



# Version 2.1 (latest)



예시 DataFrame은 모두 2021 PGI.S를 대상으로 합니다

## 의존하는 모듈

```
import requests
import pandas as pd
import datetime
import numpy as np
import os
import json
import time
```

## class: PUBG\_esports

tournament, matches, telemetry class에 api 입력 없이 접근 가능하도록 하는 클래스

- parameter
  - api\_key
- instance variable
  - api\_key
- method
  - \_init\_
  - tournament(self, tournament\_id): api key 입력 없이 tournament class 객체 획득
  - matches(self, match\_id): api key 입력 없이 matches class 객체 획득
  - telemetry(self, telemetry\_or\_match\_id): api key 입력 없이 telemetry class 객체 획득

## class: tournament

PUBG API 접근해 데이터 추출 + 토너먼트 단위의 matches 객체 획득

- parameter
  - api\_key
  - tournament\_id (리스트 가능)
- instance variable
  - api\_key
  - tournament\_id
  - match\_id
- method
  - match\_info\_list(self)
  - matches(self): tournament의 match\_id를 입력한 matches 객체 획득

## class: matches

PUBG API 접근해 데이터 추출 + telemetry object 획득

- matches 객체의 매치들은 duration 1200초 이상인 매치만 선별됨
- parameter
  - api\_key
  - match\_id (리스트 가능)
- instance variable
  - api\_key
  - match\_id
  - match\_json
  - telemetry\_link
- method
  - \_init\_ : match 길이가 1200초 이상인 match들만 선별
  - raw\_stats\_df(self)
  - participant\_df(self): matches의 참여 선수 정보 DataFrame 획득
  - telemetry(self): matches의 match\_id(리스트 가능)를 입력한 telemetry 객체 획득

## class: telemetry

PUBG API 접근해 데이터 추출

- parameter
  - api\_key
  - telemetry\_or\_match\_id (리스트 가능)

- **instance variable**
  - api\_key
  - match\_id
  - tel\_json
  - telemetry\_link
- **method**
  - \_init\_
    - telemetry json 파일을 local 경로에서 찾아 불러오거나, 없다면 API 호출하여 획득한 telemetry json 파일을 local 경로에 저장함
    - telemetry 각각의 로그에 match\_id를 삽입
  - **log\_json(self, log\_name)**: 원하는 telemetry\_log의 log\_name을 입력해 log의 list 획득
  - **kill\_log\_json(self)**: 'LogPlayerKill' log의 list 획득
  - **attack\_log\_json(self)**: 'LogPlayerAttack' log의 list 획득
  - **damage\_log\_json(self)**: 'LogPlayerTakeDamage' log의 list 획득
  - **groggy\_log\_json(self)**: 'LogPlayerMakeGroggy' log의 list 획득

class: Indicators

집계된 데이터 기반으로 지표화

- **parameter**
  - PUBG\_esports 객체
  - tournament\_id\_list (선택)
  - match\_id\_list (선택)
- **instance variable**
  - PUBG\_esports
  - match\_id\_
  - tournament\_id
  - match\_id
  - raw\_stats\_df
- **method**
  - **First\_Indicators(self)**: Indicators에 입력된 tournament\_id, match\_id들을 대상으로 집계한 1차 지표 DataFrame들을 dictionary 형태로 획득
  - ▼ stats\_df(raw\_stats\_df)

stats\_df

https://docs.google.com/spreadsheets/d/1nJPEZ9kPV3mJm-tJe8LdTxBMJ2rAECjHkUDhUI2\_LUA/edit?usp=drivesdk
  - ▼ stats\_groupby\_df(stats\_df)

stats\_groupby\_df

https://docs.google.com/spreadsheets/d/1ns66hMevW-KANbxuciPyHak6uMYxtrUpesc\_LEQgDwM/edit?usp=drivesdk
  - **Second\_Indicators(self)**: Indicators에 입력된 tournament\_id, match\_id들을 대상으로 집계한 2차 지표 DataFrame들을 dictionary 형태로 획득
  - ▼ stats\_per\_minute\_df(stats\_groupby\_df)

stats\_per\_minute\_df

https://docs.google.com/spreadsheets/d/1PKjQCyZyCOK4vVz2WkcdNF9MXw9FjHnDv5QeT3zNjv8/edit?usp=drivesdk
  - ▼ raw\_attack\_damage\_df(attack\_log\_json, damage\_log\_json)

raw\_attack\_damage\_df

https://docs.google.com/spreadsheets/d/1d6h4gNEBG1mDSUcn9LIRVqxktRN5aRCCTycl7T9QUmM/edit?usp=drivesdk
  - ▼ attack\_damage\_df(raw\_attack\_damage\_df)

attack\_damage\_df

https://docs.google.com/spreadsheets/d/1Ofnx0niURJcvNbFvhpU39G3Cz1fCGJ7JvRVpokZ2-J0/edit?usp=drivesdk
  - ▼ attack\_damage\_adj\_df(raw\_attack\_damage\_df)

attack\_damage\_adj\_df

https://docs.google.com/spreadsheets/d/1oxy2eq-WU\_fmX9UA-cSUjNEUPZrOGvn0Uwr\_wBn5Nlc/edit?usp=drivesdk
  - ▼ body\_part\_df(raw\_attack\_damage\_df)

body\_part\_df

https://docs.google.com/spreadsheets/d/1ZTCrglktxdn89wHZRSkTj3l5sshHWXaV3RwYJW0IHcU/edit?usp=drivesdk
  - ▼ steal\_kill\_df(kill\_log\_json, groggy\_log\_json)

