

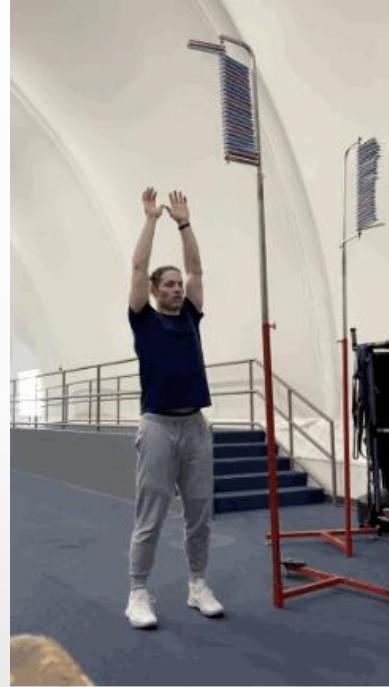
CSCI 622 – Project Recap

Athlete Performance Dashboard

Brock Gion

Project Overview – Git Repo README

- 2 Primary Goals**
 - Jump Higher
 - Target 33"
 - Current 27.5"
 - Run Faster 100M Sprint
 - Target 11.80 seconds
 - Current 12.25 seconds
- How to measure improvements with data?**
 - What data is most useful to me?
- Answer questions such as...**
 - How's my sleep?
 - Am I training at the right intensity?
 - What's my completed workout history?



brocg add in kaggle README b4b2dea last month

SupplementaryInfo Delete SupplementaryInfo/IngestionAnalysis/PutYourDataSetAnalysisCo...

src/ingestion add in kaggle README

.gitignore fitbit ingestion files. use data_lake_utils to separate out ADLS uplo...

README.md minor tweak to databricks script (limit output to 20 rows), update RE...

fitbit trackers.PNG update README with datasources, explain more in-depth the purpose/...

README.md

[Open in Visual Studio Code](#)

CSCI 622 Project - Brock

Athlete Performance Dashboard

- The Athlete Performance Dashboard aims to help individuals improve their athletic performance by making driven decisions, staying motivated, and maintaining a holistic approach to training and well-being.

2 Primary Athletic Goals

1. Jump Higher, measuring standing vertical leap (using a [vertec](#))
 - Target: 33"
 - Current: 27.5"
2. Run Faster, measuring sprint speed in 100M dash (using [handheld stopwatch](#) + [SprintTimer App](#))
 - Target: 11.80 seconds
 - Current: 12.35 seconds (as of August 2023)

These 2 performance metrics will be measured once a month and manually logged.

How to measure improvements with data?

Improving vertical jump height requires a combination of strength training, plyometric exercises, technique refinement, and consistent monitoring. Here's a list of possible types of data to collect:

Athlete Performance Dashboard

https://brocg.github.io/athlete-performance-dashboard/

Training Session Date

All

Training Session Calendar

name ● COD & Lowe... ● Primary Lower ● Run, Sprint tr... ● COD Mastery ● Morning wor... ● Lower Body ...

< > Today November 2023 Month Week Day List

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|------------------|-----|-----|-------------------|-----|-------------------|
| 29 | 30 COD & L... | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 Run, Spi... |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 COD Mas... | 21 | 22 | 23 | 24 | 25 Run, Spi... |
| 26 | 27 | 28 | 29 | 30 Morning ... | 1 | 2 Run, Spi... |
| 3 | 4 Lower Bo... | 5 | 6 | 7 | 8 | 9 |

Training Session Calendar

PolarH10 Heart Rate Data

training_session_date

Saturday, November 11, 2023

{'average': 131, 'maximum': 175}

Total

Fitbit SleepLog Data

total_hours_sleep..

9.3

Heart Rate Training Zones (1-5)

Average Heart Rate for Each Training Zone

| Training Zone | Average Heart Rate (bpm) | Max Heart Rate (bpm) |
|---------------|--------------------------|----------------------|
| Zone 1 | 114 bpm | 133 bpm |
| Zone 2 | 133 bpm | 175 bpm |
| Zone 3 | 155 bpm | |
| Zone 4 | 171 bpm | |
| Zone 5 | 190 bpm | |

Sleep Efficiency 98

Sleeping Efficiency

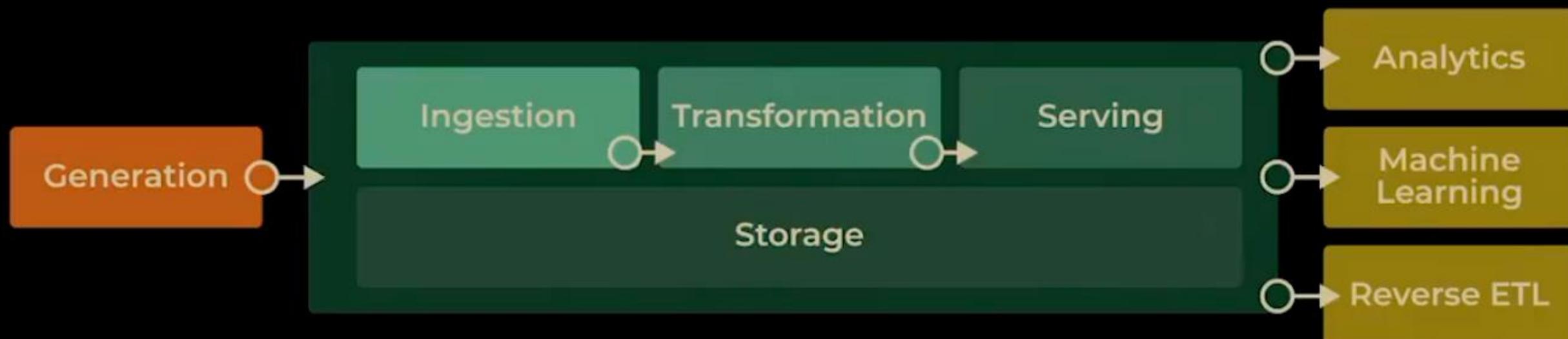
Percentage of Time

Sleep

16% (1.5 hrs)

84% (7.8 hrs)

