relying on a starting seed in order to determine subsequent numbers. An example of this is with video game speedrunners manipulating RNG by calculating the starting seed of a PRNG algorithm and then extrapolating the next values in the sequence. An excellent example of this is the manipulation of RNG in Paper Mario: The Thousand-Year Door in order to force specific item drops in order to speed up the speedrun of the game drastically. (Malleo, 2019)

John Von Neumann was the first person to generate a primitive PRNG generator using the Middle Square Method (von Neumann, 1951) which worked by taking middle values of squared numbers of some a-digit number in which: $a = 2n, n \in \mathbb{Z}^+$.

This approach is very primitive, however, with a guaranteed looping termination point within 20 iterations of the algorithm when $a = 1$, it is not an effective method for generating pseudo-random numbers.



1.3.1 Linear Congruential Generation - Initial attempts for PRNG

An LCG is an algorithm which uses a piece-wise linear equation in order to generate pseudo-random numbers. This is one of the oldest generators for PRNG which can be easily run on computer hardware due to storage-bit truncation. The generator uses a recurrence relation of the form:

$$X_{n+1} = (aX_n + c) \mod m$$

Where:

The Modulus

$$m, 0 \leq m$$

The Multiplier

$$a, 0 \leq a \leq m$$

The Increment

$$c, 0 \leq c \leq m$$

The Initial Seed

$$X_0, 0 \leq X_0 \leq m$$

If:

$$m, a, c, X_0 \in \mathbb{Z}^+$$

This generator was built by (Thomson, 1958) and became one of the initial Pseudo-random Number Generators in computer science. The reason why this PRNG method is not used largely today is due to the predictability of the outcomes given knowledge of the initial parameters.

1.3.2 Mersenne Twister

The Mersenne Twister is a PRNG algorithm developed in 1998 by Matsumoto Makoto and Nishimura Takuji whose name is derived from the use of Mersenne primes as the PRNG's period length. the generator can generate integers in the range of $[0, 2^w - 1]$ for a w-bit word length. The algorithm primarily bases itself on a matrix linear recurrence over a finite binary field F_2. The system uses Linear Feedback Shift Registers (LFSRs) to define a series x and then output them in the form x_i^T where T is an invertible F_2 matrix.

See Appendix C for a Python implementation of the Mersenne Twister with customisable starting seeds as well as iterations before the program terminates. Above the code is an explanation of the used subroutines.

The algorithm has a period of $2^{19937} - 1$.

The only argument against the use of the Mersenne Twister is its lacklustre cryptographic security (Makoto, n.d.) due to it being built using linear recursion. The Mersenne Twister algorithm passes the Diehard test but fails to many of the tests in the TestU01 Suite (L'Ecuyer and Simard, 2007).

1.4 Statistical Analysis Methods

When analysing data, a variety of tests will need to be performed in order to simulate scenarios of RNG as well as check data for correlation. An explanation of both methods will be found below.

1.4.1 Binomial Distribution

The Binomial Distribution is a statistical distribution that models the probability of a certain number of successes in a number of Bernoulli Trials (a random experiment with only 2 possible outcomes).

Bernoulli Trials can be defined as long as these 4 conditions are met:

- There are a finite number of trials
- Each trial has 2 outcomes, success and failure
- Each trial should be independent
- The trials should all have the same chance of being successful

Binomial Experiments are a set of n statistically independent Bernoulli Trials each with a probability of success p. A binomial experiment's variable is modelled in the notation $B(n, p)$. You can find the chance for there to be some k successes in n tests using the formula:

$$P(k) = \binom{n}{k} \cdot p^k q^{n-k}$$

Where $\binom{n}{k}$ denotes a binomial coefficient and, $q = 1 - p$

1.4.2 Expectation of a Binomial Distribution

The expected value of a random variable with a finite number of outcomes is a weighted average of all possible outcomes. For a binomial distribution, this can be derived using the Moment Generating Function of a binomial distribution:
Let X be a discrete random variable with a binomial distribution; as per (Earl and Nicholson, 2021):

X has probability mass function:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

From the definition of the moment-generating function:

$$M_x(t) = E(e^{tX}) = \Sigma_{k=0}^{n} P(X = k)e^{tk}$$
$$\therefore M_x(t) = \Sigma_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} e^{tk}$$
$$= \Sigma_{k=0}^{n} \binom{n}{k} (pe^t)^k (1-p)^{n-k}$$
$$= (1 - p + pe^t)^n$$

By the definition of a Moment in terms of the Moment Generating Function:

$$E(X) = M'_X(0)$$
$$M'_X(t) = \frac{d}{dt}(1 - p + pe^t)^n$$

Using the Chain Rule for Differentiation:

$$= \frac{d}{dt}(1 - p + pe^t) \frac{d}{d(1 - p + pe^t)} (1 - p + pe^t)^n$$
$$= npe^t(1 - p + pe^t)^n - 1)$$

When we set t=0, we get:

$$E(X) = npe^0(1 - p + pe^0)^{n-1}$$
$$= np(1 - p + p)^{n-1}$$
$$= np$$

Therefore we can use this to find the expectation of receiving some number of a particular item in a particular number of trials.