steal\_kill\_df

https://docs.google.com/spreadsheets/d/14dEzuqM4Vjk8T6VbUGP9o58aV0SamLATahsSW6aQ5So/edit?usp=drivesdk
  - ▼ heal\_boost\_revive\_df(raw\_stats\_df, telemetry\_json)

heal\_boost\_revive\_df

https://docs.google.com/spreadsheets/d/1AOl1\_XdEgMgCsnSdRNu2KE1orJSbNRAXs5eoFs2sBfQ/edit?usp=drivesdk
  - ▼ stats\_selected\_df(attack\_damage\_adj\_df, body\_part\_df, steal\_kill\_df):

stats\_selected\_df

https://docs.google.com/spreadsheets/d/1yc5oQ6bqrbca6mlCdAV-oM6oifo32oDjh5pP4pfIMWQ/edit?usp=drivesdk

class: Power Rank

지표 활용하여 피지컬 랭크 도출

- **parameter**

- (tournament, match id 등이 입력된) Indicators 객체
- **instance variable**
  - Indicators
  - First\_Indicators
  - Second\_Indicators
  - PUBG\_esports
  - raw\_stats\_df
- **method**
  - ▼ tournament\_rank(self, tournament\_tier): 토너먼트별로 집계한 파워랭킹 DataFrame (집계 대상 매치의 25% 초과 참여 선수)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/011cdad8-0416-44d8-801f-01b9305f2d09/pgis\\_rank.csv](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/011cdad8-0416-44d8-801f-01b9305f2d09/pgis_rank.csv)

- ingredient(raw\_stats\_df, raw\_attack\_damage\_df, steal\_kill\_df, fight\_ingredients\_df)
- indicators(ingredient)
- rank(indicators, tournament\_tier)
- \_rank(indicators, match\_groupby\_select): Z-Score활용한 이전 버전 랭킹

## class: Fight

매치를 교전 단위로 세분화


- **parameter**
  - PUBG\_esports 객체
  - tournament\_id\_list (선택)
  - match\_id\_list (선택)
- **instance variable**
  - PUBG\_esports
  - match\_id
  - tournemant\_id
  - telemetry\_object
- **method**
  - **Fights\_dfs(self)**: 교전 관련 DataFrame들을 Dictionary 형태로 획득
  - **Log\_df(self)**: Fight class의 tournament\_id\_list, match\_id\_list를 입력하여 fight\_log\_df의 결과물 획득
  - **Groupby\_df(self)**: Fight class의 tournament\_id\_list, match\_id\_list를 입력하여 fight\_log\_df를 획득 → fight\_groupby\_df 함수에 입력하여 결과물 획득
  - **Ingredients\_df(self)**: Fight class의 tournament\_id\_list, match\_id\_list를 입력하여 fight\_log\_df를 획득 → fight\_groupby\_df 함수에 입력하여 결과물 획득 → fight\_ingredients\_df 함수에 입력하여 결과물 획득
  - **Indicators\_df(self)**: Fight class의 tournament\_id\_list, match\_id\_list를 입력하여 fight\_log\_df를 획득 → fight\_groupby\_df 함수에 입력하여 결과물 획득 → fight\_ingredients\_df 함수에 입력하여 결과물 획득 → fight\_indicators\_df 함수에 획득하여 결과물 획득
  - ▼ **fight\_log\_df(telemetry\_object)**: Attack, Damage, Kill 로그를 테이블 형태로 변환한 교전 관련 로그
    - 주의! 1390000 rows (용량 큼니다)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7c7c6a35-2e1e-4a89-9fa0-495e4c206f3c/fight\\_log\\_df.csv](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/7c7c6a35-2e1e-4a89-9fa0-495e4c206f3c/fight_log_df.csv)

- LIMIT 1000

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5764533c-0ecb-498e-9d99-8dd935023686/fight\\_log\\_df\\_\\_.csv](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5764533c-0ecb-498e-9d99-8dd935023686/fight_log_df__.csv)


- ▼ **fight\_groupby\_df(fight\_log\_df)**: 교전 관련 로그를 교전별로 Group By 한 교전 발생 로그

 **fight\_groupby\_df**[https://docs.google.com/spreadsheets/d/1lTY3oun2G9d4r4VtIPsnoJty0\\_6uhA0vbxw4AvCKO2WU/edit?usp=drivesdk](https://docs.google.com/spreadsheets/d/1lTY3oun2G9d4r4VtIPsnoJty0_6uhA0vbxw4AvCKO2WU/edit?usp=drivesdk)

- ▼ **fight\_ingredients\_df(fight\_groupby\_df)**: 교전 발생 로그를 선수단위로 Group By 하여 집계

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4d4e2f16-eee7-40f0-a30e-13a4083a0a6d/pgis\\_fight\\_ingredients.csv](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/4d4e2f16-eee7-40f0-a30e-13a4083a0a6d/pgis_fight_ingredients.csv)

- ▼ **fight\_indicators\_df(fight\_indicators\_df)**: 교전 발생 로그를 선수 단위로 Group By 하여 지표화

 **fight\_indicator\_df**[https://docs.google.com/spreadsheets/d/1y27Vy4WAjvCpX\\_Mp-pggNhSVnkNR0ogKAbO6ZOWRfVl/edit?usp=drivesdk](https://docs.google.com/spreadsheets/d/1y27Vy4WAjvCpX_Mp-pggNhSVnkNR0ogKAbO6ZOWRfVl/edit?usp=drivesdk)