DATA ENGINEERING LIFECYCLE



UNDERCURRENTS

Security

Data Management

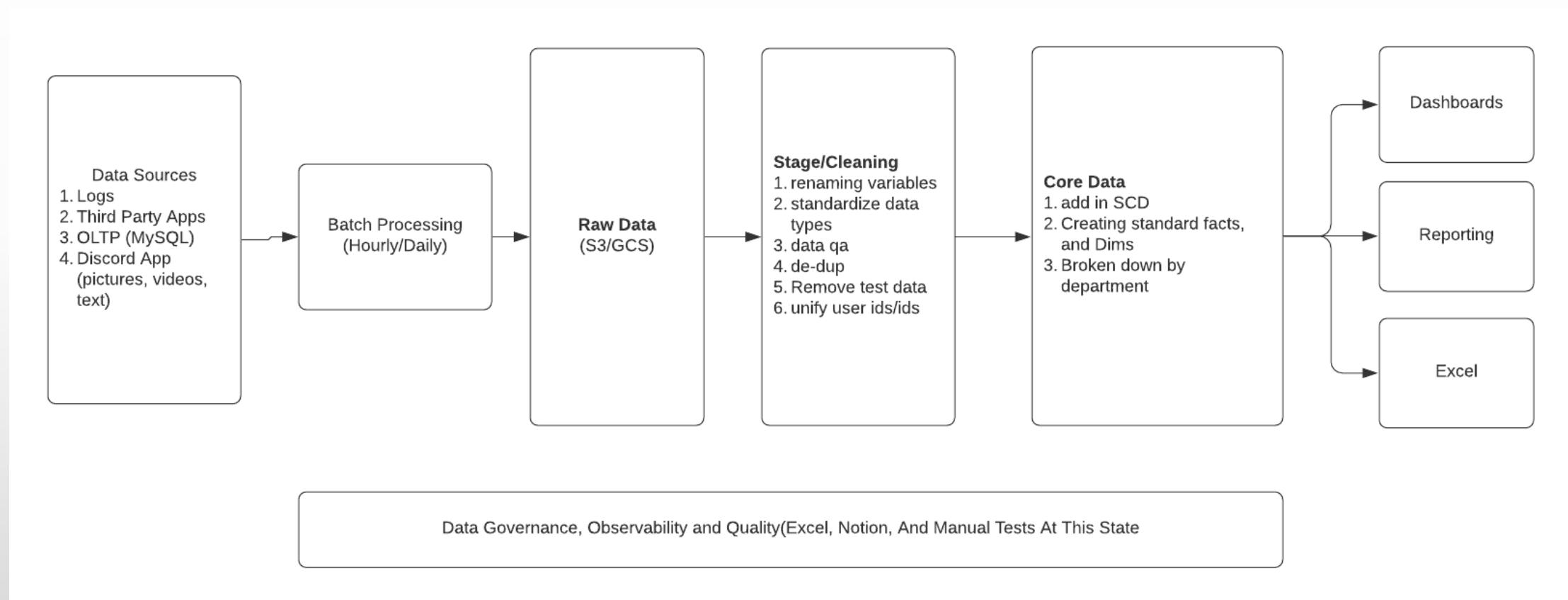
DataOps

Data Architecture

Orchestration

Software Engineering

Building Your First Baseline Data Stack



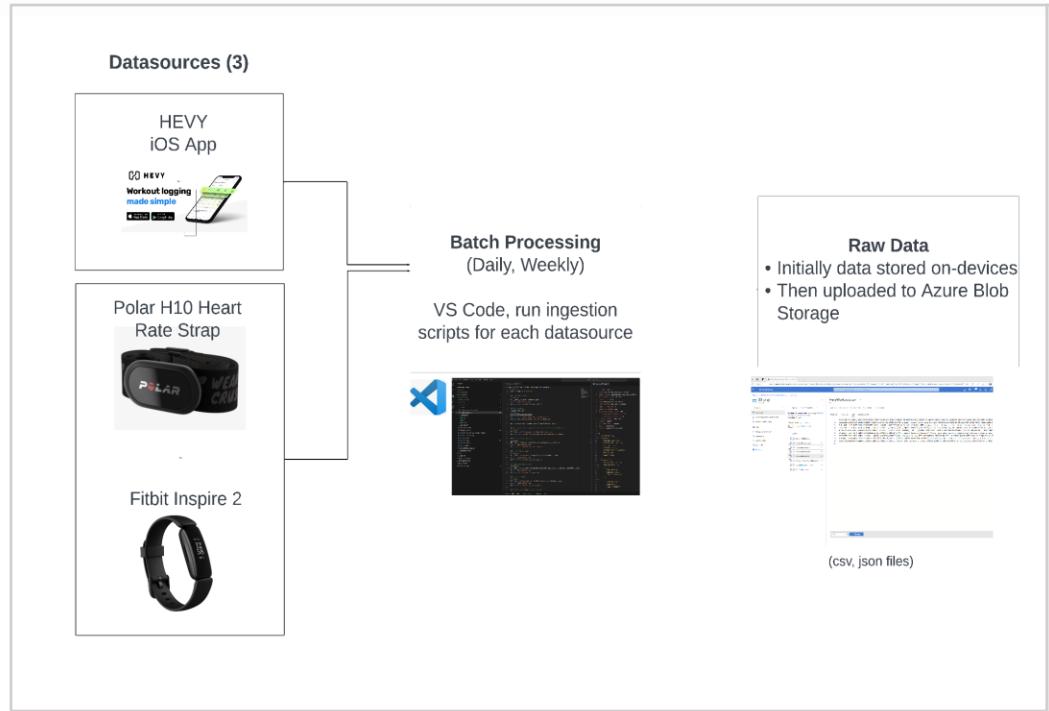
<https://seattledataguy.substack.com/p/the-baseline-datastack-going-beyond>