1.4.3 Finding the Mode of a Binomial Distribution

The mode of a Binomial Distribution can be defined using cases in the following way.

$$mode = \left\{ \begin{array}{lr} 0, & \text{if } p = 0; \\ n, & \text{if } p = 1; \\ (n+1)p \text{ and } (n+1)p-1, & \text{if } (n+1)p-1 \in \mathbb{Z} \text{ and } 0 < p < 1; \\ \lfloor (n+1)p \rfloor, & \text{if } (n+1)p-1 \notin \mathbb{Z} \end{array} \right\}$$

Note: $\mathbb{Z}$ is the set of all integers.

Proof

Let:

$$f(k) = \binom{n}{k} p^k q^{n-k}$$

For $p = 0$, $f(0)$ has a non-zero value; for $p = 1$, we find that $f(n) = 1$ and $f(n) = 0$ for all $k \neq n$

$\therefore$ the mode is 0 when $p = 0$ and n for $p = 1$

Let $0 \leq p \leq 1$:

$$\frac{f(k+1)}{f(k)} = \frac{(n-k)p}{(k+1)(1-p)}$$

Therefore we can conclude that:

$k \leq (n+1)p - 1 \implies f(k+1) \leq f(k)$
$k = (n+1)p - 1 \implies f(k+1) = f(k)$
$k \geq (n+1)p - 1 \implies f(k+1) \geq f(k)$

$\therefore$ when $(n+1)p - 1$ is an integer, both $(n+1)p - 1$ and $(n+1)p$ become modes, however when $(n+1)p - 1 \notin \mathbb{Z}$, then only $\lfloor (n+1)p - 1 \rfloor + 1 = \lfloor (n+1)p \rfloor$ is a mode (Anon, 2012).

1.4.4 Correlation Tests

For the type of data I will be using, there are 2 tests which I will be able to use to show a correlation between my data; those are the Pearson r correlation and the Spearman rank correlation:

Pearson r Correlation

Building upon Francis Galton's ideas from the 1880s and Auguste Bravais' mathematical formula of 1844, Karl Pearson developed this widely used statistical measure that we can now use to measure correlation in a data set.

It is a statistical measure that indicates the strength and direction of a linear relationship between two continuous variables between the range of -1 and 1 where -1 is a perfect negative correlation and 1 is a perfect positive correlation.
It is defined in the following way:

$$r_{xy} = \frac{n\Sigma x_i y_i - \Sigma x_i \Sigma y_i}{\sqrt{n\Sigma x_i^2 - (\Sigma x_i)^2}\sqrt{n\Sigma y_i^2 - (\Sigma y_i)^2}}$$

Where:

- r_{xy} = the Pearson r correlation coefficient between x and y
- n = the number of observations
- x_i = the value of x (for the ith observation)
- y_i = the value of y (for the ith observation)

The drawback of using the Pearson r correlation is that it can only be used if the 2 sets of data are normally distributed; therefore, for data that isn't normally distributed, then we must use Spearman's rank correlation coefficient.

Spearman's rank correlation coefficient

Spearman's rank correlation coefficient is a non-parametric statistical measure used to determine the strength and direction of the association between two sets of ranked data (Spearman, 1904). Often denoted by the Greek letter ρ or r_s, it is used when you cannot assume that your data is normally distributed. It is also effective when there are serious outliers to your data due to the data relying on the ranking of both sets of data.

For a sample set of size n, the n raw scores X_i, Y_i are converted to ranks $R(X_i), R(Y_i)$; and then r_s is computed as such:

$$r_s = \rho R(X), R(Y) = \frac{cov(R(X), R(Y)}{\sigma R(X) \sigma R(Y)}$$

Where:

- ρ denotes the Pearson correlation coefficient applied to the rank variables where:
    1. $\rho \in [-1, 1]$ where 1 denotes a perfect positive correlation and -1 denotes a perfect negative correlation
- $cov(R(X), R(Y))$ denotes the covariance of the rank variables.
- $\sigma R(X)$ and $\sigma R(Y)$ are the standard deviations of the rank variables.

If all rankings are unique integers, this formula can be simplified into the form:

$$r_s = 1 - \frac{6\Sigma d_i^2}{n(n^2 - 1)}$$

Where:

- $d_i = R(X_i) - R(Y_i)$
- n is the number of observations.

This approach is a take on the Pearson r Correlation which accounts for the lack of a uniform normal distribution. This makes it useful for analysing data such as player performance in Apex Legends via stat comparisons.

Chapter 2

Loot Spawning and Generation

2.1 Algorithmic spawning of loot

Loot Spawning and generation are determined by an RNG algorithm and weighted tables of drop spawns depending on the specific area of the map. Specific areas of the map have 'high rarity drop rates' and 'low rarity drop rates' which can impact the proportion of high or low rarity items that a player can find.

As you can tell, loot spawning weights vary depending on the map that you are playing on, and as such there are different weight tables for each area type. An example can be shown below.

