Forecasting League of Legends Player Performance through In-Game Metrics and Key Statistics*

A Hybrid Model Predict the KDA for Bilibili Gaming (BLG) team players

Colin Sihan Yang

November 22, 2024

This paper develops a hybrid model to forecast the 2024 performance of Bilibili Gaming (BLG) team players by analyzing team statistics from the League of Legends Pro League (LPL). The model focuses on predicting the Kill-Death-Assist ratio (KDA) for individual players across different games, leveraging both traditional in-game metrics such as gold difference, kills, assists, and vision score, and advanced variables like total creep score (total CS) to provide a comprehensive view of player performance. By integrating machine learning techniques with esports-specific insights, this model forecasting player KDAs, identifying performance trends, but also offers a clear assessment of individual contributions to the team. It highlights top-performing players while identifying those whose performance may negatively impact the team's success. The findings offer actionable insights, enabling the BLG organization to make informed decisions on player contracts.

1 Introduction

In recent years, esports has evolved into a global phenomenon, with professional gaming leagues such as the League of Legends Pro League (LPL) attracting millions of viewers and significant organizational investments. As the competitive landscape intensifies, teams and organizations increasingly rely on data-driven approaches to gain a strategic edge. Among these, predictive analytics has emerged as a powerful tool to evaluate player performance, optimize team strategies, and make informed management decisions.

*Code and data are available at: https://github.com/hoodiexxx/lol_win_rate_prediction.

Bilibili Gaming (BLG), a prominent team in the LPL, faces the challenge of maintaining a competitive roster in the ever-changing meta of League of Legends. Accurate performance forecasting is critical for assessing individual player contributions, identifying areas for improvement, and making decisions about player contracts and roster changes.

This paper developing a hybrid model that leverages both traditional in-game metrics and advanced variables, such as total creep score (CS), to forecast player performance for BLG in the 2024 season. The model specifically focuses on predicting the Kill-Death-Assist ratio (KDA), a widely accepted indicator of individual performance, across various matches. Finally offers actionable insights into player roles and impacts, highlighting those who can consistently carry games and identifying players whose performance may not justify their perceived value.

The remainder of this paper is structured as follows: Section 2 provides an overview of the data.

the data gathering and analysis is done in R (R Core Team 2023) with the following packages: knitr (Xie 2014), tidyverse (Wickham et al. 2019), ggplot2 (Wickham 2016), dplyr (Wickham et al. 2023), arrow (Richardson et al. 2024), here (Müller 2020), and Rstan (Stan Development Team 2024)

1.1 Estimand

the estimand for this research paper is the predicted Kill-Death-Assist ratio (KDA) for the BLG team players. The prediction is based on quantifying various in-game factors, including total cs, gold earned, vision score, damage to champion, which are used as predictors.

2 Data

2.1 Measurement

The dataset we obtain from oracleselixir (n.d.) is accumlated from multiple league of legend esport league like LCS, LEC, LCK, LPL, and the rest of global pro LoL.

The dataset utilized in this study was compiled from multiple professional League of Legends (LoL) esports leagues, including the LCS (North America), LEC (Europe), LCK (South Korea), LPL (China), and other global professional leagues. The data aggregation spans from 2023-2024 (23-24 season) and encompasses a wide variety of matches played at different competitive levels. This diversity ensures a comprehensive representation of professional gameplay styles, strategies, and performances across regions.

Each data point corresponds to a single game of one player, identified by a unique gameid, teamid and playerid and includes contextual information such as the league, split (Spring, Summer, or playoffs), split (Winter, Sping, etc), and year. The dataset captures a rich array of in-game metrics and performance indicators for teams and individual players. Team-level data includes variables such as match result (win/loss), game duration (in minutes and seconds), and pivotal game objectives (e.g., First Blood, First Tower, Baron Nashor, and Dragon counts). These metrics are critical for understanding team performance and macro-level strategies.

Player-level statistics include detailed in-game actions such as kills, deaths, assists, gold earned, damage dealt, and wards placed or cleared. These variables allow for a granular examination of individual contributions to a team's success. Additionally, positional data (e.g., top lane, jungle, mid lane, ADC, and support) is included, enabling role-specific analyses. The dataset also tracks champion selections and bans, providing insights into meta trends and drafting strategies over time.

To ensure consistency, all metrics are measured using standardized definitions as provided by the data source, oracleselixir (n.d.). This source is widely regarded as reliable within the esports analytics community. The dataset is structured to facilitate longitudinal analyses, enabling researchers to explore trends and patterns across regions, seasons, and competitive tiers. Furthermore, derived metrics, such as the champion xp difference at 25 minutes (xpdiffat25), Gold Difference at 25 minutes (golddiffat25), and Damage Per Minute (dpm), are calculated to provide advanced insights into player and team efficiency.

These variables combined allow researchers to reliably analyze and predict the 23-24 season LPL BLG team's player performance.

