Overwatch, the essence of Reinhardt: an analysis of the performance of top 100 Rein players in Season 13

Xuan Zhu

December 3rd, 2018

Abstract

Behind every competitive game on earth is a bunch of ambitious players seeking for wins. And among all the ambitious gamers there is always a group of "best of the best" people who occupy the top list in the ranking system. We know that some of those top players have already become professional players, so they usually play under the guidance of their coaches; the others are famous streamers on Twitch, or talented enough to compete with pros without prior training. Nonetheless, they are representing the highest level of the heroes they are playing in this game.

But have you ever curious about this: what are their secrets to their success? Is there anything that we, common players could learn from those pros' play? This report "The essence of Reinhardt: an analysis of the performance of top 100 Rein players in Season 13" aims at uncovering some nature of the hero Reinhardt by examining the statistics of the most skilled specialists to reveal their way of playing Reinhardt. By doing so, we are able to gain a better game understanding as an ordinary player, and possibly, to get a higher SR score in

Figure 1: “Reinhard”

the future as most of us expect.

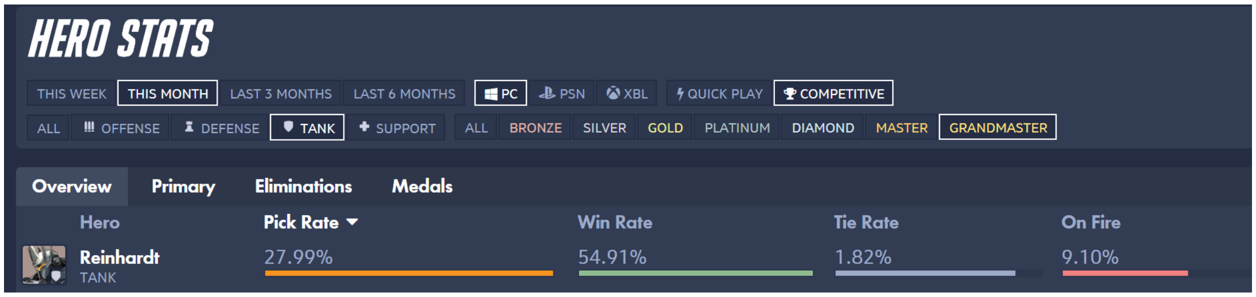
Introduction

Background Info

If you are a starter in Overwatch, for a good explanation of how overwatch is played, please go to this website:

https://www.wired.com/2017/01/overwatch-guide/

In the current competitive season 13, Reinhardt is the most popular main tank hero with the highest picking rate in all tiers, so I pick it as the sample hero to study. This means that knowing how to play Reinhardt is important if you want to rank your SR score up. From the website www.overbuff.com (one of the most authorative stats website for Overwatch), we can



HERO STATS

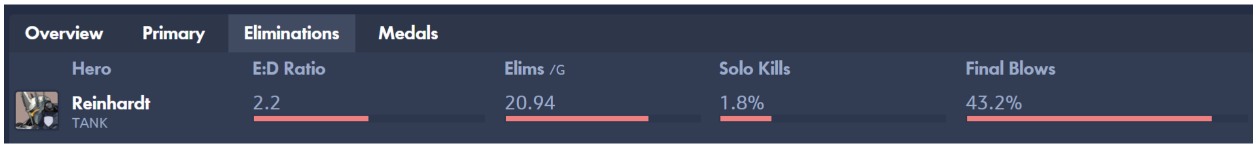
THIS WEEK [✓ THIS MONTH] LAST 3 MONTHS LAST 6 MONTHS [✓ PC] PSN XBL QUICK PLAY [✓ COMPETITIVE]

ALL OFFENSE DEFENSE [✓ TANK] SUPPORT ALL BRONZE SILVER GOLD PLATINUM DIAMOND MASTER [✓ GRANDMASTER]

Overview Primary Eliminations Medals

Hero	Pick Rate	Win Rate	Tie Rate	On Fire
Reinhardt TANK	27.99%	54.91%	1.82%	9.10%

Figure 2: "Grandmaster stats 1"



