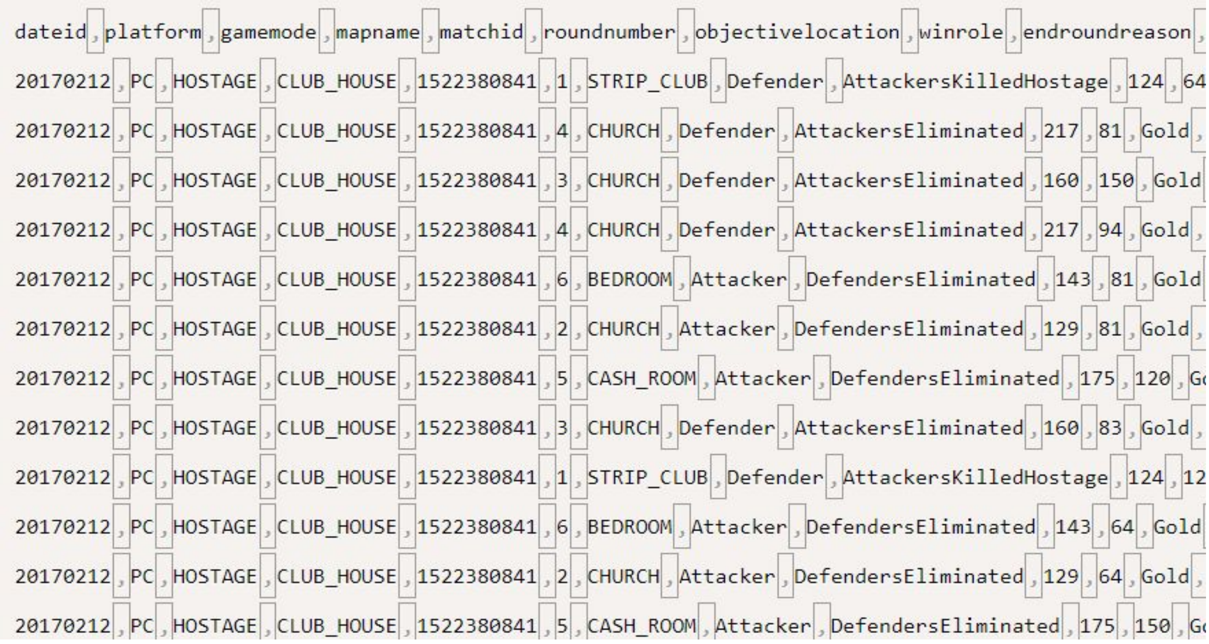
Motivation

In Rainbow Six Siege, Some operators are believed not suitable for a certain game mode, This project uses the year 2 season 5 ranked competition data to analyze whether the famous statement “doesn’t pick FUZE for the hostage” is true.

About the Data

The data come from Kaggle, the original data come from UbiSoft. The data set has 21 CSV files with total size 18.8G, 81939713 rows, and 31 columns. The figure below is a snippet of the data:



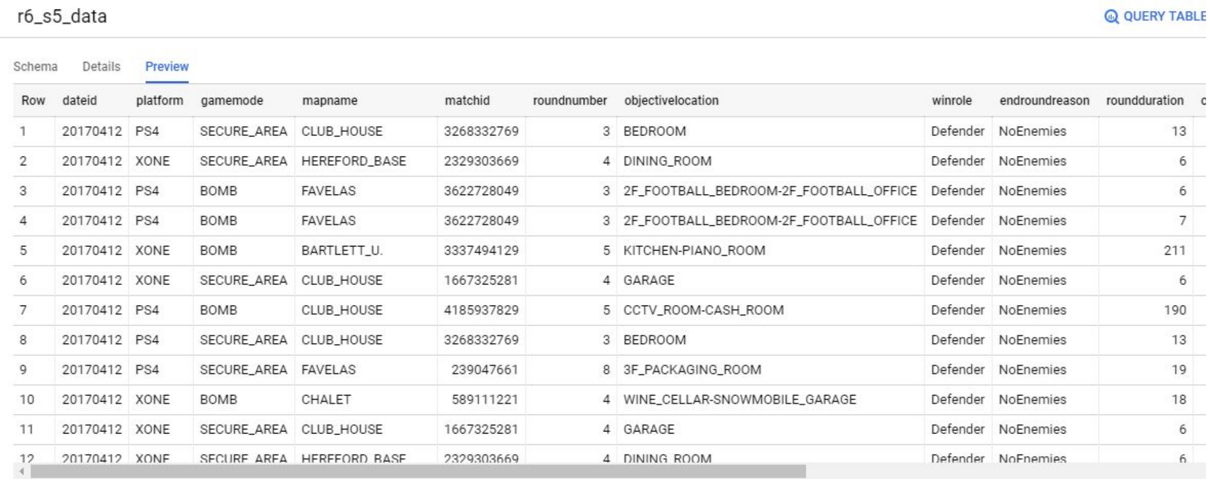
```
dateid,platform,gamemode,mapname,matchid,roundnumber,objectivelocation,winrole,endroundreason,
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,1,STRIP_CLUB,Defender,AttackersKilledHostage,124,64
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,4,CHURCH,Defender,AttackersEliminated,217,81,Gold,
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,3,CHURCH,Defender,AttackersEliminated,160,150,Gold
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,4,CHURCH,Defender,AttackersEliminated,217,94,Gold,
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,6,BEDROOM,Attacker,DefendersEliminated,143,81,Gold
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,2,CHURCH,Attacker,DefendersEliminated,129,81,Gold,
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,5,CASH_ROOM,Attacker,DefendersEliminated,175,120,G
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,3,CHURCH,Defender,AttackersEliminated,160,83,Gold,
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,1,STRIP_CLUB,Defender,AttackersKilledHostage,124,12
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,6,BEDROOM,Attacker,DefendersEliminated,143,64,Gold
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,2,CHURCH,Attacker,DefendersEliminated,129,64,Gold,
20170212,PC,HOSTAGE,CLUB_HOUSE,1522380841,5,CASH_ROOM,Attacker,DefendersEliminated,175,150,G
```

