Group Project Report

Group 126

I Introduction

In the project, Group 126 applied a variety of data wrangling and analysis techniques on a League of Legends dataset. The aim of the project is to answer the following research question:

- How do pre-game choices and configurations affect in-game performance?

In the following report, Group 126 will describe the techniques they used, as well as their thought process and final conclusions regarding the research question.

II Dataset

The dataset contains the scores and statistics of many unique matches and players from high-level matches from the popular video game league of legends. There are three sets of 5000 entries, each set from a different region, being Korea, EU and North America. The type of data ranges from categorical choices made by players before the match starts such as which champion they will play, the role they will fulfill and the spells they have equipped, as well as numerical statistics to do with in-game performance by that player for that match.

These statistics provide the ability to investigate the performance of players based on which champions they pick, which team they are on and more. Data such as kda gives insight into how many kills the player got compared to how many times they got kills, furthermore metrics like gold earned or total damage done are good indicators of overall performance.

III Target Audience

The target audience of this project includes league of legends players, particularly those who are in the higher ranks or professional play and those of them who are interested in statistics. This is because it will give an insight into what decisions can be made before a game to improve their likelihood of performing well in their matches.

With this information, they can hopefully expect to discover winning strategies and builds that they can implement to get an edge over their competitors.

IV Data Wrangling

int his section we attempted to make the data more accessible and valuable to our research.

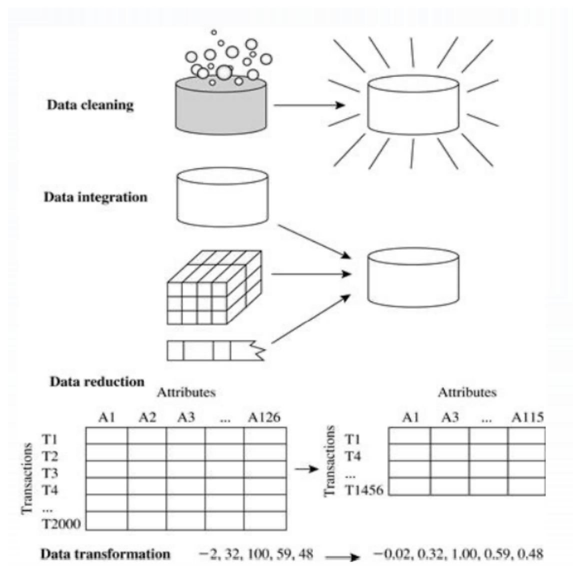


Fig. 1: Data wrangling process , Han et al 2012[1]

A. Data Cleaning

Imputation - To address missing values in the data, which are displayed as 'not a number' or NaN, we used a multivariate imputation to predict values based on patterns in multiple columns of data. This was implemented with the sklearn KNNImputer with K set to 3. This uses the K nearest neighbours algorithm, which determines missing values based on the L most similar to other rows.

Missing values - The KDA formula was used to fill in missing values in the KDA, kills, assists or deaths columns, if only one column was missing. We also used this formula after filling in one by one of these rows to make sure that the columns retained the relationship after imputation.

- KDA = (kills + assists) / (deaths + (deaths == 0))

Data splitting - TopLane_Jungle was split into its two respective roles, rather than being kept as a group. We were able to determine the role based on whether or not the player used the spell smite, as it is a definitive indicator of a jungle player. This was done so we could more accurately recommend what role a player should fulfil based on that role's performance.

Scaling -All of the entries were scaled up as if the player reached level 18 during each match. This was to ensure that the length of the game did not come into account for performance, as a player who plays poorly in a long match will get more kill participation and damage than someone who plays well in a short match. This was achieved by scaling up their stats based on the cumulative experience remaining to get to level 18.[2]

B. Data Integration

Merging region matches into a single database - we merged the three files into one single file and added a region column. The tabular data became more consistent and easier to analyse through this addition. It also simplified the code in later project stages as we were interacting with a single data source.

C. Data Reduction

Removing irrelevant columns - The damage_objectives column was merged into the damage_turrets column, which contained duplicate data. This avoids unnecessary operations later in the code, such as in the imputation, making execution faster.

D. Data Transformation

Renaming spells - Spells are an essential part of our analysis. The spells are a consequence of pre-game choices that profoundly affect in-game performance. However, in the data set the spells appeared as integers. We converted the integers to their associated, meaningful names using the below dictionary.

```
INT_TO_NAME = {
    21: "Barrier",
     1: "Cleanse",
    14: "Ignite",
     3: "Exhaust",
     4: "Flash",
     6: "Ghost",
     7: "Heal",
    13: "Clarity",
    30: "To the King!",
    31: "Poro Toss",
    11: "Smite",
    39: "Mark",
    32: "Mark",
    12: "Teleport",
    54: "Placeholder",
    55: "Placeholder and Attack-Smite"
}
```

Fig. 2: Spells dictionary , League of Legends Guide 2021

Renaming sides - we renamed side.red and side.blue to red and blue. This step improved the readability of the data.

E. Preprocessing History of the Dataset

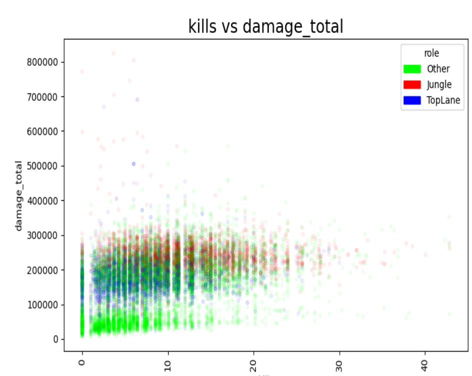
Prior to our wrangling procedure, the dataset had gone through two similar processes. Firstly, Andrew Suter[3] conducted minor quality checks on the raw data before uploading it to Kaggle. Later on, the data was modified to suit the assessment criteria by the University of Melbourne COMP20008 staff. It is important to note these preliminary processing steps as some of the original metadata could not be accessed by us. This should be taken into account when relying on the research conclusions.

V Data Analysis

In our data analysis, we used a decision tree as the supervised learning method to find the ingame performance metrics which best predicted pregame configurations. We chose this method because almost all of our data is nonlinear, with most forming clusters, but also because decision trees would be most helpful in interpreting the data. To perform the experiment we first split the data into test and train and then used the train data to create a decision tree, which was then evaluated by comparing its predicted test data with the actual test data. Evaluation was done using a confusion matrix and a simple accuracy calculation. The trees and matrices can be found in the Appendix section.

We created a collection of scatter plots to display the statistics visually with reference to which role each point came from. We split the data into three groups that represent how a player can contribute to their team's success, being how much damage they did, how much of a tank they are or how much they supported their team. Statistics such as kills, damage_total and damage_buildings show how well the player did in the damage category, statistics such as deaths, damage taken and assists show how well a player did in the tank category and statistics such as vision_score, time_cc and assists shows how much a player supported their team. In addition to this the gold_earned, minions_killed and kda metrics are all good measurements of a players overall performance in a match.

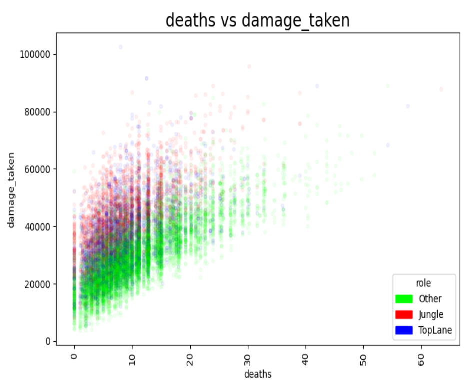
A. Damage

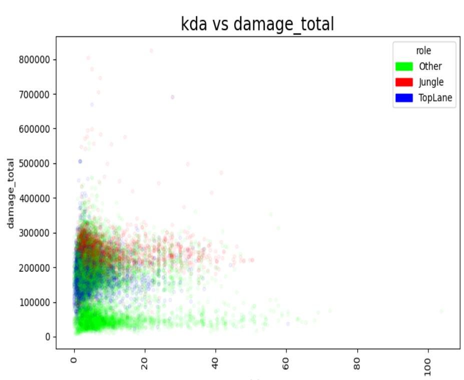


The kills vs damage_total graph show the role that does the most damage and gets the most kills is jungle, then other, then toplane, however there is then also another cluster of other below that, this is likely because of the support role included in other. The damage_building vs gold_earned shows that all the roles are performing similarly in terms of gold, but the jungle role exceeds the others in terms of building damage. As for spells, flash, smite and ignite get players the most kills, while flash and smite earn the most damage.

B. Tank

A good tank will have low deaths and high damage taken. The deaths vs damage_taken graph shows that the best tanks play jungle or toplane, since they are the points closest to the top left corner of the graph.





C. Support

A support's impact comes mostly from their assists, time_cc and vision_score, and the first three graphs in the respective appendix section which are the 3 combinations of these statistics shows that the other role clearly dominates in these fields. The spells with the highest assists are exhaust, flash and ignite, while the spells with the highest vision_score are flash, ignite, heal and exhaust.

VI Results

Our results show that the decisions made by the player before the game starts make a significant impact on their performance during the match. We were not able to force our decision trees to separate classifications with each split and because of the large number of "flash" spells in the data, this resulted in almost all predictions being this spell, which does indicate flash is a popular and powerful spell. The role decision tree shows that the other role is proficient in vision_score, which eludes to it being a support based role, and Top Lane taking the most damage of the roles which means top lane players are frequently contributing to their team by being tanks. It also shows that the players who pick jungle are the ones who are doing the most damage for their team and getting a lot of turret kills. Our champions decision tree recommended some champions that excelled in certain areas, such as Jhin which showed to have good damage output and time_cc while not taking much damage, as well as Nautilus who doesn't output as much damage but has a very large time_cc showing that they are a good support champion. However due to the extremely large number of champions the amount of information that can be extrapolated from this decision tree is limited as its prediction accuracy is around 9 The scatter plots demonstrate how the decision of role and spell affect how the player will contribute to their team, with jungle players excelling at doing damage or tanking for the team while the "other" role is more suited for playing a supporting position. In addition to this the spells chosen by the player can also have a significant impact on performance, as spells such as flash and smite are often in the top performers in most statistics, while some spells like cleanse and barrier are very frequently not used or do not perform well. The bar charts are a clear indicator that pre-game choices are important, as their average statistics vary drastically. For damage you will want to play champions such as Master Yi, Pyke, Ivern and Lillia, for tank you'll want Braum or Yuumi and for support Bard, Rakan, Nocturne and Maokai perform well. Based on this data players may chose to play as the best performing champions and roles with the best spells, however this also gives an opportunity to the developers to tune their game as many outliers such as Ivern's total_damage, the prevalence of flash used by almost everyone and the time_cc of nocturne may need to be altered in order to create a more balanced gameplay experience.

VII Limitations

Some useful statistics such as damage mitigated, damage healed, match duration and outcome were omitted from the data, which could have led to a more accurate conclusion of performance for each match. In addition to this, some data was obscured or not complete, such as the roles being split into two groups rather than all five individual roles, the minions killed statistic was reduced to "many" and "few" rather than the number itself and some of the data was simply missing. If the completed data was used it would have allowed for a more insightful analysis as it would allow the individual roles to be separated and analyzed individually. This is mostly a problem for the "other" role as it combines the middle, bottom and support roles, this is a problem since the support role is very different from the other two which is evident in the two clusters in the other role that often appears in the data. We attempted to improve the performance of the decision tree by using oversampling to make the frequency of each category in each column similar, but this was not possible because it resulted in a pandas dataframe with over 2 million rows.