Architecture Overview

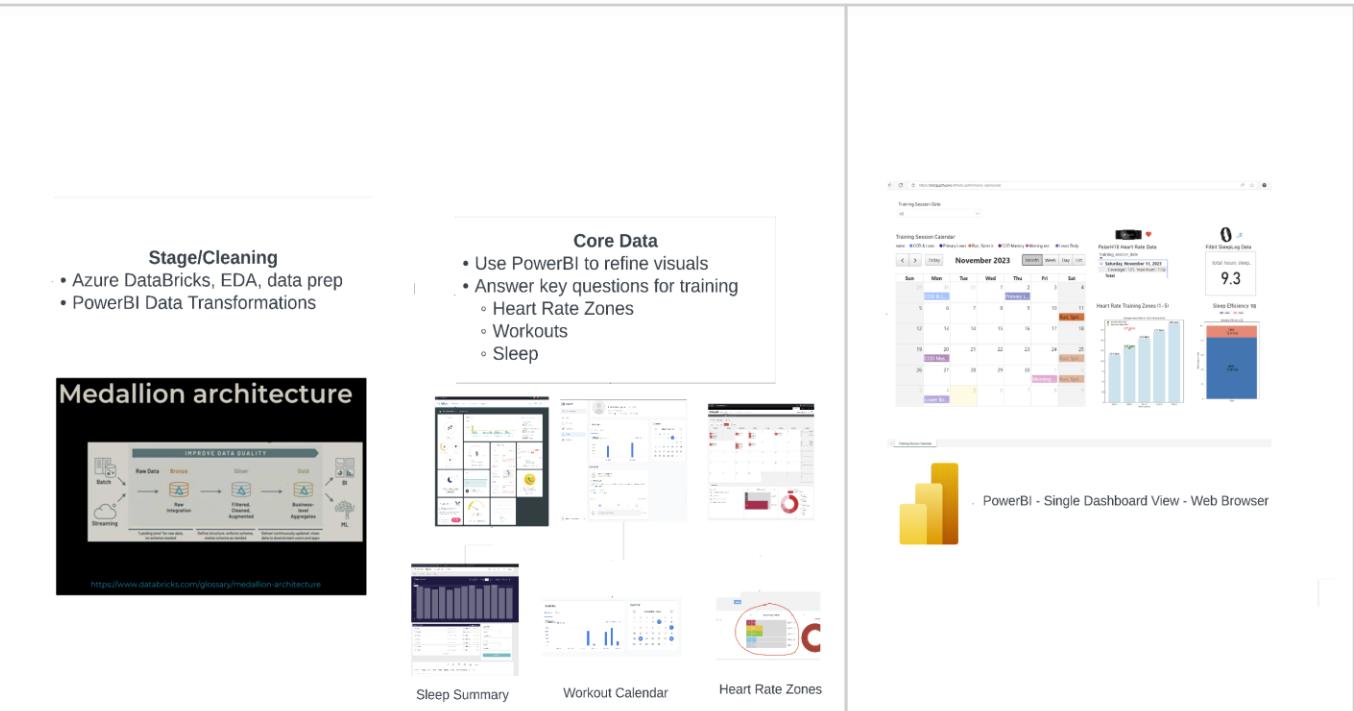
High level diagram

Data Engineering Workflow

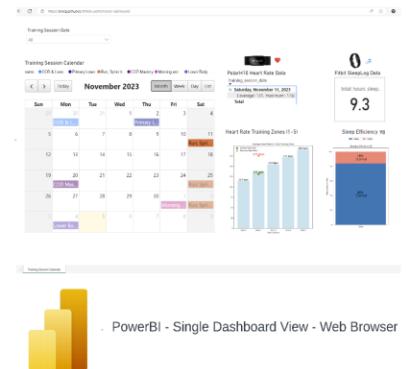
INGESTION



TRANSFORMATION



SERVING



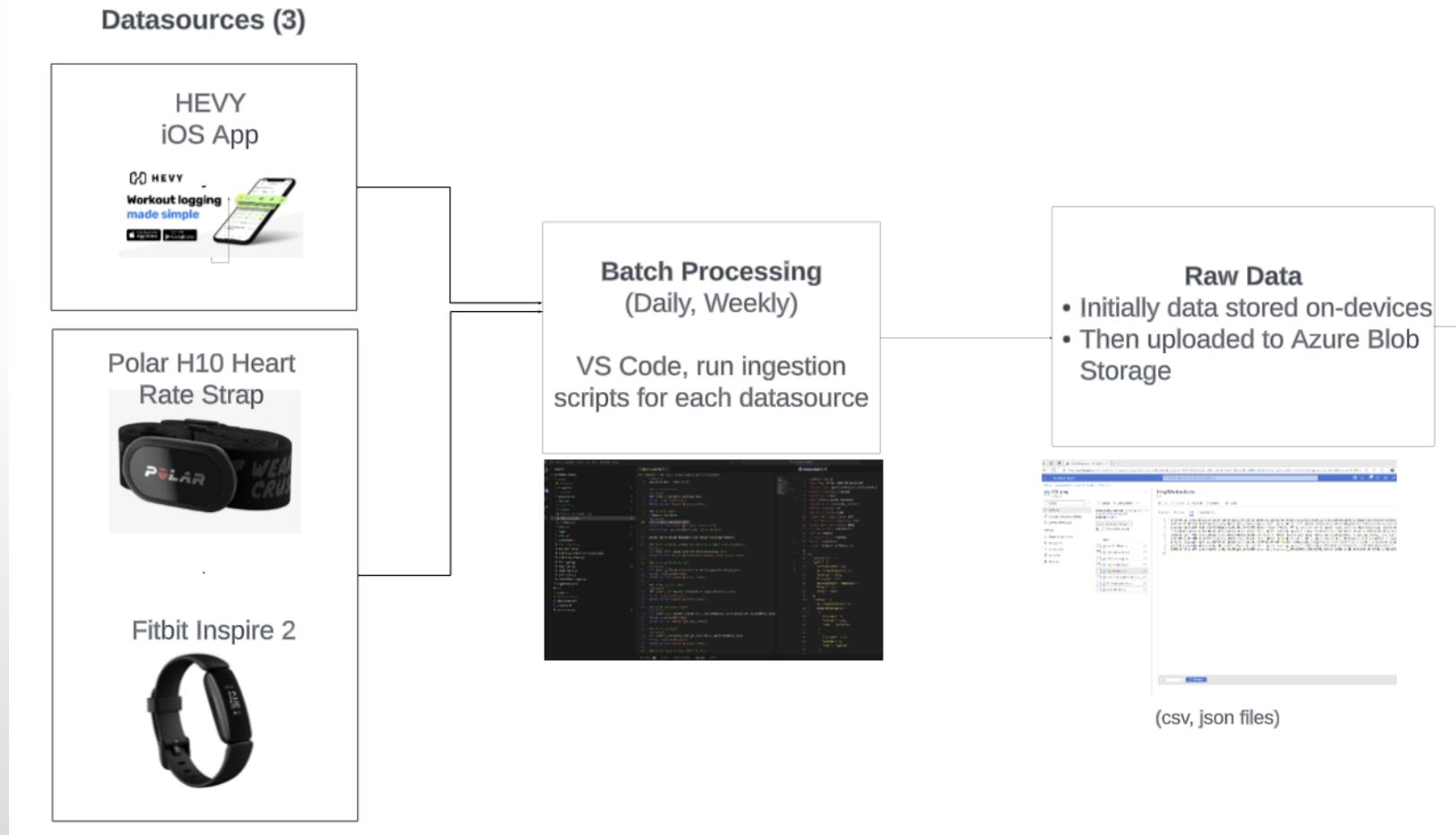
Raw Data

Dashboard

<https://lucid.app/documents/view/f57e39f2-8b43-483e-8e28-5c7cfa2553d8>

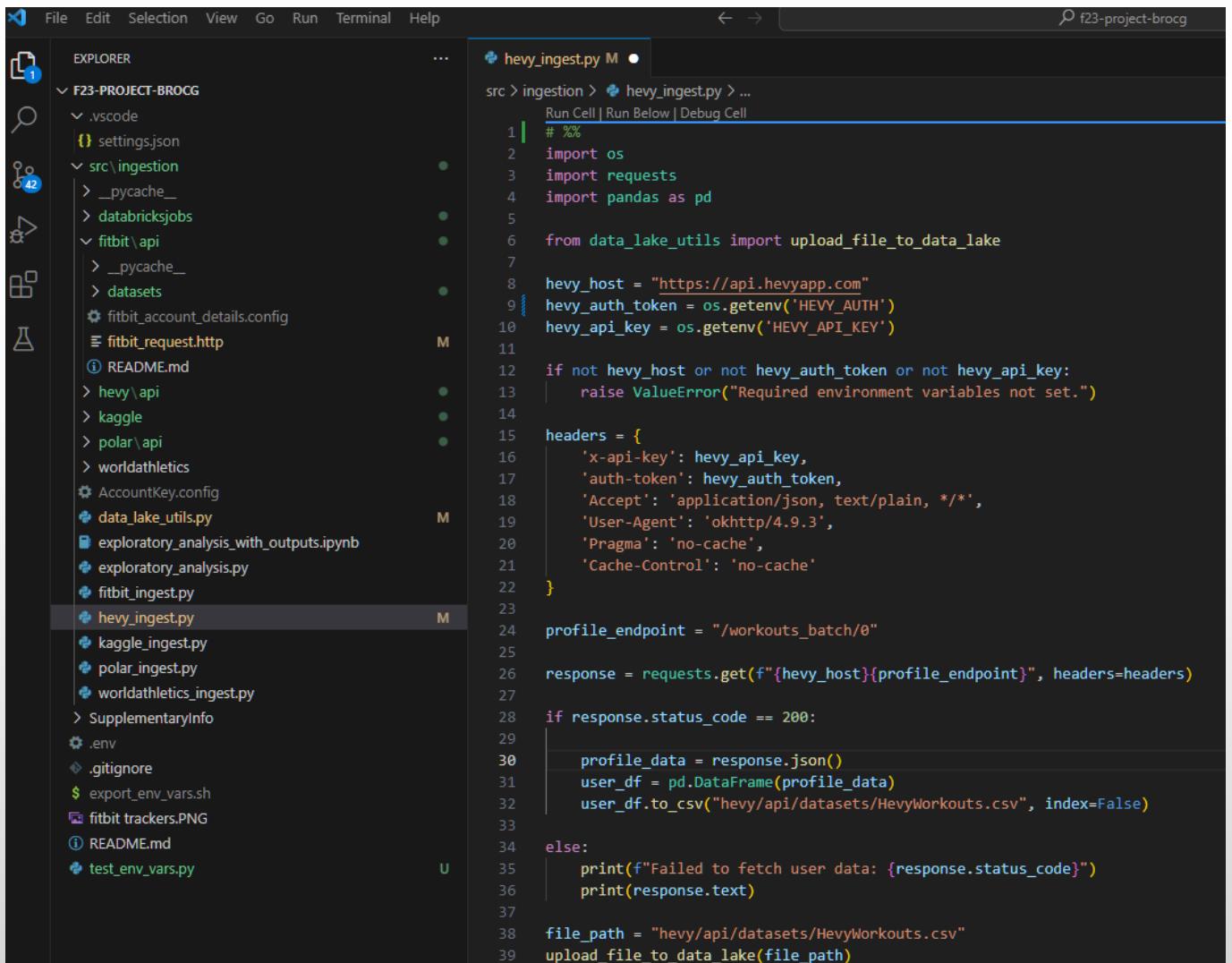
Datasources

- **Hevy**
 - Workout Data
 - **`workouts_batch/0`**
 - Provides response containing all workouts with total volume lifted
 - Saved as *HevyWorkouts.csv*
- **Polar H10**
 - Heart Rate Data
 - **`v3/exercises`**
 - Provides response containing all heart rate data (avg, max) and type of exercise
 - Saved as *PolarExercises.csv*
- **Fitbit Inspire**
 - Sleep Log Data
 - **`1.2/user/-/sleep/list.json`**
 - Provides response containing all heart rate data (avg, max) and type of exercise
 - Saved as *FitbitSleepLog.json*