Item / Group	Tier			
	Basic	Mid	High	Hot Zone
Weapon	21.35%	21.47%	21.75%	28%
Ammo	13.54%	13.61%	13.79%	8%
Attachment	16.67%	16.75%	19.1%	18%
Helmet	0.52%		1.06%	3%
Evo Shield	2.03%	2.09%	3.18%	4%
Knockdown Shield	0.52%		1.06%	2%
Backpack	4.17%	4.19%	4.24%	4%
Small Regen	10.42%	9.42%	7.96%	7%
Large Regen	3.65%	4.71%	5.31%	
Ordnance	12.5%	11.52%	9.55%	8%
Mobile Respawn Beacon	0%		0.8%	0%
Heat Shield	0%	0.52%	1.06%	0%
Nothing	14.58%	14.66%	11.14%	11%

These Drops can then be further split up into their specific categories:

Item	Tier		
	Basic	Mid	High
 Alternator SMG + 40	9.9%	8.26%	7.84%
 C.A.R. SMG + 40	3.96%	5.5%	4.9%
 Charge Rifle + 28	0.99%	1.83%	1.96%
 Devotion LMG + 40	0.99%	1.83%	1.96%
 EVA-8 Auto + 16	4.95%	4.59%	4.9%
 G7 Scout + 40	2.97%	3.67%	4.9%
 HAVOC Rifle + 40	4.95%	4.59%	4.9%
 Hemlok Burst AR + 40	4.95%	4.59%	4.9%
 Longbow DMR + 28	1.98%	2.75%	1.96%
 Mastiff Shotgun + 16	4.95%	2.75%	2.94%
 Mozambique Shotgun + 16	11.88%	9.17%	9.8%

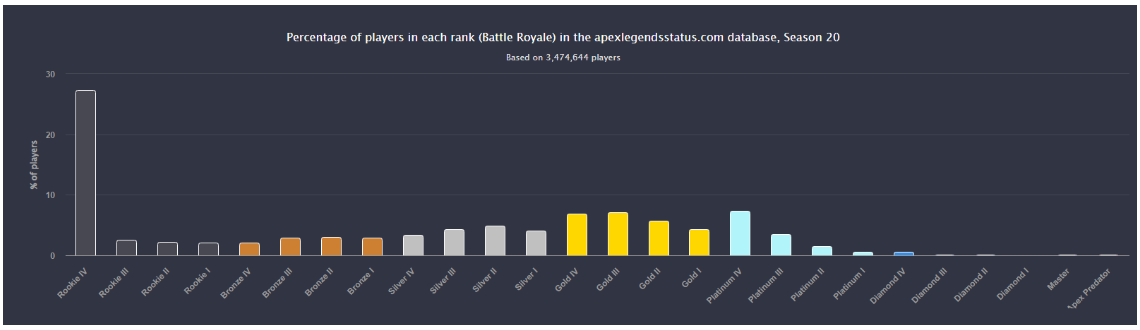
These weights can be used to determine the chance that a player finds a specific item by multiplying the values of finding a weapon and then the specific weapon.

It is worth noting that there is one exception to this rule and that is the randomly generated 'Hot Zone'. A randomly named zone is marked as a Hot Zone every match, where the overall loot quality is increased and one Fully Kitted Weapon can spawn. Fully Kitted Weapons are an upgrade on the base weapons you find with every possible attachment upgrade being pre-attached to the weapon.

Player performance tends to correlate to the performance of the weapon that they use in a game of Apex, with regards to a tiered list. The metric of the weapon used by the player can then be used to arbitrarily predict the performance of a player in a given game, however, the variation is not a massive positive correlation, and therefore more data is needed to determine the factor of likely player performance and final placement in a game of Apex.

As an example, in the recent ALGS 2023 Split 2, a high proportion of players actively picked up the Prowler SMG when possible in professional play. The active push to find certain weapons that benefit a player's play is inherently linked to the overall performance of an individual player at any given time. This led to its dominance as a weapon in professional play during the Split, finding the highest number of kills from any weapon in the game throughout the event. [Data from the Apex Legends Global Series, see Appendix B].

The implications of a specific luck element imply that Apex Legends is less focused on individual player skills. It then begs the question of how skill-based Apex is as a game based on the work discussed in 1.2.1 when discussing the work of Maboussin and their metrics of skill vs luck as a scale measure. The figure below shows the current Season 20 rank distribution of Apex Legends ranked play. (Apex Legends Status, 2024).



The ranks tend to be normally distributed if we discard the players at the lowest ranking level. I argue that discarding this data is acceptable because it accounts for unranked players who may have simply played their ranking games and then stopped playing, causing their ranking to drop over time. With that in mind, it seems that the ranks are normally distributed. With this, one can infer that the luck element of the game is negligible; otherwise, there would be a high probability of ranks being significantly less uniform in their population because the individual players are having consistently good or bad luck.

Chapter 3

Algorithmic Approaches to Circle Generation and Calculation

3.1 Vector Algebra and Apex Circles

Until 2020, Apex Legends had a randomly generated pattern of zones (also known as circles) which enclosed the remaining players to deter players from camping in one spot to secure an easy success.

After a patch in 2020, an algorithmic approach to the systems that determined future circles could give a high percentage of player prediction accuracy, requiring a brief understanding of vector addition and subtraction.