VIII Future research

Future research can examine the effect of pre-game choice on different players. For instance, are amateur players affected as much as professionals by their pre-game choices. Additionally, we could test the impact that pre-game choices have on the different stages of the game. For example, are longer games affected by pre-game choice the same as shorter games. In addition to this, more pregame factors can be included such as runes and which builds/items the player wants to run in the match. This experiment could contribute to similar research done in the field[4]

References

[1] H. et al, "Data mining concepts and techniques," 2012.
[2] "Champion experience," League of Legend Wikipediaq, 2022.
[3] A. Suter, "Lol challenger soloq data," Kaggle, 2022.
[4] R. D. Gaina, "League of legends: A study of early game impact," 2018.

Appendix

Data Transformation

```
INT_TO_NAME = {
    21: "Barrier",
     1: "Cleanse",
    14: "Ignite",
     3: "Exhaust",
     4: "Flash",
     6: "Ghost",
     7: "Heal",
    13: "Clarity",
    30: "To the King!",
    31: "Poro Toss",
    11: "Smite",
    39: "Mark",
    32: "Mark",
    12: "Teleport",
    54: "Placeholder",
    55: "Placeholder and Attack-Smite"
}
```

Data Analysis: Decision Tree

d_spell decision tree

```
|--- damage_total <= 39937.50
|   |--- vision_score <= 13.50
|   |   |--- gold_earned <= 6893.50
|   |   |   |--- class: Flash
|   |   |--- gold_earned >  6893.50
|   |   |   |--- class: Teleport
|   |--- vision_score >  13.50
|   |   |--- damage_taken <= 7319.00
|   |   |   |--- class: Flash
|   |   |--- damage_taken >  7319.00
|   |   |   |--- class: Flash
|--- damage_total >  39937.50
|   |--- damage_building <= 14182.00
|   |   |--- damage_building <= 8073.50
|   |   |   |--- class: Flash
|   |   |--- damage_building >  8073.50
|   |   |   |--- class: Flash
|   |--- damage_building >  14182.00
|   |   |--- turret_kills <= 2.50
|   |   |   |--- class: Flash
|   |   |--- turret_kills >  2.50
|   |   |   |--- class: Flash
```

Percent of correct predictions
0.4870920603994155

Confusion Matrix:
Predicted

Actual	Barrier	Cleanse	Exhaust	Flash	Ghost	Heal	Ignite	Smite	Teleport
Barrier	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Cleanse	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Exhaust	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Flash	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Ghost	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Heal	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Ignite	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Smite	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Teleport	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0

f_spell decision tree

```
|--- damage_total <= 37686.00
|   |--- vision_score <= 11.50
|   |   |--- assists <= 1.50
|   |   |   |--- class: Teleport
|   |   |--- assists >  1.50
|   |   |   |--- class: Flash
|   |--- vision_score >  11.50
|   |   |--- kills <= 7.50
|   |   |   |--- class: Ignite
|   |   |--- kills >  7.50
|   |   |   |--- class: Flash
|--- damage_total >  37686.00
|   |--- damage_building <= 14374.50
|   |   |--- damage_taken <= 22543.50
|   |   |   |--- class: Flash
|   |   |--- damage_taken >  22543.50
|   |   |   |--- class: Flash
|   |--- damage_building >  14374.50
|   |   |--- turret_kills <= 2.50
|   |   |   |--- class: Flash
|   |   |--- turret_kills >  2.50
|   |   |   |--- class: Flash
```

Percent of correct predictions
0.42815392109108624

Confusion Matrix:
Predicted

Actual	Barrier	Cleanse	Exhaust	Flash	Ghost	Heal	Ignite	Smite	Teleport
Barrier	0.0	0.0	0.0	0.800000	0.0	0.0	0.200000	0.0	0.000000
Cleanse	0.0	0.0	0.0	1.000000	0.0	0.0	0.000000	0.0	0.000000
Exhaust	0.0	0.0	0.0	0.681818	0.0	0.0	0.318182	0.0	0.000000
Flash	0.0	0.0	0.0	0.800875	0.0	0.0	0.196937	0.0	0.002188
Ghost	0.0	0.0	0.0	1.000000	0.0	0.0	0.000000	0.0	0.000000
Heal	0.0	0.0	0.0	0.861111	0.0	0.0	0.138889	0.0	0.000000
Ignite	0.0	0.0	0.0	0.472924	0.0	0.0	0.527076	0.0	0.000000
Smite	0.0	0.0	0.0	1.000000	0.0	0.0	0.000000	0.0	0.000000
Teleport	0.0	0.0	0.0	0.994695	0.0	0.0	0.002653	0.0	0.002653

```
role decision tree

|--- damage_building <= 13438.50
|   |--- damage_total <= 44343.50
|   |   |--- vision_score <= 13.50
|   |   |   |--- class: Other
|   |   |--- vision_score >  13.50
|   |   |   |--- class: Other
|   |--- damage_total >  44343.50
|   |   |--- damage_taken <= 24775.50
|   |   |   |--- class: Other
|   |   |--- damage_taken >  24775.50
|   |   |   |--- class: TopLane
|--- damage_building >  13438.50
|   |--- turret_kills <= 2.50
|   |   |--- damage_building <= 21429.00
|   |   |   |--- class: Jungle
|   |   |--- damage_building >  21429.00
|   |   |   |--- class: Jungle
|   |--- turret_kills >  2.50
|   |   |--- damage_building <= 26499.50
|   |   |   |--- class: Other
|   |   |--- damage_building >  26499.50
|   |   |   |--- class: Jungle

Percent of correct predictions
0.7082318558207501
```

Confusion Matrix:

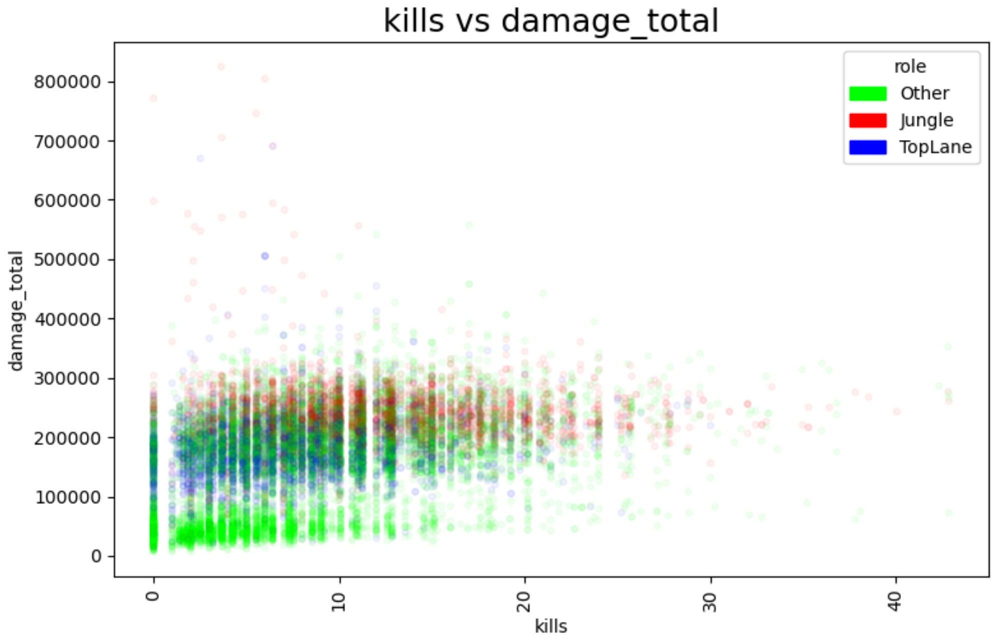
Predicted

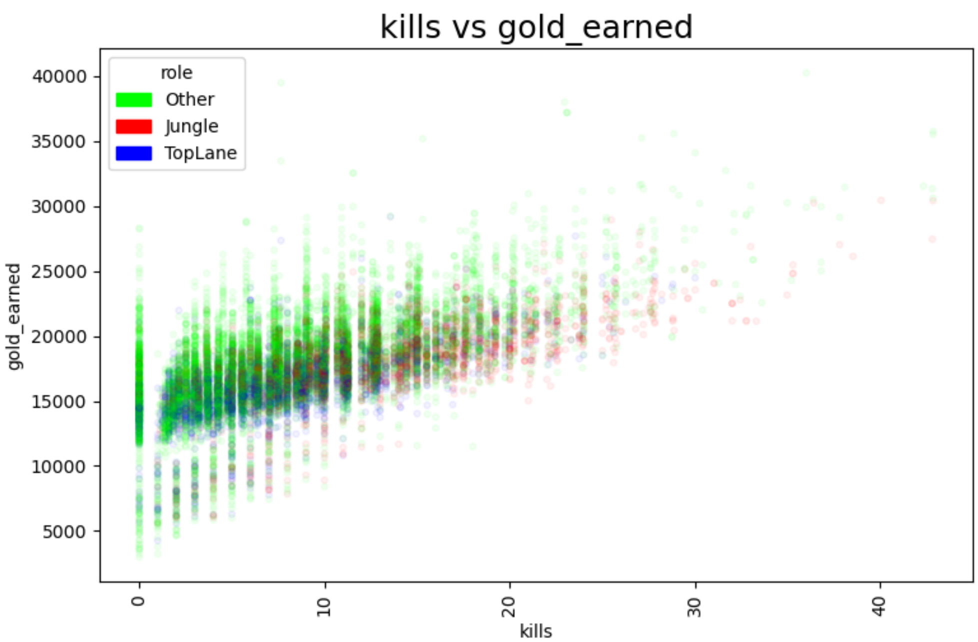
Actual	Jungle	Other	TopLane
Jungle	0.679104	0.194030	0.126866
Other	0.065182	0.863861	0.070957
TopLane	0.102506	0.592255	0.305239

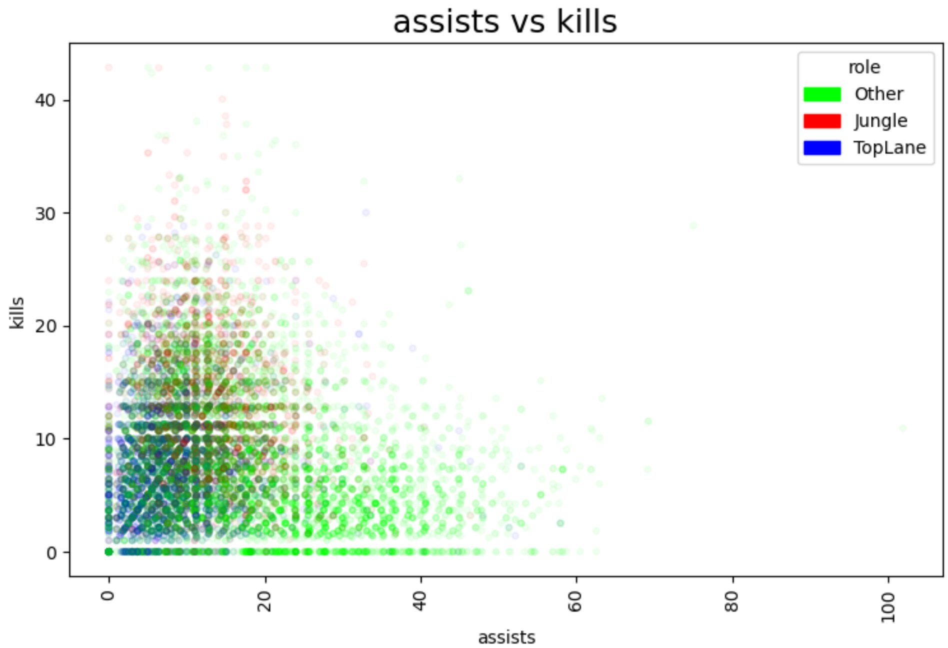
```
champion decision tree

|--- damage_total <= 33611.00
|   |--- time_cc <= 31.50
|   |   |--- kills <= 5.50
|   |   |   |--- class: Rakan
|   |   |--- kills >  5.50
|   |   |   |--- class: Pyke
|   |--- time_cc >  31.50
|   |   |--- time_cc <= 50.50
|   |   |   |--- class: Leona
|   |   |--- time_cc >  50.50
|   |   |   |--- class: Nautilus
|--- damage_total >  33611.00
|   |--- time_cc <= 4.50
|   |   |--- damage_building <= 21153.00
|   |   |   |--- class: Akali
|   |   |--- damage_building >  21153.00
|   |   |   |--- class: Nidalee
|   |--- time_cc >  4.50
|   |   |--- damage_taken <= 23419.00
|   |   |   |--- class: Jhin
|   |   |--- damage_taken >  23419.00
|   |   |   |--- class: LeeSin

Percent of correct predictions
0.08767657087189479
```

