Statbotics

Release 2.0.1

Jan 14, 2023

Contents

Statbotics.io aims to modernize FRC data analytics through developing and distributing cutting-edge metrics and analysis. This Python API makes Expected Points Added (EPA) statistics just a few Python lines away! Currently we support queries on teams, years, events, and matches. Read below for usage and documentation.

Visit https://statbotics.io for more content!

CHAPTER 1

Usage

With Python>=3.8 and pip installed, run

```
pip install statbotics==2.0.1
```

Then in a Python file, create a Statbotics object and get started!

```
import statbotics

sb = statbotics.Statbotics()
print(sb.get_team(254))

>> {'team': 254, 'name': 'The Cheesy Poofs', 'offseason': False, 'state': 'CA',
↪'country': 'USA', 'district': None, 'rookie_year': 1999, 'active': True, 'norm_epa
↪': 1961.0, 'norm_epa_recent': 1956.0, 'norm_epa_mean': 1896.0, 'norm_epa_max': 2114.
↪0, ... }
```

Read below for more methods!

CHAPTER 2

API Reference

class statbotics.main.**Statbotics**

Main Object for interfacing with the Statbotics API

get_team(*team: int, fields: List[str] = ['all']*) → Dict[str, Any]

Function to retrieve information on an individual team

Parameters

- **team** – Team Number, integer
- **fields** – List of fields to return. The default is ["all"]

Returns a dictionary with team metadata and EPA statistics

get_teams(*country: Optional[str] = None, state: Optional[str] = None, district: Optional[str] = None, active: Optional[bool] = True, metric: str = 'team', ascending: Optional[bool] = None, limit: int = 100, offset: int = 0, fields: List[str] = ['all']*) → List[Dict[str, Any]]

Function to retrieve information on multiple teams

Parameters

- **country** – Restrict based on country (select country to include)
- **state** – US States and Canada provinces only. Can infer country.
- **district** – Use 2 or 3-letter key (ex: FIM, NE, etc)
- **active** – Restrict to active teams (played most recent season)
- **metric** – Order output by field (Ex: "-norm_epa", "team", etc). Default is "team".
- **ascending** – Order output ascending or descending. Default varies by metric.
- **limit** – Limits the output length to speed up queries. Max 10,000
- **offset** – Skips the first (offset) items when returning
- **fields** – List of fields to return. Default is ["all"]

Returns A list of dictionaries, each dictionary including team metadata and EPA statistics

get_year (*year: int, fields: List[str] = ['all']*) → Dict[str, Any]
Function to retrieve information for a specific year

Parameters

- **year** – Year, integer
- **fields** – List of fields to return. The default is ["all"]

Returns a dictionary with the year, match prediction statistics, and RP prediction statistics

get_years (*metric: str = 'year', ascending: Optional[bool] = None, limit: int = 100, offset: int = 0, fields: List[str] = ['all']*) → List[Dict[str, Any]]
Function to retrieve information on multiple years

Parameters

- **metric** – Order output by field. (Ex: "epa_acc", "epa_mse", etc). Default "year"
- **ascending** – Order output ascending or descending. Default varies by metric.
- **limit** – Limits the output length to speed up queries. Max 10,000
- **offset** – Skips the first (offset) items when returning
- **fields** – List of fields to return. Default is ["all"]

Returns A list of dictionaries, each dictionary including the year and match/RP prediction statistics

get_team_year (*team: int, year: int, fields: List[str] = ['all']*) → Dict[str, Any]
Function to retrieve information for a specific team's performance in a specific year

Parameters

- **team** – Team number, integer
- **year** – Year, integer
- **fields** – List of fields to return. The default is ["all"]

Returns a dictionary with the team, year, and EPA statistics

get_team_years (*team: Optional[int] = None, year: Optional[int] = None, country: Optional[str] = None, state: Optional[str] = None, district: Optional[str] = None, metric: str = 'team', ascending: Optional[bool] = None, limit: int = 100, offset: int = 0, fields: List[str] = ['all']*) → List[Dict[str, Any]]
Function to retrieve information on multiple (team, year) pairs

Parameters

- **team** – Restrict based on a specific team number
- **country** – Restrict based on country (select countries included)
- **state** – US States and Canada provinces only. Can infer country.
- **district** – Use 2 or 3-letter key (ex: FIM, NE, etc)
- **metric** – Order output by field. (Ex: "epa_end", "team", etc). Default "team"
- **ascending** – Order output ascending or descending. Default varies by metric.
- **limit** – Limits the output length to speed up queries. Max 10,000
- **offset** – Skips the first (offset) items when returning
- **fields** – List of fields to return. Default is ["all"]

Returns A list of dictionaries, each dictionary including the team, year, and EPA statistics

get_event(*event: str, fields: List[str] = ['all']*) → Dict[str, Any]
Function to retrieve information for a specific event

Parameters

- **event** – Event key, string (ex: "2019cur")
- **fields** – List of fields to return. The default is ["all"]

Returns a dictionary with the event and EPA statistics

get_events(*year: Optional[int] = None, country: Optional[str] = None, state: Optional[str] = None, district: Optional[str] = None, type: Union[str, int, None] = None, week: Optional[int] = None, metric: str = 'year', ascending: Optional[bool] = None, limit: int = 100, offset: int = 0, fields: List[str] = ['all']*) → List[Dict[str, Any]]
Function to retrieve information on multiple events