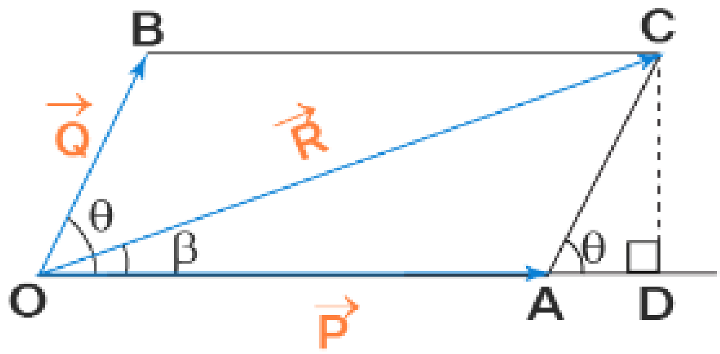
To add and subtract vectors, you need to add their corresponding components on each axis, this looks as such for vector addition:

$$\text{Let } \vec{u} = \langle u1, u2 \rangle \text{ and } \vec{v} = \langle v1, v2 \rangle\text{:}$$
$$\vec{u} + \vec{v} = \langle u_1 + v_1, u_2 + v_2 \rangle$$

And for vector subtraction:

$$\vec{u} - \vec{v} = \vec{u} + -(\vec{v}) = \langle u_1 - v_1, u_2 - v_2 \rangle$$

The Parallelogram Law of Vector Addition allows us to derive a formulaic approach for how we add vectors and how we can therefore calculate the resulting vectors and display them in a mathematical form. Consider the vectors $\vec{P}$ and $|\vec{\mathbf{Q}}|$, which in this diagram will be represented as sides OA and OB of a parallelogram as shown below.



If the vector $\vec{P}$ is extended until it reaches point D, it produces a right-angled triangle in which we can derive the magnitude of the resultant vector via the Pythagoras Theorem and further derivation. The magnitude of a vector is considered its length and therefore would be the same as the length of line OC in our diagram.

In triangle OCD, by Pythagoras Theorem we have

$$OC^2 = OD^2 + DC^2 = (OA + AD)^2 + DC^2 \quad [1]$$

In the triangle CAD, we then find:

$$\cos\theta = \frac{AD}{AC},\ \sin\theta = \frac{DC}{AC}$$
$$AD = AC\cos\theta,\ DC = AC\sin\theta$$
$$AD = |\mathbf{Q}|\cos\theta,\ DC = |\mathbf{Q}|\sin\theta \quad [2]$$

By substitution, we then prove:

$$|\mathbf{R}|^2 = (|\mathbf{P}| + |\mathbf{Q}|\cos\theta)^2 + (|\mathbf{Q}|\sin\theta)^2$$
$$|\mathbf{R}|^2 = |\mathbf{P}|^2 + |\mathbf{Q}|^2\cos^2\theta + 2|\mathbf{P}||\mathbf{Q}|\cos\theta + |\mathbf{Q}|^2\sin^2\theta$$
$$|\mathbf{R}|^2 = |\mathbf{P}|^2 + 2|\mathbf{P}||\mathbf{Q}|\cos\theta + |\mathbf{Q}|^2(\cos^2\theta + \sin^2\theta)$$
$$|\mathbf{R}|^2 = |\mathbf{P}|^2 + 2|\mathbf{P}||\mathbf{Q}|\cos\theta + |\mathbf{Q}|^2(1)$$
$$|\mathbf{R}|^2 = |\mathbf{P}|^2 + 2|\mathbf{P}||\mathbf{Q}|\cos\theta + |\mathbf{Q}|^2$$
$$\therefore |\mathbf{R}| = \sqrt{(|\mathbf{P}|^2 + 2|\mathbf{P}||\mathbf{Q}|\cos\theta + |\mathbf{Q}|^2)}$$

Vectors in this form must be defined by their magnitude but also their direction, hence we must find the angle β

Using [2] from the derivation of the magnitude $\mathbf{R}$, we can find the direction as well.

$$\tan\beta = \frac{DC}{OD}$$

$$\tan\beta = \frac{|\mathbf{Q}|\sin\theta}{OA + AD}$$

$$\tan\beta = \frac{|\mathbf{Q}|\sin\theta}{|\mathbf{P}| + |\mathbf{Q}|\cos\theta}$$

$$\beta = \arctan[\frac{|\mathbf{Q}|\sin\theta}{|\mathbf{P}| + |\mathbf{Q}|\cos\theta}]$$

It is worth noting that these formulae and derivations are not attributed to any one mathematician as these have been known for centuries in different forms; rather simply known as a fundamental principle in vector mathematics and geometry. However, the framework for working with vectors was formalised by Josiah Willard Gibbs and Oliver Heavside in the 19th Century.

ccamfpsApex (2022) is a creator who has been well-documented as pioneering the methods for circle prediction in Apex Legends using vector mathematics.
To understand how to predict these circle placements, we need to define particular vectors and points to formulate the calculations necessary.
Let:

- Point A be the Centre of the entire map
- Point B be the Centre of Circle C_1
- Point C be the where the infinite magnitude line $\vec{AB}$ meets the close edge of C_1
- Point D be the Centre of Circle C_2
- $\vec{AE} = 2\vec{AD}$
- Point F be found from point $E + \vec{AC}$
- Point G is where the tangent of the line EF intersects the line DE

At the end of this, the triangulated zone EFG will contain some amount of the following Circle to a 90+% degree accuracy.

This cycle can then be repeated to calculate the 4th and 5th circles to a relatively high degree of accuracy. For example the below image illustrates an example of this method:

At a glance, this looks to be unrelated to RNG but this doesn't account for the concept of counterpull, a randomised correction measure to ensure an adequately fair play state for all players at any given time, specifically in late-game areas.