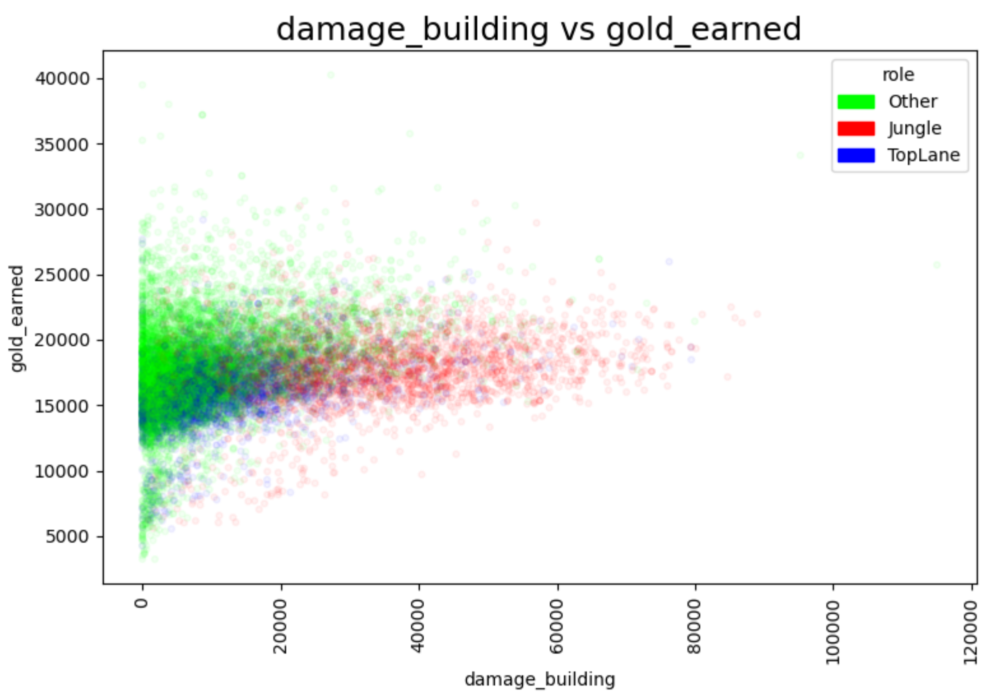
Data Analysis: Scatter Plots

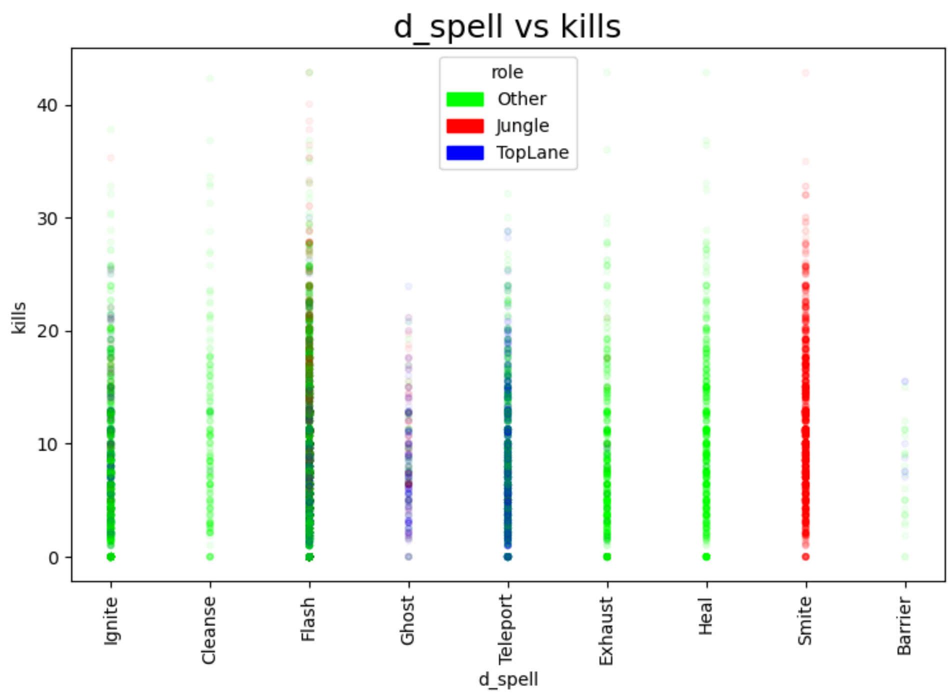
Damage Statistics


kills vs damage_total
role
Other
Jungle
TopLane
damage_total
kills
0
100000
200000
300000
400000
500000
600000
700000
800000
0
10
20
30
40

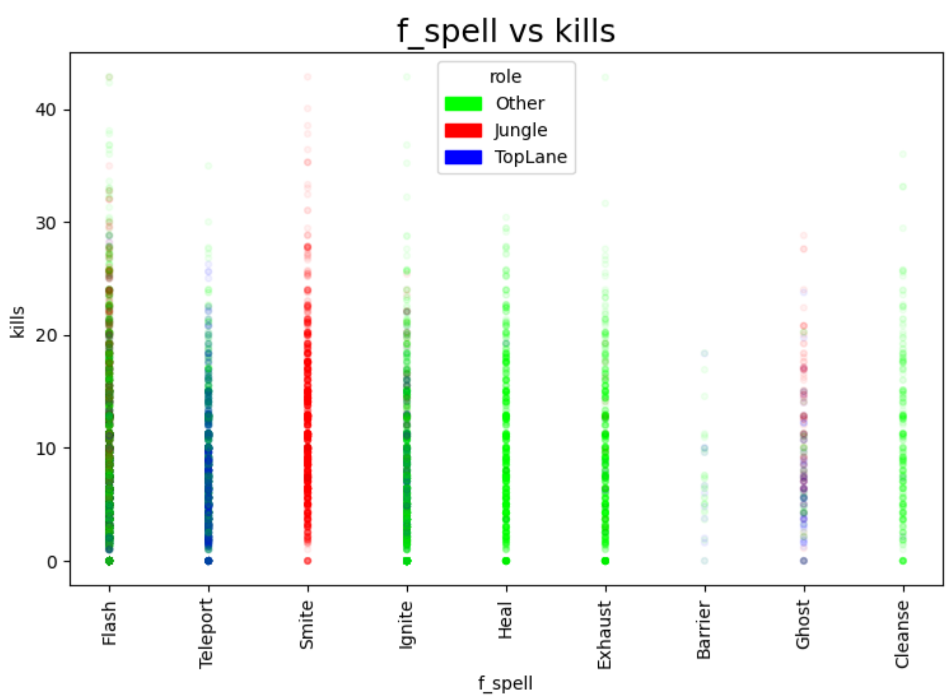

kills vs gold_earned
role
Other
Jungle
TopLane
gold_earned
kills
5000
10000
15000
20000
25000
30000
35000
40000
0
10
20
30
40


assists vs kills
role
Other
Jungle
TopLane
kills
assists
0
10
20
30
40
0
20
40
60
80
100

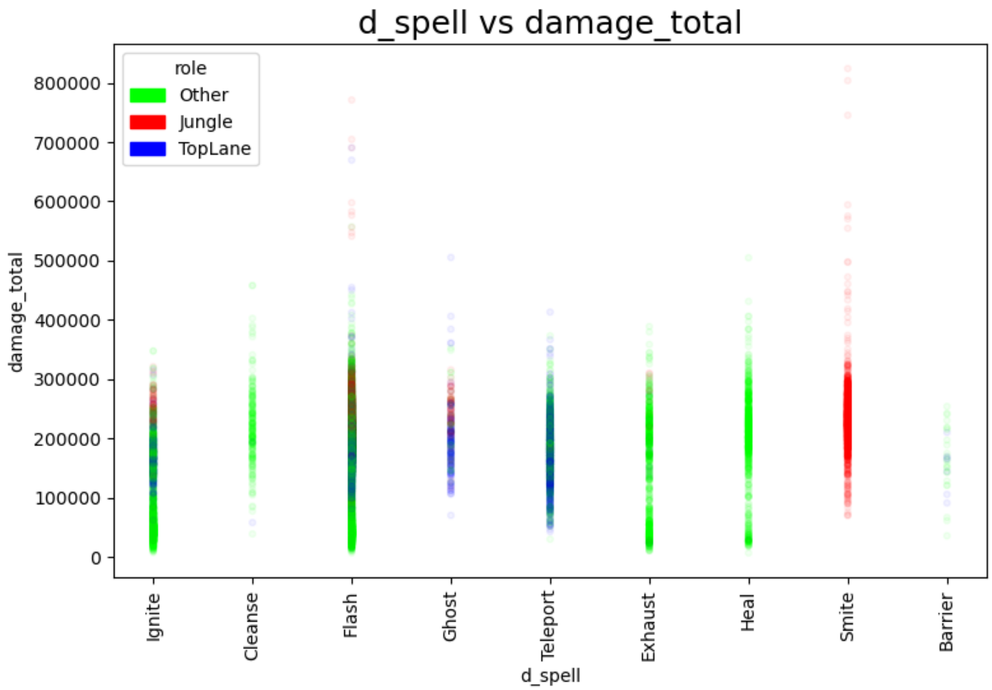

damage_building vs gold_earned
role
Other
Jungle
TopLane
gold_earned
damage_building
5000
10000
15000
20000
25000
30000
35000
40000
0
20000
40000
60000
80000
100000
120000