Batch Processing

- Ingestion files (one per datasource)**
 - hevy_ingest.py
 - polar_ingest.py
 - fitbit_ingest.py
- Helper function file**
 - data_lake_utils.py
- Each ingestion file follows similar process**
 - Authenticate w/ API endpoint
 - Capture response, check for 200
 - Save response as csv, upload to data lake
- Common pain points & pitfalls?**
 - Manually refreshing tokens
 - Fitbit = every 8 hours
 - Unknown responses
 - Hevy, unofficial API, use as-is, so have to debug yourself (status 401)



The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows the project structure under "F23-PROJECT-BROCG". It includes a ".vscode" folder, a "src\ingestion" folder containing "hevy_ingest.py", and other subfolders like ".pycache", "databricksjobs", "fitbit\api", "hevy\api", "kaggle", "polar\api", "worldathletics", and "SupplementaryInfo". There are also ".env", ".gitignore", and "README.md" files.
- Code Editor:** The "hevy_ingest.py" file is open. The code uses Python to interact with the Hevy API. It imports os, requests, and pandas, and uses data_lake_utils to upload files. It defines headers for the API calls and handles a 200 status code by parsing the JSON response and saving it to a CSV. It also handles a 401 status code by printing an error message and the response text.

```

File Edit Selection View Go Run Terminal Help
EXPLORER
F23-PROJECT-BROCG
  .vscode
    settings.json
  src\ingestion
    __pycache__
    databricksjobs
    fitbit\api
      __pycache__
      datasets
      fitbit_account_details.config
      fitbit_request.http
      README.md
      hevy\api
      kaggle
      polar\api
      worldathletics
      AccountKey.config
      data_lake_utils.py
      exploratory_analysis_with_outputs.ipynb
      exploratory_analysis.py
      fitbit_ingest.py
      hevy_ingest.py
      kaggle_ingest.py
      polar_ingest.py
      worldathletics_ingest.py
      SupplementaryInfo
      .env
      .gitignore
      export_env_vars.sh
      fitbit_trackers.PNG
      README.md
      test_env_vars.py
hevy_ingest.py M ...
src > ingestion > hevy_ingest.py > ...
Run Cell | Run Below | Debug Cell
1 # %%
2 import os
3 import requests
4 import pandas as pd
5
6 from data_lake_utils import upload_file_to_data_lake
7
8 hevy_host = "https://api.hevyapp.com"
9 hevy_auth_token = os.getenv('HEVY_AUTH')
10 hevy_api_key = os.getenv('HEVY_API_KEY')
11
12 if not hevy_host or not hevy_auth_token or not hevy_api_key:
13     raise ValueError("Required environment variables not set.")
14
15 headers = {
16     'x-api-key': hevy_api_key,
17     'auth-token': hevy_auth_token,
18     'Accept': 'application/json, text/plain, */*',
19     'User-Agent': 'okhttp/4.9.3',
20     'Pragma': 'no-cache',
21     'Cache-Control': 'no-cache'
22 }
23
24 profile_endpoint = "/workouts_batch/0"
25
26 response = requests.get(f"{hevy_host}{profile_endpoint}", headers=headers)
27
28 if response.status_code == 200:
29
30     profile_data = response.json()
31     user_df = pd.DataFrame(profile_data)
32     user_df.to_csv("hevy/api/datasets/HevyWorkouts.csv", index=False)
33
34 else:
35     print(f"Failed to fetch user data: {response.status_code}")
36     print(response.text)
37
38 file_path = "hevy/api/datasets/HevyWorkouts.csv"
39 upload_file_to_data_lake(file_path)

```

Raw Data

HevyWorkouts.csv ...

Blob

Save Discard Download Refresh Delete

Overview Versions Edit Generate SAS

```

1 id,short_id,index,name,description,start_time,end_time,created_at,updated_at,routine_id,apple_watch,user_id,username,profile_image,verified,nth_workout,like_count,is_liked_by_user,is_private,like_images,comments,comment_count,media,image
2 c02ee31b-e638-4ba-f-262910dff221,MdpC7ruF1Z,30426659,COD & Lower Focus,"RPE 4, isos hardest",1698717635,1698719486,2023-10-31T02:31:14.152Z,2023-10-31T02:32:24.874Z,f58cb5b0-d6b9-4d7c-9ba-f-d5b560ee9df,True,c36b0a00-49d2-414a-9c78
3 4b1cd2b-b2a9-d3d8-9ab-3eb7a00544d,2vdpxVjfcA,30716596,Primary Lower,"Solid, RPE 4, only one set of squats feels weird on right hamstring, worked thru it no issues",1698869885,1698972149,2023-11-03T00:41:31.826Z,2023-12-04T06:30:34.4
4 3772598-DE16-424E-A52D-408F842F8B6C,q29du07ceER,31549723,"Run, Sprint training Session",3 sets 75M,1699748342,1699748787,2023-11-12T00:26:28.105Z,be2204be-105f-4f05-96e0-40e8719116d,True,c36b0a00-49d2-414a-9c7
5 e5cf9297-1b8f-4423-87ea-1bb010ffdad,w5610khUE,32398459,COD Mastery,"RPE 5 (paired with bball warmup and stairclimber), no iso leg drives, late night workout",1700540795,1700542150,2023-11-21T04:49:08.017Z,2023-12-04T06:30:22.031Z,f58
6 e4a99eaf-d8af-46ff-a958-f97cc98812ce,COVYH817Vv,32870142,"Run, Sprint training Session","RPE 5, good interval run, followed up with iso drives in squat rack",1700958207,1700960717,2023-11-26T01:05:09.168Z,2023-11-30T19:18:16.771Z,be220
7 49727763-7744-4489-b405-4131efdcce0,dQLNvmC1Q5,33485045,Morning workout 🌞,H1IT,1701428295,1701431107,2023-12-01T12:59:56.357Z,2023-12-04T05:08:12.431Z,False,c36b0a00-49d2-414a-9c7
8 e9194403-c2a4-44a1-902c-bc6c4a5c0f5,1hINT1dJw81,33533739,"Run, Sprint training Session",1701557787,1701560018,2023-12-02T23:33:39.418Z,be2204be-105f-4f05-96e0-40e8719116d,True,c36b0a00-49d2-414a-9c70-e549f9c
9 029954d2-19ab-4074-aabe-82b3b9efca88,159r60AgLN,33610873,Lower Body Secondary,,1701683668,1701684872,2023-12-04T10:14:33.051Z,2023-12-04T10:14:33.051Z,a33ed5b0-184b-4f8d-8d7b-cd411f203f67,False,c36b0a00-49d2-414a-9c70-e549f9c3b46,heal

```

PolarExercises.csv ...

