MA678 Project Report on
A study of Overwatch Statistics: Reinhardt

Name: Xuan Zhu

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 been 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 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 the future as most of us expect.

Introduction

Background Information: If you are a starter in Overwatch, for a good explanation of how Overwatch is played (e.g. how wins and losses are defined), please go to this website: https://www.wired.com/2017/01/overwatch-guide/

For evaluating the performance of a single hero, there are many statistics existing for your reference: Elimination/ Death Ratio, Win Rate, Damage, Damage Blocked, etc. As a main tank with strong control skills, Reinhardt does not only need to make and absorb damage as usual, but also to control as many as enemies he can. The three skills he has, Charge, Fire and the ultimate ability “Earth Shatter” are related to measuring the control abilities. Other statistics often indicate how well a tank can survive during the team fight. From the website www.overbuff.com (one of the most authorative statistics website for Overwatch), we can quickly grab the average stats of evaluating Reinhardt for each tier, separately. Take the highest tier Grandmaster (SR score>4000) as an example:

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%

Overview | Primary | Eliminations | Medals

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

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

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 summarizes the average level of players with >4000 SR score. In other words, some data used to calculate the means is from those who are not rein mains, but are forced to fill in the tank position if 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. We only want to study on the stats of pro Reinhardt players because their Reins are representative.

Method

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.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
Arty-1346	0.5000	3	3	9881	0.0	15130	1.63
Arty-1346	1.0000	1	0	30743	1.0	53074	2.29
Arty-1346	0.2500	1	3	9672	0.0	17676	2.35
Arty-1346	0.6667	2	1	10031	0.0	10663	2.75
Arty-1346	0.5333	8	7	12190	0.4	17895	2.46
Arty-1346	1.0000	2	0	13080	0.5	12870	2.72

In total, there are 547 observations of 23 variables. 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 the variables we are interested in:

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

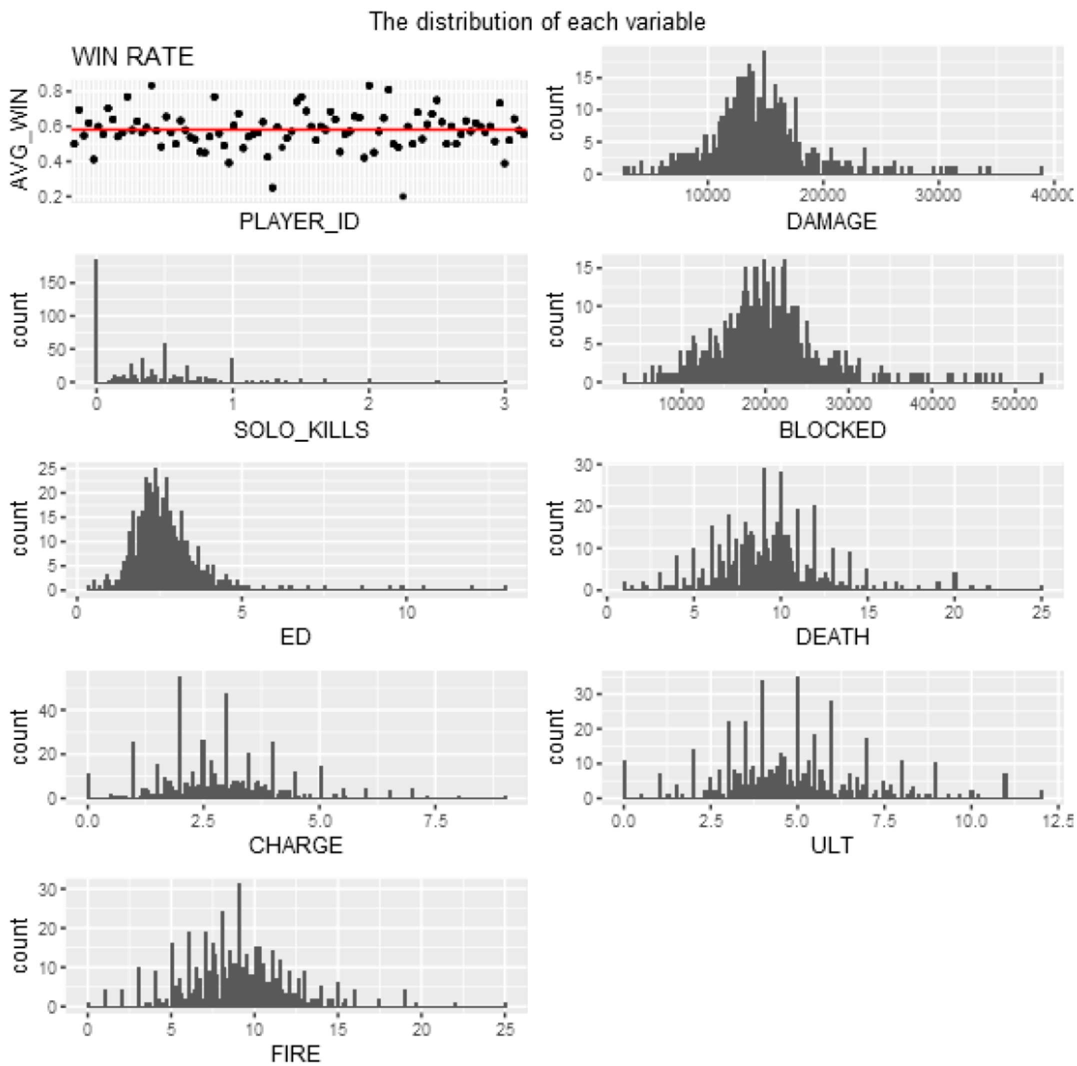
DEATH | The death rate per game. Why death is also important rather than simply picking ED(Killing/Death ratio) as the predictor? Because killing enemies is not the only job a tank is expected to do from a team perspective. Sometimes Rein's teammates need Rein to protect his teammate by shield first. 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. Higher kills gained by using this skill indicate a better master of Rein.

