Monitoring Game Updates

Character balances for the online game—Battlerite

Ryan Tran, 05/10/18

Introduction

Proposal

- Client: Game Developers
- Problem:
  - Online games require constant character balances
- Goal:
  - Use the Battlerite API as a demonstration
  - Determine which characters need changes
  - Monitor character statistics before/after balance changes

Data Wrangling

Collect data

Data Collection

- Create function to extract necessary information
  - Store in easily accessible DataFrame
- Create second function to get chunk of data
  - Uses first function
  - Gathers data for specified date
- Collected data from various days and hours

Character Mappings

- Mapped character IDs to corresponding names
  - gameplay.json
  - English.ini
- gameplay.json –> COLLECTED DATA –> English.ini –> Character Names

Exploratory Data Analysis

Check for missing data

Check if match data includes every character

- Many characters missing in higher leagues
  - Especially in 3v3 game mode

Check if match data includes every character (2)

```
Format : bronze
Game mode: overall
   Missing characters: None
Game mode: 2v2
   Missing (amount): 1
   Missing characters: Taya
Game mode: 3v3
   Missing (amount): 2
   Missing characters: Lucie, Varesh

Format : silver
Game mode: overall
   Missing characters: None
Game mode: 2v2
   Missing characters: None
Game mode: 3v3
   Missing (amount): 1
   Missing characters: Varesh

Format : platinum
Game mode: overall
   Missing characters: None
Game mode: 2v2
   Missing (amount): 1
   Missing characters: Taya
Game mode: 3v3
   Missing characters: None

Format : diamond
Game mode: overall
   Missing characters: None
Game mode: 2v2
   Missing (amount): 1
   Missing characters: Zander
Game mode: 3v3
   Missing (amount): 3
   Missing characters: Blossom, Iva, Pestilus

Format : champion
Game mode: overall
   Missing (amount): 9
   Missing characters: Ashka, Bakko, Blossom, Destiny, Jamila, Jumong, Oldur, Sirius, Ulric
Game mode: 2v2
   Missing (amount): 11
   Missing characters: Ashka, Bakko, Blossom, Destiny, Jamila, Jumong, Oldur, Pestilus, Sirius, Ulric, Zander
Game mode: 3v3
   Missing (amount): 20
   Missing characters: Alysia, Ashka, Bakko, Blossom, Destiny, Ezmo, Freya, Iva, Jamila, Jumong, Lucie, Oldur, Pearl, Poloma, R
aigon, Ruh Kaan, Sirius, Thorn, Ulric, Varesh
Format : grand_champ
Game mode: overall
   Missing (amount): 20
   Missing characters: Alysia, Ashka, Blossom, Destiny, Freya, Iva, Jade, Jamila, Lucie, Pearl, Poloma, Raigon, Rook, Ruh Kaan,
Shifu, Sirius, Taya, Thorn, Ulric, Varesh
Game mode: 2v2
   Missing (amount): 20
```

Import newly collected data, and repeat

- Collected new data
  - Included few more days and hours to collection
- Only one combination had missing characters
  - Satisfactory

```
Format : grand_champ
Game mode: overall
   Missing characters: None
Game mode: 2v2
   Missing characters: None
Game mode: 3v3
   Missing (amount): 1
   Missing characters: Jamila
```

Character Statistics

View win rates and pick rates

Win Rates

- Collect the number of wins for each character
  - Divide by number of times each character was played
- Done for each format and game mode combination
- Characters with missing data
  - Assigned a win rate of 0

Pick Rates

- Collect the number of times each character was played
  - Divide by total number of characters played
- Done for each format and game mode combination
- Characters with missing data
  - Assigned a pick rate of 0

Statistics Table

- Display both win rates and pick rates
  - Rearranged the indexes and columns
  - Then merged the two together

	stat	win_rate										pick_rate		
	format	overall	casual	ranked	bronze	silver	gold	platinum	diamond	champion	grand_champ	overall	casual	ranked
character	game_mode													
Alysia	2V2	0.493289	0.501329	0.482524	0.432343	0.504769	0.487903	0.464891	0.425926	0.133333	1.0	0.034475	0.031549	0.039363
	3V3	0.503205	0.514226	0.482873	0.517442	0.488333	0.482927	0.477419	0.404255	0.500000	0.0	0.040585	0.037334	0.048352
	overall	0.497414	0.507089	0.482653	0.463158	0.499462	0.485999	0.470263	0.415842	0.333333	0.5	0.036779	0.033895	0.042265
Ashka	2V2	0.494462	0.508079	0.465495	0.469363	0.448759	0.474916	0.480315	0.500000	0.612903	1.0	0.056853	0.061825	0.048548
	3V3	0.508435	0.513175	0.495421	0.468384	0.481959	0.530541	0.483516	0.547170	0.518519	0.0	0.064475	0.067042	0.058343