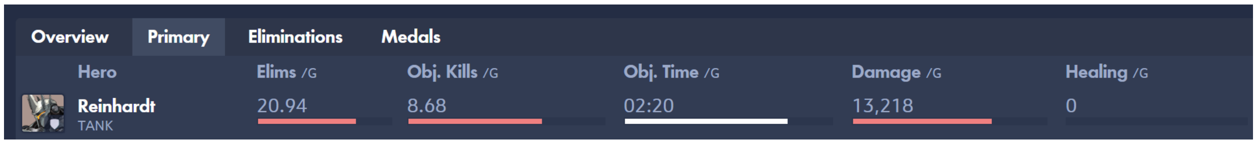
Overview Primary Eliminations Medals

Hero	E:D Ratio	Elims /G	Solo Kills	Final Blows
Reinhardt TANK	2.2	20.94	1.8%	43.2%

Figure 3: "Grandmaster stats 2"

quickly grab the average stats for each tier, separately. Take the highest tier(SR score>4000) as an example:

The average win rate of Grandmaster level Reins is 54.91%. The E/D ratio is 2.2, and eliminations per game is 22,etc. But this only summerizes the average level of players with >4000 SR score. In other words, some data used to calcualate the means is from those who are not rein mains, but are forced to fill in the tank position as their teammates request. As a result, their stats are not as good as the Rein mains', pulling the numbers shown down at some level. The situation happens a lot in Overwatch. But we only want to study on the stats of pro Reinhardt players because their Reins are representative.



Overview Primary Eliminations Medals

Hero	Elims /G	Obj. Kills /G	Obj. Time /G	Damage /G	Healing /G
Reinhardt TANK	20.94	8.68	02:20	13,218	0

Figure 4: "Grandmaster stats 3"

Method

i. Data Source

I used the public data from www.overbuff.com. For detailed procedure on how to web scrape statistics from the website, please refer to the Appendix: Data Mining Instruction. Several things need to be mentioned before you dive into the model:

a) The game records are taken in the time period of late-Nov,2018, corresponding to the game version of PATCH 1.30.0.1.The variables I choose to put in my model are usually not sensitive to new patches. But this is not 100% guaranteed. Please be careful.

b) The game records are from top 100 players rated by Overbuff instead of Blizzard. The difference is subtle, but Overbuff's rating system is able to filter out those who play a lot and get a relatively stable high skill rating score. For example, a top 500 support player in Blizzard's rating system is required to play at least 50 games to be shown on the top list, but we do not know how many games he was playing on Ana. Maybe he just played 2 or 3 games as Ana and played the rest 48 games as Zenyatta. But Overbuff's hero ranking system can ensure each player recorded in my dataset is particularly pro in his/her chosen heroes.

The source data is as the following in R:

```
##   PLAYER_ID WIN_RATE WIN LOSS DAMAGE SOLO_KILLS BLOCKED   ED DEATH CHARGE
## 1 Arty-1346   0.5000   3    3   9881        0.0   15130 1.63 10.00   2.00
## 2 Arty-1346   1.0000   1    0  30743        1.0   53074 2.29 21.00   6.00
## 3 Arty-1346   0.2500   1    3   9672        0.0   17676 2.35  8.50   2.75
## 4 Arty-1346   0.6667   2    1  10031        0.0   10663 2.75  6.67   1.67
## 5 Arty-1346   0.5333   8    7  12190        0.4   17895 2.46  8.93   2.93
## 6 Arty-1346   1.0000   2    0  13080        0.5   12870 2.72  9.00   4.50
##     ULT  FIRE GAME_PLAYED
## 1  4.83  7.00            6
## 2 10.00 16.00            1
## 3  4.00  8.00            4
## 4  1.33  6.00            3
## 5  3.93  7.13           15
## 6  3.50  8.50            2
```

Each row in the data set represents each time the player opens Overwatch. For example, Arty-1346 logged into Overwatch last night, playing 6 games as Rein in total and the win-loss record is 3-3. The numbers following the LOSS columns are the average stats per game for the 6 games he played. Here's an explanation list of variables:

DAMAGE The damage made by Rein.