Parameters

- **year** – Restrict by specific year, integer
- **country** – Restrict based on country (select countries included)
- **state** – US States and Canada provinces only. Can infer country.
- **district** – Use 2 or 3-letter key (ex: FIM, NE, etc)
- **type** – 0=regional, 1=district, 2=district champ, 3=champs, 4=einstein
- **week** – Week of play, generally between 0 and 8
- **metric** – Order output bu field. (Ex: "epa_pre_playoffs", "epa_end", etc). Default "year"
- **ascending** – Order output ascending or descending. Default varies by metric.
- **limit** – Limits the output length to speed up queries. Max 10,000
- **offset** – Skips the first (offset) items when returning
- **fields** – List of fields to return. Default is ["all"]

Returns A list of dictionaries, each dictionary including the team, event and EPA statistics

get_team_event(*team: int, event: str, fields: List[str] = ['all']*) → Dict[str, Any]
Function to retrieve information for a specific (team, event) pair

Parameters

- **team** – Team number, integer
- **event** – Event key, string (ex: "2019cur")
- **fields** – List of fields to return. The default is ["all"]

Returns a dictionary with the event and EPA statistics

get_team_events(*team: Optional[int] = None, year: Optional[int] = None, event: Optional[str] = None, country: Optional[str] = None, state: Optional[str] = None, district: Optional[str] = None, type: Union[str, int, None] = None, week: Optional[int] = None, metric: str = 'year', ascending: Optional[bool] = None, limit: int = 0, offset: int = 0, fields: List[str] = ['all']*) → List[Dict[str, Any]]
Function to retrieve information on multiple (team, event) pairs

Parameters

- **team** – Restrict by team number, integer
- **year** – Restrict by specific year, integer
- **country** – Restrict based on country (select countries included)
- **state** – US States and Canada provinces only. Can infer country.
- **district** – Use 2 or 3-letter key (ex: FIM, NE, etc)
- **type** – 0=regional, 1=district, 2=district champ, 3=champs, 4=einstein
- **week** – Week of play, generally between 0 and 8
- **metric** – Order output by field. (Ex: "epa_pre_playoffs", "epa_end", etc). Default "year"
- **ascending** – Order output ascending or descending. Default varies by metric.
- **limit** – Limits the output length to speed up queries. Max 10,000
- **offset** – Skips the first (offset) items when returning
- **fields** – List of fields to return. Default is ["all"]

Returns A list of dictionaries, each dictionary including the team, event and EPA statistics

get_match (*match: str, fields: List[str] = ['all']*) → Dict[str, Any]

Function to retrieve information for a specific match

Parameters

- **match** – Match key, string (ex: "2019cur_qm1", "2019cmptx_f1m3")
- **fields** – List of fields to return. The default is ["all"]

Returns a dictionary with the match, score breakdowns, and predictions

get_matches (*team: Optional[int] = None, year: Optional[int] = None, event: Optional[str] = None, week: Optional[int] = None, elims: Optional[bool] = None, metric: str = 'time', ascending: Optional[bool] = None, limit: int = 200, offset: int = 0, fields: List[str] = ['all']*) → List[Dict[str, Any]]

Function to retrieve information on multiple matches

Parameters

- **team** – Restrict by team number, integer
- **year** – Restrict by specific year, integer
- **event** – Restrict by specific event key, string
- **week** – Week of play, generally between 0 and 8
- **elims** – Restrict to only elimination matches, default False
- **metric** – Order output by field. (Ex: "time", "epa_pre_playoffs", "epa_end", etc). Default "time"
- **ascending** – Order output ascending or descending. Default varies by metric.
- **limit** – Limits the output length to speed up queries. Max 10,000
- **offset** – Skips the first (offset) items when returning
- **fields** – List of fields to return. Default is ["all"]

Returns A list of dictionaries, each dictionary including the match, score breakdowns, and predictions

get_team_match(*team: int, match: str, fields: List[str] = ['all']*) → Dict[str, Any]
Function to retrieve information for a specific (team, match) pair

Parameters

- **team** – Team number, integer
- **match** – Match key, string (ex: "2019cur_qm1", "2019cmptx_f1m3")
- **fields** – List of fields to return. The default is ["all"]

Returns a dictionary with the team, match, alliance, and EPA statistics

get_team_matches(*team: Optional[int] = None, year: Optional[int] = None, event: Optional[str] = None, week: Optional[int] = None, match: Optional[str] = None, elims: Optional[bool] = None, metric: str = 'time', ascending: Optional[bool] = None, limit: int = 100, offset: int = 0, fields: List[str] = ['all']*) → List[Dict[str, Any]]
Function to retrieve information on multiple (team, match) pairs

Parameters

- **team** – Restrict by team number, integer
- **year** – Restrict by specific year, integer
- **event** – Restrict by specific event key, string
- **week** – Week of play, generally between 0 and 8
- **elims** – Restrict to only elimination matches, default False
- **metric** – Order output by field. (Ex: "time", "auto_epa", etc). Default "time"
- **ascending** – Order output ascending or descending. Default varies by metric.
- **limit** – Limits the output length to speed up queries. Max 10,000
- **offset** – Skips the first (offset) items when returning
- **fields** – List of fields to return. Default is ["all"]

Returns A list of dictionaries, each dictionary including the team, match, alliance, and then elo

CHAPTER 3

Contribute

If you are interested in contributing, reach out to Abhijit Gupta (avgupta456@gmail.com). Source code is available at github.com/avgupta456/statbotics.

CHAPTER 4

Support

If you are having issues, please let us know. We welcome issues and pull requests at github.com/avgupta456/statbotics.

CHAPTER 5

License

The project is licensed under the MIT license.

Index

G

S