However there are limitations in these measurements.The dataset combines data from multiple leagues (LCS, LEC, LCK, LPL, and others), each of which has distinct playstyles, metas, and competitive environments. These regional differences might introduce biases, as strategies or performance metrics effective in one league may not translate directly to another. For instance, the LCK may prioritize slower, macro-focused playstyles, whereas the LPL might exhibit faster, skirmish-heavy games. This variation could confound cross-regional comparisons unless properly controlled for (Tan 2020).

2.2 Raw Data

The raw data from oracleselixir@oracleselixir_oracles_nodate contains 52 columns, all the column headers are displayed below:

gameid	datacompleteness	url
league	year	split
playoffs	date	game
patch	participantid	side
position	playername	playerid
teamname	teamid	champion
ban1	ban2	ban3
ban4	ban5	pick1
pick2	pick3	pick4
pick5	gamelength	result
kills	deaths	assists
teamkills	teamdeaths	doublekills
triplekills	quadrakills	pentakills
firstblood	firstbloodkill	firstbloodassist
firstbloodvictim	team kpm	ckpm
firstdragon	dragons	opp_dragons
elementaldrakes	opp_elementaldrakes	infernals
mountains	clouds	oceans
chemtechs	hextechs	dragons (type unknown)
elders	opp_elders	firstherald
heralds	opp_heralds	void_grubs
opp_void_grubs	firstbaron	barons
opp_barons	firsttower	towers
opp_towers	firstmidtower	firsttothreetowers
turretplates	opp_turretplates	inhibitors
opp_inhibitors	damagetochampions	dpm
damageshare	damagetakenperminute	damagemitigatedperminute
wardsplaced	wpm	wardskilled
wcpm	controlwardsbought	visionscore
vspm	totalgold	earnedgold
earned gpm	earnedgoldshare	goldspent
gspd	gpr	total cs
minionkills	monsterkills	monsterkillsownjungle
monsterkillsenemyjungle	cspm	goldat10
xpat10	csat10	opp_goldat10
opp_xpat10	opp_csat10	golddiffat10
xpdiffat10	csdiffat10	killsat10
assistsat10	deathsat10	opp_killsat10
opp_assistsat10	opp_deathsat10	goldat15
xpat15	csat15	opp_goldat15
opp_xpat15	opp_csat15	golddiffat15
xpdiffat15	csdiffat15	killsat15
assistsat15	deathsat15	opp_killsat15
opp_assistsat15	opp_deathsat15	goldat20
xpat20	csat20	opp_goldat20
opp_xpat20	opp_csat20	golddiffat20
xpdiffat20	csdiffat20	killsat20
assistsat20	deathsat20	opp_killsat20
opp_assistsat20	opp_deathsat20	goldat25

These columns can be categorized into three types, elements of response variable, numeric predictors, and categorical predictors.

the elements of response variable are: - kills: The number of enemy champions defeated by the player. - deaths: The number of times the player was defeated by opponents. - assists: The number of assists a player contributes to a teammate's kills.

These three attributes will later be combined to calculate the Kill-Death-Assist ratio (KDA), which serves as the response variable for the model in the cleaned data section Section 2.3.

example of Numeric Predictors are: - totalgold: Total gold earned by the player during the game. - total.cs: Total number of minions or monsters killed by the player. - visionscore: A composite measure of vision provided and vision denied. - damagetochampions: Total damage inflicted by the player on opponents. - golddiffat25: Gold advantage or deficit compared to the opposing player/team at the 25-minute mark.

example of categorical predictors are: - position: The player's role in the game (e.g., top, jungle, mid, ADC, support). - league: The league where the match was played (e.g., LPL, LCK). - side: team or the blue side or the red side

2.3 Cleaned Data

The cleaned dataset used in this study focuses on matches from the League of Legends Pro League (LPL) involving Bilibili Gaming (BLG) during the 2024 season. To ensure the data aligns with the research objectives, the dataset was filtered to include only games played by BLG, with key variables retained for performance analysis. The cleaning process involved removing rows with missing values and selecting relevant columns, including contextual information such as game ID, league, year, split, and side, as well as in-game performance metrics like kills, deaths, assists, total gold, and vision score. A critical addition to the dataset was the calculation of the Kill-Death-Assist ratio (KDA), a widely used indicator of individual performance. The KDA was derived using the formula:

$$KDA = \frac{Kills + Assists}{Max(1, Deaths)} \tag{1}$$

ensuring robustness even when deaths were zero.

and we mutate the gamelength unit from second into minute.

The cleaned dataset was saved in Parquet formats to support flexibility in analysis and compatibility with various tools. This focused and structured dataset ensures reliability and completeness, providing a robust foundation for forecasting player performance, analyzing trends, and identifying key contributors to the analysis of BLG's players performance.

The first 6 rows of the dataset are displayed in Table 1.