SOLO_KILLS The eliminations that Rein get without teammates' help.This variable is usually considered unimportant for Rein. We will decide whether to exclude it after the EDA process later.

BLOCKED The blocked damage by Rein's shield as tank.

ED The elimination/death ratio

DEATH The death rate per game. Why death is also important rather than simply picking ED(Killing/Death ratio) as the predictor? Because killing enemies is not the only job a tank is expected to do from a team perspective. Sometimes Rein's teammates need Rein to protect his teammate by shield first. A good Rein can have a low KD. In other words, when converting the death rate to KD ratio, we lose some key information. Besides, we expect there is interaction between ED and DEATH.

CHARGE One of the unique skills of Rein. Higer kills gained by using this skill indicate a better master of Rein.

ULT One of the unique skills of Rein.Higer kills gained by using this skill indicate a better master of Rein.

FIRE One of the unique skills of Rein. Higer kills gained by using this skill indicate a better master of Rein.

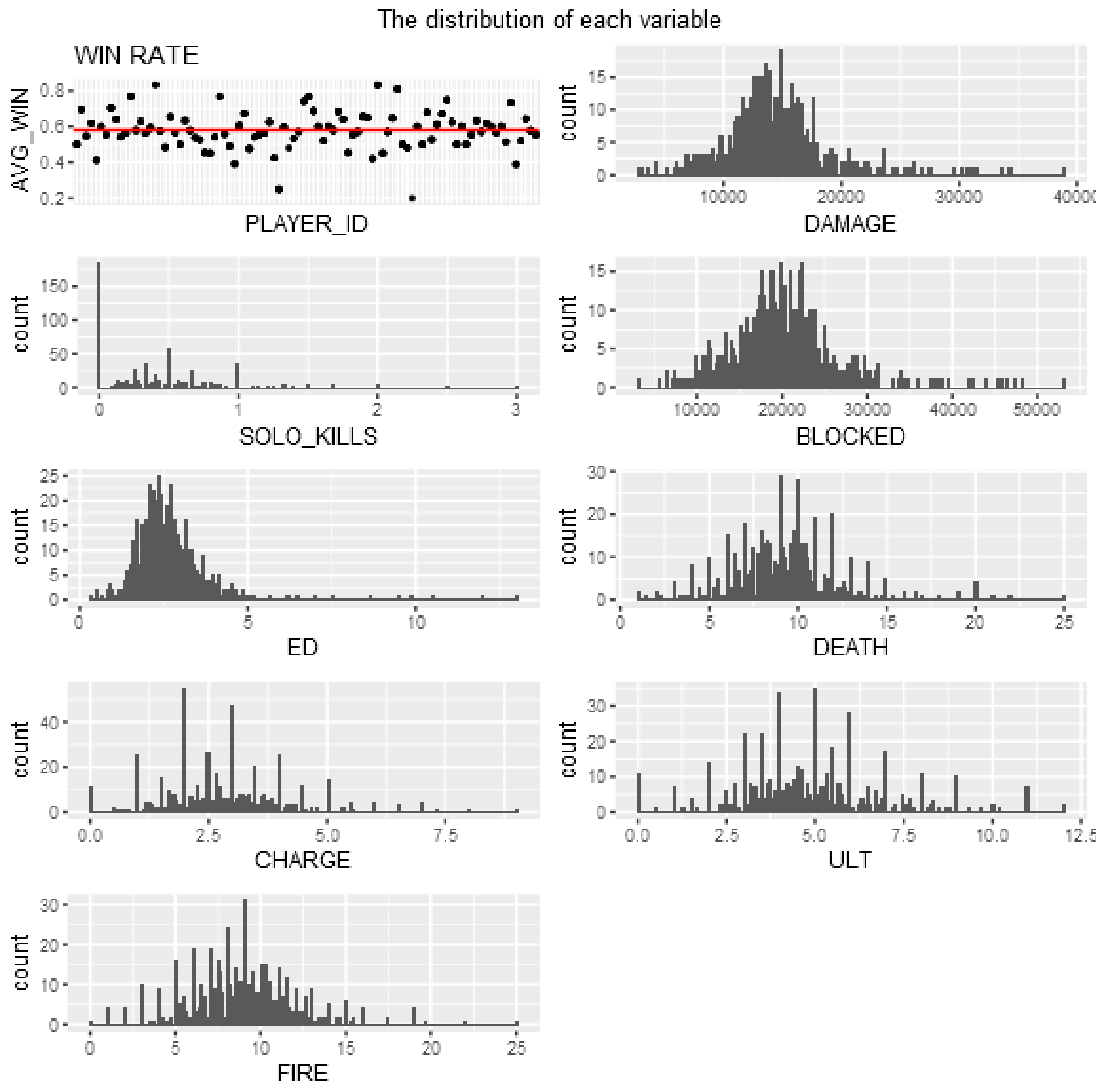
Basic Data Visualizaition & Exploratory Data Analysis