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

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

Basic Data Visualization & Exploratory Data Analysis: First, we take a look at the distribution of each variable in the dataset.

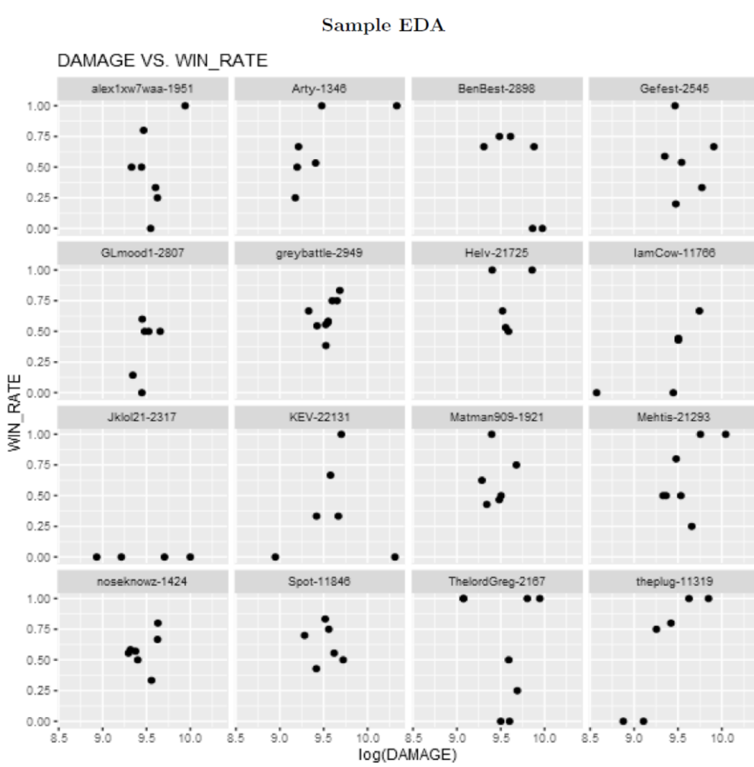


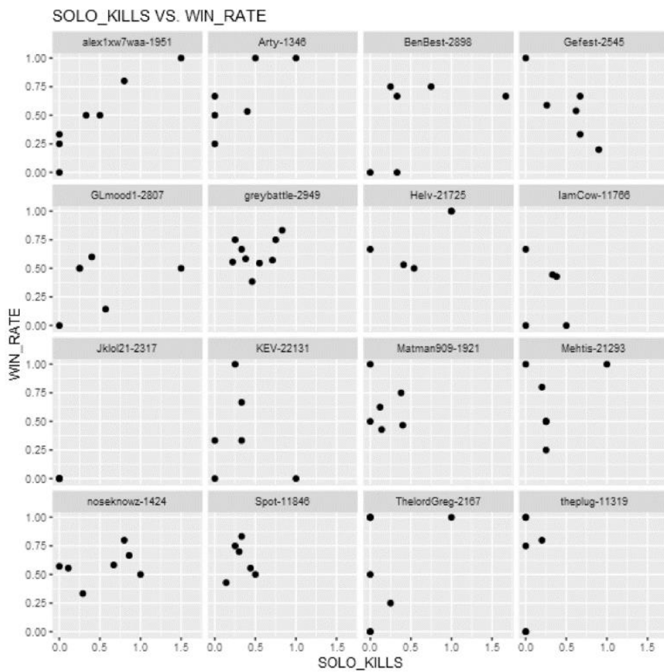
The average win rate for each player is calculated by Sum (Wins)/ Sum (Game Played). The red line in the plot represents the average level of all players, which is about 58%. So the individual win rate is distributed randomly around the average line. One may notice that there are outliers.