Table 1: Sample of cleaned league of legend esport data

side	position	playername	champion	kills	deaths	assists	visionscore	totalgold	total.cs	damagetochampions	KDA	gamelength	result
Blue	top	Bin	K'Sante	1	0	6	21	10913	251	10197	7.0	27.05000	1
Blue	jng	Xun	Brand	8	0	8	34	12311	200	18832	16.0	27.05000	1
Blue	mid	knight	LeBlanc	4	0	6	19	13156	247	26196	10.0	27.05000	1
Blue	bot	Elk	Varus	1	2	8	34	11495	244	12877	4.5	27.05000	1
Blue	sup	ON	Ashe	3	1	10	140	9104	43	10592	13.0	27.05000	1
Red	top	Bin	Aatrox	0	2	2	35	11821	319	17950	1.0	30.56667	0

2.4 DATA Exploration

2.4.1 Champion Pool for players

Table 2: the table that shows the champion pool for the BLG players

Table 2: Top 5 Most Played Champions by Player with Win Rate

Player Name	Champion	Position	Count	Win Rate
Bin	Renekton	top	18	0.78
Bin	Kennen	top	12	0.92
Bin	Gnar	top	11	0.73
Bin	K'Sante	top	11	0.55
Bin	Aatrox	top	10	0.70
Elk	Kai'Sa	bot	10	0.80
Elk	Kalista	bot	10	0.90
Elk	Lucian	bot	10	0.90
Elk	Ezreal	bot	9	0.78
Elk	Senna	bot	9	0.78
ON	Rell	sup	15	0.80
ON	Rakan	sup	13	0.69
ON	Nautilus	sup	11	0.73
ON	Alistar	sup	9	1.00
ON	Renata Glasc	sup	9	0.78
Wei	Sejuani	jng	6	0.83
Wei	Maokai	jng	5	1.00
Wei	Lillia	jng	4	1.00
Wei	Zyra	jng	4	0.75
Wei	Brand	jng	2	1.00

Player Name	Champion	Position	Count	Win Rate
Xun	Kindred	jng	12	0.83
Xun	Vi	jng	11	0.55
Xun	Xin Zhao	jng	8	0.88
Xun	Brand	jng	6	0.50
Xun	Lee Sin	jng	6	0.83
knight	Corki	mid	14	0.93
knight	Yone	mid	10	0.70
knight	Ahri	mid	8	0.88
knight	Karma	mid	7	0.86
knight	Neeko	mid	7	1.00
knight	Taliyah	mid	7	0.86
knight	Tristana	mid	7	0.57

The table Table 2 summarizes the top 5 most played champions for each Bilibili Gaming (BLG) player, including their positions, usage counts, and win rates. It highlights the players' preferences and effectiveness with specific champions, offering insights into individual strengths and team strategies.

- Bin (top lane) favors durable, aggressive champions like Renekton (18 games, 77.8% win rate) and Kennen (12 games, 91.7%). While most of his champions are highly effective, K'Sante has a lower win rate of 54.5%, suggesting Bin may not good at Tank champion.
- Elk (bot lane) excels on meta ADCs like Kai'Sa, Kalista, and Lucian, all boasting win rates of 80-90%, with consistent performance on utility champions like Senna and Ezreal (77.8%).
- ON (support) shines with engage-heavy champions such as Rell (80%) and Rakan (69.2%), but his Alistar stands out with a perfect 100% win rate, indicating strong synergy in specific scenarios. Wei (substitute jungle) shows exceptional results on niche picks like Maokai, Lillia, and Brand, each with 100% win rates, while his most-played Sejuani achieves 83.3%.
- Xun (main jungle) demonstrates a balanced pool with high win rates on Kindred (83.3%) and Xin Zhao (87.5%), though his performance on Vi (54.5%) is less consistent. Knight (mid lane) showcases versatility, excelling on Corki (92.9%) and Neeko (100%), but his Tristana struggles with a 57.1% win rate.

Wei's (Secondary jungle) limited champion pool suggests fewer games played, with Sejuani (6 games) and Maokai (5 games) leading the count. Wei appears to specialize in tanky jungle champions and niche picks like Zyra and Brand. it may shows BLG want wei play tank champion to protect other teammates.

Overall, most players exhibit strong performance with their preferred champions, though some picks, like K'Sante and Vi, show room for improvement. These insights can guide BLG's drafting strategies and optimize player-champion pairings for better outcomes in competitive play.