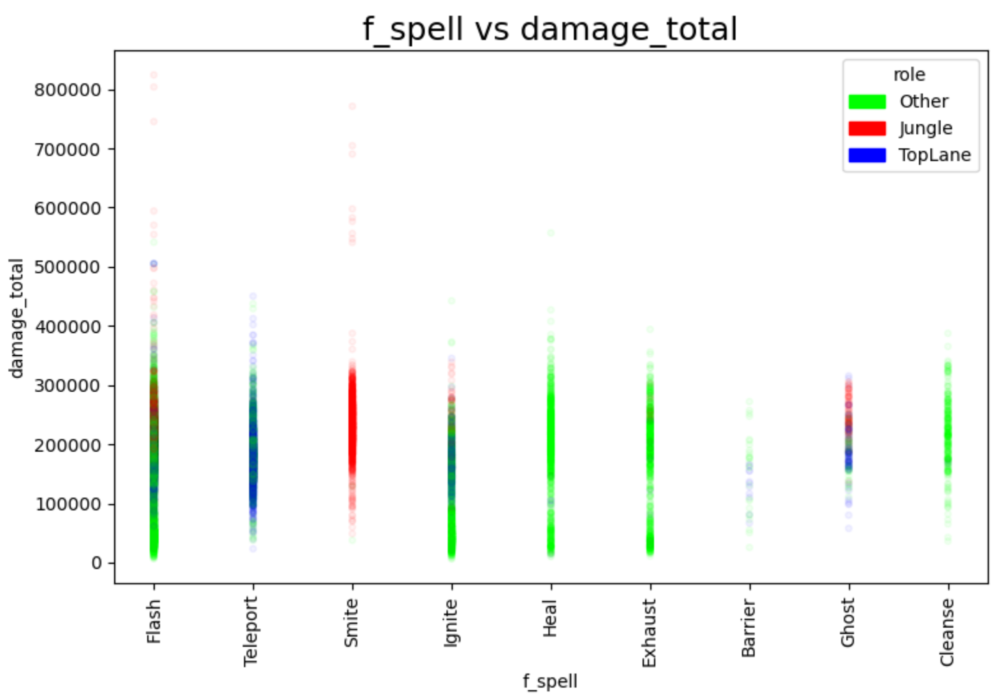
d_spell vs kills
role
Other
Jungle
TopLane
kills
0
10
20
30
40
Ignite
Cleanse
Flash
Ghost
Teleport
Exhaust
Heal
Smite
Barrier
d_spell



f_spell vs kills
role
Other
Jungle
TopLane
kills
0
10
20
30
40
Flash
Teleport
Smite
Ignite
Heal
Exhaust
Barrier
Ghost
Cleanse
f_spell

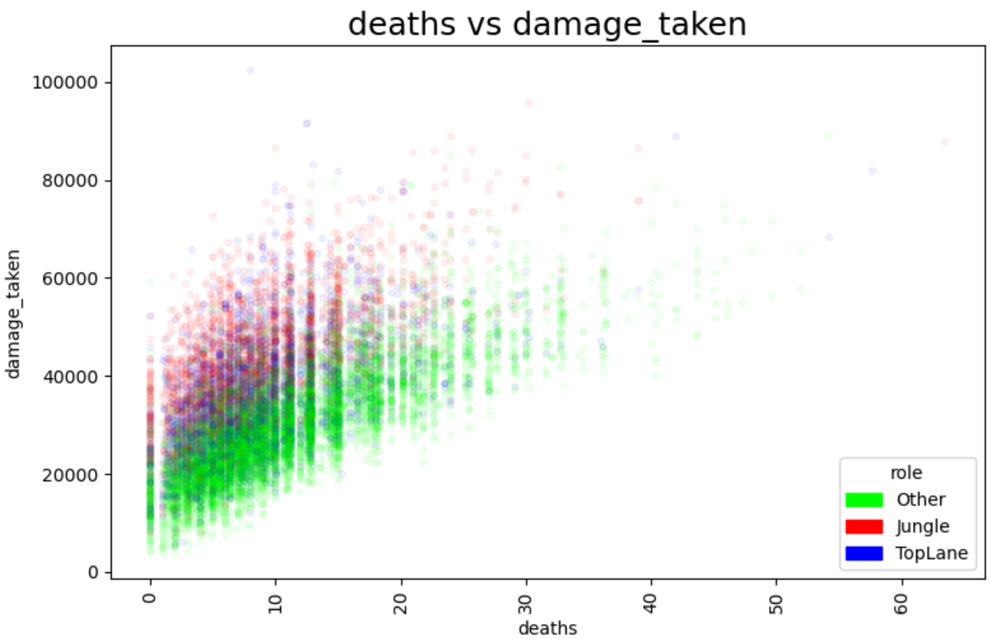


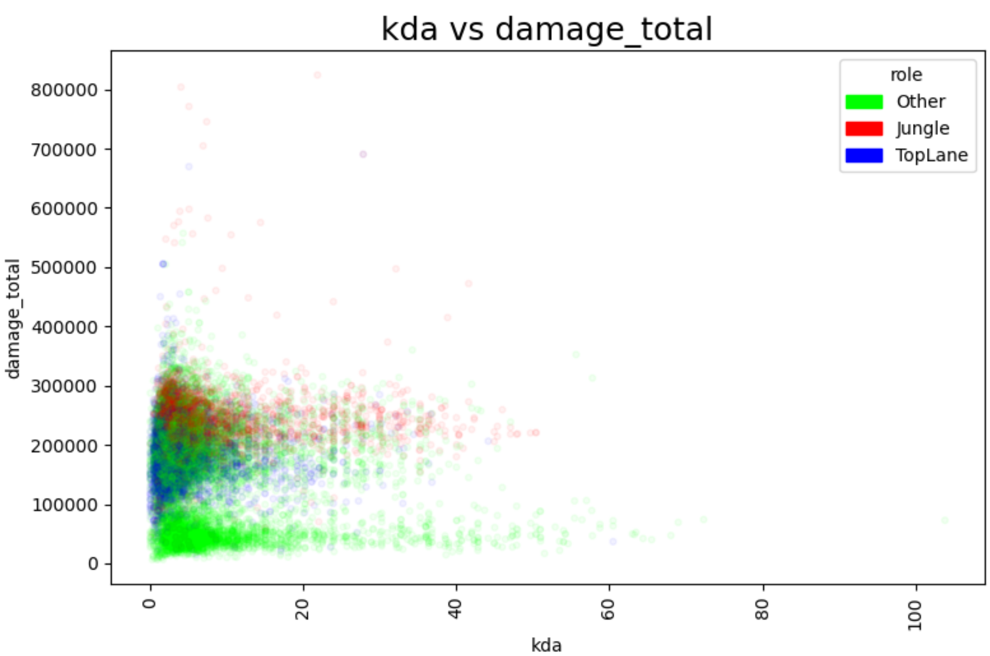
d_spell vs damage_total
role
Other
Jungle
TopLane
damage_total
800000
700000
600000
500000
400000
300000
200000
100000
0
Ignite
Cleanse
Flash
Ghost
Teleport
Exhaust
Heal
Smite
Barrier
d_spell

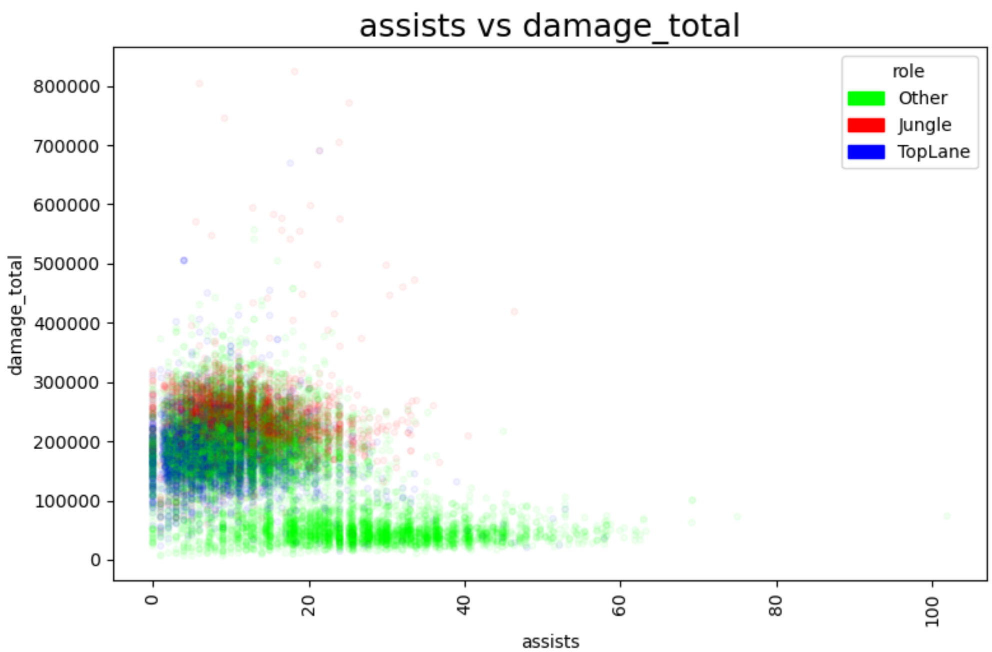


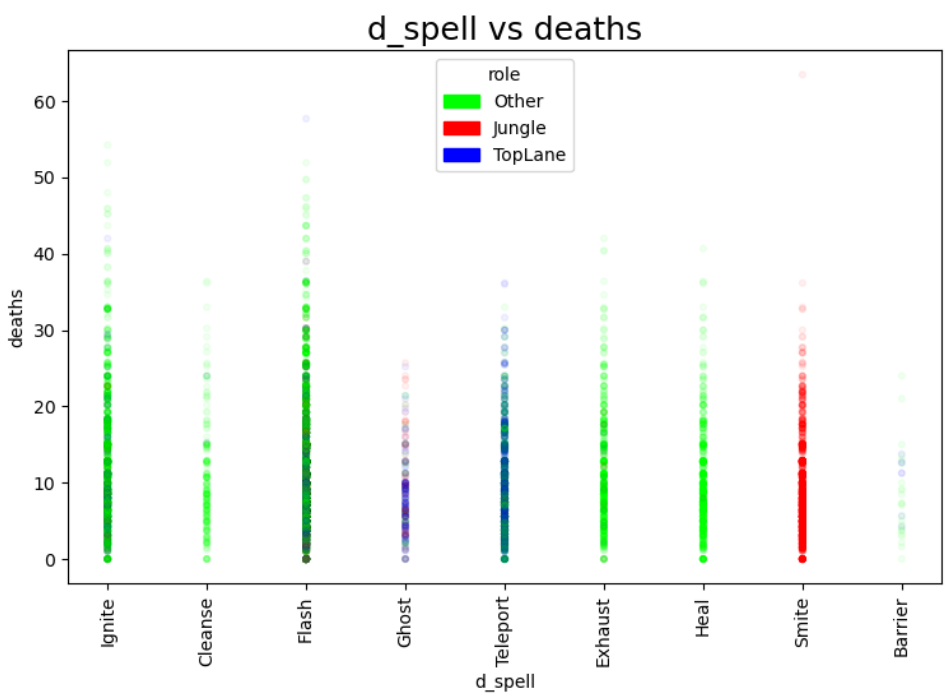
f_spell vs damage_total
role
Other
Jungle
TopLane
damage_total
800000
700000
600000
500000
400000
300000
200000
100000
0
Flash
Teleport
Smite
Ignite
Heal
Exhaust
Barrier
Ghost
Cleanse
f_spell

