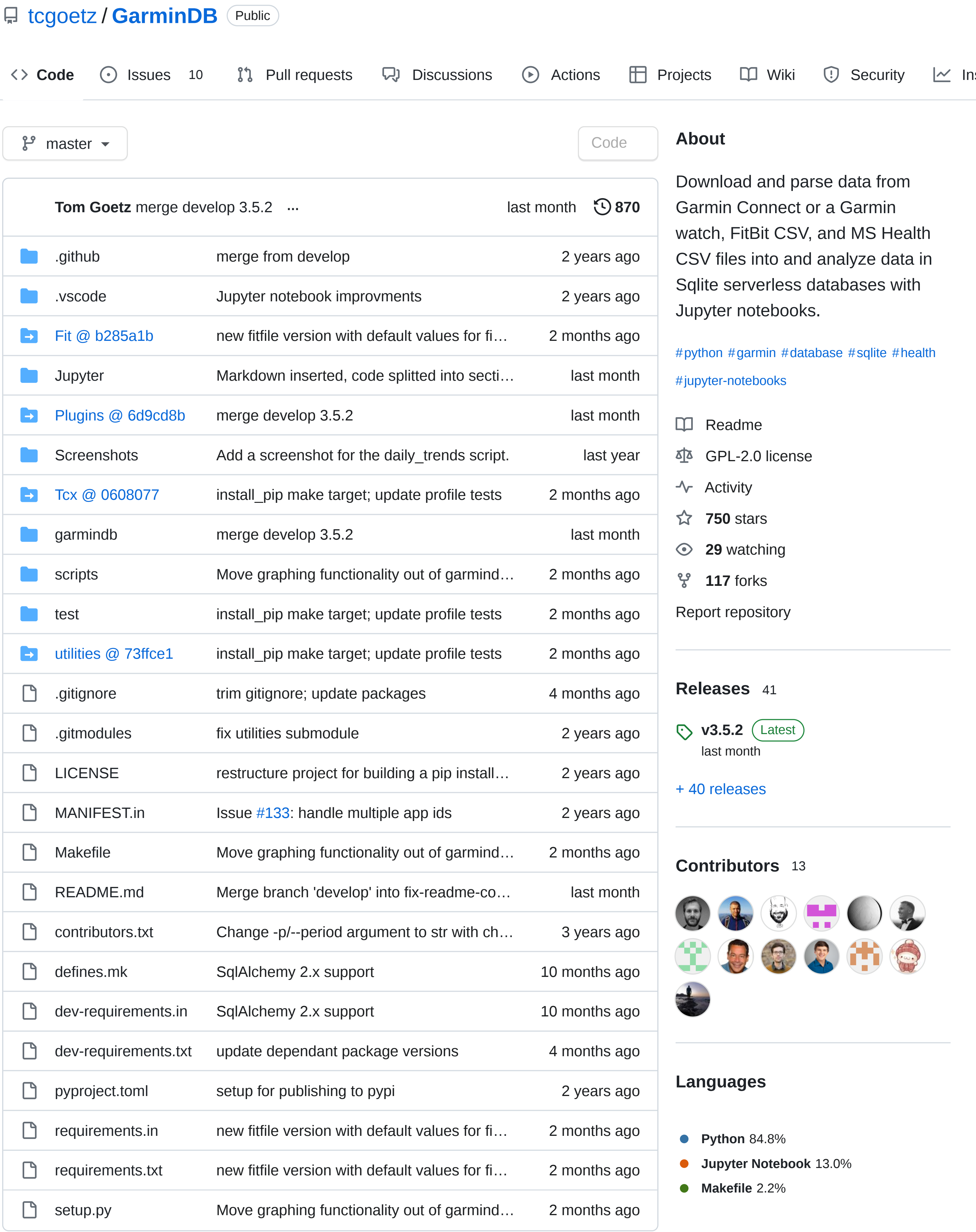


tcgoetz / GarminDB Public

Tom Goetz merge develop 3.5.2 ... last month 870

.github	merge from develop	2 years ago
.vscode	Jupyter notebook improvments	2 years ago
Fit @ b285a1b	new fitfile version with default values for fi…	2 months ago
Jupyter	Markdown inserted, code splitted into secti…	last month
Plugins @ 6d9cd8b	merge develop 3.5.2	last month
Screenshots	Add a screenshot for the daily_trends script.	last year
Tcx @ 0608077	install_pip make target; update profile tests	2 months ago
garmindb	merge develop 3.5.2	last month
scripts	Move graphing functionality out of garmind…	2 months ago
test	install_pip make target; update profile tests	2 months ago
utilities @ 73ffce1	install_pip make target; update profile tests	2 months ago
.gitignore	trim gitignore; update packages	4 months ago
.gitmodules	fix utilities submodule	2 years ago
LICENSE	restructure project for building a pip install…	2 years ago
MANIFEST.in	Issue #133: handle multiple app ids	2 years ago
Makefile	Move graphing functionality out of garmind…	2 months ago
README.md	Merge branch 'develop' into fix-readme-co…	last month
contributors.txt	Change -p/--period argument to str with ch…	3 years ago