2.4.2 Players VisionScore

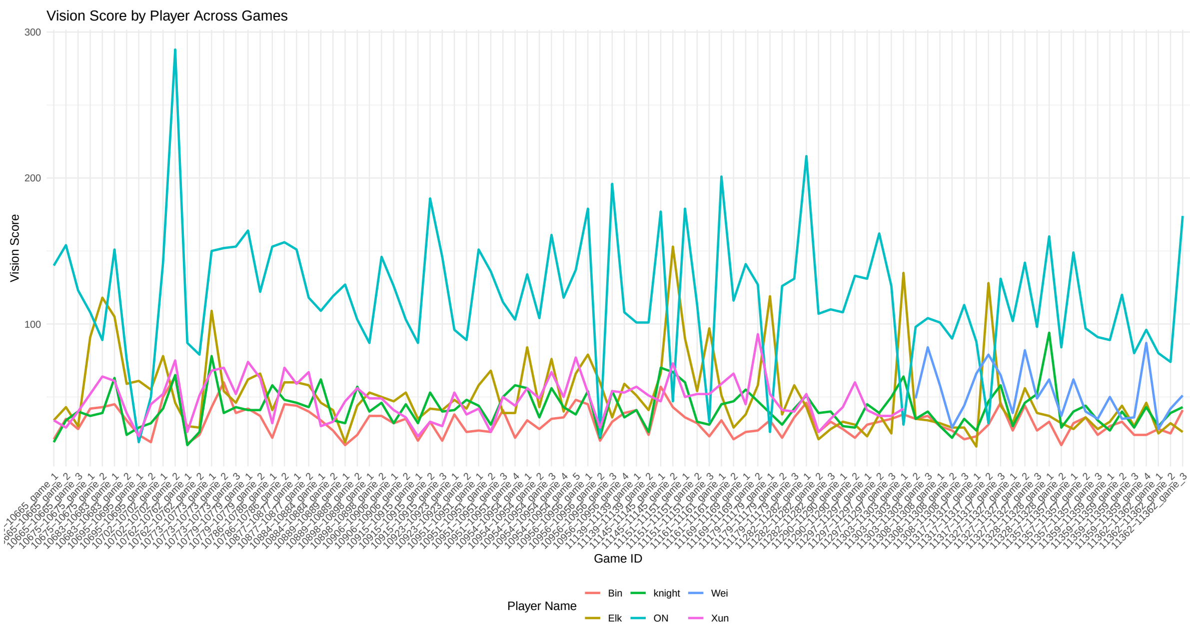


Figure 1: BLG players visionscore over game

The line chart Figure 1 visualizes the vision scores of individual players on the Bilibili Gaming (BLG) team across multiple games, offering insights into their vision control performance and consistency. Each line represents a player's cumulative vision score over the course of several matches, helping to compare and analyze their contributions to map awareness and team strategy.

Key observations highlight that ON, the team's support player, consistently records the highest vision scores across games. This is expected, as supports typically focus on warding and vision control, critical for map dominance. ELK, the ADC, also shows considerable vision score variability. Other roles, such as Bin (top lane), and Knight (mid lane), have lower and steadier vision scores, aligning with their roles' primary focus on laning and damage output rather than vision control.

2.4.3 Game duration

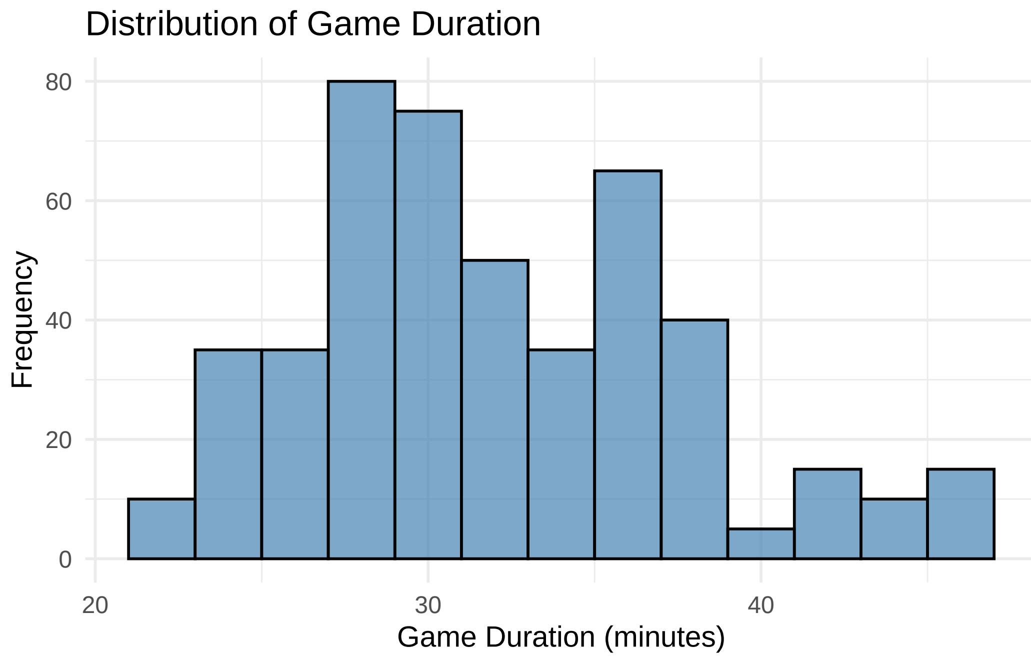


Figure 2: BLG team generally end the game around 30 minutes

The histogram Figure 2 depicts the distribution of game durations for the Bilibili Gaming (BLG) team, measured in minutes. The graph reveals that most games tend to last between 28 to 36 minutes, with a peak frequency around the 30-minute mark. This suggests that BLG typically ends their games within the expected duration for competitive matches.