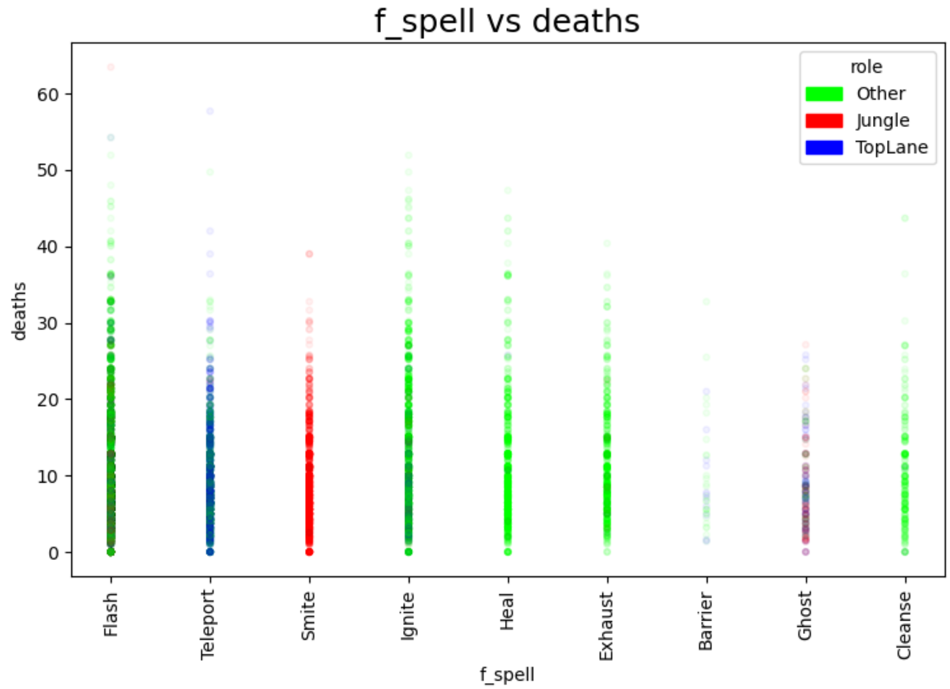
Tank Statistics



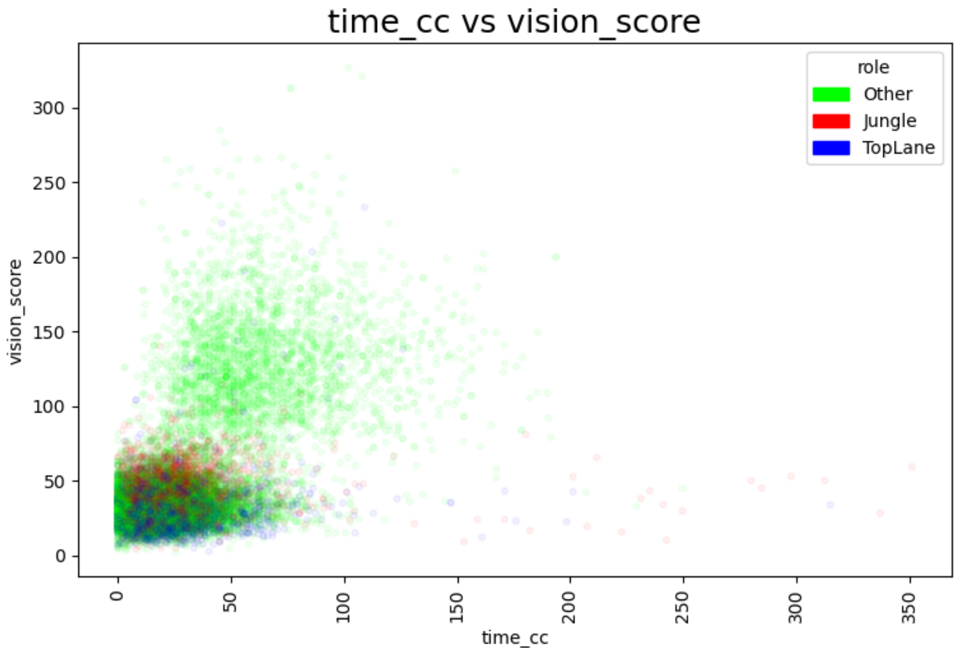


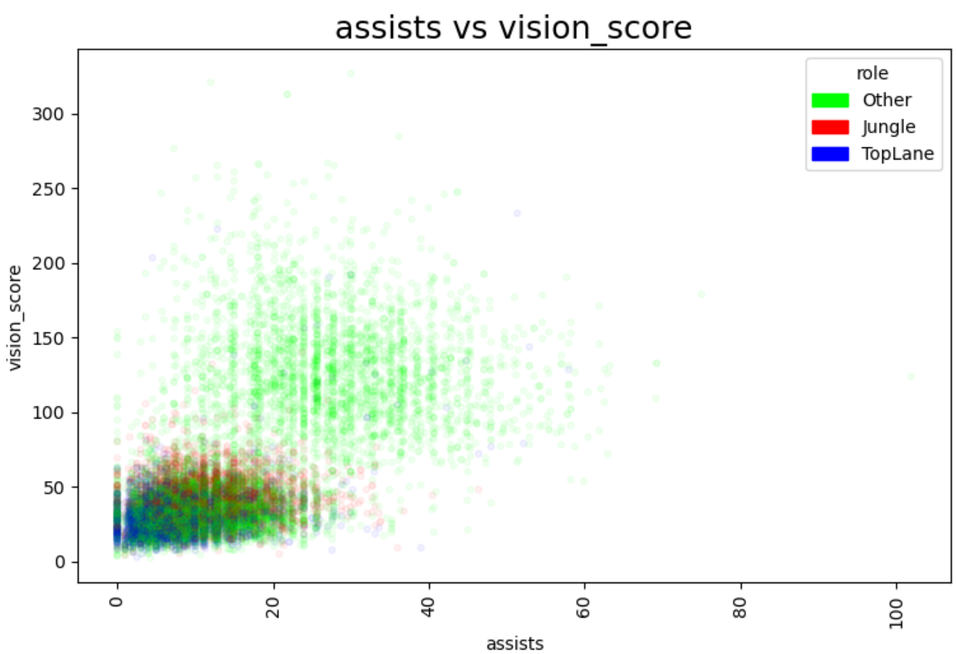

assists vs damage_total
role
Other
Jungle
TopLane
damage_total
assists
0
100000
200000
300000
400000
500000
600000
700000
800000
0
20
40
60
80
100

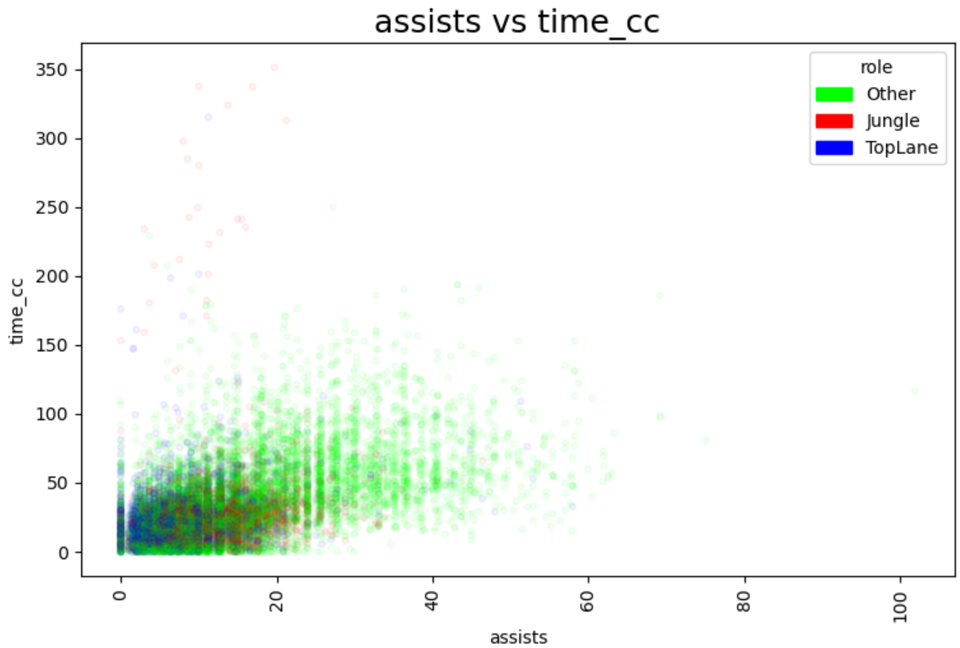

d_spell vs deaths
role
Other
Jungle
TopLane
deaths
d_spell
0
10
20
30
40
50
60
Ignite
Cleanse
Flash
Ghost
Teleport
Exhaust
Heal
Smite
Barrier

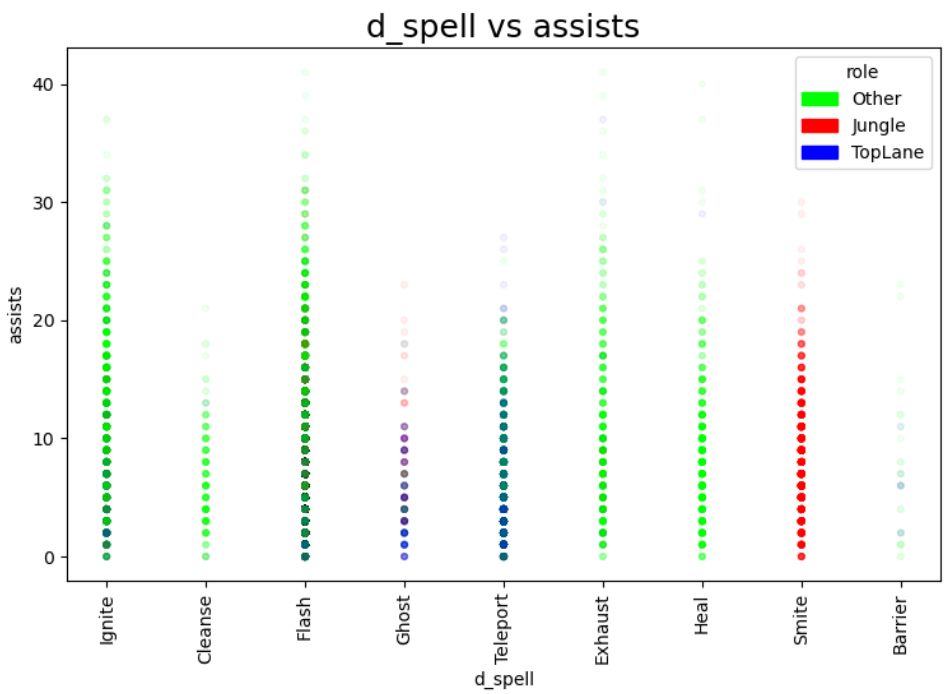

f_spell vs deaths
deaths
0
10
20
30
40
50
60
f_spell
Flash
Teleport
Smite
Ignite
Heal
Exhaust
Barrier
Ghost
Cleanse
role
Other
Jungle
TopLane