defines.mk	SqlAlchemy 2.x support	10 months ago
dev-requirements.in	SqlAlchemy 2.x support	10 months ago
dev-requirements.txt	update dependant package versions	4 months ago
pyproject.toml	setup for publishing to pypi	2 years ago
requirements.in	new fitfile version with default values for fi…	2 months ago
requirements.txt	new fitfile version with default values for fi…	2 months ago
setup.py	Move graphing functionality out of garmind…	2 months ago

About

Download and parse data from Garmin Connect or a Garmin watch, FitBit CSV, and MS Health CSV files into and analyze data in Sqlite serverless databases with Jupyter notebooks.

#python #garmin #database #sqlite #health #jupyter-notebooks

Readme
GPL-2.0 license
Activity
750 stars
29 watching
117 forks
Report repository

Releases 41

v3.5.2 Latest
last month

+ 40 releases

Contributors 13

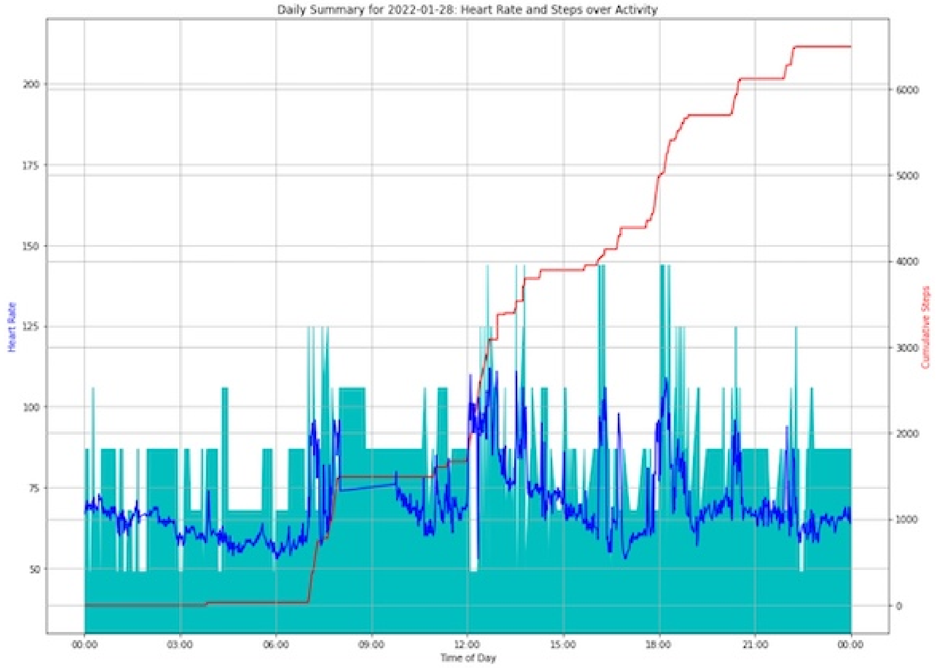
Languages

- ● **Python** 84.8%
- ● **Jupyter Notebook** 13.0%
- ● **Makefile** 2.2%

README.md

Summary of 2022-01-28

Weight	Resting HR	Max HR	Waking Avg RR	Steps	Floors	Intensity Mins	Calories	Sleep	REM Sleep	Stress
163.0	56.0	112.0	12.0	6490	28.1	00:00:00	2128	08:04:00	01:46:00	27



Sandwich Canal Walk

With Michele and Maeve.