First, we take a look at the distribution of each variable in the dataset.

The average win rate for each player is calculated by $sum(Wins)/Sum(GamePlayed)$. The red line in the plot represents the average level of all players, which is about 58%. So the individual win rate is distributed randomly around the average line.

One may notice that there are outliers. After checking the source data, I confirm that the data is recorded correctly and hence we should take them into account. The game records were taken recently, so it is possible that there exist two guys rarely win within several days.

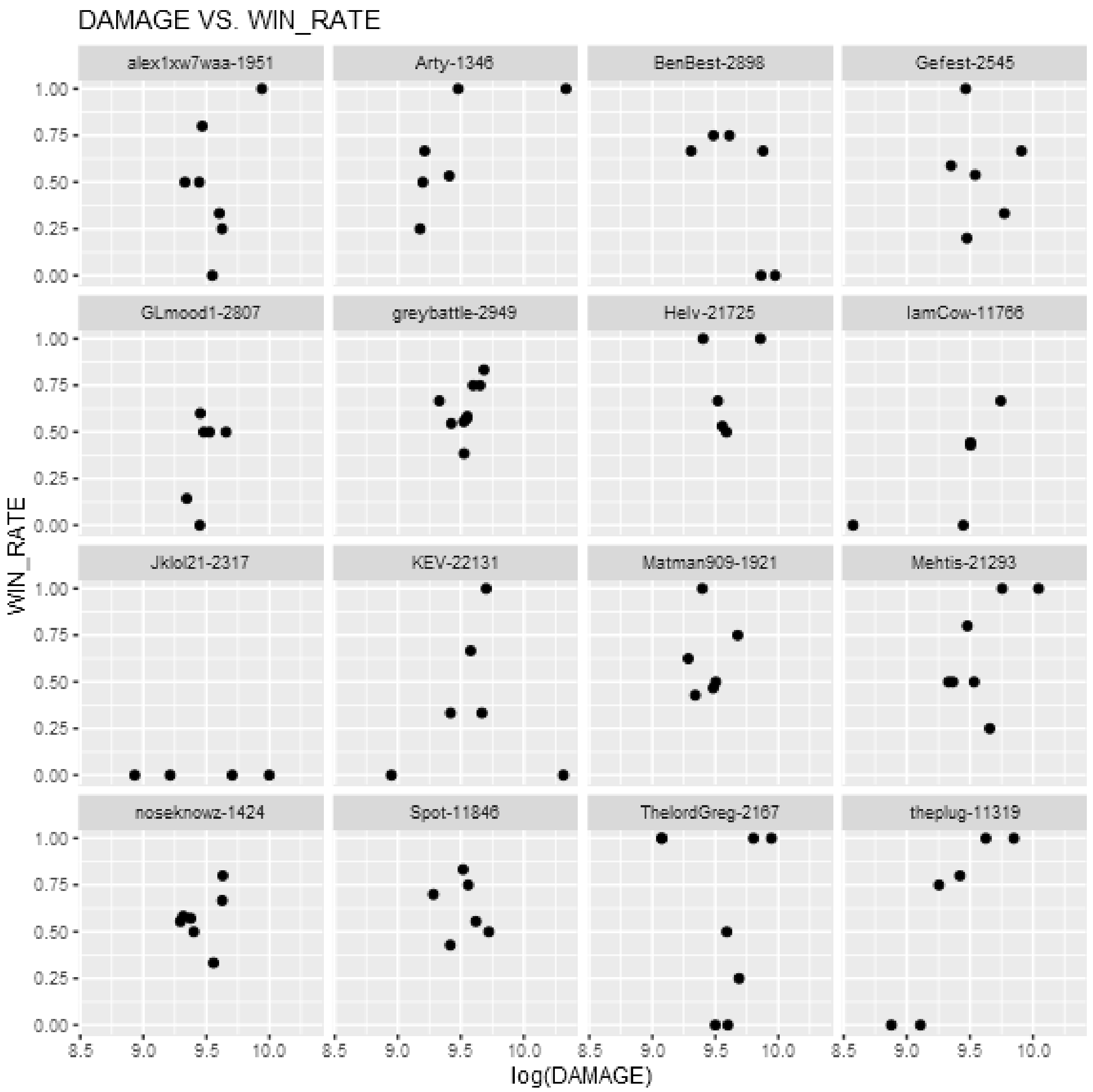
```
## [1] 0.582827
```

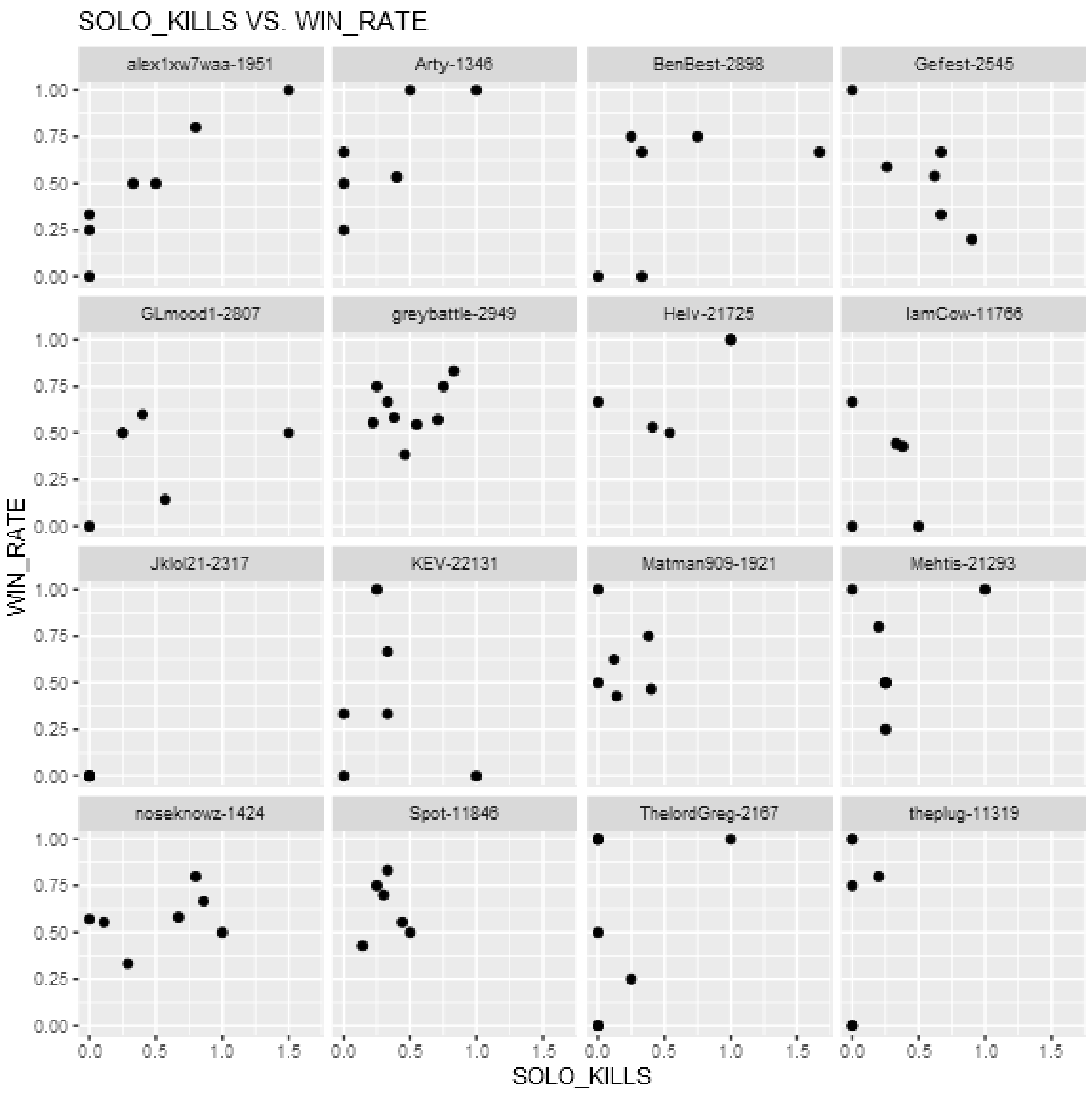


The variables DAMAGE, BLOCKED and ED look normal but they are kind of right skewed. Other variables have some values that occur multiple times. For example, SOLO_KILLS has an issue of zero inflation.

Next, we display several plots to exhibit the relationships between the possible predictors and the response variable. For example, p1 shows that high win rates occur more when log(DAMAGE) is high than it does when log(DAMAGE) is low. In p2 the trend is not that obvious, so we decide not to include SOLO_KILLS in the model.