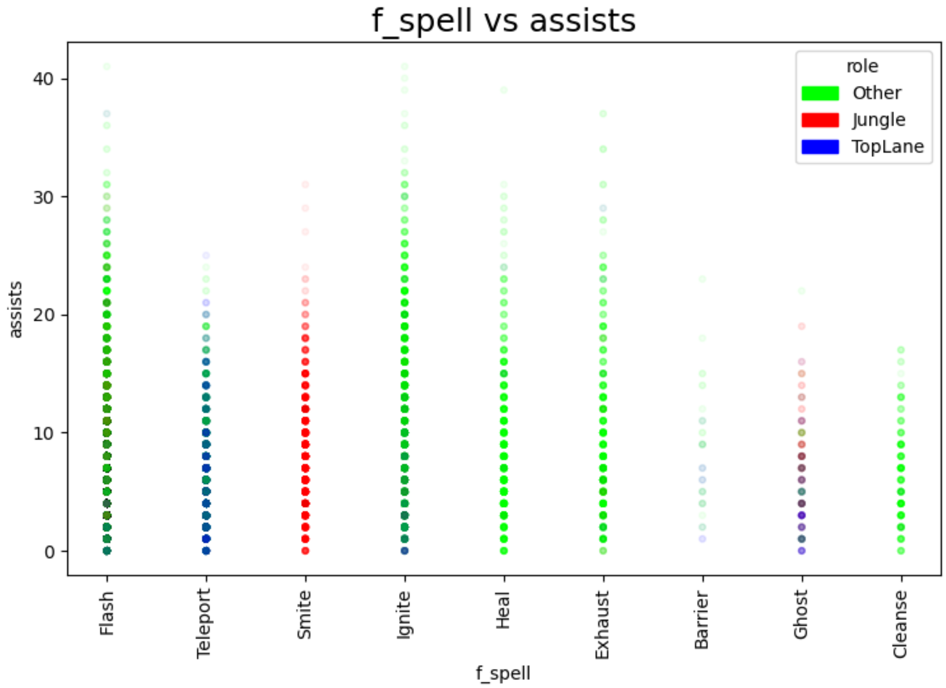
Support Statistics

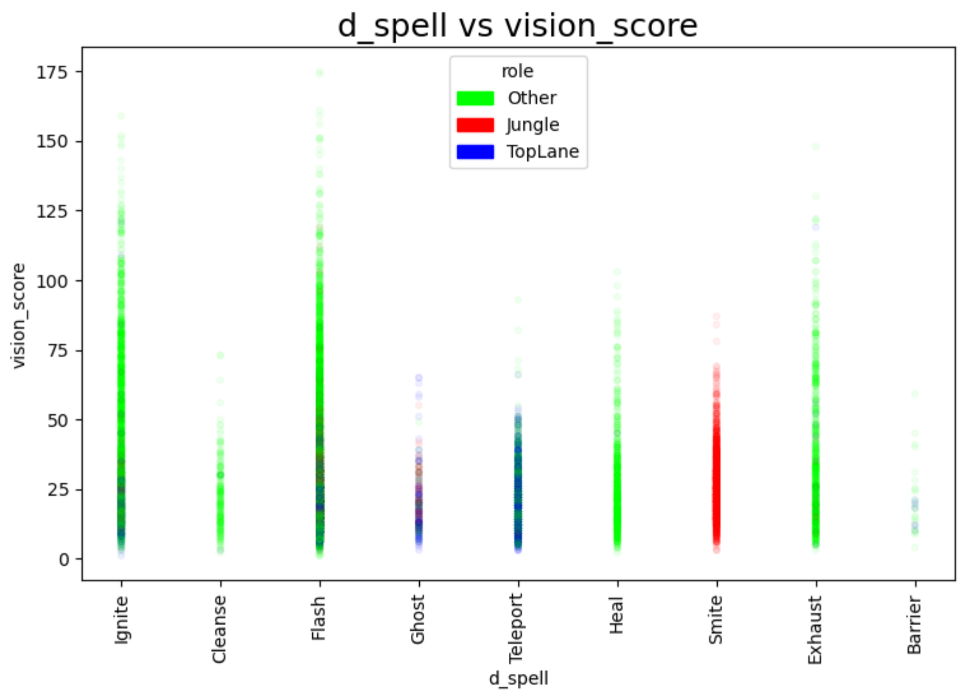


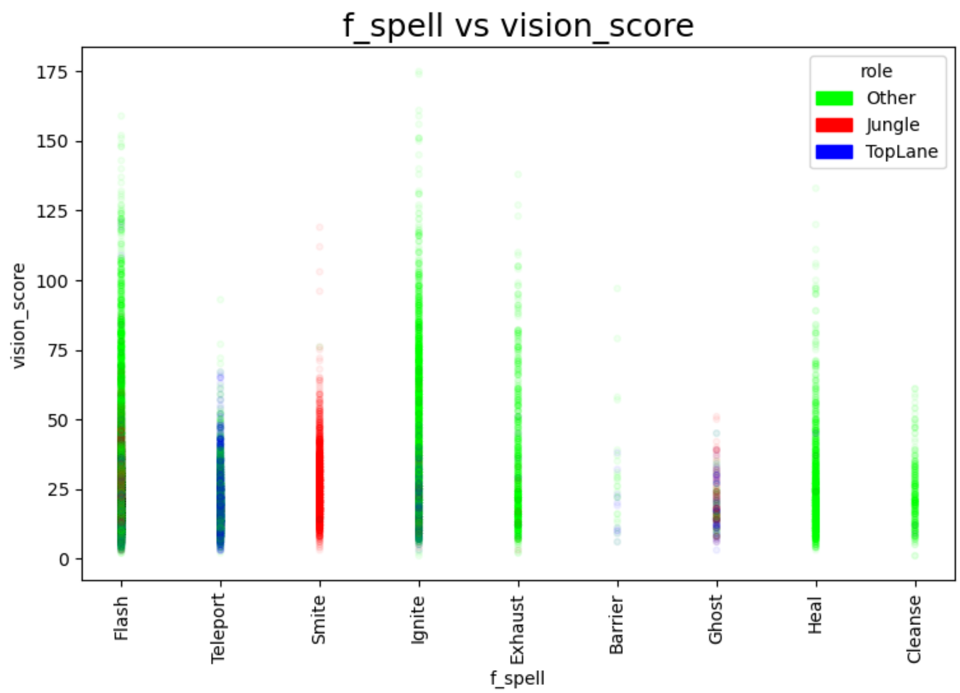



assists vs time_cc
role
Other
Jungle
TopLane
time_cc
0
50
100
150
200
250
300
350
0
20
40
60
80
100
assists

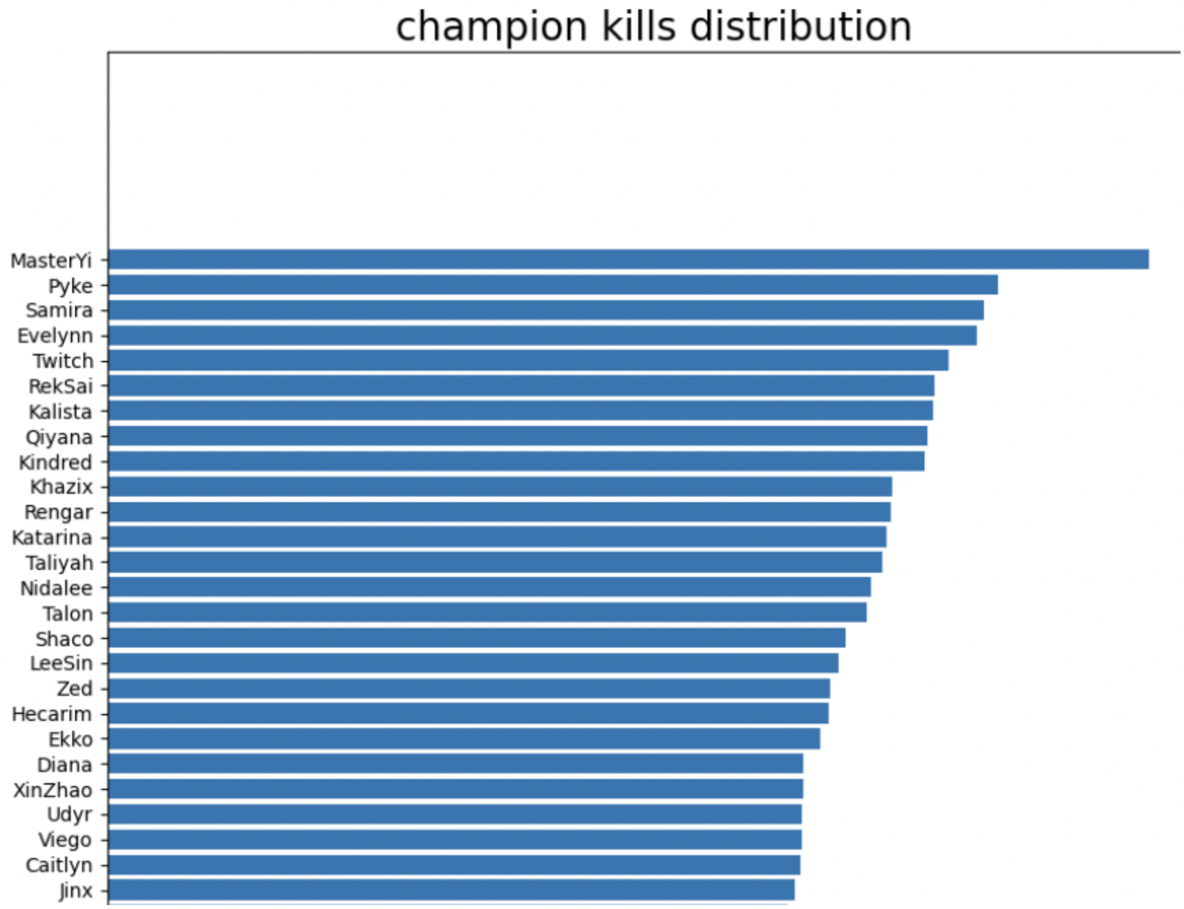

d_spell vs assists
role
Other
Jungle
TopLane
assists
0
10
20
30
40
Ignite
Cleanse
Flash
Ghost
Teleport
Exhaust
Heal
Smite
Barrier
d_spell