Blob

Save Discard Download Refresh Delete

Overview Versions Edit Generate SAS

```

1 id,upload_time,polar_user,device,start_time,start_time_utc_offset,duration,heart_rate,sport,has_route,detailed_sport_info,training_load_pro,distance,calories
2 yMkd06X,2023-10-24T08:22:26Z,https://www.polaraccesslink.com/v3/users/60755152,Polar Flow app,2023-10-24 3:22:05,-300,PT10.51S,{},OTHER,FALSE,YOGA,"{'cardio-load': 0.0, 'cardio-load-interpretation': 'NOT_AVAILABLE', 'muscle-load': -1.0, 'muscle-load-interpretation': 'NOT_AVAILABLE'}"
3 0rpz408o,2023-10-24T08:26:38Z,https://www.polaraccesslink.com/v3/users/60755152,Polar Flow app,2023-10-24 3:26:18,-300,PT12.80S,{},OTHER,TRUE,PARASPORTS_WHEELCHAIR,"{'cardio-load': 0.0, 'cardio-load-interpretation': 'NOT_AVAILABLE', 'muscle-load': -1.0, 'muscle-load-interpretation': 'NOT_AVAILABLE'}"
4 y8uro660,2023-10-24T19:45:38Z,https://www.polaraccesslink.com/v3/users/60755152,Polar BEAT,2023-10-24 14:32:46,-300,PT757S,{},{"average": 82, "maximum": 117},OTHER,TRUE,WALKING,,671,39
5 y78vx7w0,2023-10-25T19:06:39Z,https://www.polaraccesslink.com/v3/users/60755152,Polar BEAT,2023-10-25 11:53:31,-300,PT454S,{},{"average": 89, "maximum": 126},OTHER,TRUE,WALKING,,478,28
6 yNkdxlV0,2023-10-25T23:32:58Z,https://www.polaraccesslink.com/v3/users/60755152,Polar Flow app,2023-10-25 17:38:36,-300,PT3732.154S,{},{"average": 73, "maximum": 122},OTHER,FALSE,YOGA,"{'cardio-load': 8.0185, 'cardio-load-interpretation': 'NOT_AVAILABLE'}"
7 0rpzqEEa,2023-10-29T23:12:26Z,https://www.polaraccesslink.com/v3/users/60755152,Polar BEAT,2023-10-29 18:00:17,-300,PT714S,{},{"average": 63, "maximum": 76},OTHER,TRUE,WALKING,,4842,18
8 Pkv4EJvL,2023-10-29T23:32:49Z,https://www.polaraccesslink.com/v3/users/60755152,Polar Flow app,2023-10-29 18:15:03,-300,PT1057.897S,{},{"average": 67, "maximum": 88},OTHER,TRUE,WALKING,"{'cardio-load': 0.0415559, 'cardio-load-interpretation': 'NOT_AVAILABLE'}"
9 y3GjhjQw,2023-10-31T01:47:36Z,https://www.polaraccesslink.com/v3/users/60755152,Polar Flow app,2023-10-30 20:38:55,-300,PT995.753S,{}, {"average": 112, "maximum": 160},CYCLING,TRUE,CYCLING,"{'cardio-load': 11.8723, 'cardio-load-interpretation': 'NOT_AVAILABLE'}"
10 PogXGjmW,2023-10-31T02:50:42Z,https://www.polaraccesslink.com/v3/users/60755152,Polar BEAT,2023-10-30 21:03:37,-300,PT1779S,{}, {"average": 132, "maximum": 169},OTHER,FALSE,MOBILITY_DYNAMIC,,,333

```

FitbitSleepLog.json ...

Blob

Save Discard Download Refresh Delete

Overview Versions Edit Generate SAS

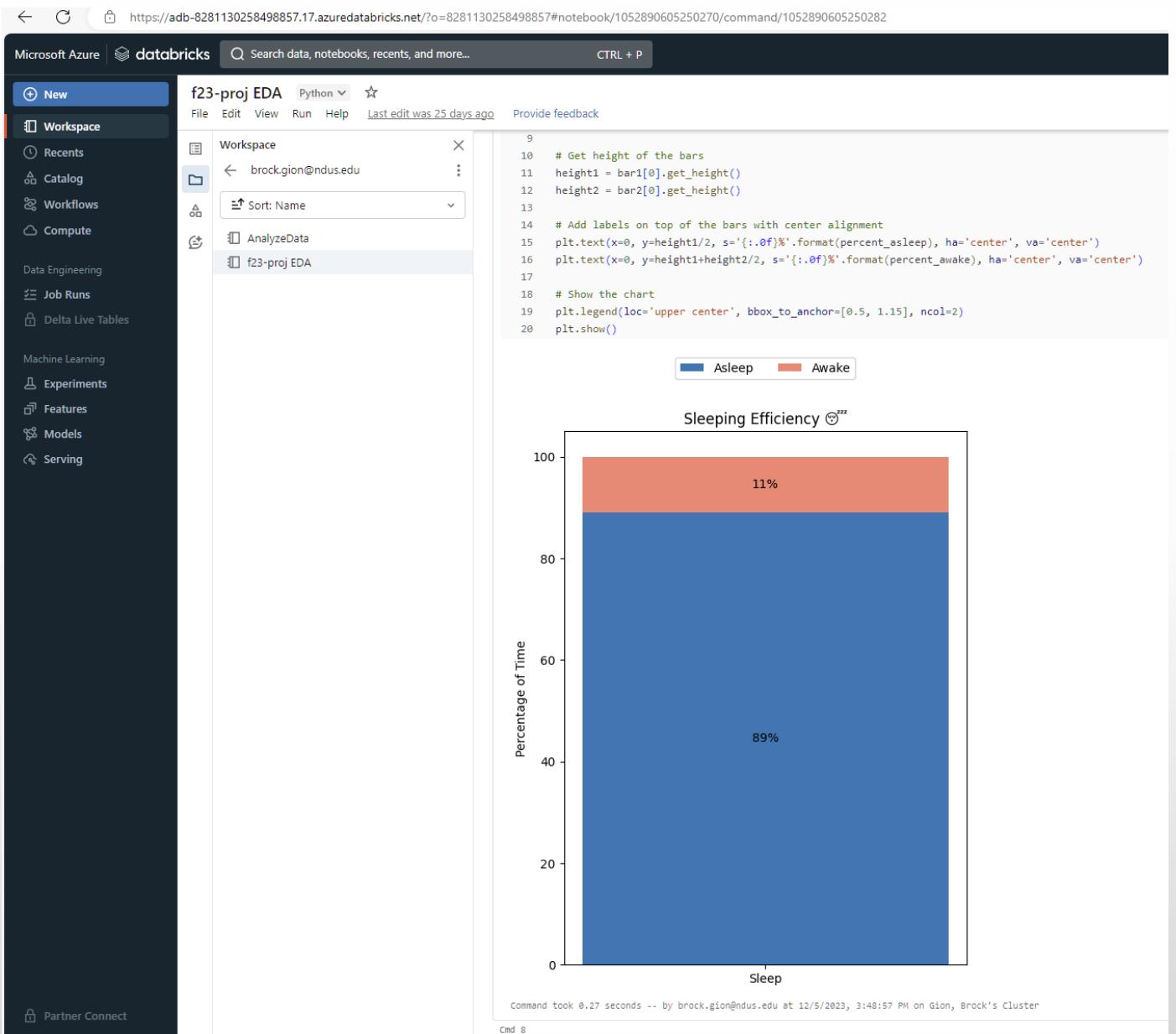
```

1 [{"pagination":{"afterDate":"2023-10-01","limit":90,"next":"","offset":0,"previous":"","sort":"desc"}, "sleep":[{"dateOfSleep":"2023-12-03","duration":30540000,"efficiency":98,"endTime":"2023-12-03T13:51:00.000","infoCode":0,"isMainSleep":true,"levels":{},"data":[{"dateTime": "2023-12-03T05:22:00.000","level": "wake","seconds":160}, {"dateTime": "2023-12-03T05:23:00.000","level": "light","seconds":1050}, {"dateTime": "2023-12-03T05:24:00.000","level": "deep","seconds":1590}, {"dateTime": "2023-12-03T06:03:00.000","level": "light","seconds":1260}, {"dateTime": "2023-12-03T06:13:00.000","level": "deep","seconds":1290}, {"dateTime": "2023-12-03T07:05:30.000","level": "deep","seconds":1200}, {"dateTime": "2023-12-03T07:27:00.000","level": "light","seconds":1470}, {"dateTime": "2023-12-03T07:51:30.000","level": "rem","seconds":360}, {"dateTime": "2023-12-03T07:57:30.000","level": "light","seconds":1420}, {"dateTime": "2023-12-03T08:04:30.000","level": "rem","seconds":1470}, {"dateTime": "2023-12-03T08:29:00.000","level": "light","seconds":360}, {"dateTime": "2023-12-03T08:35:00.000","level": "wake","seconds":1110}, {"dateTime": "2023-12-03T08:53:30.000","level": "light","seconds":13600}, {"dateTime": "2023-12-03T09:53:30.000","level": "wake","seconds":210}, {"dateTime": "2023-12-03T09:57:00.000","level": "light","seconds":1260}, {"dateTime": "2023-12-03T10:18:00.000","level": "rem","seconds":570}, {"dateTime": "2023-12-03T10:27:30.000","level": "light","seconds":2520}, {"dateTime": "2023-12-03T11:09:30.000","level": "deep","seconds":300}, {"dateTime": "2023-12-03T11:26:00.000","level": "wake","seconds":300}, {"dateTime": "2023-12-03T11:32:00.000","level": "light","seconds":480}, {"dateTime": "2023-12-03T11:40:00.000","level": "deep","seconds":330}, {"dateTime": "2023-12-03T11:45:30.000","level": "light","seconds":1260}, {"dateTime": "2023-12-03T12:06:30.000","level": "wake","seconds":510}, {"dateTime": "2023-12-03T12:15:00.000","level": "light","seconds":300}, {"dateTime": "2023-12-03T12:21:00.000","level": "wake","seconds":360}, {"dateTime": "2023-12-03T12:27:00.000","level": "light","seconds":210}, {"dateTime": "2023-12-03T12:30:30.000","level": "wake","seconds":330}, {"dateTime": "2023-12-03T12:36:00.000","level": "light","seconds":120}, {"dateTime": "2023-12-03T12:38:00.000","level": "rem","seconds":1140}, {"dateTime": "2023-12-03T12:57:00.000","level": "light","seconds":120}], "date": "2023-12-03T13:51:00.000"}]