Summary

Id	Sport	Type	Distance (miles)	Elapsed Time	Moving Time	Avg HR	Max HR	Avg RR	Max RR	Gained (feet)	Lost (feet)	Calories	Avg Temp (deg F)	Start Location	End Location
7749731018	walking	recreation	4.9	01:43:22.157000	01:43:22.157000	113	129			131.0	449.0	481	82.4	41.77340229973197, -70.49947930499911	41.77317707799375, -70.49924075603485

Laps

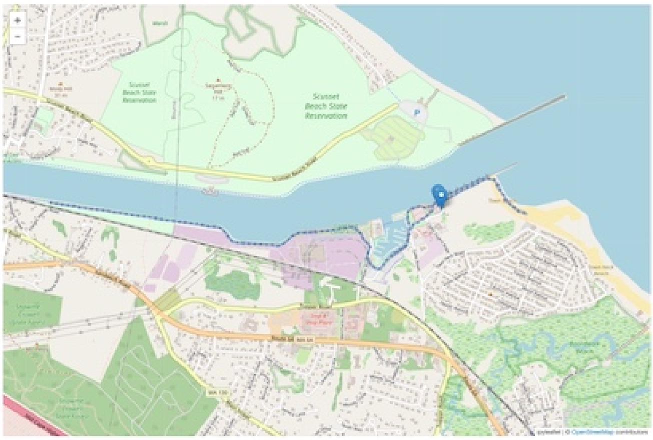
Lap	Distance (miles)	Elapsed Time	Moving Time	Avg HR	Max HR	Avg RR	Max RR	Gained (feet)	Lost (feet)	Calories	Avg Temp (deg F)	Start Location	End Location
0	1.0	00:21:31.009000	00:21:31.009000	110	128			39.0	164.0	111	82.4	41.77340229973197, -70.49947930499911	41.77154855802655, -70.51114565692842
1	1.0	00:19:01.873000	00:19:01.873000	115	129			36.0	108.0	90	82.4	41.77154310978949, -70.51115571521223	41.773504643769694, -70.52956966683269
2	1.0	00:18:52.059000	00:18:52.059000	117	129			7.0	56.0	90	80.6	41.773506151512265, -70.52959430962801	41.771040949970484, -70.51243286579847
3	1.0	00:19:09.725000	00:19:09.725000	115	128			3.0	69.0	87	80.6	41.77104304544628, -70.51241870038211	41.77395097911358, -70.49895359303322
4	0.9	00:24:47.391000	00:24:47.391000	110	128			43.0	46.0	103	82.4	41.773957181721926, -70.49894420430064	41.77317707799375, -70.49924075603485

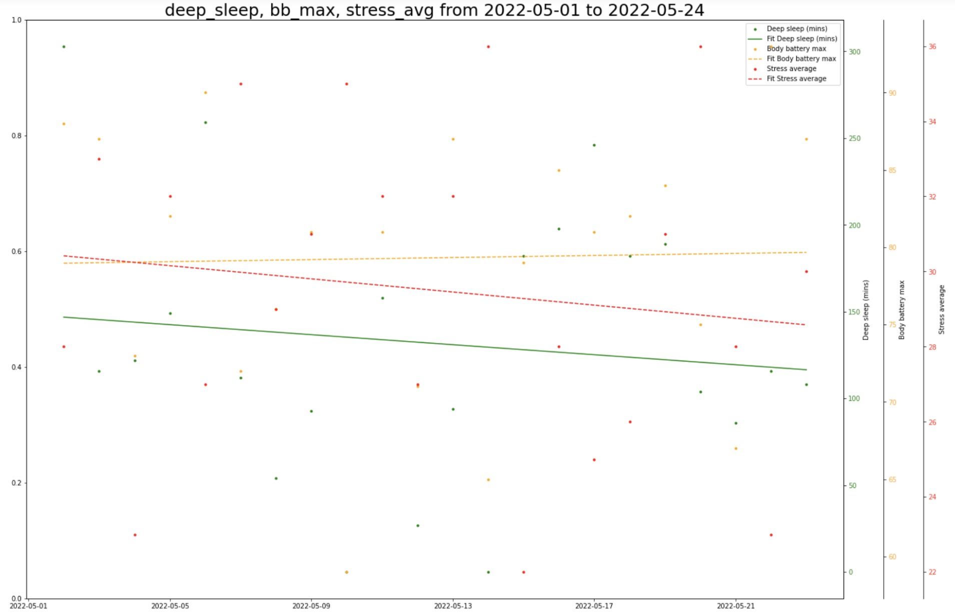
Time In Heart Rate Zones

Zone	HR Threshold	Time
1	112	00:59:36.476000
2	124	00:04:03.005000
3	135	00:00:00
4	146	00:00:00
5	158	00:00:00

Training Effect

Type	Rating
Aerobic	2.5
Anaerobic	0.0


+
−
Scusset Beach State Reservation
Scusset Beach State Reservation
Route 6A
Leaflet | © OpenStreetMap contributors



GarminDB

[Python](#) scripts for parsing health data into and manipulating data in a [SQLite](#) database. SQLite is a light weight database that doesn't require a server.