3.2 Counterpull

Counterpull is the random variance that is applied regarding generating a circle when there is some form of map-based obstruction. In rare cases, vectors need to be subtracted rather than added to maximise the size of a given circle. These numbers are calculated based on the analysis of a random sample of 250 games from the Apex Legends Global Series from 2022-2024.

Given 2 possible circle outcomes x and y:

- where x and y have a percentage x_c and y_c of playable space within it, where $x_c, y_c \in \mathbb{R}^+$
- x_c and y_c are multiplied by some constant λ where $\lambda x_c + \lambda y_c = 100$
- $\therefore \lambda = 100 \cdot (x_c + y_c)^{-1}, \lambda \in \mathbb{R}$
- A random number corresponding to a given circle is generated to determine the counterpull of the next circle.

3.3 Implications of Circle Prediction

It is a well known fact in competitive circles within Apex Legends that circle prediction is a viable option in order to strategize for late-game decision-making. However it has an impact in our scale of luck vs skill as laid out in the introduction.

3.4 Strategic Approaches to Map rotation

Due to the random nature of circle generation, players have devised the method that I had shown above in order to predict later circles in real time. The random nature of the circles leads to a strategic decision-making process that professional players need to maintain their performance. When referring back to the work of Maboussin, it begs the question of whether the impact of the randomly generating circles are actually wholly random, or whether the impact is negligible due to the decision making process regarding rotation.

Chapter 4

Analysis of Player Performance between Professional Players

4.1 Testing

In the best interests of getting data which is most reliable, I will be using the ALGS Year 3 Split 2 Championships Data (2023) in which I will be using random sampling of 50 individual team performances from the event. With this in mind, I will then focus on team performance regarding loot level of the team's drop location. This will give valuable insight into the correlation of loot quality with regard to the team's final placement. Unfortunately, the fact that the variety of circle possibilities are so large in Apex, means that testing with regard to data such as their initial closeness to the final circles as an example, is unable to be tested. As well as this, there would only be 1 instance of each case of circles combination which means that the data set is minuscule and therefore very likely to test within the critical regions of any statistical test and therefore be rejected.

I intend to investigate whether a team in each of our 50 instances is able to receive 3 complete sets of each player's highest-performing weapon according to ALGS (2023). and then comparing the damage output of the teams that obtained all of their preferred weapons to those who were not. As well as testing the average damage output of players who were able to find their preferred weapons against those who were unable to. This builds on the concepts discussed by Novak et al., (2019) and their research into analysis of esports statistics. However, a lot of their work also discusses Notational Analysis which, due to the size of my data set, was not a feasible approach.

4.2 Formulation of Null Hypothesis

To mathematically formulate my hypothesis and alternative, I must generate my null hypothesis defined as H_0, and then my alternative hypothesis, known as H_1.

I hypothesise the following.

$$H_0 : -0.1 \geq \rho \geq 0.1$$

$$H_1 : \rho \notin [-0.1, 0.1]$$

In this particular instance, we will use Spearman's rank-order correlation. This is due to the fact that the variable we use to check if a team has 3 complete sets of data is binary. Whilst limiting, this will still give us the data necessary in order to determine a correlation.

4.3 Results for individual players

Of the 150 players tested, 113 of them were able to obtain full loadouts within the team's first loot area. And of those who did receive full loadouts, they had an average damage output 29% higher than those who did not. Using Spearman's rank-order correlation, I was able to determine that the correlation between the binary variable 'Obtained Weapons' and the player's damage output was 0.223, as seen below:

Correlations

			Zone_1_Fullguns	Dmg_Out
Spearman's rho	Zone_1_Fullguns	Correlation Coefficient	1.000	.223**
		Sig. (2-tailed)	.	.006
		N	150	150
	Dmg_Out	Correlation Coefficient	.223**	1.000
		Sig. (2-tailed)	.006	.
		N	150	150

**. Correlation is significant at the 0.01 level (2-tailed).

This data is promising in providing an argument to reject my H_0 with regards to individual player performance.

4.4 Results for Teams

Of all 50 games analysed, teams received 3 full loadouts in the first loot zone a total of 23 times, a completion rate of 46%. And in the games in which teams were able to obtain their full loadouts, it was found that the average damage output of the team was 27% higher than the teams who did not receive their full loadouts. The Spearman's correlation came out to 0.210 as shown below:

Correlations

			Zone_1_Fullguns	Dmg_Out
Spearman's rho	Zone_1_Fullguns	Correlation Coefficient	1.000	.210
		Sig. (2-tailed)	.	.143
		N	50	50
	Dmg_Out	Correlation Coefficient	.210	1.000
		Sig. (2-tailed)	.143	.
		N	50	50

Chapter 5

Implications of Correlation

5.1 Findings

From the tests performed, there was a clear correlation between individual player performance and the weapons that they obtained. Depending on whether players obtained their 'preferred weapons', there was an average damage output reduction of roughly 29%. This reduction in player statistics can allow us to infer that there is a tangible impact on player performance when they have poor luck. The impact of player performance drop-off is highly impactful, nearly to the degree that it will impact the entirety of a game.

