

# Increasing Portability of Strength Training Programs

A close-up, low-angle shot of a person's legs and torso as they perform a deadlift with a barbell. The person is wearing dark shorts with orange stripes and grey athletic shoes. The barbell has large black weight plates. The background is a dark, textured floor.

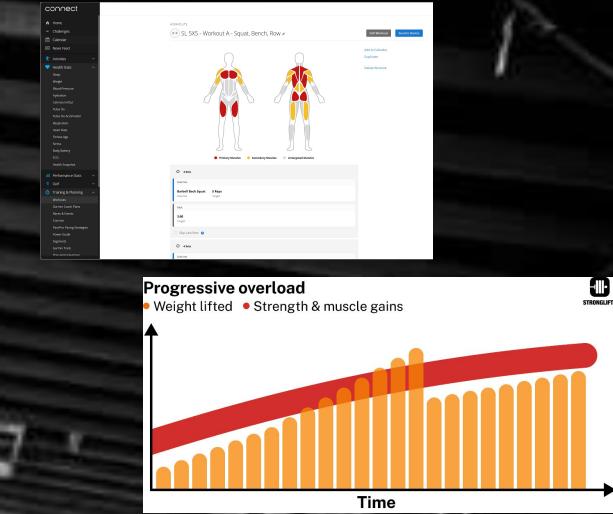
Christopher Phillips  
IMT 542



# Portable Information System:

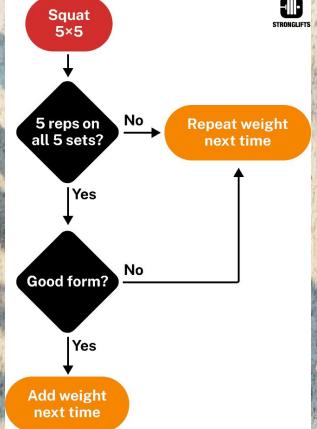
Goal: Create a modular, Python-powered system for tracking and analyzing weightlifting workouts, with a focus on the StrongLifts 5x5 program.

- Many systems do not include robust portability options for weightlifting, especially as they relate to a consistent weightlifting program
  - Moreover, confusing gym jargon and program names can make it seem intimidating
- The scripts and data model strictly follow the weightlifting scope:
- No cardio or other activity types.
- No user accounts or cloud integrations.
- No extraneous metadata fields (like HRV, sleep, or other biometrics).
- Everything implemented ties directly to the core requirement: weightlifting data capture

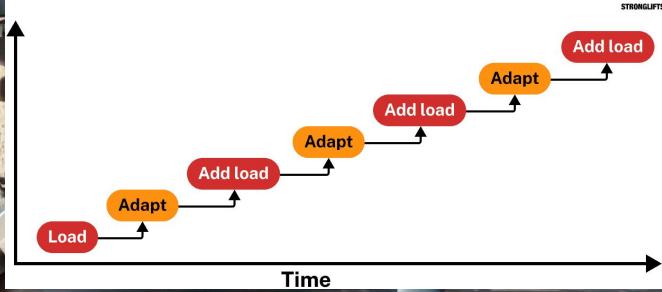


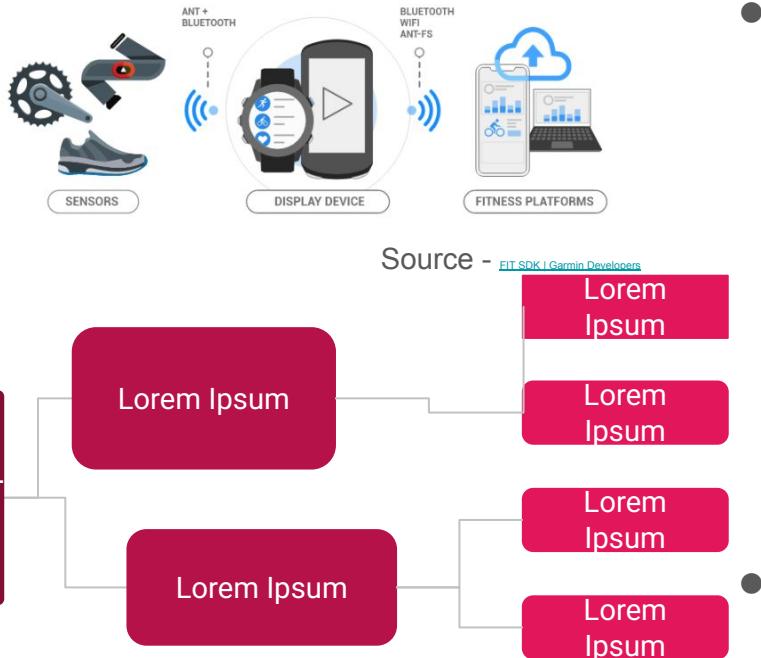
## Stronglifts 5×5: The Complete Guide to Building Strength and Muscle

Discover the ultimate resource for Stronglifts 5×5, the proven strength training program to help you build strength and muscle. Learn everything about how the program works to maximize your results and achieve your fitness goals.