Sample EDA





After doing EDA, we select out the predictors that we think display a trend with the win rate: DAMAGE,DEATH,FIRE and CHARGE.

ii.Model Used

The model used for analysis is a multilevel logistic model, which takes the variablity of personal performance into account.

$$logit^{-1}(\beta_0 + \beta_{charge}Charge + \beta_{Damage}log(Damage) + \beta_{Fire}Fire + \beta_{Death}Death = P_{Win}$$

```
## Linear mixed-effects model fit by maximum likelihood
##  Data: GameRecordS13R
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | PLAYER_ID
##         (Intercept)  Residual
## StdDev:   0.1826185 0.9101132
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: cbind(WIN, LOSS) ~ log(DAMAGE) + c.DEATHR + c.CHARGE + c.FIRE
##                  Value Std.Error  DF   t-value p-value
## (Intercept) -9.191686 2.8530179 449 -3.221742  0.0014
## log(DAMAGE)  0.994733 0.2987552 449  3.329593  0.0009
## c.DEATHR    -0.208157 0.0233581 449 -8.911572  0.0000
## c.CHARGE     0.117042 0.0447512 449  2.615393  0.0092
## c.FIRE       0.093413 0.0247169 449  3.779319  0.0002
##  Correlation:
##             (Intr) l(DAMA c.DEAT c.CHAR
## log(DAMAGE) -1.000
## c.DEATHR     0.488 -0.488
## c.CHARGE     0.160 -0.161 -0.083
## c.FIRE       0.631 -0.630  0.128 -0.138
##
```

```
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.97352213 -0.70179517  0.03135311  0.72570494  3.71044845
##
## Number of Observations: 547
## Number of Groups: 94
```

Result

i. Model Choice

The choice of model predictors:

From the EDA process we have observed that among the three unique skills of Rein, ULT kills do not demonstrate an obvious trend with win rate. One explanation is that in games of top players, the efficiency of ultiminate skill 'shatter' does not depend on the personal ability of players much but on the cooperation of team, because it is a skill that can be easily disrupted by enemies without teammates' help. On contrary, even though the skills CHARGE and FIRE are not as powerful as ULT, they have less cooldown times, faster launch time and are harder to be interrupted so that they are more related to personal level of performance.