However, when we look beyond an individual level and to the level of a team, there is a stark contrast. We calculated that on average, a team obtains 2.3 completed sets of 'preferred weapons' per drop zone; this means that most of the time, a team will only obtain 2 completed sets of weapons. Therefore, this balances a level of consistency to a team in a competitive setting since if 1 player is unable to obtain an optimal loadout, their teammates are likely to be able to do so. However, the overall impact of this is highly dependent on team rotations and circle generation. The numerical impact is almost impossible to calculate due to the nature of the generation of circles in Apex Legends as seen in Chapter 3. The fact that there is a 360° possibility of where any circle can generate after a completely random initial circle, at this stage, it is too complicated a problem to solve without more computational power at hand.

5.2 Implications of Findings

In the current state of the game, the data that has been gathered and the conclusions that we've been able to draw imply that there is more to simply the factor of luck at play here. Circle Generation and Weapon Drops are not the only factor that impacts competitive play at a professional level; since there is of course the ever-alarming presence of individual mechanical player performance which will shape the end of each game of Apex Legends. It is a factor of the BR genre that teams will become better equipped with weapons and equipment as the game reaches its conclusion. But because of this as well, the factor of luck is mitigated further and further to the point that its impact is negligible on the outcome of a competitive match. The gap in player performance, therefore, becomes more wholly the factor of individual player mechanical skill and the strategic element of team positioning, rotation etc.

Referring back to the work of Maboussin, he defines an observed outcome of a random system as '$Skill + Luck$' and we can use this definition in our example of Apex Legends. The combination of individual player skill and luck of loot drops and circle generation can combine to produce the outcome which we as viewers see. The aspect of luck and the extent of its importance is very difficult to predict (Aoki, Assuncao and Vaz de Melo, 2017), and the same goes for the players themselves. There is only so much that can be done in order to predict the outcome of a game with so many random aspects, no matter the size of the impact of each random element. Due to this, it is near impossible to determine who the best players and teams are in terms of their skill (Lopez, Matthews and Baumer, 2018).

Chapter 6

Conclusion & Further Research

6.1 Conclusion

Throughout this report, I have been able to mathematically demonstrate the correlation and impact that randomness has on professional play in Apex Legends. The data clearly demonstrates there to be an impact on player performance with regards to weapon acquisition in the early stages of the match at the individual level.

In contrast however, the data also shows there to be a lesser impact on team dynamics due to the likelihood of at least 2 members of the team obtaining the weapons they prefer in the first loot zone. Over time, as the distribution of loot tends to lessen and each player has become better equipped, the team impact of the weapons dropped in the first loot zone reduce significantly, however, the difference between teams that are better equipped in the start of the game are not negligible.

Being the first research into the matter of randomness on the Battle Royale FPS genre, there is much more that needs to be researched and that will be enabled with more computational power dedicated to this research. My inability to research trends with circle generation comes down to the lack of computational power available and access to the algorithms used to generate unique circles for each game of Apex Legends. On top of this, the lack of a large enough data set to digest this data means that there is unfortunately no possible way that any of the data that could be used at this current time would yield results for analysis.

6.2 Future Research Opportunities

My research has been limited by many aspects such as a lack of computational power and access to the algorithms which define circle generation in Apex Legends. As well as this, the issues that come with Apex Legends rotating the weapons that are obtainable in each Season of the game make it difficult to use data across a vast number of seasons where the weapons that enter and exit rotation are exceptionally powerful.

With regard to further research, I see 3 potential avenues for further research into this subject and they are as such:

- Computational Modelling: Modelling techniques could be used in order to understand the complexities between loot acquisition, circle generation and a team's performance in response to both random elements. This would give a better insight and a more advanced one in comparison to what I have been able to produce. This data would be able to include stimuli such as late-game supply drops as well as 'third-partying' in which one team waits in order to ambush 2 weak opponents who are fighting one another.
- Psychological Impacts: Another possible avenue would be a detailed look into how players are impacted mentally and psychologically when placed in a situation with very poor or very good luck. This insight could demonstrate whether the correlation between loot acquisition and damage output are mental or whether it is purely due to the difference in weaponry. A discussion into 'tilt' mentality, following the work of Wu, Lee and Steinkuehler (2021)
- Game Design: An investigation into how best to combat luck in the Battle Royale genre whilst also keeping it's characteristic traits could provide an opportunity to see a game that is more skill-based but true to the original aspects of what makes Apex Legends such an enjoyable game for many.

Further research into the impact of randomness in esports could open up a new horizon into how we consider esports statistics and possibly redefine the metrics which we use to define player and team performance in the genre.

Whilst this study is an important first step, much is left to be done in understanding the complexities between luck and skill within Apex Legends, the broader FPS genre as well as the rest of the esports titles between different genres. The potential avenues for further research allow for us to understand soon the intricacies that make up individual decision-making in situations of good or bad luck in a particular situation. The better we understand the complex systems which make up competitive play in esports, the better that we can illustrate this data not only in an academic setting but also in a commercial setting, providing better insights to players through statistics.

Eventually, the convoluted concepts that have been defined in this report will be better understood and allow us to do many things; such as refining the way in which we perceive performance metrics, as well as gaining a better understanding of the mental decisions which professional players make in response to random stimuli.

Chapter 7

References

Anon (2012). *Finding mode in Binomial distribution.* [online] Mathematics Stack Exchange. Available at: https://math.stackexchange.com/questions/117926/finding-mode-in-binomial-distribution/117940#117940 [Accessed 28 Mar. 2024].