```

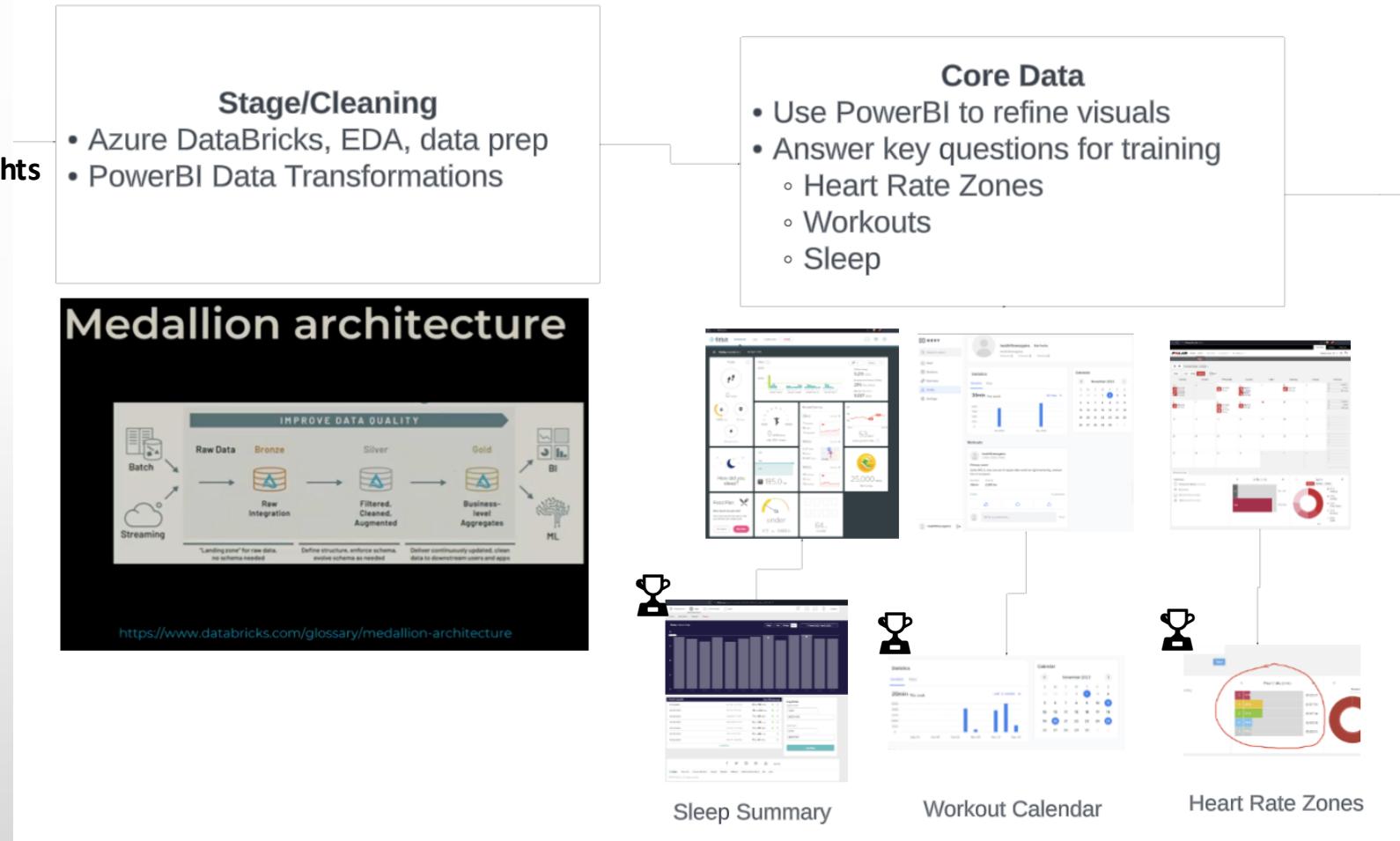
Stage Cleaning

- Azure Databricks**
 - Verifying structure/shape of datasets
 - Initial Exploratory Data Analysis (EDA)
- Automated “Job Runs” triggered when CSV files get uploaded**
 - Experimented with using a trigger in Azure Data Factory
 - Really useful (not sure on cost?), saves me the effort of manually spinning up a cluster
- CI/CD Integration**
 - Explored the idea of having the output of Databricks push to a linked repo
 - Conclusion? Not as simple as it sounds. Hit limits within Azure Data Factory, better suited for Azure Dev Ops.
 - Concern over \$\$ cost, prefer to keep free (or under \$5/month)



Stage Cleaning

- **Medallion architecture**
 - Bronze = Raw
 - Silver = Filtered, Cleaned
 - **Gold = Business-level, valuable insights**
- **3 Core Visuals (Gold-level)**
 - Sleep Summary
 - Fitbit Dashboard
 - Workout Calendar
 - Hevy Dashboard
 - Heart Rate Zones
 - Polar Dashboard