Inferential Statistics

Look for ways to improve performance

Create Functions

- First function: difference
  - Compute the difference between desired statistic
    - Win Rate or Pick Rate
- Second function: draw_bs_pairs
  - Bootstrap the data several times
    - Bootstrapping refers to a test in which dataset is resampled
    - Then a statistic is computed for resampled data

Hypothesis Tests

- Statistically significant difference in win/pick rates?
  - 2v2 vs 3v3
  - Bronze vs Champion league
- Ranked format most important
  - Played by serious players

Hypothesis Tests (2)

Game Modes:

- There is indeed a difference between 2v2 and 3v3 game modes
  - 44% of characters have significantly different win rates
  - 78% of characters have significantly different pick rates
- Game developers must take game modes into account

Hypothesis Tests (3)

Leagues:

- There is indeed a difference between leagues
  - 55% of characters have significantly different win rates
  - 85% of characters have significantly different pick rates
- Game developers must take player skill into account

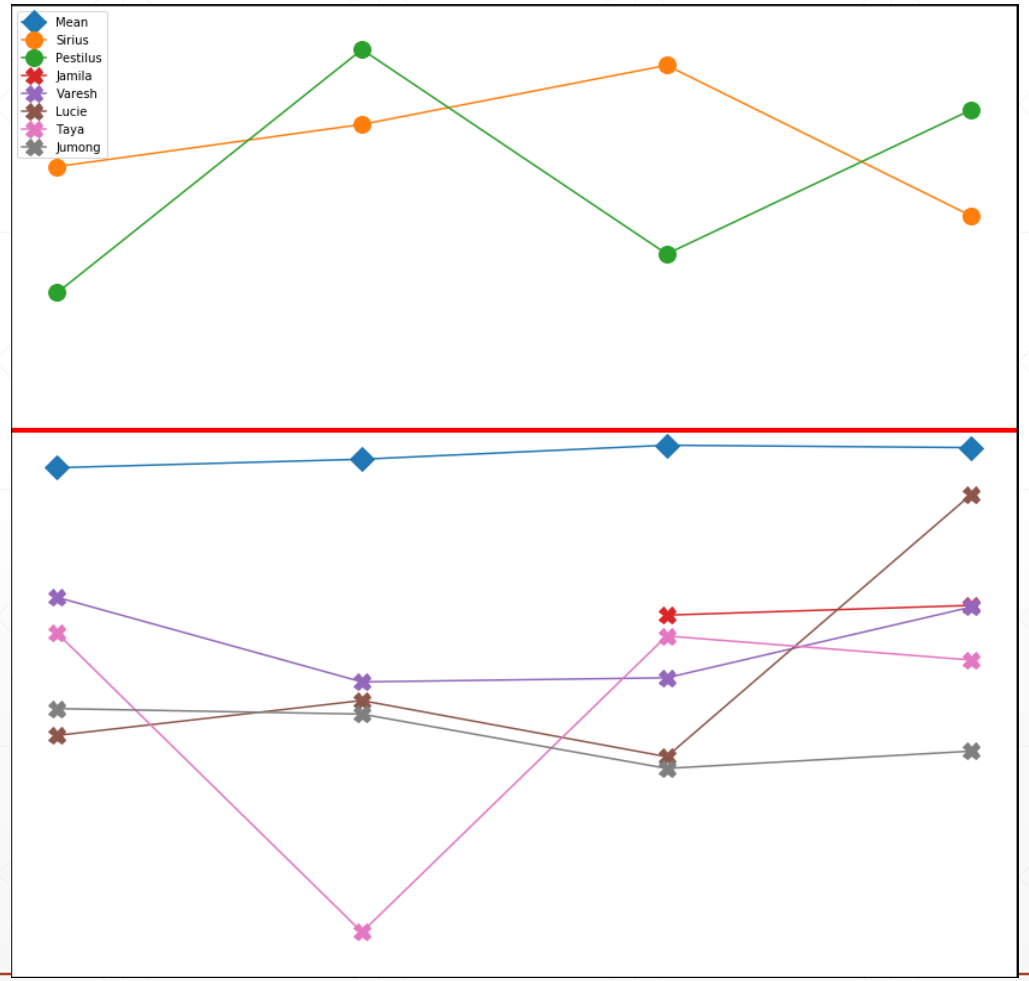
Monitoring Changes

Monitor imbalanced characters before and after changes

Find Imbalanced Characters

- Imbalanced characters
  - Characters with excessively high/low win rates
  - At least one standard deviation from the mean
- Plot the monthly win rate of each character
- REMINDER:
  - Ranked format most important

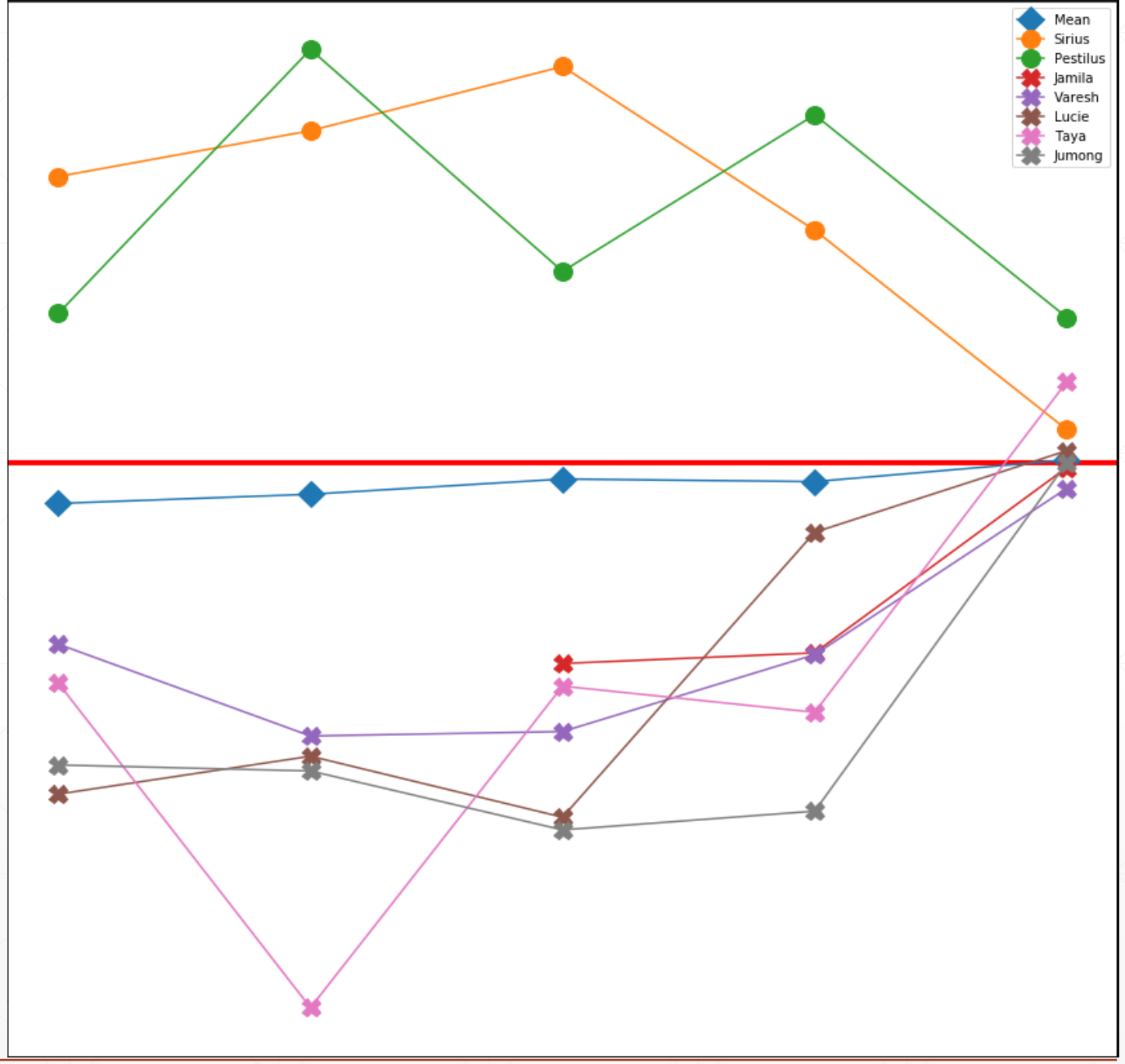
Find Imbalanced Characters (2)



