BaseballCZ-Statistics Documentation

Release 0.0.1

Zuzana Ferkova

Jul 09, 2018

Contents

BaseballCZ-Statistics is an open-source API, written in Python that allows you to download and process statistics from the baseball.cz page.

To install the API please follow the instructions on Installation page.

For a quick start guide please visit Getting started page.

To find a specific functionality of the API visit Documentation page.

If you have any questions, or issues with the API, visit Contact page. If you enjoy using the API and would like to contribute, or have more functionality you'd like to add, please visit the Contribute page.

Last, but not least, I'd like to thank you for trying out and using the API!

CHAPTER 1

Introduction

1.1 Idea

BaseballCZ-Statistics aims to provide an easy to use Python API that could be used to automatize pipelines for statistics computation on the data from baseball.cz.

The API allows to either directly use the data downloaded on the remote server, or to automatically download the current CSV files locally, and work with those further.

1.2 Used Technologies

The API is build on several other Python libraries that are used to speed up the development process.

1.2.1 Data Download and Remote Server

As of now baseball.cz doesn't provide an easy access for automatic data download. Selenium is therefore used to simulate clicking the download button on the statistics page. The CSVs are downloaded locally and then sent to a remote FTP server.

To retrieve the data, API communicates via requests module with remote Flask server that sends back the loaded data.

1.2.2 Statistics

Data received from a remote server are parsed into Pandas Dataframe. Along with the Numpy the API provides vectorized computation of the requested statistics.

The API computes most statistics described at baseball-stat.cz, and can access all statistics provided in CSV files.

CHAPTER 2

Installation

The *baseballcz-statistics* API provides two ways to work with the data. First off, you can access a remote server and compute the statistics without needing to download the CSV data locally. This will use CSV files from a remote server that are downloaded daily.

The second option is to download the CSV files locally and work with the data.

2.1 Downloads

As this is a Python API, make sure you have Python 3+ installed. You can download Python from the official page. You can also download Anaconda distribution to have some of the required libraries installed right away.

To install the API first download the source codes from the baseballcz-statistics.github.com github page. You can either clone the repository, or click on the green *Clone or Download* button, then click *Download ZIP*.

If you downloaded ZIP file, extract it to your disk.

2.2 Modules

There are two modules available. *Remote* module contains scripts that work with data on the remote server. *Local* module contains scripts that allow you to download and work with the data locally.

2.3 Required Libraries

To install the required libraries, open the command line in the main directory of the API.

If you'd like to install and use *remote* module, run the following command:

```
pip install /remote/requirements.txt
```

If you'd like to install and use *local* module, run the following command:

```
pip install /local/requirements.txt
```

If you are installing *local* module you will need to set up Firefox webdriver. Please follow the instruction on Selenium page to correctly set them up. BaseballCZ-Statistics currently doesn't support downloading via other browsers.

2.4 Test Installation

To test whether all the parts of the installation process were successful, open the command line in the main directory of the API. Then start Python interpreter by typing:

```
python
```

and try to import a module from the API. If you installed *remote* module type:

```
import remote.client.stats
```

or if you installed *local* module type:

```
import local.stats
```

If you didn't encounter any errors, the API is ready to be used.

CHAPTER 3

Getting Started

Firstly, make sure the API is correctly installed and go through the installation process. If the installation was successful, open the command line in the main directory of the API and start Python with command:

```
python
```

Now we will discuss using the *remote* module, with specifications on the *local* module following later. At the end of the page we will also discuss how to set your own remote server to download the data on your own if necessary.

3.1 Using Data from Remote server

With the Python running in the command line, you can, for example, download the individual batting statistics, by running following commands:

```
import remote.client.download_stats as dw

# download individual batting statistics
data = dw.load_individual_batters()

# access downloaded data
data.player_data

# access summed data
data.summed_data

# access file name
data.file_name

# access time when the data was downloaded to the server
data.last_modified
```

This will download the most recent individual batter statistics and prepare them in a format that can be further processed. For more options on the kind of statistics that can be accessed please read the download_stats page.

3.2 Computing Statistics

If you have data loaded in the *data* variable, as in the section above, you can start computing the statistics:

```
import remote.client.stats

# get number of at-bat(AB) per player
stats.AB(data.player_data)

# compute isolated power(ISO) for player with
# at least 20 plate appearances
stats.ISO(data.player_data, 20)
```

If you have *batters* data loaded, but try to compute *pitcher* statistics an error may occur:

```
data.file_name
# bat_individual

stats.ERA(data.player_data)
# AttributeError: 'NoneType' object has no attribute 'dtype'
```

So make sure you are always only computing the valid statistics. Read stats documentation page to see, which methods are available for which data.

Computing statistics returns *Series* object that contains the computed statistics for every available player (this can either be all players, or only the players that fulfill the minimal plate appearances requirement). However, the result does not contain the names of the players the statistics belong to. You can therefore associate the computed statistics with the names as follows:

```
# get players at-bat
res = stats.AB(data.player_data)

# get names of the players associated with the computed statistics
stats.names_to_data(data.player_data, res)

# get names of the players associated with the computed statistics
# and sort the data in descending order
stats.sort_computed_data(data.player_data, res)

# get names of the players associated with the computed statistics
# and sort the data in ascending order
stats.sort_computed_data(data.player_data, res, ascending = True)
```

For more information on statistics that can be computed via the API read stats documentation page.

3.3 Downloading Data Locally

You can use *local* module to automatically download the data locally. To download tthe CSV files use following commands:

```
import local.download as dw
import local.constants as cs

# download all individual statistics
dw.download_single_stats()
```

(continues on next page)

(continued from previous page)

```

# download all team statistics
dw.download_team_stats()

# download all statistics
dw.download_all()

# download single statistic, individual batters
dw.download_stats(category = cs.CATEGORIES[0], team_stats = False)
```

If successful the data should be saved in the "/data/" directory.

You can then load the downloaded data as *Data_CSV* class:

```
import local.load_file

# load individual batters stored in /data/ directory
data = load_file.load_individual_batters()
```

This will allow you to use the loaded data the same way as in the *remote* module.

3.4 Setting Up Remote Server

You can also set your own server that will download the CSV data automatically and store them to the server's local drive, or a remote FTP. The provided scripts should allow you to easily deploy the server to any remote server with Python and Linux distribution on it.

Use *Dockerfile* to install all required dependencies.

Script *server.py* is the main server script that accepts the requests from the clients and returns requested files, if available.

Script *worker.py* is set up to automatically download the statistics locally, and send them to a FTP server.

For safety reason, information about the FTP server, such as server URL, or login information, are set locally as environmental variables. For more information read the *Data scraping <data_scraping.html>_* documentation page.

CHAPTER 4

Documentation

4.1 Remote Module

4.2 Local Module

CHAPTER 5

Contacts

In case you have any issue with the API, or would like to ask any related questions, there are several ways to do so.

Fig. 1: Direct contact via @FerkiZ

Fig. 2: Raise issue on GitHub page.

CHAPTER 6

Contribute

If you'd like to help out with the project, feel free to contact me! You can find the contact information on Contacts page. Some of the ways to contribute are:

- Adding/requesting new features
- Bug fixing
- Testing functionality
- Code review
- Anything else you can think of!

You can take a look at source codes at baseballcz-stats.github.com. Use the Issues page to request a new functionality, report bugs, and more.

If you decide to contribute to the project in any way, I'd like to heartfully thank you. I hope this project will serve you well!

CHAPTER 7

Indices and tables

- genindex
- modindex
- search