TRANSFORMATION

Stage Cleaning

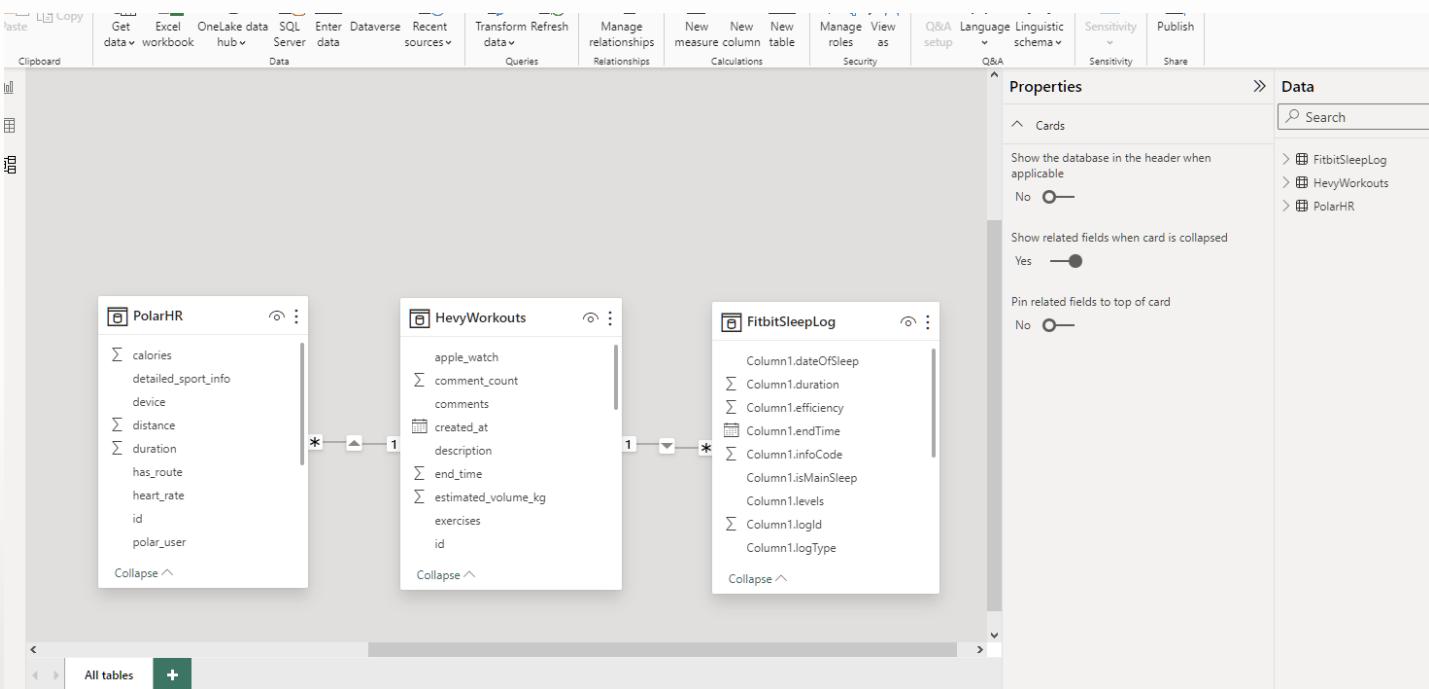
- **Using PowerBI for data transformations**
 - Example – calculating “total_hours_sleep” from Fitbit data
 - taking JSON file, converted to Table, ran Python script to get total hours rounded to nearest tenth
- **Challenges**
 - PowerBI GUI can be hard to navigate sometimes
 - Found myself gravitating toward running python scripts to transform data vs. using PowerBI functions
 - In the end, used a hybrid approach
- **Improvements**
 - Understand how to form the right relationships between the datasets
 - Key point – everything revolves around using dates
 - Created “training_session_date” as field to join the tables on

The screenshot shows the Microsoft Power BI desktop application. The main workspace displays a table with 17 columns and 51 rows. The table includes columns for minutesToFallAsleep, startTime, timeInBed, type, and total_hours_sleep. The 'total_hours_sleep' column contains numerical values rounded to one decimal place. The 'APPLIED STEPS' pane on the right details the data processing workflow, starting from the source and including steps like 'Imported JSON', 'Converted to Table', 'Run Python script', and 'Changed Type'.

| | minutesToFallAsleep | startTime | timeInBed | type | total_hours_sleep |
|----|---------------------|-----------|------------------------|-------------|-------------------|
| 1 | 62 | 0 | 12/3/2023 5:22:00 AM | 509 stages | 8.5 |
| 2 | 123 | 0 | 12/1/2023 11:09:00 PM | 773 stages | 12.9 |
| 3 | 57 | 0 | 11/30/2023 11:35:30 PM | 291 stages | 4.8 |
| 4 | 90 | 0 | 11/29/2023 10:38:00 PM | 493 stages | 8.2 |
| 5 | 89 | 0 | 11/28/2023 9:21:30 PM | 635 stages | 10.6 |
| 6 | 67 | 0 | 11/23/2023 12:42:00 AM | 451 stages | 7.5 |
| 7 | 70 | 0 | 11/20/2023 3:33:30 AM | 442 stages | 7.4 |
| 8 | 84 | 0 | 11/19/2023 2:54:00 AM | 547 stages | 9.1 |
| 9 | 126 | 0 | 11/17/2023 10:51:00 PM | 702 stages | 11.7 |
| 10 | 62 | 0 | 11/17/2023 1:36:30 AM | 396 stages | 6.6 |
| 11 | 52 | 0 | 11/16/2023 1:31:00 AM | 460 stages | 7.7 |
| 12 | 67 | 0 | 11/14/2023 11:28:30 PM | 515 stages | 8.6 |
| 13 | 53 | 0 | 11/13/2023 10:55:30 PM | 563 stages | 9.4 |
| 14 | 50 | 0 | 11/12/2023 11:59:00 PM | 370 stages | 6.2 |
| 15 | 103 | 0 | 11/11/2023 11:07:00 PM | 709 stages | 11.8 |
| 16 | 90 | 0 | 11/11/2023 2:30:30 AM | 556 stages | 9.3 |
| 17 | 64 | 0 | 11/10/2023 2:00:30 AM | 371 stages | 6.2 |
| 18 | 61 | 0 | 11/9/2023 2:10:30 AM | 422 stages | 7 |
| 19 | 52 | 0 | 11/8/2023 12:55:30 AM | 397 stages | 6.6 |
| 20 | 3 | 6 | 11/7/2023 2:00:00 AM | 371 classic | 6.2 |
| 21 | 44 | 0 | 11/6/2023 2:26:30 AM | 348 stages | 5.8 |
| 22 | 59 | 0 | 11/3/2023 10:39:00 PM | 540 stages | 9 |
| 23 | 66 | 0 | 11/3/2023 12:13:00 AM | 477 stages | 8 |
| 24 | 33 | 0 | 11/2/2023 1:02:00 AM | 416 stages | 6.9 |
| 25 | 71 | 0 | 10/31/2023 11:47:30 PM | 489 stages | 8.2 |
| 26 | 56 | 0 | 10/31/2023 3:15:30 AM | 375 stages | 6.2 |
| 27 | 34 | 0 | 10/30/2023 2:58:30 AM | 306 stages | 5.1 |
| 28 | 85 | 0 | 10/29/2023 4:35:00 AM | 436 stages | 7.3 |
| 29 | 88 | 0 | 10/28/2023 12:28:30 AM | 651 stages | 10.8 |
| 30 | 59 | 0 | 10/27/2023 2:26:00 AM | 354 stages | 5.9 |
| 31 | 79 | 0 | 10/26/2023 12:30:00 AM | 507 stages | 8.4 |
| 32 | 38 | 0 | 10/25/2023 5:19:30 AM | 315 stages | 5.2 |
| 33 | 79 | 0 | 10/24/2023 3:53:00 AM | 501 stages | 8.4 |
| 34 | 0 | 2 | 10/23/2023 10:15:00 AM | 20 classic | 0.3 |
| 35 | 43 | 0 | 10/23/2023 3:43:00 AM | 307 stages | 5.1 |
| 36 | 50 | 0 | 10/17/2023 1:32:00 AM | 394 stages | 6.6 |
| 37 | 50 | 0 | 10/16/2023 12:54:30 AM | 454 stages | 7.6 |
| 38 | 92 | 0 | 10/15/2023 3:30:00 AM | 572 stages | 9.5 |
| 39 | 62 | 0 | 10/10/2023 11:56:00 PM | 455 stages | 7.6 |
| 40 | 77 | 0 | 10/10/2023 12:49:00 AM | 449 stages | 7.4 |