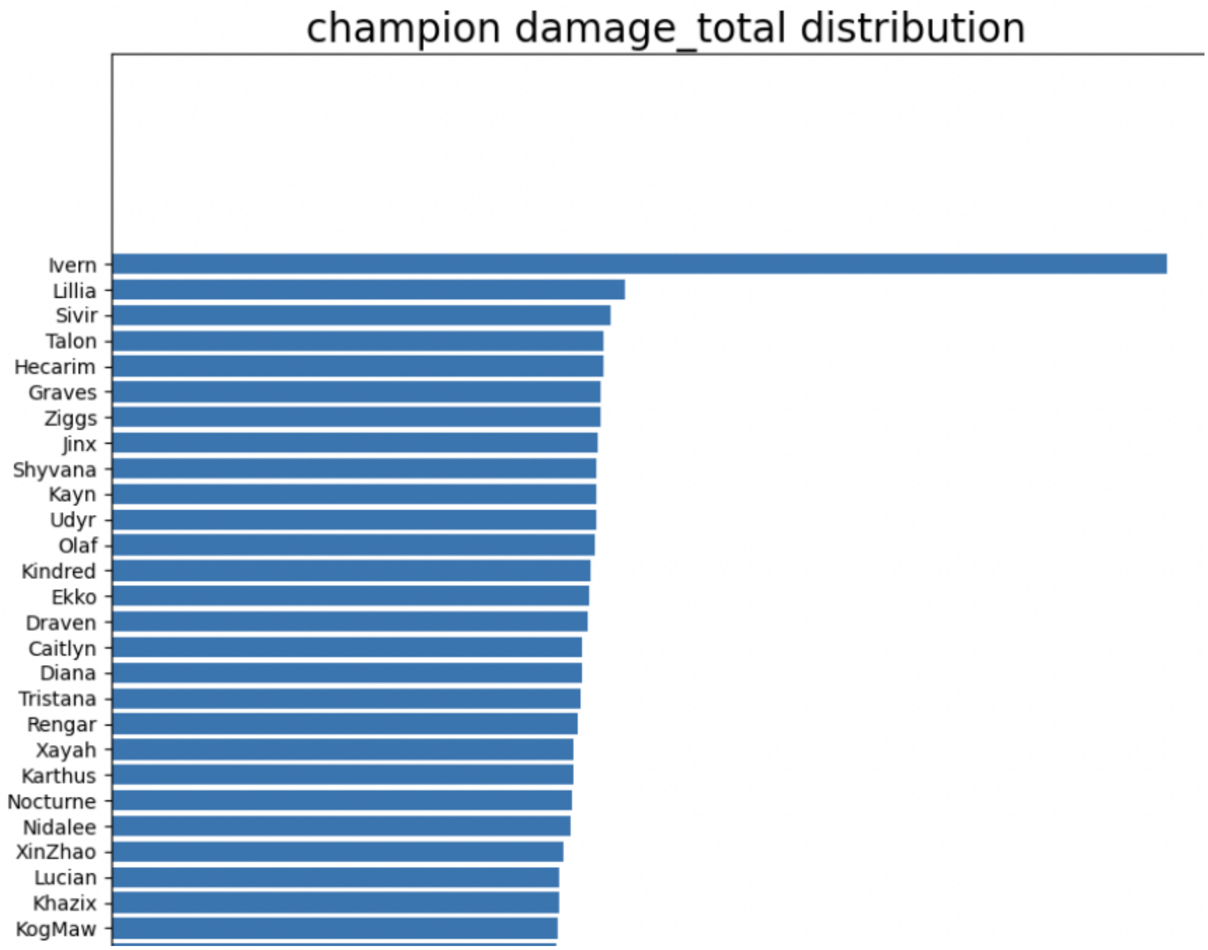

f_spell vs assists
role
Other
Jungle
TopLane
assists
0
10
20
30
40
Flash
Teleport
Smite
Ignite
Heal
Exhaust
Barrier
Ghost
Cleanse
f_spell

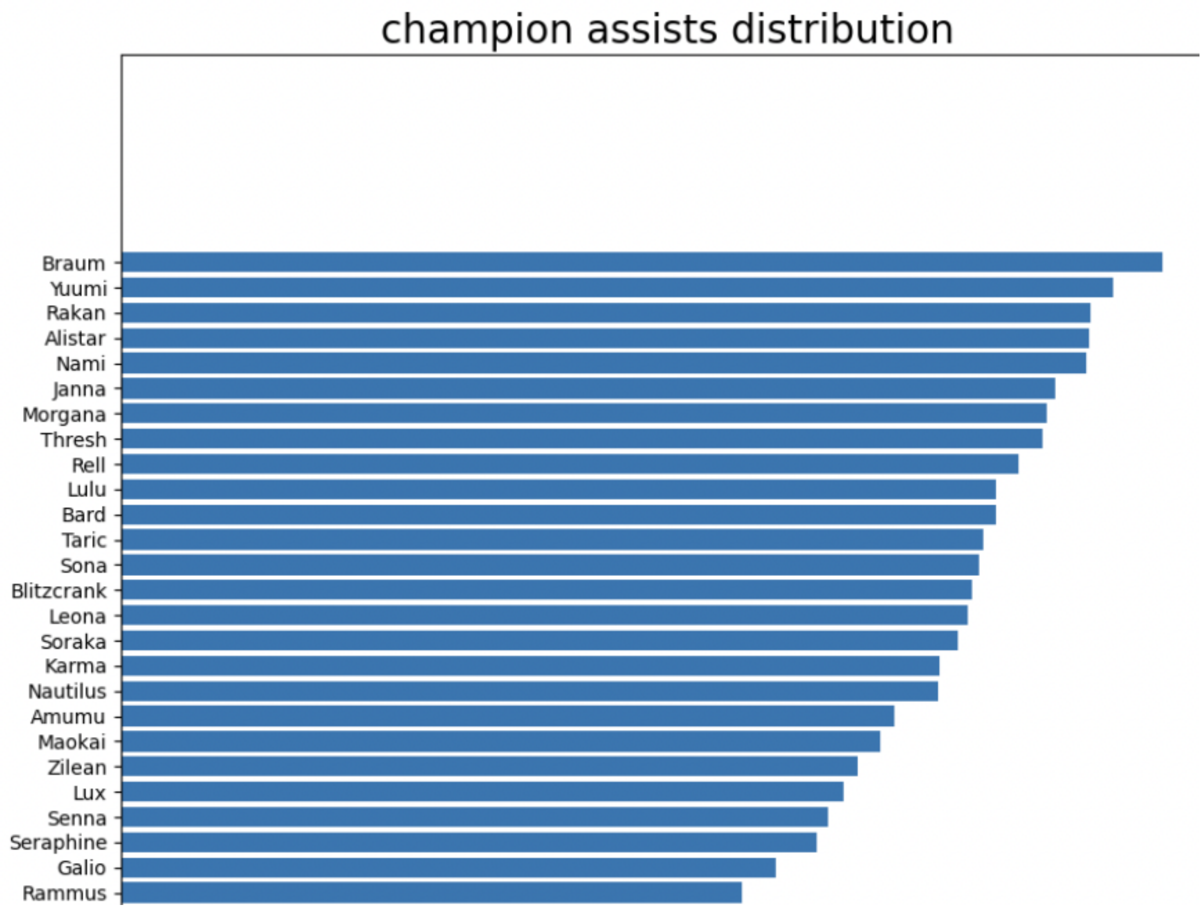

d_spell vs vision_score
role
Other
Jungle
TopLane
vision_score
175
150
125
100
75
50
25
0
Ignite
Cleanse
Flash
Ghost
Teleport
Heal
Smite
Exhaust
Barrier
d_spell