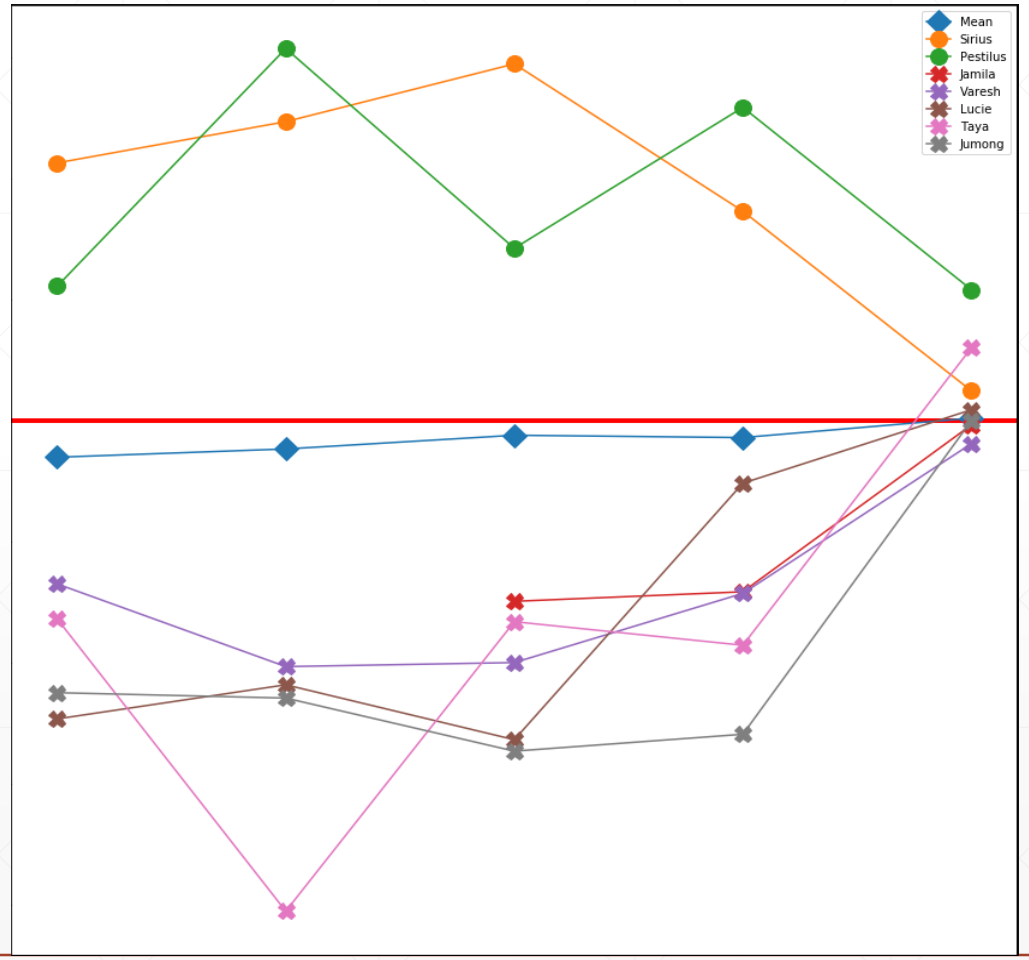

```
Excessively high win rates:
 character  game_mode
Sirius      overall      0.547324
Pestilus    overall      0.540463
Name: ranked, dtype: float64

Excessively low win rates:
 character  game_mode
Jamila      overall      0.470655
Varesh      overall      0.465686
Lucie       overall      0.458049
Taya        overall      0.453735
Jumong      overall      0.451389
```

Simulated Updates

- Simulate update
  - Generate random data
- Game developers can monitor characters
  - Before and after changes



Simulated Updates (2)



Conclusions

Conclusions

- Pulled data from Battlerite API
  - Created table to easily view win/pick rates
- Determined statistical difference between win/pick rates
  - Game modes AND leagues
  - Must target each character in different ways
    - Reduce imbalances between game modes or player skill
- Monitor changes
  - Identify imbalanced characters
  - Visualize win rates over time
  - Monitor changes before/after changes