What they can do:

- Automatically download and import Garmin daily monitoring files (all day heart rate, activity, climb/descend, stress, and intensity minutes) from the user's Garmin Connect "Daily Summary" page.
- Extract sleep, weight, and resting heart rate data from Garmin Connect, store it as JSON files, and import it into the DB.
- Download and import activity files from Garmin Connect. A summary table for all activities and more detailed data for some activity types. Lap and record entries for activities.
- Summarizing data into a DB with tables containing daily, weekly, monthly, and yearly summaries.
- Graph your data from the commandline or with Jupyter notebooks.
- Retain downloaded JSON and FIT files so that the DB can be regenerated without connecting to or redownloading data from Garmin Connect.
- Export activities as TCX files.

Once you have your data in the DB, I recommend using a supplied Jupyter notebooks, third party Jupyter notebooks, and/or SQLite browser like SQLite Studio or DB Browser for SQLite for browsing and working with the data. The scripts create some default views in the DBs that make browsing the data easier.

Using It

Releases

GarminDb releases are hosted on PyPI. GarminDb requires Python 3.x. With Python installed, install the latest release with pip by running `pip install garmindb` in a terminal.

- Copy `GarminConnectConfig.json.example` to `~/.GarminDb/GarminConnectConfig.json`, edit it, and add your Garmin Connect username and password and adjust the start dates to match the dats of your data in Garmin Connect.
- Starting out: download all of your data and create your db by running `garmindb_cli.py --all --download --import --analyze` in a terminal.
- Incrementally update your db by downloading the latest data and importing it by running `garmindb_cli.py --all --download --import --analyze --latest` in a terminal.
- Ocassionally run `garmindb_cli.py --backup` to backup your DB files.

Update to the latest release with `pip install --upgrade garmindb`.

From Source

The scripts are automated with Make. Run the Make commands in a terminal window.

- Git clone GarminDB repo using the SSH clone method. The submodules require you to use SSH and not HTTPS. Get the command from the green button on the project home page.
- Run `make setup` in the cloned tree to get the scripts ready to process data.
- Copy `GarminConnectConfig.json.example` to `~/.GarminDb/GarminConnectConfig.json`, edit it, and add your Garmin Connect username and password and adjust the start dates to match the dats of your data in Garmin Connect.
- Run `make create_dbs` once to fetch and process for you data.
- Keep all of your local data up to date by periodically running only one command: `make`.

There is more help on using the program in the wiki.

Jupyter Notebooks

Jupyter notebooks for analzing data from the database can be found in the 'Jupyter' directory in the source tree. Links to user submitted notebooks can be found in the wiki.

Plugins

Plugins allow the user to expand the types of data that are processed and stored in the database. GarminDb already has a number of plugins for handling data from third-party Connect IQ apps and data fields. Read more about plugins here.

Success Stories

Find out who's using GarminDb on what platforms, OSes, and python versions here. If you're using GarminDB and your scenario isn't listed send me a message or file an issue with your success case.

Notes

- You may get a DB version exception after updating the code, this means that the DB schema was updated and you need to rebuild your DBs by running `garmindb_cli.py --rebuild_db` . Your DBs will be regenerated from the previously downloaded data files. All of your data will not be redownloaded from Garmin.
- The scripts were developed on MacOS. Information or patches on using these scripts on other platforms are welcome.
- When a database update finishes, a summary of the data in the DB will be saved to stats.txt. The output includes the date ranges included in the downloaded daily monitoring files and activities. It includes the number of records for daily monitoring, activities, sleep, resting heart rate, weight, etc. Use the summary information to determine if all of your data has been downloaded from Garmin Connect. If not, adjust the dates in GarminConnectConfig.json and runt he download again.
- In `GarminConnectConfig.json` the "steps" element of the "course_views" is list of course ids that per course database views will be generated for. The database view allows you to compare all activities from that course.

Bugs and Debugging

- If you have issues, file a bug here on the project. See the Issues tab at the top of the project page. Run `make bugreport` or `garmindb_bug_report.py` and include bugreport.txt in your bug report.
- Besides errors that appear on the screen, one of the first places to look for more information is the log files (garmindb.log).
- If your having issues with a particular data files, please considering sharing so I can debug it and add support.

Contributing

Please submit a pull request targeting the develop branch and add your self to the contributors file. Run `make flake8` at the top level and fix all errors before submitting your pull request.