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 unlucky guys rarely win within several days.

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.





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

Model Used: The model used for analysis is a multilevel logistic model, which takes the variability of personal performance into account.

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

Result

Model Choice： The output from R is as the following.

```
## 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
```

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 ultimate skill 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 not actually as important as we usually think. (p=0.8337>0.05)

```
##   Structure: fixed weights
##   Formula: ~invwt
## Fixed effects: cbind(WIN, LOSS) ~ log(DAMAGE) + log(BLOCKED) + c.DEATHR + c.CH
##                   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:
```

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* package) 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)

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.

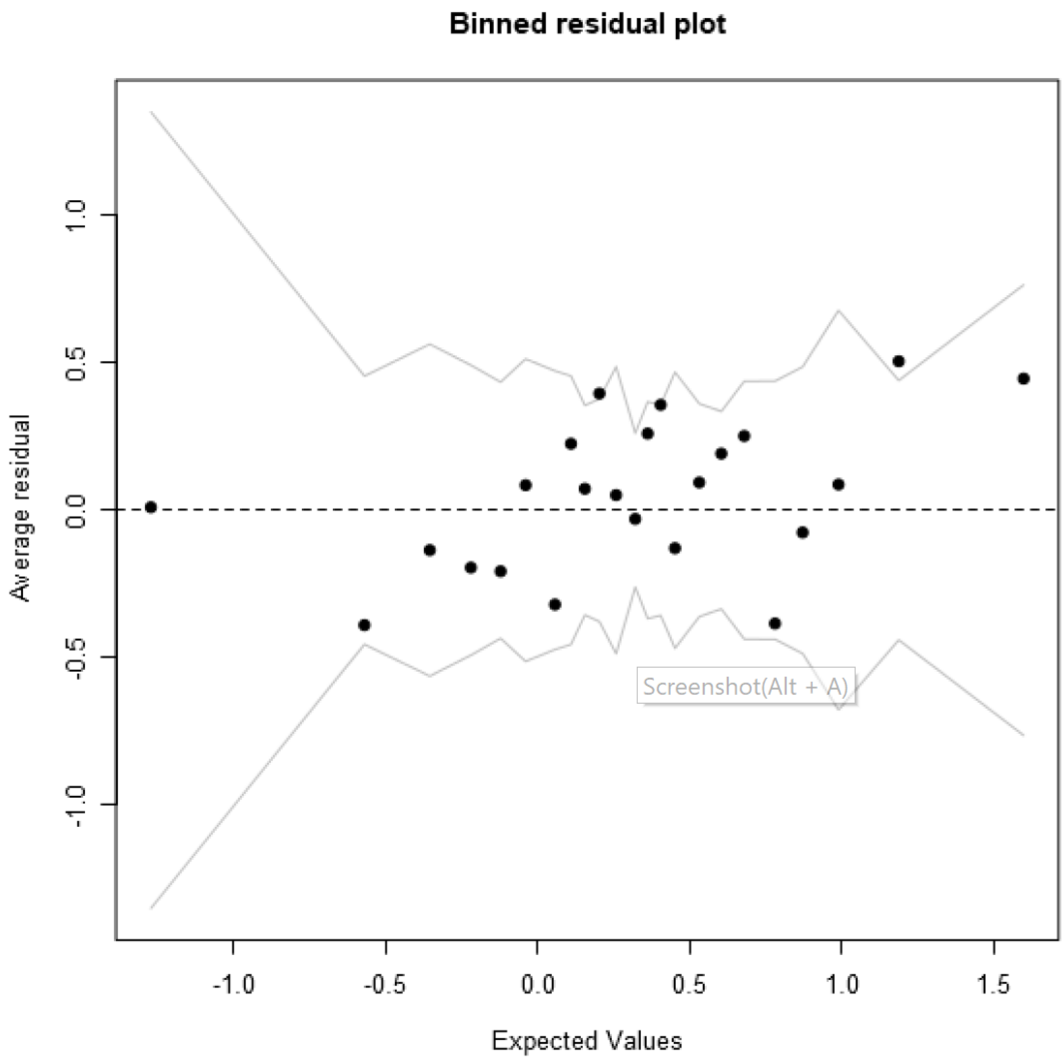
If you make a mistake in game and this leads you to die once more, your odds of win will decrease by exp (0.2) -1=22%. Similarly, we can calculate the odds change for other predictors. The increase or decrease in odds of wins seems large, as it will be super hard to increase your performance beyond even 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.

Model Checking: All the predictors are significant at 95% confidence level, as we expected. No Confidence Interval contains 0.

```
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
```

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

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.