### Progressive overload





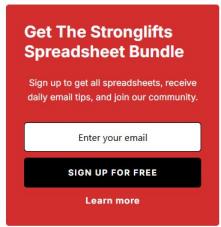
- Many fitness companies have their own device solution each of which records health/fitness data in their own portability solution

- Examples include: Garmin, Fitbit, Apple
- My current solution involves manually entering data on my hardware sensor while working out, resulting in poor data quality and lack of gym ergonomics.
- Lots of data handling/manual entry makes it burdensome for users to ensure clean data is uploaded.



Goal: Create a modular, Python-powered system for tracking and analyzing weightlifting workouts, with a focus on the StrongLifts 5x5 program.

Not being tied within a vendors workout program or the available tools allow you to customize your tool in useful ways.



## Stronglifts 5×5: The Complete Guide to Building Strength and Muscle



## Portable information structure

The system defines **JSON-based workout records** with clear, explicit field names (`exercise_name`, `sets`, `weight_kg`, `reps`, `successful`, etc.).

## Focus on weightlifting (StrongLifts 5x5)

The structure and scripts are **tightly aligned** with StrongLifts 5x5 principles:

- **Workout A / Workout B rotation**
- **5 sets of 5 reps** (except deadlift: 1 set of 5)
- **Gradual progression logic** (+2.5 kg or +1.25 kg)
- **Weight caps** consistent with realistic maximums
  - a. (primarily encountered when generating example data for visualization)





# Functional Exercise?



## Data Flow

Input: User logs workout in `generate_workout.py`

Output: Structured JSON file saved in `workouts/`

Analysis: `visualize_workouts.py` loads these JSON files, parses them into a pandas DataFrame, and creates visual plots.

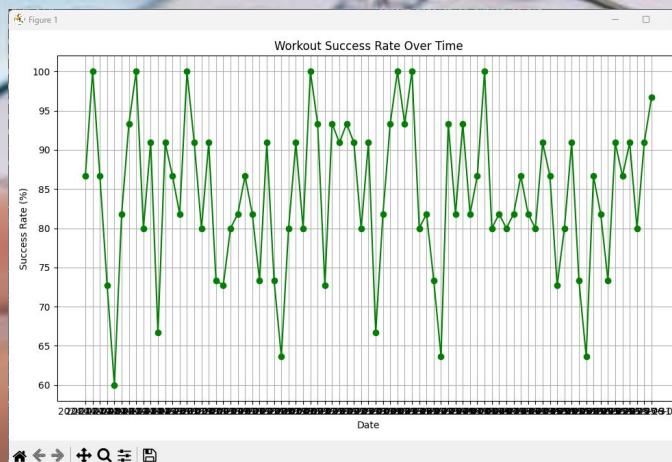
```
Enter workout type to record (A or B): A
Was set 1 of Squat at 55.0 kg successful? (y/n): y
Was set 2 of Squat at 55.0 kg successful? (y/n): y
Was set 3 of Squat at 55.0 kg successful? (y/n): y
Was set 4 of Squat at 55.0 kg successful? (y/n): y
Was set 5 of Squat at 55.0 kg successful? (y/n): n
Was set 1 of Bench Press at 42.5 kg successful? (y/n): y
Was set 2 of Bench Press at 42.5 kg successful? (y/n): y
Was set 3 of Bench Press at 42.5 kg successful? (y/n): y
Was set 4 of Bench Press at 42.5 kg successful? (y/n): y
Was set 5 of Bench Press at 42.5 kg successful? (y/n): y
Was set 1 of Barbell Row at 41.25 kg successful? (y/n): y
Was set 2 of Barbell Row at 41.25 kg successful? (y/n): y
Was set 3 of Barbell Row at 41.25 kg successful? (y/n): y
Was set 4 of Barbell Row at 41.25 kg successful? (y/n): y
Was set 5 of Barbell Row at 41.25 kg successful? (y/n): y
How did you feel? (e.g., strong, tired): Better at python, but tired.
Where did you train? (e.g., Home Gym): Office.
```

Workout 2025-06-03-A summary:

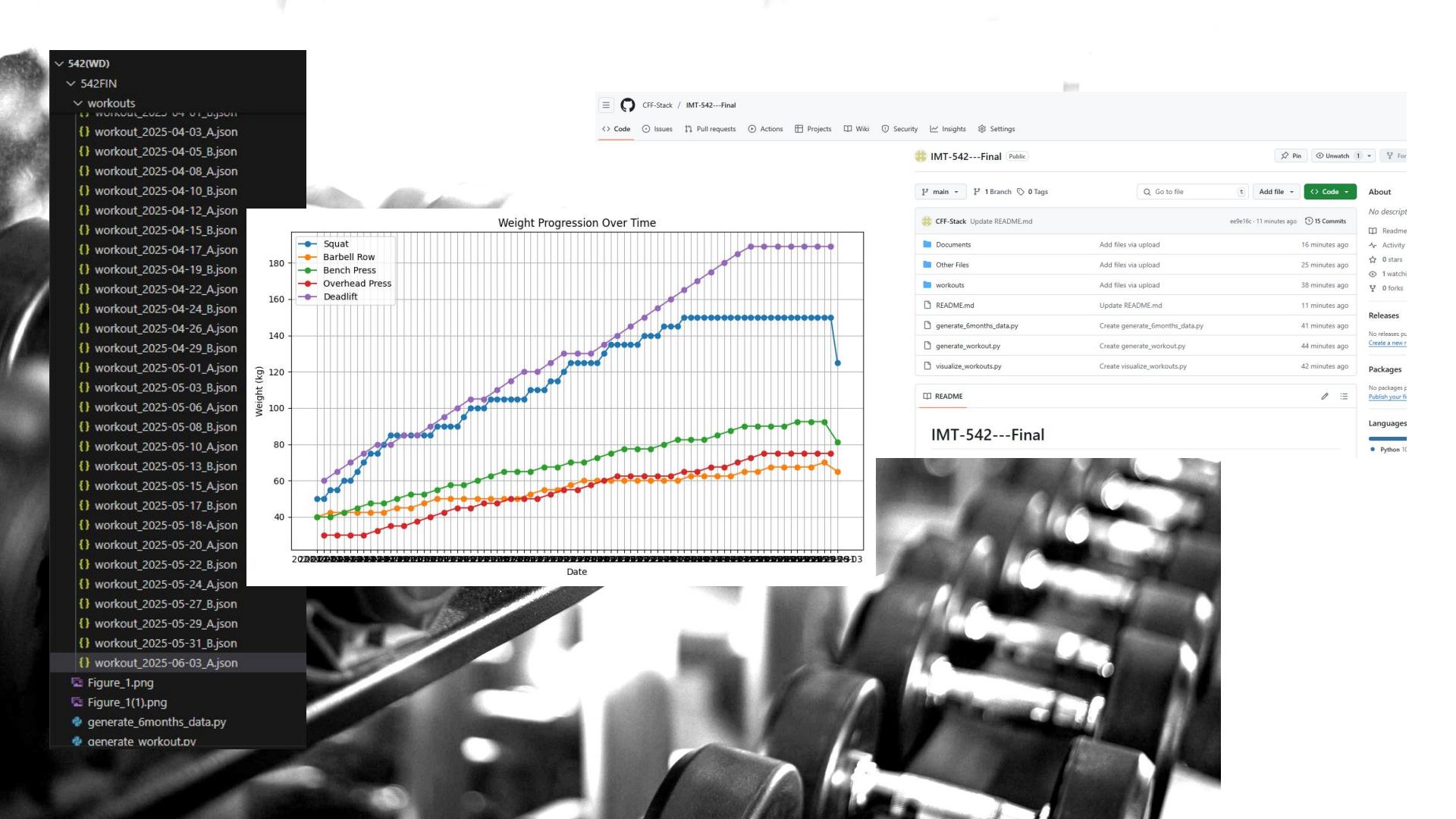
- Squat: 4/5 sets successful
- Bench Press: 5/5 sets successful
- Barbell Row: 5/5 sets successful