Why BLOCKED is not included either? Originally, we expect this would be significant, but both EDA and the summary of the following model show that it is actually as important as we usually think. (p=0.8337>0.05)

```
mR2 <-glmmPQL(cbind(WIN,LOSS)~log(DAMAGE)+log(BLOCKED)+c.DEATHR+c.CHARGE+c.FIRE,~1|PLAY
```

```
## iteration 1
## iteration 2
## iteration 3
```

```r
summary(mR2)
```

```
## Linear mixed-effects model fit by maximum likelihood
##  Data: GameRecordS13R
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | PLAYER_ID
##          (Intercept)  Residual
## StdDev:   0.1836015 0.9099454
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: cbind(WIN, LOSS) ~ log(DAMAGE) + log(BLOCKED) + c.DEATHR + c.CHARGE +
##                   Value Std.Error  DF   t-value p-value
## (Intercept)  -9.464394 3.1139122 448 -3.039390  0.0025
## log(DAMAGE)   0.972314 0.3197002 448  3.041330  0.0025
## log(BLOCKED)  0.049204 0.2341614 448  0.210127  0.8337
## c.DEATHR     -0.210267 0.0252582 448 -8.324689  0.0000
## c.CHARGE      0.117485 0.0448649 448  2.618636  0.0091
## c.FIRE        0.093346 0.0247467 448  3.772040  0.0002
##  Correlation:
##              (Intr) l(DAMA l(BLOC c.DEAT c.CHAR
## log(DAMAGE)  -0.718
## log(BLOCKED) -0.398 -0.353
## c.DEATHR      0.565 -0.289 -0.378
## c.CHARGE      0.127 -0.168  0.050 -0.096
## c.FIRE        0.586 -0.583 -0.018  0.126 -0.139
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.97731170 -0.70409602  0.02687136  0.72099005  3.72464792
##
## Number of Observations: 547
## Number of Groups: 94
```

The package I am using:

I used the function "glmmPQL" from the package MASS. When using "glmer" function instead, I got a warning message saying that it fails to converge. It may be a technical issue from the package, but one should realize that the approximation method to build the model is based on the function I am applying. The major difference between glmer (which is provided by the package lme4) and glmmPQL (which relies on function lme, from the nlme pacakge) is that the parameter estimation algorithm used in nlme is not optimized for dealing with crossed random effects, which are associated with a sparse design matrix, while lme4 takes advantage of this structure. (Pinheiro & Bates, "Mixed-Effects Models in S and S-PLUS", Springer, 2000, pp. 163)

ii.Interpretation

```
The $\sigma x^2$ is 0.18, which measures the variability within groups.
```

The intercept is not meaningful because log(DAMAGE) will not be zero.