Aoki, R.Y.S., Assuncao, R.M. and Vaz de Melo, P.O.S. (2017). Luck is Hard to Beat. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* doi:https://doi.org/10.1145/3097983.3098045.

Apex Legends Global Series (2023). *Apex Legends Global Series Year 3 Split 2 Championships Data.* [online] Available at: https://apexlegendsstatus.com/algs/Y3-Split2/ALGS-Championships/Global/Overview#tab-scoring.

Blackman, D. and Vigna, S. (2022). *Scrambled Linear Pseudorandom Number Generators.* [online] arXiv.org. doi:https://doi.org/10.48550/arXiv.1805.01407.

ccamfpsApex (2022). *Endzone & Ring Prediction.* [online] GitHub. Available at: https://github.com/ccamfpsApex/ApexLegendsGuide/wiki/Endzone-&-Ring-Prediction.

Choi, G. and Kim, M. (2018). Gameplay of Battle Royale Game by Rules and Actions of Play. *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE).* doi:https://doi.org/10.1109/gcce.2018.8574781.

Drahi, D., Walk, N., Hoban, M.J., Fedorov, A.K., Shakhovoy, R., Feimov, A., Kurochkin, Y., Kolthammer, W.S., Nunn, J., Barrett, J. and Walmsley, I.A. (2020). Certified Quantum Random Numbers from Untrusted Light. *Physical Review X*, 10(4). doi:https://doi.org/10.1103/physrevx.10.041048.

Earl, R. and Nicholson, J. (2021). *The concise Oxford dictionary of mathematics.* Oxford, United Kingdom: Oxford University Press.

Electronic Arts (2019). *Apex Legends - The Next Evolution of Battle Royale - Free on PS4, Xbox One, and PC.* [online] Apex Legends. Available at: https://www.ea.com/en-gb/games/apex-legends.

Feller, W. (1957). *An Introduction to Probability Theory and Its Applications.*

Gnedenko, B.V. (1950). *The theory of probability : and the elements of statistics.* Providence, R.I.: Chelsea Pub. Co.

Hans Jürgen Prömel (2005). Complete Disorder is Impossible: The Mathematical Work of Walter Deuber. 14(1), pp.3–16. doi:https://doi.org/10.1017/s0963548304006674.

John von Neumann, "Various techniques used in connection with random digits", in A. S. Householder, G. E. Forsythe, and H. H. Germond, eds., *Monte Carlo Method, National Bureau of Standards Applied Mathematics Series*, vol. 12 (Washington, D.C.: U.S. Government Printing Office, 1951): pp. 36–38.

Kolmogorov, A.N. (1950). *Grundbegriffe der Wahrscheinlichkeitrechnung.*

L'Ecuyer, P. and Simard, R. (2007). TestU01: A C library for empirical testing of random number gen-erators. *ACM Trans. Math. Softw*, [online] 33(4). Available at: https://www.iro.umontreal.ca/l̃ecuyer/myftp/papers/testu01.pdf.

Lemire, D. (2020). *The Xorshift128+ random number generator fails BigCrush.* [online] Daniel Lemire's blog. Available at: https://lemire.me/blog/2017/09/08/the-xorshift128-random-number-generator-fails-bigcrush/.

Lopez, M.J., Matthews, G.J. and Baumer, B.S. (2018). How often does the best team win? A unified approach to understanding randomness in North American sport. *The Annals of Applied Statistics*, [online] 12(4), pp.2483–2516. doi:https://doi.org/10.1214/18-aoas1165.

Makoto, M. (n.d.). *Mersenne Twister: A random number generator (since 1997/10).* [online] www.math.sci.hiroshima-u.ac.jp. Available at: http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/mt.html.

Malleo (2019). *Manipulating a 1 in 46,000 Chance - When Luck Just Isn't Enough.* [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=-9YLCoK5K6o [Accessed 21 Mar. 2024].

Marletto, C. (2016). Constructor theory of probability. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472(2192), p.20150883. doi:https://doi.org/10.1098/rspa.2015.0883.

Matsumoto, M. and Nishimura, T. (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1), pp.3–30. doi:https://doi.org/10.1145/272991.272995.

Mauboussin, M.J. (2012). *The success equation : untangling skill and luck in business, sports, and investing.* Boston, Mass.: Harvard Business Review Press.

Novak, A., Bennett, K., Pluss, M. and Fransen, J. (2019). Performance analysis in esports: part 1 - the validity and reliability of match statistics and notational analysis in League of Legends. doi:https://doi.org/10.31236/osf.io/sm3nj.

O'Neill, M.E. (2018). *Xoroshiro128+ Fails PractRand Even When Truncated*. [online] PCG. Available at: https://www.pcg-random.org/posts/xoroshiro-fails-truncated.html [Accessed 22 Feb. 2024].

Pearson, K. (1895). *'Note on Regression and Inheritance in the Case of Two Parents'.*.

Prohorov, Yu.V. and Rozanov, Yu.A. (1969). *Probability Theory, basic concepts*. Springer.

Rescher, N. (2021). *Luck Theory*. Springer Nature.

Simon de Laplace, P. (1812). *Théorie Analytique des Probabilités*.

Spearman, C. (1904). The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1), p.72. doi:https://doi.org/10.2307/1412159.