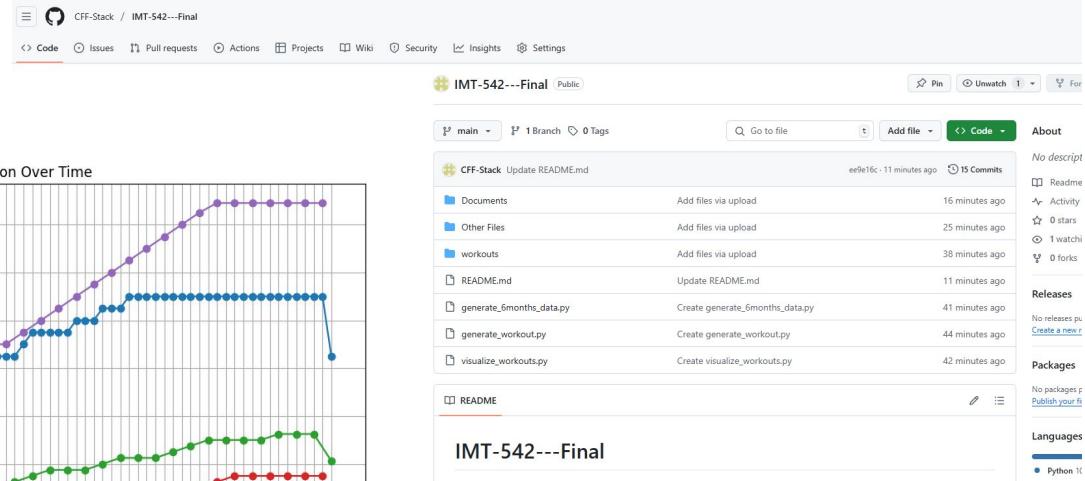
Workout record saved to: `workouts\workout_2025-06-03_A.json`

PS C:\Users\cphil\OneDrive\UW\SPR 24\INT 542\542(WD)\542FIN>



```
542FIN > workouts > {} workout_2025-06-03_A.json > ...
1
2   "workout_id": "2025-06-03-A",
3   "date": "2025-06-03",
4   "program": "StrongLifts 5x5",
5   "workout_type": "A",
6   "exercises": [
7     {
8       "exercise_name": "Squat",
9       "sets": [
10         {
11           "set_number": 1,
12           "weight_kg": 150,
13           "reps": 5,
14           "successful": true
15         },
16         {
17           "set_number": 2,
18           "weight_kg": 150,
19           "reps": 5,
20           "successful": true
21         },
22         {
23           "set_number": 3,
24           "weight_kg": 150,
25           "reps": 5,
26           "successful": true
27         },
28         {
29           "set_number": 4,
30           "weight_kg": 150,
31           "reps": 5,
32           "successful": true
33         },
34         {
35           "set_number": 5,
36           "weight_kg": 150,
37           "reps": 5,
38           "successful": true
39         }
40     ],
41     "notes": ""
42   },
43   {
44     "exercise_name": "Bench Press",
45     "sets": [
46       {
47         "set_number": 1,
48         "weight_kg": 92.5,
49         "reps": 5,
50         "successful": true
51       },
52       {
53         "set_number": 2,
54         "weight_kg": 92.5,
55         "reps": 5,
56         "successful": true
57       },
58       {
59         "set_number": 3,
60         "weight_kg": 92.5,
61         "reps": 5,
62         "successful": true
63       }
64     ]
65   }
66 ]
```

✓ 542(WD)  
✓ 542FIN  
✓ workouts  
↳ workout\_2025-04-03\_A.json  
↳ workout\_2025-04-05\_B.json  
↳ workout\_2025-04-08\_A.json  
↳ workout\_2025-04-10\_B.json  
↳ workout\_2025-04-12\_A.json  
↳ workout\_2025-04-15\_B.json  
↳ workout\_2025-04-17\_A.json  
↳ workout\_2025-04-19\_B.json  
↳ workout\_2025-04-22\_A.json  
↳ workout\_2025-04-24\_B.json  
↳ workout\_2025-04-26\_A.json  
↳ workout\_2025-04-29\_B.json  
↳ workout\_2025-05-01\_A.json  
↳ workout\_2025-05-03\_B.json  
↳ workout\_2025-05-06\_A.json  
↳ workout\_2025-05-08\_B.json  
↳ workout\_2025-05-10\_A.json  
↳ workout\_2025-05-13\_B.json  
↳ workout\_2025-05-15\_A.json  
↳ workout\_2025-05-17\_B.json  
↳ workout\_2025-05-18\_A.json  
↳ workout\_2025-05-20\_A.json  
↳ workout\_2025-05-22\_B.json  
↳ workout\_2025-05-24\_A.json  
↳ workout\_2025-05-27\_B.json  
↳ workout\_2025-05-29\_A.json  
↳ workout\_2025-05-31\_B.json  
↳ workout\_2025-06-03\_A.json  
  
Figure\_1.png  
Figure\_1(1).png  
generate\_6months\_data.py  
generate workout.py



# References

1. [Download the FIT SDK | Garmin Developers](#)
2. [FIT SDK | Garmin Developers](#)
3. [Strength Training & Muscle Building | Stronglifts: Strength Training and Muscle Building](#)
4. [CFF-Stack/IMT-542---Final](#)

```
3     notes :  
4   }  
5   ],  
6   "metadata": {  
7     "duration_min": null,  
8     "user_feeling": "Better at python, but tired.",  
9     "location": "Office. "  
10    }  
1 }
```