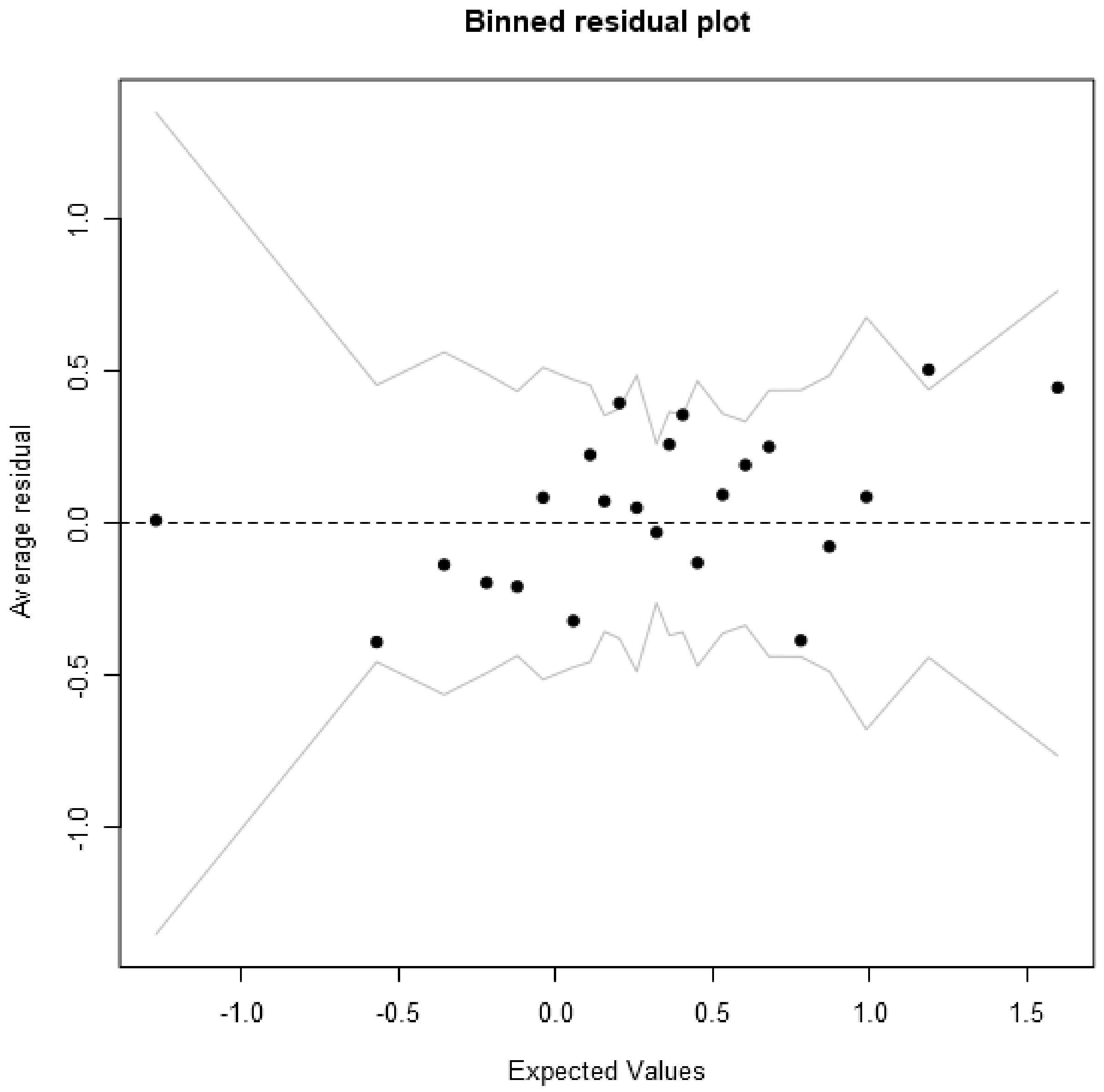
If you make a mistake in game and this leads you to die once more, your odds of win will decrease by $e^{0.2}-1=$22%. Similarly, we can calculate the odds change for other predictors. The increase or decrease in odds of wins seems large, as it will be super hard to increase your perfermance beyond even the average level of the best players. However, the magnitude of the parameters would give us a idea of which stats affect the win rate more.

iii.Model Checking

All the predictors are significant at 95% confidence level, as we expected.

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                     lower        est.      upper
## (Intercept) -14.77292759 -9.19168617 -3.6104448
## log(DAMAGE)   0.41029081  0.99473333  1.5791758
## c.DEATHR     -0.25385145 -0.20815704 -0.1624626
## c.CHARGE      0.02949707  0.11704202  0.2045870
## c.FIRE        0.04506042  0.09341309  0.1417658
## attr(,"label")
## [1] "Fixed effects:"
```

We apply binned residual plot instead of the simple residual plots because y has repeated patterns.



The binned residual looks good. There is a subtle right upper trend, but it is acceptable. One might notice that the expected values are kind of centered between 0.5-0.6, and this is because the current average win rate is ~55%. Usually the win rate would not be too high or too low.

Conclusion & Discussion

In conclusion, at a top level of Rein games, the effect of SOLO_KILLS, ULT and BLOCKED is not that significant, but a better use of FIRE & CHARGE, and keeping your death rate low with a relatively high damage are the direction we should strive for.

Appendix

Data Mining Instruction.rmd:https://github.com/xuz057/overwatch/blob/master/Data_Mining_Instruction.Rmd

The major difference between glmer and glmmPQL: Pinheiro & Bates, "Mixed-Effects Models in S and S-PLUS", Springer, 2000, pp. 163.