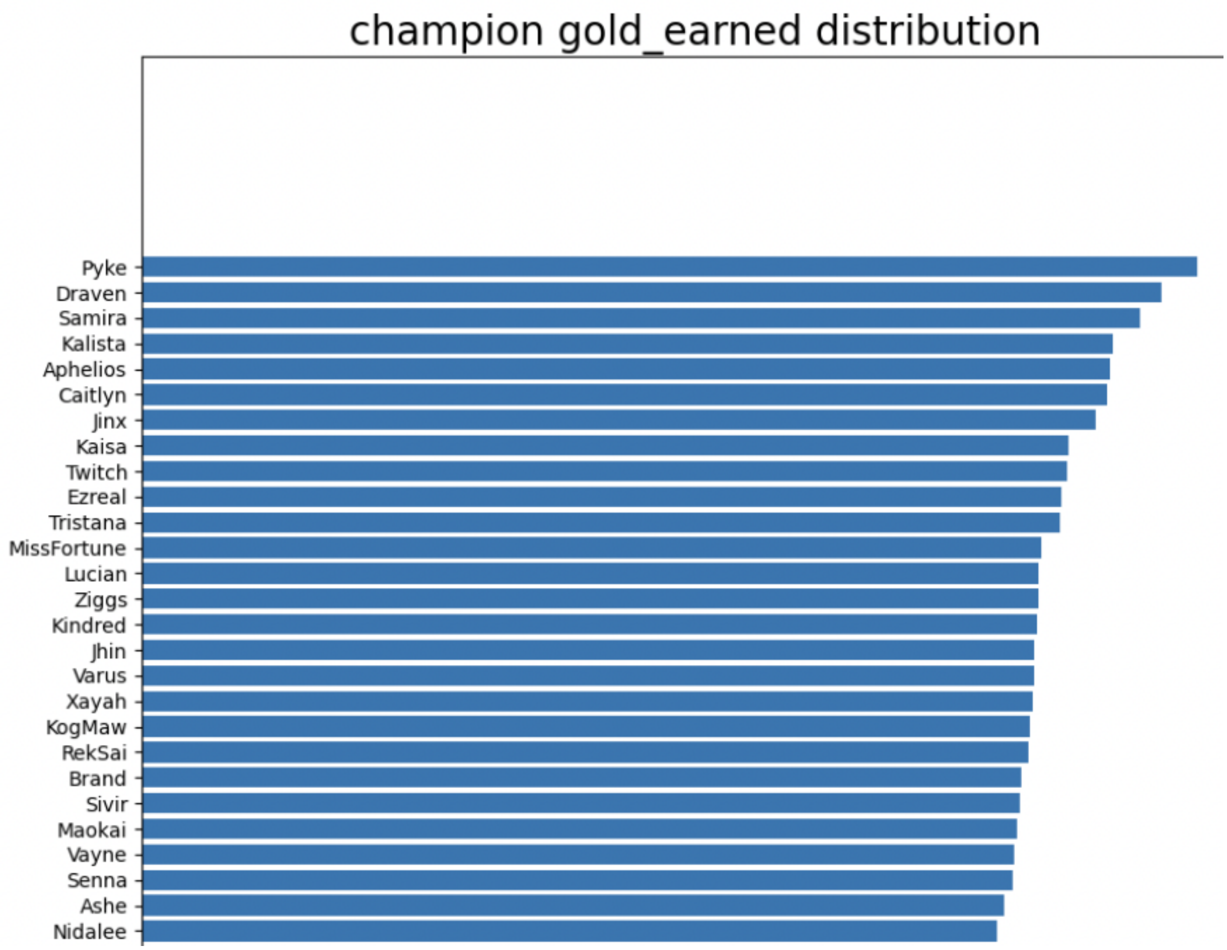


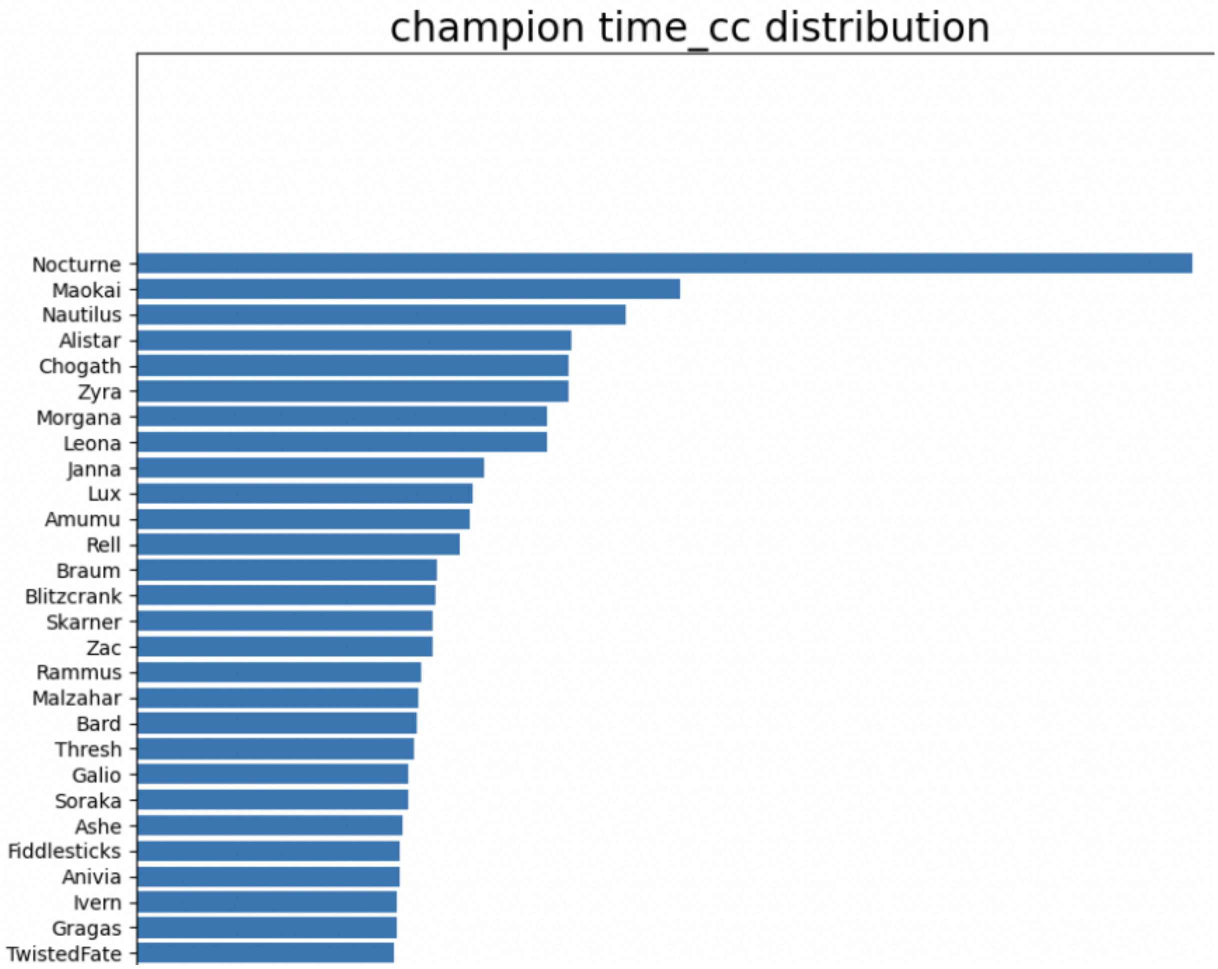
Data Analysis: Partial Bar Charts

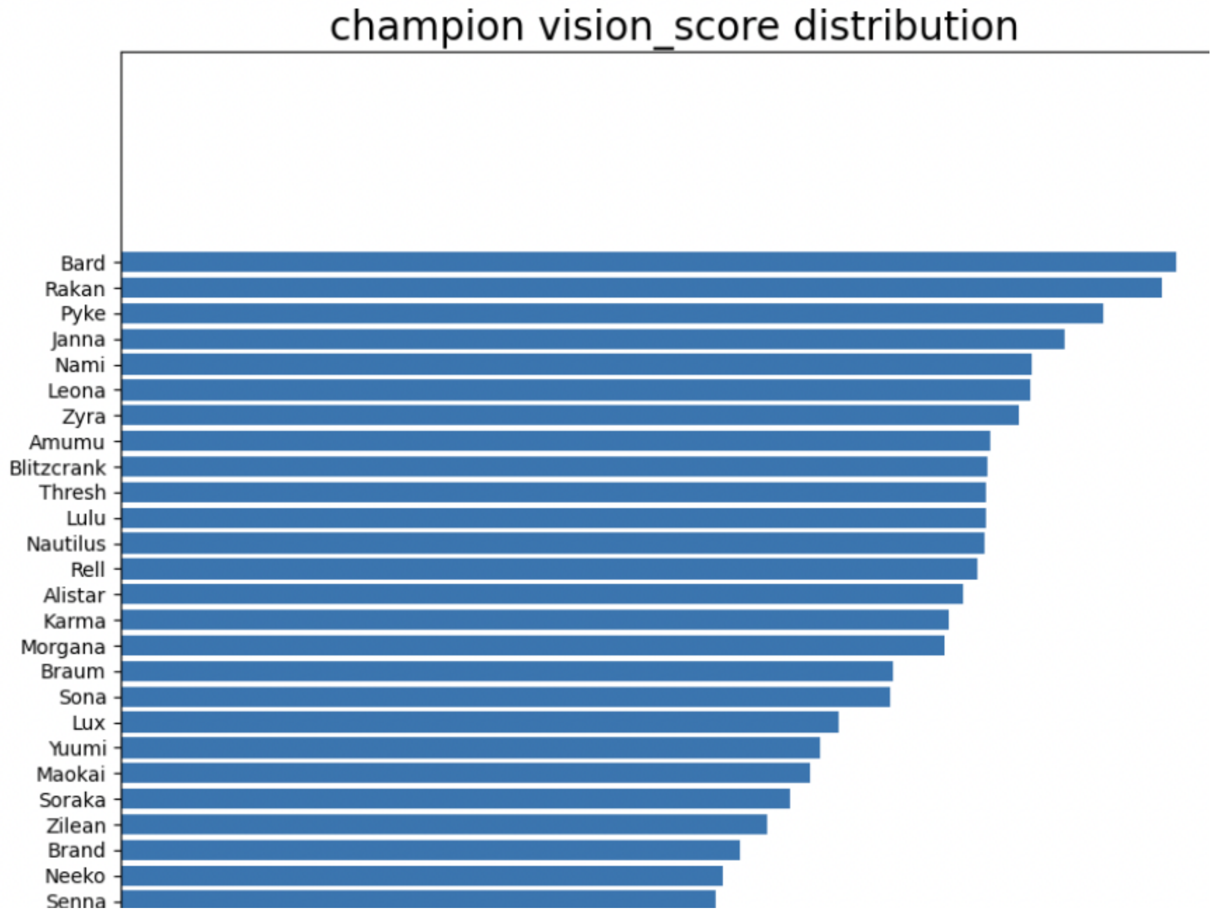

champion kills distribution
MasterYi
Pyke
Samira
Evelynn
Twitch
RekSai
Kalista
Qiyana
Kindred
Khazix
Rengar
Katarina
Taliyah
Nidalee
Talon
Shaco
LeeSin
Zed
Hecarim
Ekko
Diana
XinZhao
Udyr
Viego
Caitlyn
Jinx


champion damage_total distribution
Ivern
Lillia
Sivir
Talon
Hecarim
Graves
Ziggs
Jinx
Shyvana
Kayn
Udyr
Olaf
Kindred
Ekko
Draven
Caitlyn
Diana
Tristana
Rengar
Xayah
Karthus
Nocturne
Nidalee
XinZhao
Lucian
Khazix
KogMaw

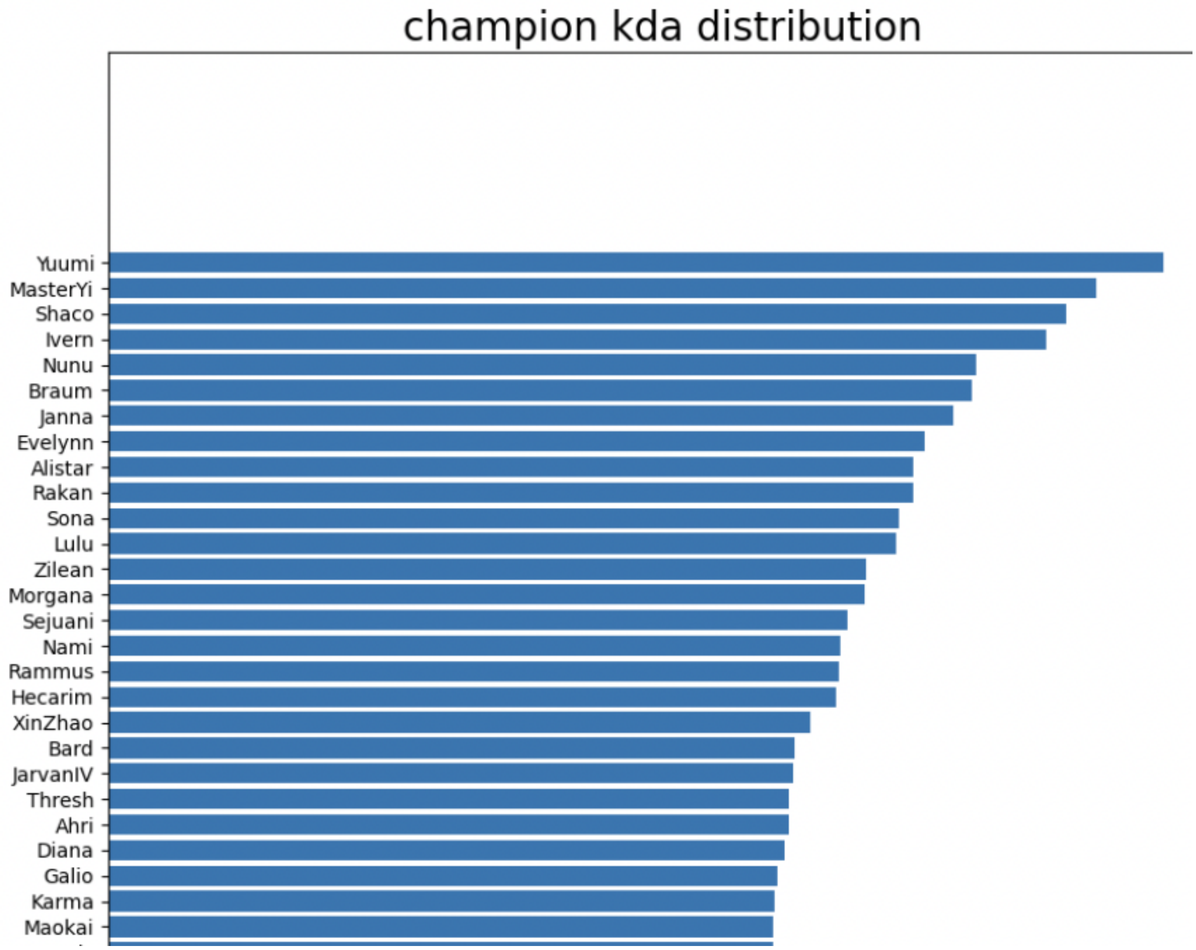

champion assists distribution
Braum
Yuumi
Rakan
Alistar
Nami
Janna
Morgana
Thresh
Rell
Lulu
Bard
Taric
Sona
Blitzcrank
Leona
Soraka
Karma
Nautilus
Amumu
Maokai
Zilean
Lux
Senna
Seraphine
Galio
Rammus


champion gold_earned distribution
Pyke
Draven
Samira
Kalista
Aphelios
Caitlyn
Jinx
Kaisa
Twitch
Ezreal
Tristana
MissFortune
Lucian
Ziggs
Kindred
Jhin
Varus
Xayah
KogMaw
RekSai
Brand
Sivir
Maokai
Vayne
Senna
Ashe
Nidalee


champion time_cc distribution
Nocturne
Maokai
Nautilus
Alistar
Chogath
Zyra
Morgana
Leona
Janna
Lux
Amumu
Rell
Braum
Blitzcrank
Skarner
Zac
Rammus
Malzahar
Bard
Thresh
Galio
Soraka
Ashe
Fiddlesticks
Anivia
Ivern
Gragas
TwistedFate



champion vision_score distribution
Bard
Rakan
Pyke
Janna
Nami
Leona
Zyra
Amumu
Blitzcrank
Thresh
Lulu
Nautilus
Rell
Alistar
Karma
Morgana
Braum
Sona
Lux
Yuumi
Maokai
Soraka
Zilean
Brand
Neeko
Senna


champion kda distribution
Yuumi
MasterYi
Shaco
Ivern
Nunu
Braum
Janna
Evelynn
Alistar
Rakan
Sona
Lulu
Zilean
Morgana
Sejuani
Nami
Rammus
Hecarim
XinZhao
Bard
JarvanIV
Thresh
Ahri
Diana
Galio
Karma
Maokai