2.4.4 players gold earned and players damage

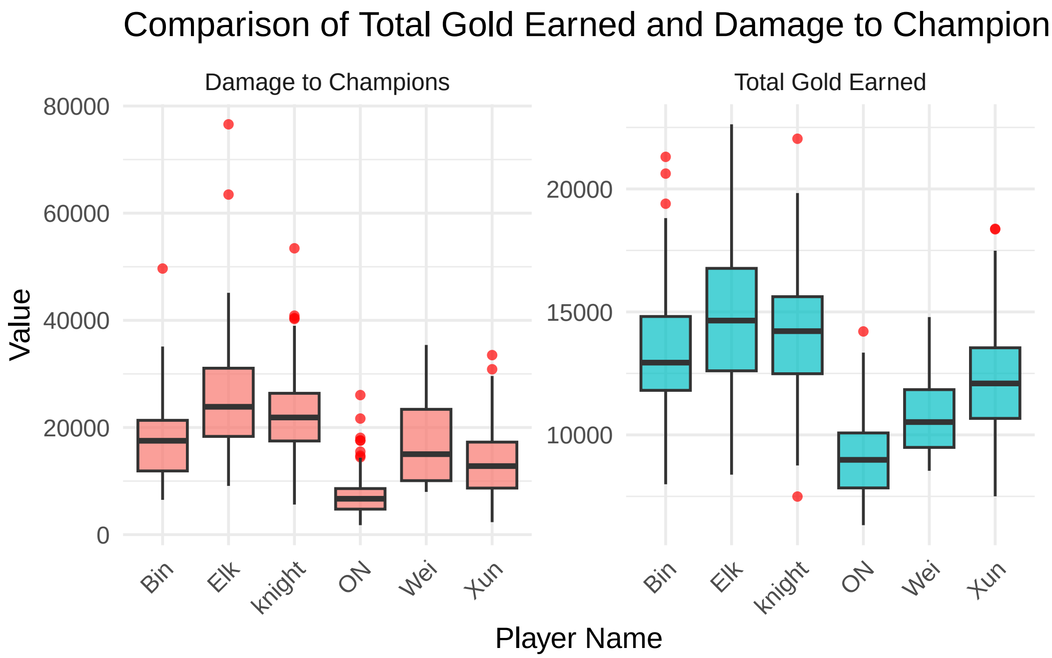


Figure 3: In BLG team, ELK is the highest damage and gold earned player.

The boxplots Figure 3 compare the total gold earned and damage to champions by each player on the Bilibili Gaming (BLG) team, offering insights into their in-game contributions and roles.

Key observations highlight that Elk, the ADC, consistently earns the most gold and deals the highest damage to champions. This reflects the ADC role's reliance on gold to build damage-focused items and their critical role in team fights. However, Wei stands out by earning a comparatively high amount of gold while contributing less damage, suggesting a potential mismatch between resource allocation and damage output. This could indicate a focus on tank or utility-oriented champions or inefficiency in converting gold into impactful performance.

3 Model

The goal of our modelling strategy is twofold. Firstly, we aim to accurately predict the player KDA performance based on relevant game data and key influencing factors. Secondly, we seek to evaluate the efficacy of different modeling approaches—from simple linear regression (SLR) to multiple linear regression (MLR) and Bayesian hierarchical models—to understand their predictive capabilities and assess the underlying relationships between variables. By comparing these models, we can determine which approach provides the most robust and reliable predictions, while considering the variability and potential uncertainty in the data.

Here we briefly describe the Bayesian analysis model used to investigate… Background details and diagnostics are included in Appendix B.

3.1 Precaution for Building Model

According to the KDA formula in cleaned data section Section 2.3, the response variable KDA is directly depend on three attributes, kills, deaths and assists, so we cannot use kills, deaths and assists as predictors in our model formula Since the response variable is mathematically derived from these three attributes, including kills, deaths, and assists as predictors in the model would create perfect multicollinearity. This means the predictors would have a deterministic relationship with the response variable, violating the assumption of independence between the predictors and the response variable in regression models. For a Bayesian model, this also violates the assumption that the prior information (or likelihood) is not over-specified. In Bayesian inference, incorporating variables that deterministically define the response variable introduces redundancy and inflates the certainty of the posterior distribution, making the model overly confident and prone to poor generalization.

3.2 Basic Simple Linear Regresion Model

3.2.1 Basic Simple Linear Regression Model Set-Up

$$Y = \beta_0 + \beta_1 \cdot X + \epsilon$$

Where:

- Y: The dependent variable (response variable) you are trying to predict.
- X: The independent variable (predictor variable) used to predict Y, in this case, X represent the predictor variable - total CS.
- β_0: The intercept of the regression line, representing the value of Y when $X = 0$.

- β_1: The slope of the regression line, representing the change in Y for a one-unit increase in X.
- ϵ: The error term, accounting for the variability in Y that X does not explain.