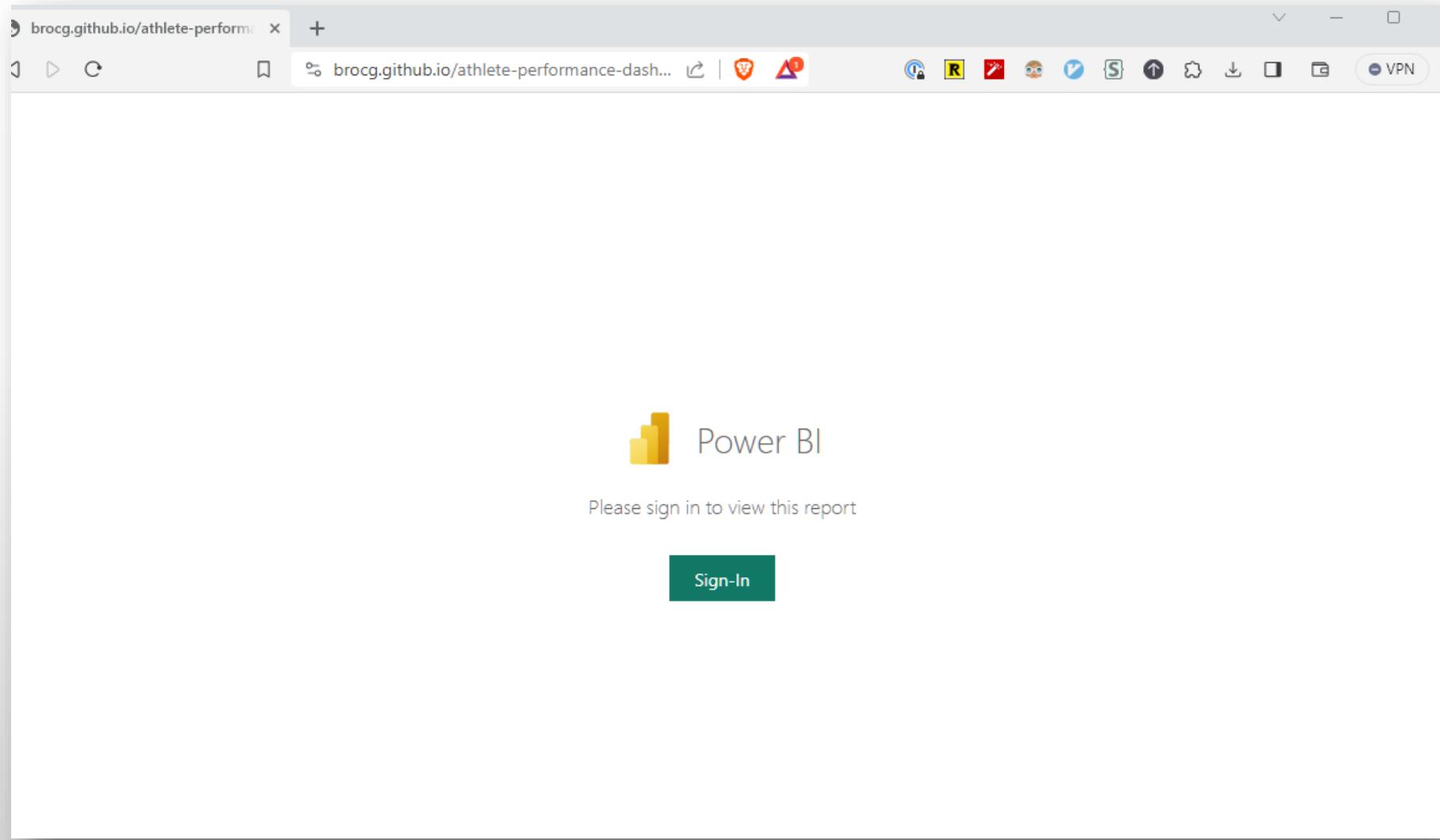
Stage Cleaning

- Data model – 3 tables**
 - PolarHR
 - HevyWorkouts
 - FitbuitSleepLog
- Each table has one thing in common – datetime of training session**
 - Used this to join on so can use the full benefit of PowerBI when filtering/slicing data (everything stays in sync)
- Ideas for improvement, explore more relationships and clean invalid data**
 - For instance, if I'm wearing a Fitbit and heart rate strap, does the data line up?
 - Have more robust ways to clear out blanks, invalid values, etc.



SERVING

PowerBI



SERVING

PowerBI - Components Under the hood

The screenshot displays the Microsoft PowerBI desktop application interface. On the left, a sidebar shows the 'AppSource' section for 'Calendar by MAQ Software'. It includes a preview image, a 3.7 (64) rating, a 'PBI Certified' badge, and sections for 'Overview', 'Ratings + reviews', 'Business Uses', 'Key Features', and 'What's New in 4.0.5.0'. The main workspace shows a calendar view for November 2023 with highlighted days. To the right, there are two visualizations: a bar chart titled 'Heart Rate Training Zones (1-5)' showing average heart rates (114, 133, 155, 171, 190 bpm) for zones 1 through 5, and a stacked area chart titled 'Sleep Efficiency' showing sleep stages across the night. A Python script editor window is open at the bottom, displaying code to create a DataFrame and remove duplicates, followed by a script area with imports and data definition. The ribbon menu at the top includes options like 'Format painter', 'Get data', 'Excel', 'OneLake data hub', 'SQL Server', 'Enter Data', 'Dataverse', 'Recent sources', 'Transform Refresh data', 'New visual', 'Text box', 'More visuals', 'New measure', 'Quick measure', 'Insert', 'Sensitivity', 'Publish', 'Sensitivity', and 'Share'.

SERVING

Athlete Performance Dashboard

https://brocg.github.io/athlete-performance-dashboard/

Training Session Date

All

Training Session Calendar

name ● COD & Lowe... ● Primary Lower ● Run, Sprint tr... ● COD Mastery ● Morning wor... ● Lower Body ...

< > Today November 2023 Month Week Day List

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-------------|-----|--------------|-----|-----|-------------|-----|
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| COD & L... | | Primary L... | | | | |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| | | Run, Spi... | | | | |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| COD Mas... | | | | | Run, Spi... | |
| 26 | 27 | 28 | 29 | 30 | 1 | 2 |
| | | Morning ... | | | Run, Spi... | |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Lower Bo... | | | | | | |

PolarH10 Heart Rate Data

training_session_date

Saturday, November 11, 2023

{'average': 131, 'maximum': 175}

Total

Fitbit SleepLog Data

total_hours_sleep..

9.3

Heart Rate Training Zones (1-5)

Average Heart Rate for Each Training Zone

| Training Zone | Average Heart Rate | Maximum Heart Rate |
|---------------|--------------------|--------------------|
| Zone 1 | 114 bpm | 175 bpm |
| Zone 2 | 137 bpm | 175 bpm |
| Zone 3 | 155 bpm | |
| Zone 4 | 171 bpm | |
| Zone 5 | 190 bpm | |

Sleep Efficiency 98

Sleeping Efficiency

Sleep

Percentage of time

84% (7.8 hrs)

16% (1.5 hrs)

Final Demo – Browsing calendar to see stats

<https://brocg.github.io/athlete-performance-dashboard/>

The dashboard features a central calendar for December 2023. Training sessions are color-coded: purple for COD & Lower Body, blue for Primary Lower, orange for Run, Sprint, and Tr., pink for COD Mastery, and green for Morning work. Specific sessions like 'Morning ...' and 'Run, Spr...' are highlighted. To the right, three cards provide detailed data: 'PolarH10 Heart Rate Data' shows heart rate zones with a Polar H10 icon; 'Fitbit SleepLog Data' shows sleep duration with a Fitbit icon; and a chart titled 'Sleep Efficiency' shows the percentage of time asleep versus awake.

| Training Session | Date | Category |
|------------------|------------|---------------------------|
| Morning ... | 2023-10-30 | Pink (Morning work) |
| Run, Spr... | 2023-10-30 | Orange (Run, Sprint, Tr.) |
| Lower Bo... | 2023-10-04 | Purple (Lower Body) |

PolarH10 Heart Rate Data

- Tuesday, October 24, 2023
- Wednesday, October 25, 2023
- Sunday, October 29, 2023
- Monday, October 30, 2023
- Wednesday, November 01, 2023

Fitbit SleepLog Data

total_hours_sleep.. **10.2**

Heart Rate Training Zones (1-5)

Average Heart Rate for Each Training Zone

| Training Zone | Avg Heart Rate (bpm) |
|---------------|----------------------|
| Zone 1 | 114 bpm |
| Zone 2 | 133 bpm |
| Zone 3 | 155 bpm |
| Zone 4 | 171 bpm |
| Zone 5 | 190 bpm |

Sleep Efficiency

Percentage of Time

| State | Percentage | Hours |
|--------|------------|-----------|
| Asleep | 80% | (0.0 hrs) |
| Awake | 20% | (0.3 hrs) |