Data Obtain and Process Method

The Data obtained by download from the website. It was loaded into GCS then import into Big Query due to data set size are too big. This project chose the big query as data analytic tool because it provides fast process speed with easy to use standard SQL syntax.

Data Preview

The 21 CSV files are loaded into one single table in Big Query, Below is the data preview on Big Query:

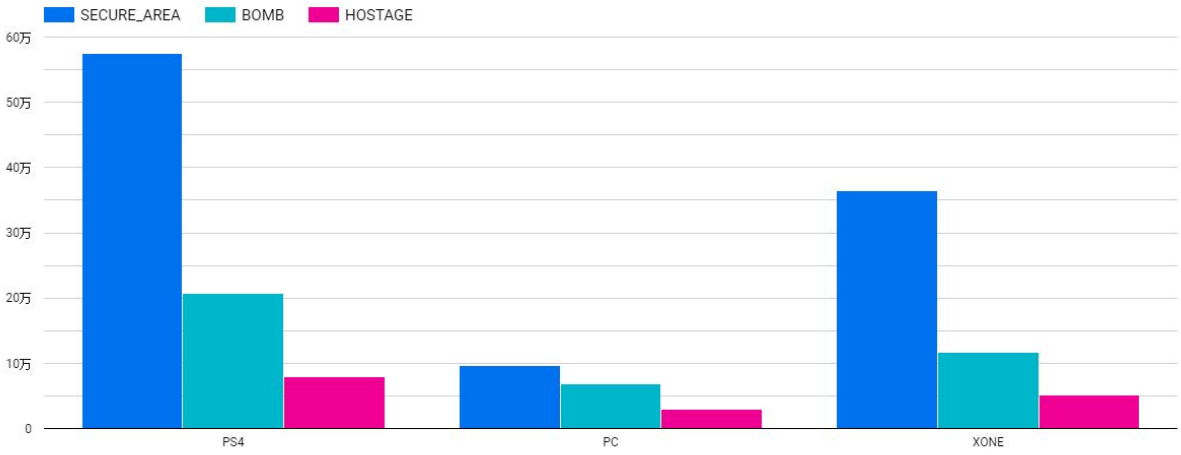


Row	dateid	platform	gamemode	mapname	matchid	roundnumber	objectivelocation	winrole	endroundreason	roundduration	c
1	20170412	PS4	SECURE_AREA	CLUB_HOUSE	3268332769	3	BEDROOM	Defender	NoEnemies	13	
2	20170412	XONE	SECURE_AREA	HEREFORD_BASE	2329303669	4	DINING_ROOM	Defender	NoEnemies	6	
3	20170412	PS4	BOMB	FAVELAS	3622728049	3	2F_FOOTBALL_BEDROOM-2F_FOOTBALL_OFFICE	Defender	NoEnemies	6	
4	20170412	PS4	BOMB	FAVELAS	3622728049	3	2F_FOOTBALL_BEDROOM-2F_FOOTBALL_OFFICE	Defender	NoEnemies	7	
5	20170412	XONE	BOMB	BARTLETT_U.	3337494129	5	KITCHEN-PIANO_ROOM	Defender	NoEnemies	211	
6	20170412	XONE	SECURE_AREA	CLUB_HOUSE	1667325281	4	GARAGE	Defender	NoEnemies	6	
7	20170412	PS4	BOMB	CLUB_HOUSE	4185937829	5	CCTV_ROOM-CASH_ROOM	Defender	NoEnemies	190	
8	20170412	PS4	SECURE_AREA	CLUB_HOUSE	3268332769	3	BEDROOM	Defender	NoEnemies	13	
9	20170412	PS4	SECURE_AREA	FAVELAS	239047661	8	3F_PACKAGING_ROOM	Defender	NoEnemies	19	
10	20170412	XONE	BOMB	CHALET	589111221	4	WINE_CELLAR-SNOWMOBILE_GARAGE	Defender	NoEnemies	18	
11	20170412	XONE	SECURE_AREA	CLUB_HOUSE	1667325281	4	GARAGE	Defender	NoEnemies	6	
12	20170412	XONE	SECURE_AREA	HEREFORD_BASE	2329303669	4	DINING_ROOM	Defender	NoEnemies	6	

Each row represents one round in a match, rounds belong to the same matches will have the same value in `matcheid`.

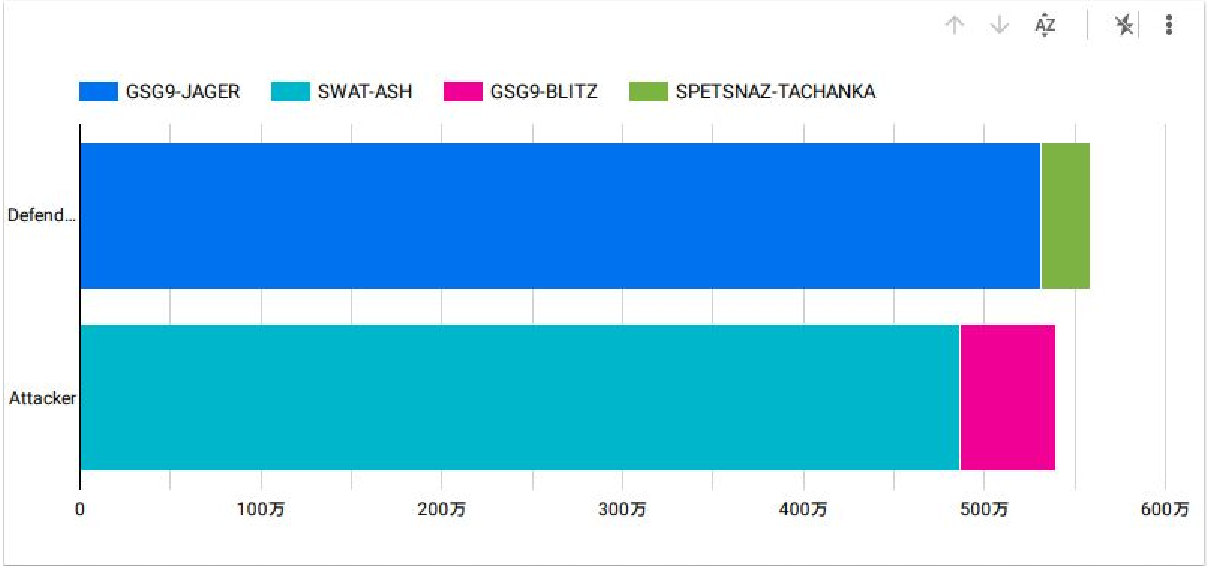
Game Mode

According to the data (see Appendix Query1), Hostage is the least played game mode while Secure area is the most played among all platform:

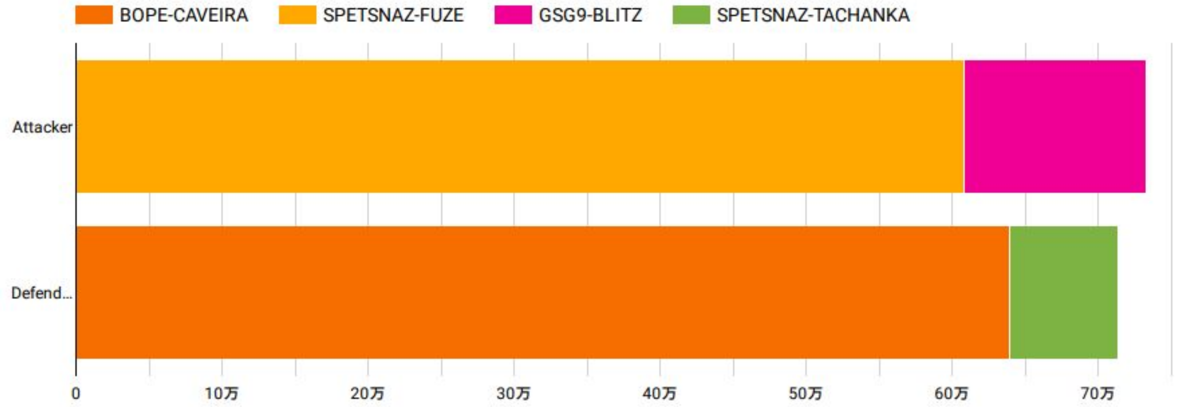


Most Popular Operators v.s. Least Popular Operators

According to the data (Appendix I Query 2), For most of the skill level, The most popular operator for Attater is Ash while the least popular operator is Blitz. For the defender, the most popular operator is JAGER while the least popular operator is TACHANKA :



But Things are different for Coper Players, they like Fuze for attack and Caveira for Defend:

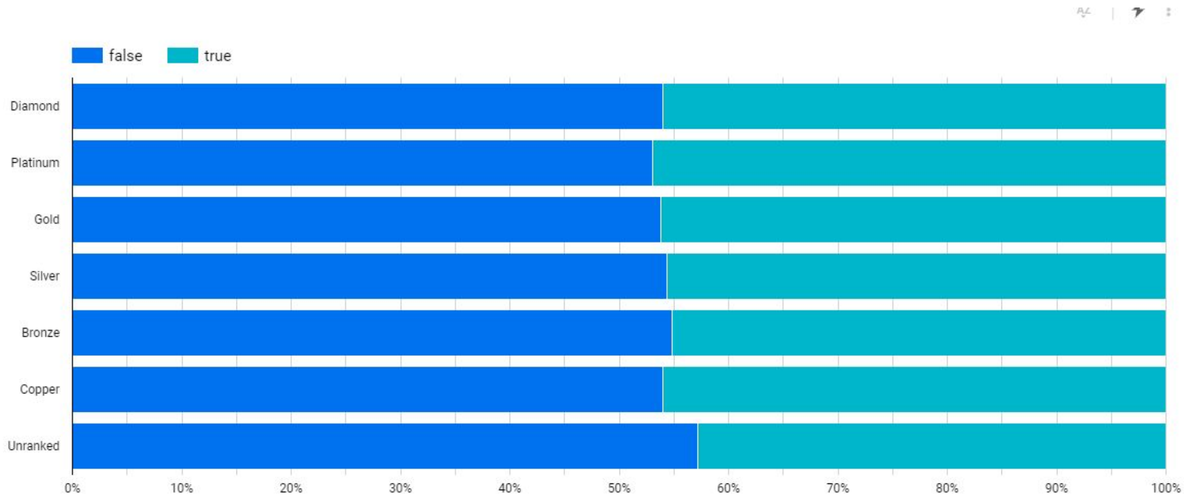


This statistic does not include operator with “RESERVED” (The recruiter with certain weapons)

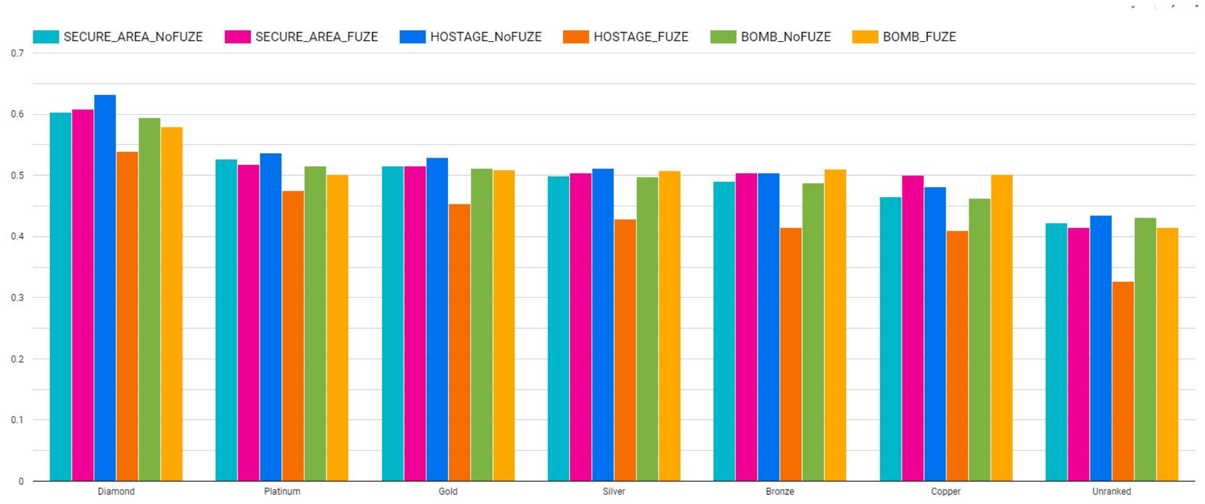
Data Analysis

Fuze Pick and Win Rate

The first analysis focuses on the relationship between win ratio and picks FUZE in the “hostage” game mode. The result shows that for most rank levels, pick FUZE will make the team less possible to win the hostage round. (See Appendix I Query 3) The only exception is Diamond matches on XONE, pick FUZE will lead to a higher win rate (66.7%) compare to not pick FUZE (62.2%). However, the number of matches are related small, only 377 matches (See Appendix I query 4) happened in that season. Below chat shows the win rate for pick/not pick FUZE in each rank level regardless platform: (false/true means with_fuze = false/true)



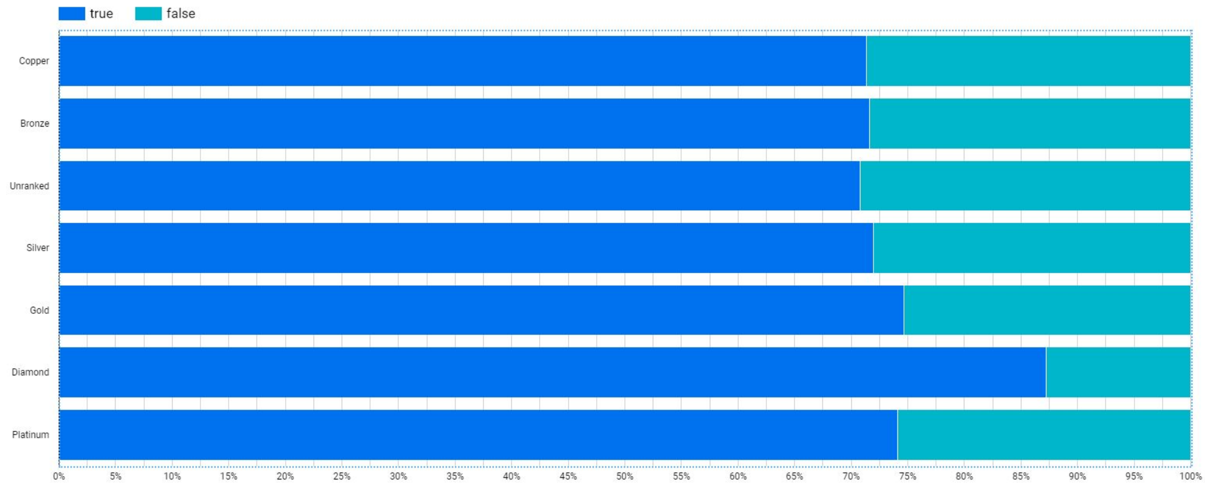
To exclude the possibility that fuze is a real bad operator, the below analysis (See Appendix I query 4) includes the other mode and demonstrate that fuze will harm the team only in hostage mode:



Even the result complies with the statement, one argument can be proposed: the main reason that Fuze is not suitable for the hostage is that his gadget designed for area control and mass damage, it can easily kill the hostage and result in a round loss. This situation usually comes from the player not familiar with maps, game mechanisms, and operators. It should happen more frequently in lower level rank than the higher rank. However, in the chart, there is no significant difference between these two categories of players. So, does the lower win rate really cause by Fuze accidentally kill the hostage?

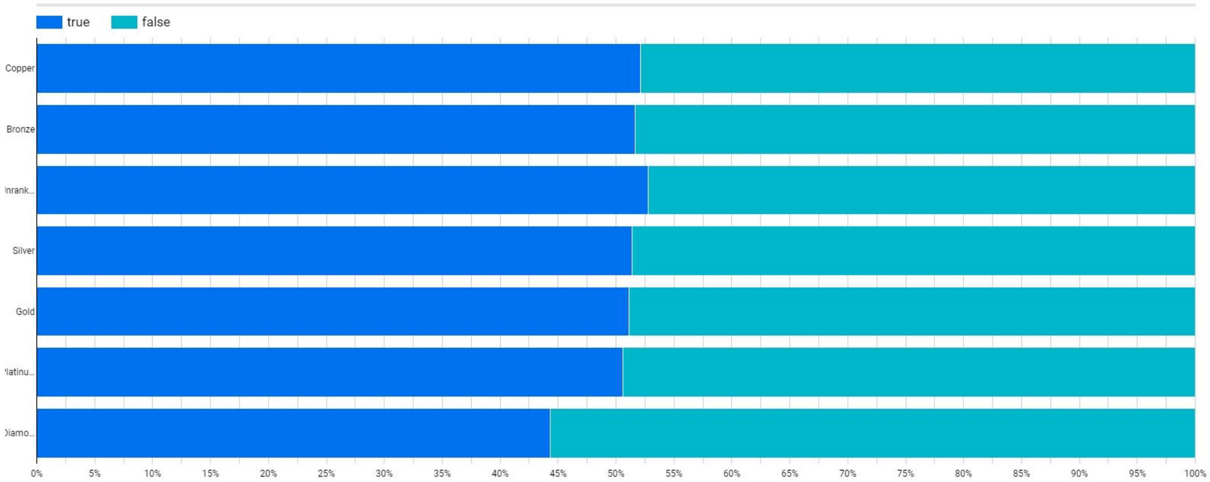
Does Hostage Actually be Fuzed?

From the analysis (See Appendix I query 6), all rank level suffered a higher change of hostage KIA loss when the team picks fuze:



Again, the result supports the statement. The main abnormality is the most skilled player even has the highest chance to kill the hostage when they pick fuze.

Consider a similar gadget with FUZE, the frag grenade, it can cause accidental hostage kill. However, chose frag grenade will not significantly increase hostage KIA loss (Appendix I query 7). In Diamond, choose frag grenade even lower the possibility of a hostage kill:



these analyses indicate that choose fuze will highly increase the risk of kill hostage, but there is no direct data support the kills are caused by FUZE’s gadget.

Summarize

Pick Fuze for Hostage game mode is clearly hurt the team, but the exact reason it not clear.

Conclusion and Future Works

The statement “do not choose fuze for hostage” is supported by the data in terms of win rate and risk of kill hostage. Future works may include:

- Find out if pick fuze affect other teammates’ operator pick
- Consider the map/objective location as an additional dimension for data analysis
- The reason for Diamond player Fuze more hostage
- Improve query efficiency

Reference

1. How to find the most frequent item of an array in js：https://medium.com/@AmJustSam/how-to-find-most-frequent-item-of-an-array-12015df68c65

Appendix I BQ Queries

Nums of competitions in each Platform for each mode:

```
SELECT platform, gamemode, count(DISTINCT matchid) as num_of_matches from
`Doc_me.r6_s5_data` group by platform, gamemode
```

Most Popular v.s Least Popular Operator

```
WITH
  oprator_stat AS(
  SELECT
    skillrank,
    operator,
    role,
    count(*) as pick
  FROM `Doc_me.r6_s5_data`
  WHERE operator NOT LIKE '%RESERVE%'
  group by 1, 2, 3

),
operator_pop AS(
  SELECT
    skillrank,
    operator,
    role,
    pick,
    RANK() OVER (PARTITION BY skillrank, role ORDER BY pick DESC) AS pop,
    RANK() OVER (PARTITION BY skillrank, role ORDER BY pick ASC) AS unpop
  FROM oprator_stat
)

SELECT skillrank, role, operator, pick FROM operator_pop WHERE pop = 1 OR unpop = 1
```

Win ratio with/without fuze in hostage among all ranks

```
# below code come from
#
https://medium.com/@AmJustSam/how-to-find-most-frequent-item-of-an-array-12015df68c6
5
# Each mathes has 5 player per team, they may has different rank level, this query use the
# most rank level as their "overall rank level"
CREATE TEMP FUNCTION
  find_rank(params ARRAY<string>)
  RETURNS string
  LANGUAGE js AS """
  var counts = {};
  var compare = 0;
  var mostFrequent;
  for(var i = 0, len = params.length; i < len; i++){
      var word = params[i];
      if(counts[word] === undefined){
         counts[word] = 1;
      }else{
         counts[word] = counts[word] + 1;
      }
      if(counts[word] > compare){
         compare = counts[word];
         mostFrequent = params[i];
      }
 }
  return mostFrequent;
""";

# group rounds
WITH
  round_info AS(
  SELECT
    platform,
    matchid,
    gamemode,
    roundnumber,
    haswon,
    ARRAY_AGG(operator) AS operators,
    ARRAY_AGG(skillrank) AS players_rank
  FROM `Doc_me.r6_s5_data`
  group by 1, 2, 3, 4,5
),
  fuze_the_hostage AS(
  SELECT
    platform,
    find_rank(players_rank) as ranklevel,
    gamemode,
    haswon,
    'SPETSNAZ-FUZE' IN UNNEST(operators) AS with_fuze,
  FROM round_info
  )

SELECT
  platform,
  ranklevel,
  with_fuze,
  SUM(haswon) / COUNT(haswon) AS win_rate,
FROM fuze_the_hostage
WHERE gamemode = 'HOSTAGE'
GROUP BY 1,2,3
ORDER BY platform, ranklevel, with_fuze
```

Include all Mode for win rate with/without FUZE

Above query delete `WHERE gamemode = 'HOSTAGE'` and export output into table “win_with_fuze_all_mode”

```
select
  ranklevel,
  win_rate,
  CASE
    WHEN with_fuze THEN CONCAT(gamemode, '_FUZE')
    ELSE CONCAT(gamemode, '_NoFUZE')
  END AS category
FROM `Doc_me.win_with_fuze_all_mode`
```

Count XONE Diamond Hostage mathes:

```
select count(DISTINCT matchid) from `Doc_me.r6_s5_data` where platform = 'XONE' AND
skillrank = 'Diamond' AND gamemode = 'HOSTAGE'
```

Hostage KIA rate with/without FUZE

```
# below code come from
#
https://medium.com/@AmJustSam/how-to-find-most-frequent-item-of-an-array-12015df68c6
5
# Each mathes has 5 player per team, they may has different rank level, this query use the
# most rank level as their "overall rank level"
CREATE TEMP FUNCTION
  find_rank(params ARRAY<string>)
  RETURNS string
  LANGUAGE js AS """
  var counts = {};
  var compare = 0;
  var mostFrequent;
  for(var i = 0, len = params.length; i < len; i++){
      var word = params[i];
      if(counts[word] === undefined){
         counts[word] = 1;
      }else{
         counts[word] = counts[word] + 1;
      }
      if(counts[word] > compare){
         compare = counts[word];
         mostFrequent = params[i];
      }
 }
  return mostFrequent;
""";

# group rounds
WITH
  round_info AS(
  SELECT
    matchid,
    gamemode,
    endroundreason,
    roundnumber,
    haswon,
    ARRAY_AGG(operator) AS operators,
    ARRAY_AGG(skillrank) AS players_rank
  FROM `Doc_me.r6_s5_data`
  group by 1, 2, 3, 4,5
),
  fuze_the_hostage AS(
  SELECT
    find_rank(players_rank) as ranklevel,
    gamemode,
    CASE
      WHEN endroundreason = 'AttackersKilledHostage' THEN 1
      ELSE 0
    END AS hostage_fuzed,
    'SPETSNAZ-FUZE' IN UNNEST(operators) AS with_fuze,
  FROM round_info
  )

SELECT
  ranklevel,
  with_fuze,
  SUM(hostage_fuzed) / COUNT(hostage_fuzed) AS hostage_KIA_rate,
FROM fuze_the_hostage
WHERE gamemode = 'HOSTAGE'
GROUP BY 1,2
ORDER BY ranklevel, with_fuze
```

Hostage KIA rate with/without Nade

```
 # below code come from
#
https://medium.com/@AmJustSam/how-to-find-most-frequent-item-of-an-array-12015df68c6
5
# Each mathes has 5 player per team, they may has different rank level, this query use the
# most rank level as their "overall rank level"
CREATE TEMP FUNCTION
  find_rank(params ARRAY<string>)
  RETURNS string
  LANGUAGE js AS """
  var counts = {};
  var compare = 0;
  var mostFrequent;
  for(var i = 0, len = params.length; i < len; i++){
      var word = params[i];
      if(counts[word] === undefined){
         counts[word] = 1;
      }else{
         counts[word] = counts[word] + 1;
      }
      if(counts[word] > compare){
         compare = counts[word];
         mostFrequent = params[i];
      }
 }
  return mostFrequent;
""";

# group rounds
WITH
  round_info AS(
  SELECT
    matchid,
    gamemode,
    secondarygadget,
    endroundreason,
    roundnumber,
    haswon,
    ARRAY_AGG(secondarygadget) AS secondary_gadget,
    ARRAY_AGG(skillrank) AS players_rank
  FROM `Doc_me.r6_s5_data`
  group by 1, 2, 3, 4,5,6
),
  fuze_the_hostage AS(
  SELECT
    find_rank(players_rank) as ranklevel,
    gamemode,
    CASE
      WHEN endroundreason = 'AttackersKilledHostage' THEN 1
      ELSE 0
    END AS hostage_killed,
    'FRAG_GRENADE' IN UNNEST(secondary_gadget) AS with_nade,
  FROM round_info
  )

SELECT
  ranklevel,
  with_nade,
  SUM(hostage_killed) / COUNT(hostage_killed) AS hostage_KIA_rate,
FROM fuze_the_hostage
WHERE gamemode = 'HOSTAGE'
GROUP BY 1,2
ORDER BY ranklevel, with_nade
```

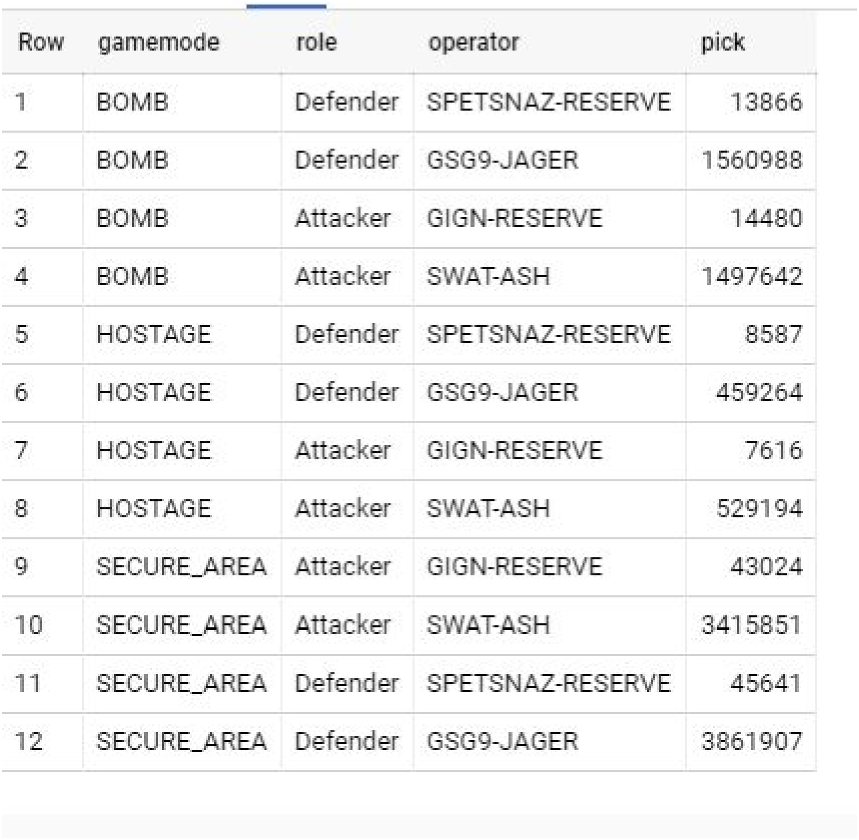
Appendix II Experiments BQ

Most/Least Pop operators among all game mode

```
WITH
  oprator_stat AS(
  SELECT
    gamemode,
    operator,
    role,
    count(*) as pick
  FROM `Doc_me.r6_s5_data`
  group by 1, 2,3
),
operator_pop AS(
  SELECT
    gamemode,
    operator,
    role,
    pick,
    RANK() OVER (PARTITION BY gamemode, role ORDER BY pick DESC) AS pop,
    RANK() OVER (PARTITION BY gamemode, role ORDER BY pick ASC) AS unpop
  FROM oprator_stat
)

SELECT gamemode, role, operator, pick FROM operator_pop WHERE pop = 1 OR unpop =
1
```

Eliminated because no difference between game mode:



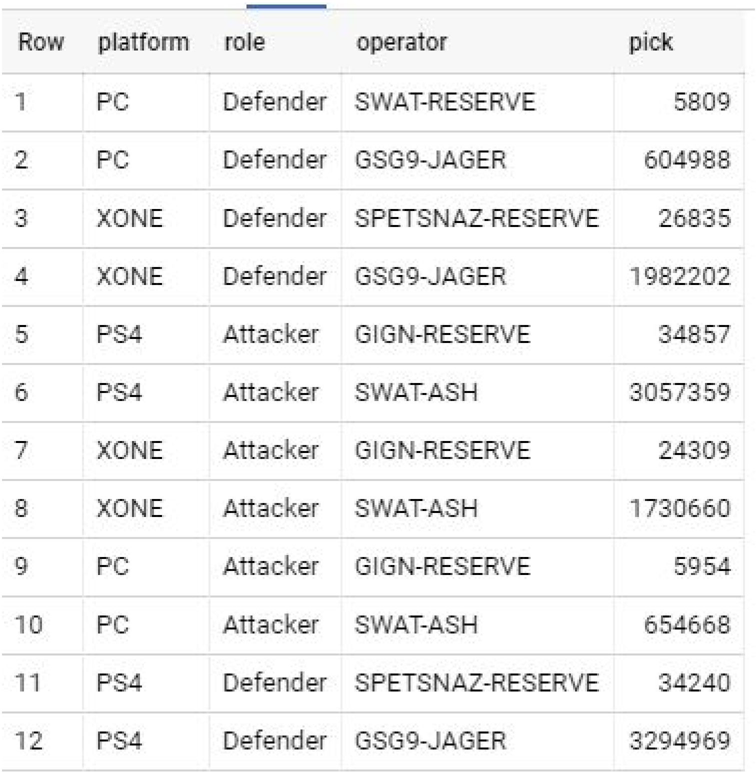
Row	gamemode	role	operator	pick
1	BOMB	Defender	SPETSNAZ-RESERVE	13866
2	BOMB	Defender	GSG9-JAGER	1560988
3	BOMB	Attacker	GIGN-RESERVE	14480
4	BOMB	Attacker	SWAT-ASH	1497642
5	HOSTAGE	Defender	SPETSNAZ-RESERVE	8587
6	HOSTAGE	Defender	GSG9-JAGER	459264
7	HOSTAGE	Attacker	GIGN-RESERVE	7616
8	HOSTAGE	Attacker	SWAT-ASH	529194
9	SECURE_AREA	Attacker	GIGN-RESERVE	43024
10	SECURE_AREA	Attacker	SWAT-ASH	3415851
11	SECURE_AREA	Defender	SPETSNAZ-RESERVE	45641
12	SECURE_AREA	Defender	GSG9-JAGER	3861907

Most/Least Pop operators among all platform

```
WITH
  oprator_stat AS(
  SELECT
    platform,
    operator,
    role,
    count(*) as pick
  FROM `Doc_me.r6_s5_data`
  group by 1, 2,3
),
operator_pop AS(
  SELECT
    platform,
    operator,
    role,
    pick,
    RANK() OVER (PARTITION BY platform, role ORDER BY pick DESC) AS pop,
    RANK() OVER (PARTITION BY platform, role ORDER BY pick ASC) AS unpop
  FROM oprator_stat
)

SELECT platform, role, operator, pick FROM operator_pop WHERE pop = 1 OR unpop = 1
```

Eliminated because no difference between platform:



Row	platform	role	operator	pick
1	PC	Defender	SWAT-RESERVE	5809
2	PC	Defender	GSG9-JAGER	604988
3	XONE	Defender	SPETSNAZ-RESERVE	26835
4	XONE	Defender	GSG9-JAGER	1982202
5	PS4	Attacker	GIGN-RESERVE	34857
6	PS4	Attacker	SWAT-ASH	3057359
7	XONE	Attacker	GIGN-RESERVE	24309
8	XONE	Attacker	SWAT-ASH	1730660
9	PC	Attacker	GIGN-RESERVE	5954
10	PC	Attacker	SWAT-ASH	654668
11	PS4	Defender	SPETSNAZ-RESERVE	34240
12	PS4	Defender	GSG9-JAGER	3294969