3.2.2 SLR Model

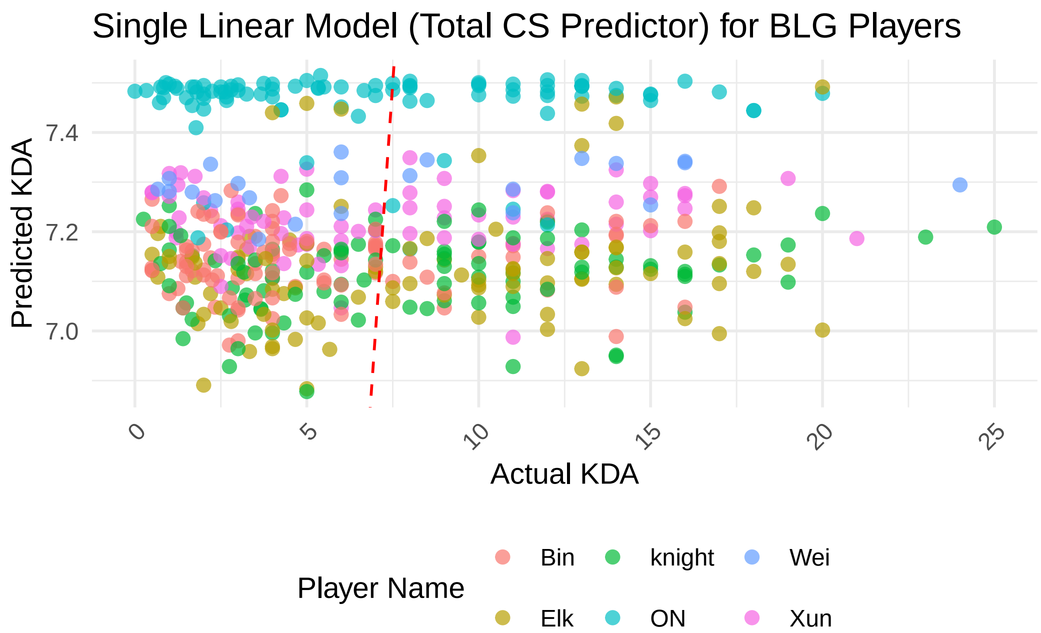
In this SLR model, the response variable is KDA and the only one predictor is the total CS. Figure 4a visualizes the relationship between the actual and predicted values of KDA for BLG players, based on a simple linear regression model with total cs as the sole predictor. eaach points represent individual comparisons between actual and predicted values, with the color of the dots representing different players. The red dashed line represents the line of perfect prediction, where actual values would equal predicted values.

Figure 4b examines the relationship between total CS and KDA for BLG players, with each point representing a game and each player distinguished by color. Separate regression lines for each player highlight role-specific trends.

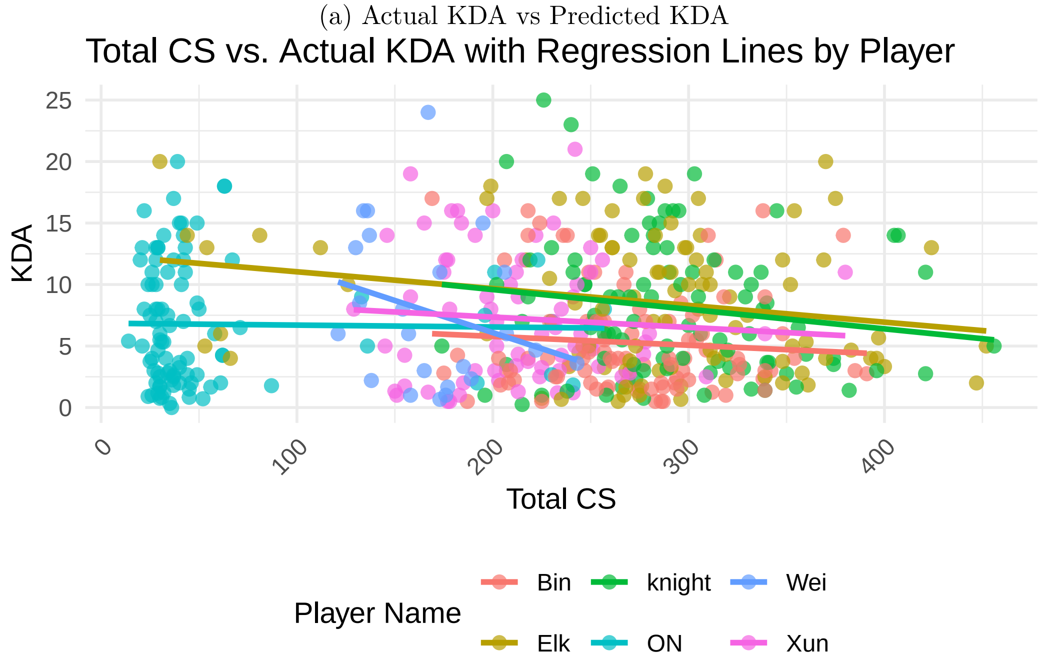
The primary concern lies in the evident dispersion of data points, which are widely spread and do not cluster closely around the line of perfect prediction (the dashed red line). This suggestslimited dependence of KDA on total CS, total CS does not adequately explain the variability in KDA. The observed inconsistencies between actual and predicted values indicate that the relationship between KDA and total CS is not sufficiently captured by a linear model with just one predictor.