Thomson, W.E. (1958). A Modified Congruence Method of Generating Pseudo-random Numbers. *The Computer Journal*, 1(2), pp.83–83. doi:https://doi.org/10.1093/comjnl/1.2.8

Weisstein, E.W. (n.d.). *Negative Binomial Distribution*. [online] mathworld.wolfram.com. Available at: https://mathworld.wolfram.com/NegativeBinomialDistribution.html.

Wu, M., Lee, J.S. and Steinkuehler, C. (2021). Understanding Tilt in Esports: A Study on Young League of Legends Players. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. doi:https://doi.org/10.1145/3411764.3445143.

Zhang, Y., Zhu, M., Yang, B. and Liu, L. (2021). Study on post-processing algorithms for true random number generators. doi:https://doi.org/10.1109/iscipt53667.2021.00067.

Appendix A

Approval of Ethics Documentation

School of Digital,Technologies and Arts

Ethics Review Report

STAFFORDSHIRE UNIVERSITY

Researcher Name:	Callum Clark
Title of Study:	A statistical investigation into the real-world impact of Random Number Generation (RNG) in competitive Apex Legends
Panel Reviewer:	Dr Ying-Ying Law

Comments:

Your disclaimer ethics form has been approved, but the proportionate ethics form has not been approved (due to insufficient supporting documents provided) – you can only have ONE ethics form approved – so if you wish to change this, please consider resubmitting.

<u>Dataset analysis:</u>
As long as the dataset you are analysing is accessible via the general public (posted online for all Apex Legends community members), you may use this dataset. However, if it requires you to ask players for their dataset, then this will be a proportionate ethics form, which you will need to resubmit (because it involves gathering data from other people).

Regarding analysis of your own 400 games as a researcher – I would suggest providing a good justification why these are worthwhile to research over other datasets.

Please arrange a meeting with your supervisor.

Recommendation:
(please indicate recommendation)

Approved

Date: 10th Nov 2023.

James Foster
To: Callum Clark
Mon 13/11/2023 14:55

Ethics reviewer's report - 1.d...
204 KB

Hi Callum,

Your ethics has been approved. See attached.

The comments made by the reviewer about additional data and proportionate ethics forms can be ignored as they are no longer relevant within for this study and added in error.

Kind regards,
James

Appendix B

Statistics of Weapon Kills from the 2023 ALGS Split 2 Playoffs [July 12-13 2023]

Weapon	Kills
Prowler	846
Nemesis	390
CAR	379
R-99	226
Flatline	192
Volt	186
30-30	172
G7Scout	108
ArcStar	100
R-301	71
ChargeRifle	69
TripleTake	61
Wingman	59
Alternator	57
L-STAR	41

Appendix C

Python Mersenne Twister Algorithm

C.1 Explanation of subroutines

- _int32(x) - This function guarantees that we are using the necessary 32-bit unsigned integers.
- initialise_generator(self, seed) - This function initialises the Mersenne Twister array with the seed value from the user input and then populates the array.
- extract_number(self) - Generates the next number and then performs the tempering process by using bitwise XOR oprations as well as bit shifting.
- generate_numbers(self): - Refills the Mersenne Twister array when the index reaches 624 in order to not repeat numbers.

We Generate a Mersenne Twister class in order to encapsulate and reuse the code in order to make it more efficient.

The algorithm has a time complexity of O(1) since the time is constant for the generation of an array of random numbers.

C.2 Implementation in Python 3.11

```
def _int32(x):
    # Get the 32-bit integer representation of x
    return x & 0xFFFFFFFF

class MersenneTwister:
    def __init__(self, seed):
        self.w, self.n, self.m, self.r = 32, 624, 397, 31
        self.a = 0x9908B0DF
        self.u, self.d = 11, 0xFFFFFFFF
        self.s, self.b = 7, 0x9D2C5680
        self.t, self.c = 15, 0xEFC60000
        self.l = 18
```

```
        self.f = 1812433253
        self.MT = [0] * self.n
        self.index = self.n + 1
        self.lower_mask = (1 << self.r) - 1
        self.upper_mask = (~self.lower_mask) & 0xFFFFFFFF
        self.seed_mt(seed)

    def seed_mt(self, seed):
        self.index = self.n
        self.MT[0] = seed
        for i in range(1, self.n):
            # Slightly complex linear recurrence; consult reference for
                details
            self.MT[i] = _int32(self.f * (self.MT[i - 1] ^ (self.MT[i - 1]
                >> (self.w - 2))) + i)

    def extract_number(self):
        if self.index >= self.n:
            self.twist()

        y = self.MT[self.index]
        y = y ^ (y >> self.u)
        y = y ^ ((y << self.s) & self.b)
        y = y ^ ((y << self.t) & self.c)
        y = y ^ (y >> self.l)

        self.index += 1
        return _int32(y)

    def twist(self):
        for i in range(self.n):
            x = (self.MT[i] & self.upper_mask) + (self.MT[(i + 1) %
                self.n] & self.lower_mask)
            xA = x >> 1
            if x % 2 != 0:
                xA = xA ^ self.a
            self.MT[i] = self.MT[(i + self.m) % self.n] ^ xA
        self.index = 0


# Items belwo can be altered to set starting seeds and the iterations
    before completion
if __name__ == "__main__":
    generator = MersenneTwister(5489) # Set a seed
    for i in range(10):
        print(generator.extract_number())
```