3.2.3 Compute R Squared for SLR Model

$$SS_{\text{total}} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$
$$SS_{\text{residual}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
$$R^2 = 1 - \frac{SS_{\text{residual}}}{SS_{\text{total}}}$$
$$\text{R-squared} = 0.0008659$$



(a) Actual KDA vs Predicted KDA



(b) predictor total CS vs Actual KDA with linear regression line

Figure 4: total cs has limited predictive accuracy for league of legend esport players KDA performance.

3.3 Multiple Linear Regression Model

3.3.1 Multiple Linear Regression Model Set-up

The formula is:

$$Y = \beta_0 + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6 + \epsilon$$

Where:

- Y: The dependent variable (response variable), representing the KDA.
- X_1: Side (categorical variable: Blue or Red), representing the team side during the game.
- X_2: Vision Score, measuring the player's vision contribution during the game.
- X_3: Total Gold, representing the total gold earned by the player.
- X_4: Total CS, representing the total creep score of the player.
- X_5: Damage to Champions, indicating the total damage dealt to enemy champions.
- X_6: Game Duration, representing the length of the game in seconds.
- β_0: The intercept, representing the expected value of Y when all predictors are 0.
- $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6$: The coefficients, representing the change in Y for a one-unit increase in the corresponding predictor while holding other predictors constant.
- ϵ: The error term, accounting for the variability in Y not explained by the predictors.

This multiple linear regression model accounts for both categorical and continuous variables to predict the player's **KDA**, capturing the combined influence of gameplay factors such as team side, vision control, resource management, damage output, and game duration.

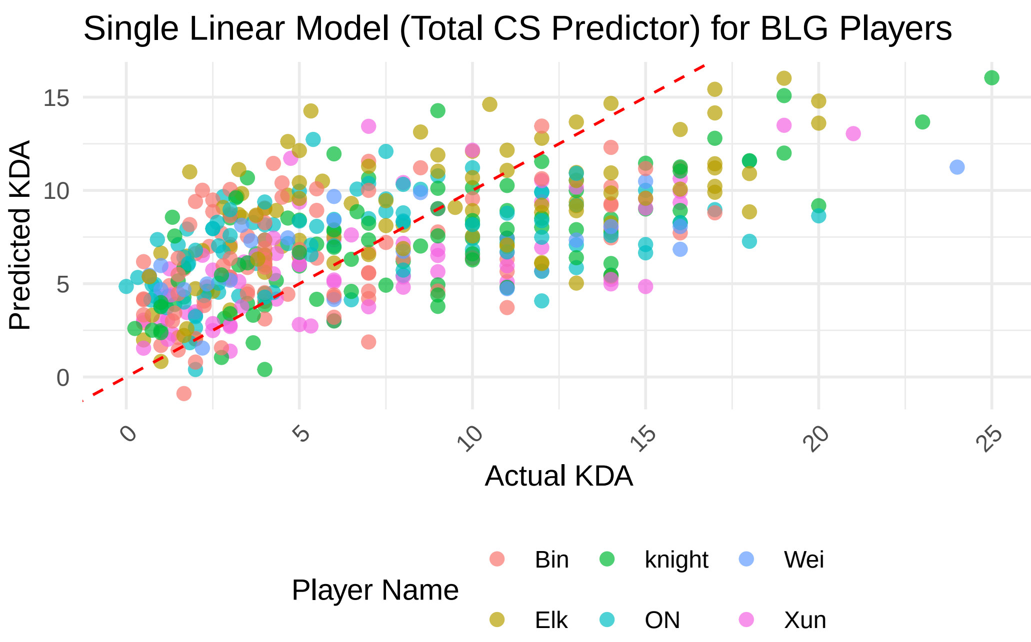
3.3.2 MLR

```
model_MLR = lm(KDA ~ side + visionscore + totalgold + total.cs + damagetochampions + gameleng

summary(model_MLR)
```

```
Call:
lm(formula = KDA ~ side + visionscore + totalgold + total.cs +
    damagetochampions + gamelength, data = cleaned_lol_data)

Residuals:
   Min     1Q Median     3Q    Max
-9.162 -3.081 -0.809  3.071 12.751

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        1.091e+01  1.219e+00   8.948   <2e-16 ***
sideRed           -1.831e-01  4.007e-01  -0.457    0.648
visionscore       -6.786e-03  9.706e-03  -0.699    0.485
totalgold          2.160e-03  1.833e-04  11.783   <2e-16 ***
total.cs          -4.659e-02  5.665e-03  -8.224    2e-15 ***
damagetochampions -3.080e-05  3.082e-05  -0.999    0.318
gamelength        -6.189e-01  5.450e-02 -11.355   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.294 on 463 degrees of freedom
Multiple R-squared:  0.337, Adjusted R-squared:  0.3284
F-statistic: 39.23 on 6 and 463 DF,  p-value: < 2.2e-16
```

```r
model_MLR_KDA_predictions = predict(model_MLR, cleaned_lol_data)

model_MLR_visulization_data <- cleaned_lol_data |>
  mutate(KDA_predictions = predict(lm_model1),
         playername = as.factor(playername)) # Ensure player names are treated as factors

ggplot(model_MLR_visulization_data, aes(x = KDA, y = model_MLR_KDA_predictions, color = playe
  geom_point(size = 2, alpha = 0.7) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(
    title = "Single Linear Model (Total CS Predictor) for BLG Players",
    x = "Actual KDA",
    y = "Predicted KDA",
    color = "Player Name"
  ) +
  theme_minimal() +
  theme(
```

```
    legend.position = "bottom",
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```




```
# Multiple R-squared:  0.337
```

3.4 Bayesian Model set-up

The Bayesian model is implemented in R (R Core Team 2023) using the rstanarm package as described by Brilleman et al. (2018). We run the model with the following specifications:

- Formula: pct $\sim$ pollscore + days taken from election + sample size + (1|methodology) + (1|state)
- Priors: Normal(0, 2.5) for all coefficients and intercept, Exponential(1) for σ
- Settings: Seed = 123, Cores = 4, Adapt delta = 0.95

We run the model in R (R Core Team 2023) using the `rstanarm` package of Brilleman et al. (2018). We use the default priors from `rstanarm`.

3.5 Model set-up

Define y_i as the number of seconds that the plane remained aloft. Then β_i is the wing width and γ_i is the wing length, both measured in millimeters.

$$
\begin{aligned}
y_i|\mu_i, \sigma &\sim \text{Normal}(\mu_i, \sigma) && (2)\\
\mu_i &= \alpha + \beta_i + \gamma_i && (3)\\
\alpha &\sim \text{Normal}(0, 2.5) && (4)\\
\beta &\sim \text{Normal}(0, 2.5) && (5)\\
\gamma &\sim \text{Normal}(0, 2.5) && (6)\\
\sigma &\sim \text{Exponential}(1) && (7)
\end{aligned}
$$

We run the model in R (R Core Team 2023) using the `rstanarm` package of Brilleman et al. (2018). We use the default priors from `rstanarm`.

3.5.1 Model justification

We expect a positive relationship between the size of the wings and time spent aloft. In particular…

We can use maths by including latex between dollar signs, for instance θ.

4 Results

Our results are summarized in **?@tbl-modelresults**.

5 Discussion

5.1 First discussion point

If my paper were 10 pages, then should be be at least 2.5 pages. The discussion is a chance to show off what you know and what you learnt from all this.

5.2 Second discussion point

Please don't use these as sub-heading labels - change them to be what your point actually is.

5.3 Third discussion point

5.4 Weaknesses and next steps

Weaknesses and next steps should also be included.

Appendix

A Additional data details

B Model details

B.1 Posterior predictive check

In **?@fig-ppcheckandposteriorvsprior-1** we implement a posterior predictive check. This shows…

In **?@fig-ppcheckandposteriorvsprior-2** we compare the posterior with the prior. This shows…

B.2 Diagnostics

?@fig-stanareyouokay-1 is a trace plot. It shows… This suggests…

?@fig-stanareyouokay-2 is a Rhat plot. It shows… This suggests…

References

Brilleman, SL, MJ Crowther, M Moreno-Betancur, J Buros Novik, and R Wolfe. 2018. "Joint Longitudinal and Time-to-Event Models via Stan." https://github.com/stan-dev/stancon_talks/.

Müller, Kirill. 2020. *Here: A Simpler Way to Find Your Files*. https://CRAN.R-project.org/package=here.

oracleselixir. n.d. "Oracle's Elixir - LoL Esports Stats." Accessed November 21, 2024. https://oracleselixir.com/tools/downloads.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Richardson, Neal, Ian Cook, Nic Crane, Dewey Dunnington, Romain François, Jonathan Keane, Dragoș Moldovan-Grünfeld, Jeroen Ooms, Jacob Wujciak-Jens, and Apache Arrow. 2024. *Arrow: Integration to 'Apache' 'Arrow'*. https://CRAN.R-project.org/package=arrow.

Stan Development Team. 2024. "RStan: The R Interface to Stan." https://mc-stan.org/.

Tan, Amanda "Tania Mae". 2020. "Aggression or Control: A Deep Dive into the Huge Difference Between the LPL and LCK | ONE Esports." https://www.oneesports.gg/league-of-legends/aggression-or-control-a-deep-dive-into-the-huge-difference-between-the-lpl-and-lck/.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Romain François, Lionel Henry, Kirill Müller, and Davis Vaughan. 2023. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC.