



DOCUMENTATION
Data Procurement, Cleansing and Organisation

Purpose	1
Scope	2
Wahoo and Garmin fitness activity data procurement	2
What is / Who are Garmin and Wahoo?	2
File output - FIT files	2
Where was the data procured from?	3
Accessing the data	3
Other information collected	3
Privacy	3
FIT File Conversion, Data cleansing and Organisation	4
FIT Conversion	4
Data Cleansing and Organisation	4
Key Operations (Wahoo)	5
Key Operations (Garmin)	13
Access to BigQuery	13
Outcome	13

Purpose

Redback Operation core product will collect performance (output) data from a number of sensors while the user is using the bike and wahoo kickr trainer. Moreover, establishing a working environment that will align with the core product is critical in establishing continuity in data-related tasks and advising other core teams about Data Science and Analytics (DSA) Team's requirements. The initial effort is to set up a core data location for the team that house all key datasets.

By the end of this task, further work on data analysis, visualisation and using a number of machine learning algorithms will be possible.

Scope

Initialise a BigQuery (Google) environment that houses all of the DSA's data; three key datasets have been identified:

1. Sales device data
2. Oxygen (O2) Uptake related data
3. Wahoo/Garmin workout data.

Sales and O2 datasets were located within the DSA team's GitHub repository, however, the Garmin and Wahoo data was procured, cleansed and organised independently. This document outlines the specific details relating to independent data procurement.

Wahoo and Garmin fitness activity data procurement

What is / Who are Garmin and Wahoo?

Wahoo

Founded in 2009 by Chip Hawkins in Atlanta, GA, Wahoo creates innovative solutions to make hard-fought goals attainable and lives better. Wahoo was built on the foundation of simplicity and the mindset that "there's got to be a better way. Moreover, Wahoo produces a suite of devices that record/collect performance-related data from Athletes.

Key devices involved in the data procurement process:

1. Wahoo Bolt
2. Wahoo Element

Garmin

Garmin makes products that are engineered on the inside for life on the outside. We do this so our customers can make the most of the time they spend pursuing their passions. Garmin brings GPS navigation and wearable technology to the automotive, aviation, marine, outdoor and fitness markets.

Key devices involved in the data procurement process:

1. Edge 1030
2. Edge 1040

File output - FIT files

The Flexible and Interoperable Data Transfer (FIT) protocol is designed specifically for the storing and sharing of data that originates from sport, fitness and health devices. The FIT protocol defines a set of data storage templates (FIT messages) that can be used to store

information such as activity data, courses, and workouts. It is designed to be compact, interoperable and extensible. Extensive documentation is available [here](#) regarding FIT files.

The FIT file protocol was designed to provide:

- Interoperability of device data across various platforms
- Scalability from small embedded devices to cloud platforms
- Forward compatibility, allowing the protocol to grow and retain existing functionality
- Automated compatibility across platforms of different native endianness

Referenced from [Garmin](#)

FIT files are not immediately in a format that a human can read such that a conversion to a Comma-separated values file (CSV) is required. This document outlines the steps required to conduct a CSV conversion.

Where was the data procured from?

I provided over 400 FIT files to the project and also procured data from our Cyclists in Melbourne by way of convenience sampling. Therefore, the activity files can't be used for population sampling activity; demographics are extremely narrow i.e., Male, 30-60 years etc.

Accessing the data

Computer used to pull FIT file: Macbook Pro (iOS)

Garmin Devices: A USB-c cable was used to access the files on the device. The device is immediately accessible in the device listing; files are then transferred (copied) off the device.

Wahoo: To access a Wahoo device, [Android File Transfer](#) is required. The device also needs to be turned ON. Thereafter, the device is immediately accessible in the device listing; files are then transferred (copied) off the device. More information is available [here](#) about connecting a Wahoo device to a computer.

Other information collected

In addition to acquiring FIT files from a participant, their age, biological gender (male/female), weight and functional threshold power (FTP) were also recorded.

Privacy

FIT files contain GPS data and device data such as serial numbers; these data points have been considered sensitive and have been removed from the datasets entirely. Moreover, participants signed a statement of release which outlined what the data was going to be used for and how the data was going to be anonymised.

FIT File Conversion, Data cleansing and Organisation

FIT Conversion

Initially, Garmin Development [documentation](#) was assessed to understand the basics. Garmin's FIT software development kit (SDK) was installed and the FitCSVTool was used to convert FIT files to CSV format. For a demonstration, please watch this [video](#). Note: the FIT SDK was downloaded and used on a Computer running a Windows operating system.

Testing was conducted initially using a Wahoo-derived FIT file. On review, the output wasn't organised in a way that instantly supported data analysis-related tasks. Further research was conducted to refine the conversion process.

A key [resource](#) used established a pathway to securing improved conversion techniques using a python script to convert FIT files in a highly organised manner. Moreover, the python script for wahoo-derived FIT files also included two additional functions that renamed the files to secure standardised naming conventions, removed sensitive data fields and also combined all of the CSV files into one master CSV file i.e, one master file per participant.

Key resources:

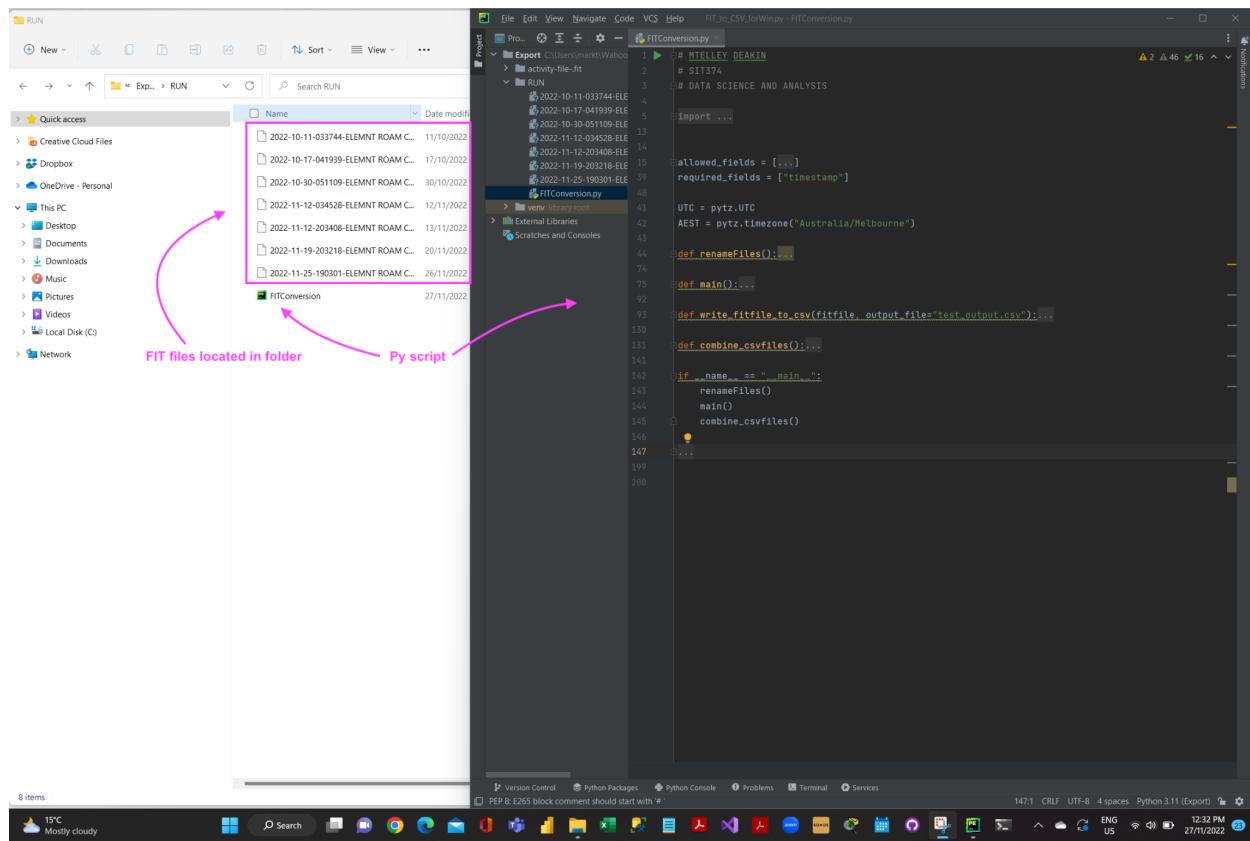
- <https://github.com/rdchip/FIT-to-CSV-converter-for-windows>
- <https://www.python.org/downloads/release/python-3110/>
- <https://www.jetbrains.com/edu-products/download/#section=pycharm-edu>
- <https://pypi.org/project/fitparse/#description>
- <https://github.com/dtcooper/python-fitparse>
- https://github.com/ekapope/Combine-CSV-files-in-the-folder/blob/master/Combine_CSVs.py

Data Cleansing and Organisation

Given the scale of data being procured, BigQuery (Google) was used to house and organise the data. Due to the size of the user CSV files, they were stored in a Google Cloud Storage bucket in preparation for housing the data in a data table. On review, the schema of the CSV file relating to data types also presented challenges; all data types were set as STRING values initially to avoid data table creation errors. The schema file used is located within the [Github](#) repo.

Key Operations (Wahoo)

Open the script file, drop FIT files into the correct folder and run.



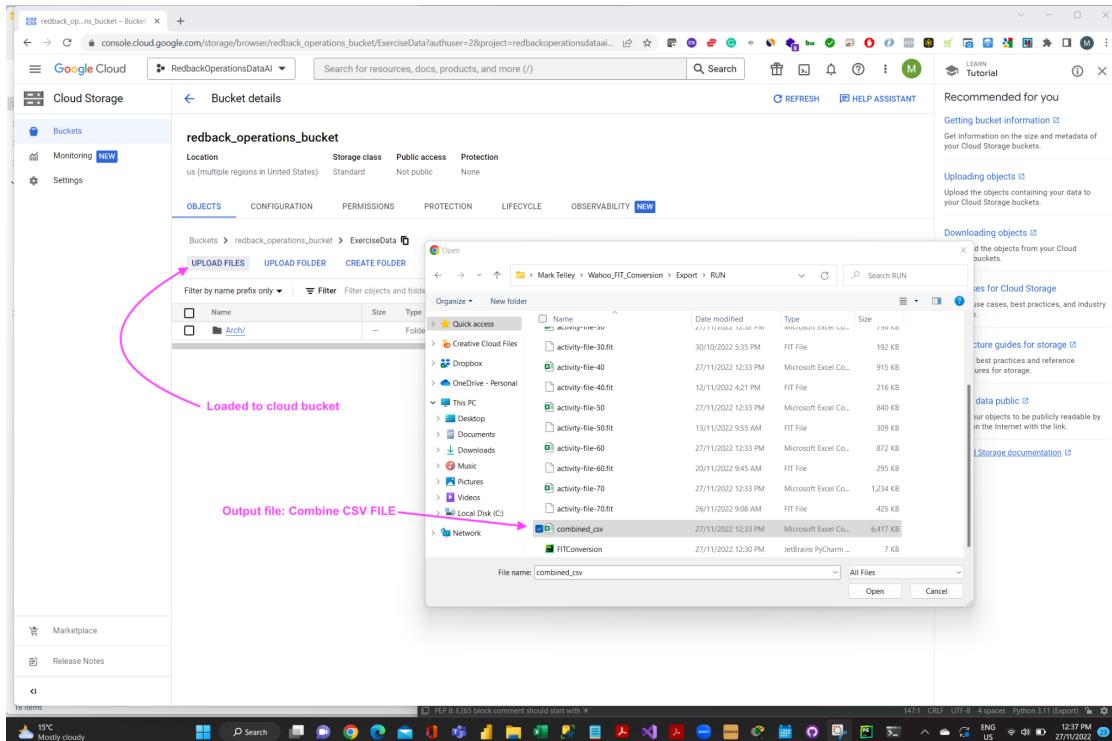
After the script has finished, a combined CSV file will be available in the same location + all single FIT files have been renamed and converted. Upload the combined CSV file to a Google Cloud storage bucket:

The screenshot shows a Microsoft Visual Studio Code interface with several panes:

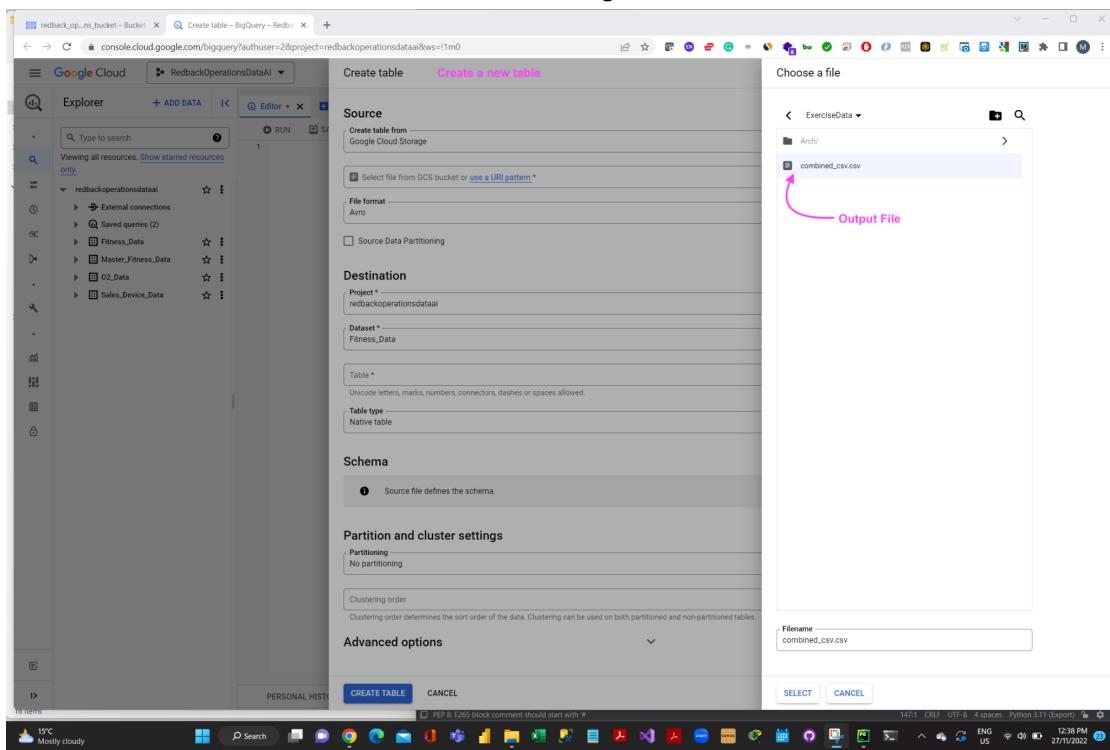
- Left Sidebar:** Quick access, Creative Cloud Files, Dropbox, OneDrive - Personal, This PC (Desktop, Documents, Downloads, Music, Pictures, Videos, Local Disk (C:)), Network.
- File Explorer:** Shows files in the current directory: activity-file-10ft, activity-file-20ft, activity-file-30ft, activity-file-40ft, activity-file-50ft, activity-file-60ft, activity-file-70ft, FITConversion.py, activity-file-10, activity-file-20, activity-file-30, activity-file-40, activity-file-50, activity-file-60, activity-file-70, combined.csv, and combined.csv.
- Code Editor:** The file `FITConversion.py` is open, containing Python code for file conversion. It includes imports for `pytz`, `os`, and `datetime`. The code defines functions for renaming files, writing files to CSV, and combining CSV files. A breakpoint is set at line 167. The code editor also shows a warning about EOL conversion.
- Terminal:** The terminal window shows the output of the script execution. It starts with "Renaming Started" and lists 7 files renamed. It then shows "File Converted" for each of the 7 files, followed by "finished conversions" and "Combining Files". The process finished with an exit code of 0.
- Bottom Status Bar:** Version control (git), Run, Python Package, Python Console, Problems, Terminal, Services. It also shows system information like CPU, RAM, and disk usage.

A pink arrow points from the text "Script Execution Output" to the terminal window.

Upload the CSV file to a Google Cloud Storage bucket:



Create a data table: Load from the cloud storage bucket:



When creating a data table, local files up to 100MB can only be loaded, hence the need to store them in a Cloud bucket.

Source

Create table from _____
Upload _____

Select file * _____ [BROWSE](#) [?](#)

File format _____ Avro

Upload a local file of up to 100 MB. For larger files, first upload data into [Google Cloud Storage](#), and then create a table from GCS. [Learn more](#)

Copy in the table schema JSON code:

The screenshot shows the 'Create Table' dialog in the Google Cloud BigQuery interface. The 'Source' section is set to 'Create table from Google Cloud Storage' with the path 'redbackoperations:redbackoperationsdata/fitness/_tableau/fitness_data.csv'. The 'Format' is set to 'CSV'. The 'Destination' section shows the table name 'redbackoperations:redbackoperationsdata/fitness-data' and the dataset 'Fitness_Data'. The 'Schema' section contains the following JSON schema:

```
1  {
2    "name": "creation_time",
3    "type": "TIMESTAMP",
4    "mode": "NULLABLE"
5  },
6  {
7    "name": "device_id",
8    "type": "STRING",
9    "mode": "NULLABLE"
10 },
11 {
12   "name": "distance",
13   "type": "FLOAT",
14   "mode": "NULLABLE"
15 },
16 {
17   "name": "distance_kilometers",
18   "type": "FLOAT",
19   "mode": "NULLABLE"
20 },
21 {
22   "name": "duration",
23   "type": "FLOAT",
24   "mode": "NULLABLE"
25 },
26 {
27   "name": "lat",
28   "type": "FLOAT",
29   "mode": "NULLABLE"
30 },
31 {
32   "name": "long",
33   "type": "FLOAT",
34   "mode": "NULLABLE"
35 },
36 {
37   "name": "location",
38   "type": "STRING",
39   "mode": "NULLABLE"
40 },
41 {
42   "name": "location_accuracy",
43   "type": "FLOAT",
44   "mode": "NULLABLE"
45 },
46 {
47   "name": "location_elevation",
48   "type": "FLOAT",
49   "mode": "NULLABLE"
50 },
51 {
52   "name": "location_heading",
53   "type": "FLOAT",
54   "mode": "NULLABLE"
55 },
56 {
57   "name": "location_pitch",
58   "type": "FLOAT",
59   "mode": "NULLABLE"
60 },
61 {
62   "name": "location_roll",
63   "type": "FLOAT",
64   "mode": "NULLABLE"
65 },
66 {
67   "name": "pace",
68   "type": "FLOAT",
69   "mode": "NULLABLE"
70 },
71 {
72   "name": "speed",
73   "type": "FLOAT",
74   "mode": "NULLABLE"
75 },
76 {
77   "name": "step_count",
78   "type": "INT64",
79   "mode": "NULLABLE"
80 },
81 {
82   "name": "steps",
83   "type": "FLOAT",
84   "mode": "NULLABLE"
85 },
86 {
87   "name": "user",
88   "type": "STRING",
89   "mode": "NULLABLE"
90 },
91 {
92   "name": "user_agent",
93   "type": "STRING",
94   "mode": "NULLABLE"
95 },
96 {
97   "name": "version",
98   "type": "STRING",
99   "mode": "NULLABLE"
100 }
```

The table has been created, with singular data types:

The screenshot shows the 'Schema' view for the 'fitness-data' table in the Google Cloud BigQuery interface. The table has 21 columns with the following data types:

Field name	Type	Mode	Description
creation_time	STRING	NULLABLE	time UTC
device_id	STRING	NULLABLE	
distance	STRING	NULLABLE	
distance_kilometers	STRING	NULLABLE	km
duration	STRING	NULLABLE	m
lat	STRING	NULLABLE	
long	STRING	NULLABLE	
location	STRING	NULLABLE	
location_accuracy	FLOAT	NULLABLE	
location_elevation	FLOAT	NULLABLE	
location_heading	FLOAT	NULLABLE	
location_pitch	FLOAT	NULLABLE	
location_roll	FLOAT	NULLABLE	
pace	FLOAT	NULLABLE	
speed	FLOAT	NULLABLE	
step_count	INT64	NULLABLE	
steps	FLOAT	NULLABLE	
user	STRING	NULLABLE	
user_agent	STRING	NULLABLE	
version	STRING	NULLABLE	

Preview of data:

BigQuery		Explorer	+ ADD DATA	fitness-activity-data	G QUERY	SHARE	COPY	SNAPSHOT	DELETE	EXPORT									
		Viewing all resources	[New started resources]	SCHEMA	DETAILS	PREVIEW													
Analysis																			
SQL workspace																			
Data transfers																			
Scheduled queries																			
Analytics Hub																			
Dataform																			
Migration																			
SQL translation																			
Administrators																			
Monitoring																			
Capacity management																			
BI Engine																			
Policy tags																			
Release Notes																			
PERSONAL HISTORY		PROJECT HISTORY																	
Results per page: 50 ▾ 101 – 200 of 99222																			
REFRESH																			

Run script to clean up the data table; The script will correct data types, key user data manually added, sensitive data removed and some filtering logic applied:

BigQuery		Explorer	+ ADD DATA	webui_cleaner.sql	G QUERY	SHARE	COPY	SNAPSHOT	DELETE	EXPORT									
		Viewing all resources	[New started resources]	SCHEMA	DETAILS	PREVIEW													
Analysis																			
SQL workspace																			
Data transfers																			
Scheduled queries																			
Analytics Hub																			
Dataform																			
Migration																			
SQL translation																			
Administrators																			
Monitoring																			
Capacity management																			
BI Engine																			
Policy tags																			
Release Notes																			
PERSONAL HISTORY		PROJECT HISTORY																	
Results per page: 50 ▾ 101 – 200 of 99222																			
REFRESH																			

Schedule the query to create (overwrite) a new table

Then run the transfer:

When complete, the green tick will indicate the transfer is complete:

Filter transfer configs							
	Display name	Source	Schedule (UTC)	Region	Destination dataset	Next scheduled	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/> convert_	Scheduled Query	None	US	Fitness_Data	None	⋮

Updated Schema (new data types etc):

The screenshot shows the Redshift Data API Explorer interface. On the left is the 'Explorer' sidebar with a search bar and a tree view of resources under 'redbackoperationsdataai'. The main area shows the 'SCHEMA' tab for the 'fitness-activity-user2' table. The schema table lists the following fields:

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
timestamp	STRING	NULLABLE				
timestamp_AEST	TIMESTAMP	NULLABLE				
date_AEST	DATE	NULLABLE				
distance	FLOAT	NULLABLE				
enhanced_altitude	FLOAT	NULLABLE				
ascent	FLOAT	NULLABLE				
grade	FLOAT	NULLABLE				
calories	FLOAT	NULLABLE				
enhanced_speed	FLOAT	NULLABLE				
heart_rate	FLOAT	NULLABLE				
temperature	INTEGER	NULLABLE				
cadence	FLOAT	NULLABLE				
power	FLOAT	NULLABLE				
left_right_balance	FLOAT	NULLABLE				
gps_accuracy	FLOAT	NULLABLE				
sessionID	STRING	NULLABLE				
userID	STRING	NULLABLE				
age	INTEGER	NULLABLE				
gender	STRING	NULLABLE				
weight	INTEGER	NULLABLE				
FTP	INTEGER	NULLABLE				

At the bottom are 'EDIT SCHEMA' and 'VIEW ROW ACCESS POLICIES' buttons.

The following query is then scheduled: Given the consistent naming conventions, a wildcard operation can be used to join user tables together:

The screenshot shows the Redshift Data API Query Editor. At the top are buttons for RUN, SAVE, SHARE, SCHEDULE, and MORE. The query editor contains the following SQL code:

```

1 -- MASTER FILE OPERATIONS
2 SELECT
3 *
4 FROM
5 `redbackoperationsdataai.Fitness_Data.fitness-activity-*` -- WILDCARD
6

```

The master table is created which houses all user data:

Row	timestamp	timestamp_AEST	date_AEST	distance	enhanced_alfity	ascent	grade	calories	enhanced_speed	heart_rate	userID	gender	age	weight	power	cadence	ftp	wpkrg	wperkg
1	2021-03-30 19:11:59+00:00	2021-03-31 06:11:59 UTC	2021-03-31	27.73852	30.600000...	98.0	1.41	582.0	42.2172000...	2									
2	2021-03-31 18:36:15+00:00	2021-04-01 05:36:15 UTC	2021-04-01	0.0	null	null	null	0.0	0.0	0.0									1
3	2021-03-31 18:36:17+00:00	2021-04-01 05:36:17 UTC	2021-04-01	0.00158	null	null	null	1.0	9.66240000...	1									
4	2021-03-31 18:36:19+00:00	2021-04-01 05:36:19 UTC	2021-04-01	0.01517	null	null	null	1.0	27.198	1									
5	2021-03-31 18:36:31+00:00	2021-04-01 05:36:31 UTC	2021-04-01	0.15164	null	null	null	7.0	45.9684	1									
6	2021-03-31 18:36:35+00:00	2021-04-01 05:36:35 UTC	2021-04-01	0.19857	null	null	null	9.0	39.9708	1									
7	2021-03-31 18:36:36+00:00	2021-04-01 05:36:36 UTC	2021-04-01	0.2098	null	null	null	9.0	40.8276	1									
8	2021-03-31 18:36:37+00:00	2021-04-01 05:36:37 UTC	2021-04-01	0.22154	null	null	null	9.0	40.8348	1									
9	2021-03-31 18:36:40+00:00	2021-04-01 05:36:40 UTC	2021-04-01	0.25423	null	null	null	11.0	40.6044	1									
10	2021-03-31 18:36:45+00:00	2021-04-01 05:36:45 UTC	2021-04-01	0.31568	null	null	null	13.0	41.9868	1									
11	2021-03-31 18:36:47+00:00	2021-04-01 05:36:47 UTC	2021-04-01	0.34064999...	null	null	null	14.0	41.0436	1									
12	2021-03-31 18:36:52+00:00	2021-04-01 05:36:52 UTC	2021-04-01	0.39262	null	null	null	15.0	34.2216	1									
13	2021-03-31 18:36:54+00:00	2021-04-01 05:36:54 UTC	2021-04-01	0.4093	null	null	null	15.0	27.8784	1									
14	2021-03-31 18:36:55+00:00	2021-04-01 05:36:55 UTC	2021-04-01	0.41716	null	null	null	15.0	24.3072	1									
15	2021-03-31 18:36:56+00:00	2021-04-01 05:36:56 UTC	2021-04-01	0.42144	null	null	null	15.0	20.7684	1									
16	2021-03-31 18:37:14+00:00	2021-04-01 05:37:14 UTC	2021-04-01	0.47644	null	null	null	17.0	12.3624	1									
17	2021-04-06 06:46:54+00:00	2021-04-06 06:46:54 UTC	2021-04-06	25.52177	13.799999...	252.0	0.0	565.0	30.6864	1									
18	2021-04-16 21:44:52+00:00	2021-04-17 07:44:52 UTC	2021-04-17	19.42806	44.200000...	261.0	3.55	491.0	23.291999...	1									
19	2021-04-17 00:11:00+00:00	2021-04-17 10:11:00 UTC	2021-04-17	77.01831	null	1194.0	null	1934.0	null										

This master table will be used as a key resource; The table can be queried, connected to Tableau etc for data analysis and modelling efforts:

```

-- MASTER FILE OPERATIONS
SELECT
  userID,
  ROUND(AVG(heart_rate),2) AS avg_heart_rate,
  ROUND(AVG(enhanced_speed),2) AS avg_speed,
  ROUND(AVG(power),2) AS avg_power,
  ROUND(AVG(cadence),2) AS avg_cadence,
  MAX(age) AS age,
  MAX(FTP) AS FTP,
  MAX(weight) AS weight,
  trim(gender) as gender,
  AVG(wperkg) AS wperkg,
  FROM
    `redbackoperationsdataai.Master_Fitness_Data.master-fitness-activity`
  WHERE
    cadence != 0
    AND enhanced_speed != 0
    AND power != 0
  GROUP BY
    userID,
    gender
  
```

Row	userID	avg_heart_rate	avg_speed	avg_power	avg_cadence	age	FTP	weight	gender	wperkg
1	U1000000	156.8	30.43	224.82	80.39	33	301	80	MALE	
2	U1000003	162.92	30.94	255.64	81.84	33	310	81	MALE	
3	U1000002	164.55	26.86	203.3	85.51	46	270	75	MALE	

The process was replicated for additional users.

Key Operations (Garmin)

There was a slight change in the set of operations for Garmin-derived data. As a result, certain attributes were omitted i.e., Left/Right balance. To simplify operations and in the interest of time, only one user with Garmin-derived data was included in the final data set. Further work is required to support and update key operations for Garmin-derived data.

Access to Google Console Assets

Google Console Storage

Access to Google Cloud Storage and the relevant project buckets has been restricted due to data privacy requirements and billing controls; I'm using my own personal Google Cloud Storage account to store combined CSV files.

Google Bigquery

Relevant project team members have had access enabled for all datasets - Team members only have viewing privileges. Access details below:

Admin/Owner: mtelley@deakin.edu.au

Access information - watch this Loom [video](#).

Other Notes

Please note, there is availability to query a data table when saving the query results to another data table. This method was not used for a range of reasons, mainly due to the transfer method providing added debugging capabilities should errors occur.

There is an outstanding task that requires some minor SQL development to allocate session IDs to workouts i.e., there can be more than one session in a day, such that each session must have a unique ID. This will be achieved using lag/over operations:

```
4 WITH
5   proto_1 AS (
6     SELECT
7       timestamp,
8       LAG(Distance) OVER(ORDER BY timestamp ASC ) AS PreviousRow,
9     CASE
10      WHEN Distance < LAG(Distance) OVER(ORDER BY timestamp ASC ) THEN 1
11      ELSE
12        0
13    END
14    AS endSession,
15  FROM
16    `redbackoperationsdataai.Fitness_Data.fitness-activity-user1`
17  ORDER BY
18    Timestamp ASC)
19 SELECT
20   *,
21  FROM
22  proto_1
23 WHERE endSession != 0
```

[← Query results](#)

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	timestamp	PreviousRow	endSession			
1	2021-03-09 22:22:07+00:00	62.928	1			
2	2021-03-11 22:43:39+00:00	17.13455	1			
3	2021-03-13 19:53:22+00:00	10.30559	1			
4	2021-03-14 20:54:21+00:00	73.90983	1			
5	2021-03-15 07:33:59+00:00	5.18281000...	1			

Outcome

Over 15,740,893 seconds (+4300 hours) of exercise data has been collated and stored in a single data set. This data set will be used to create data visualisations, train/test models etc. Moreover, the outcome of this exercise will help inform key data collection efforts for the in-game experience i.e., how to collect data, deploy models and then show the users (at the end of their workout or session) a data visualisation that represents their efforts.

Next page for a summary of the master table:

Explorer + ADD DATA ?

Type to search

Viewing all resources. [Show starred resources only.](#)

- ▶ Project queries
 - ▶ Lag_proto
 - ▶ Master_Ops
 - ▶ User_ID
- ▼ Fitness_Data
 - ▶ fitness-activity-user1
 - ▶ fitness-activity-user10
 - ▶ fitness-activity-user2
 - ▶ fitness-activity-user3
 - ▶ fitness-activity-user4
 - ▶ fitness-activity-user5
 - ▶ fitness-activity-user6
 - ▶ fitness-activity-user7
 - ▶ fitness-activity-user8V2
 - ▶ fitness-activity-user9V2
- ▼ Master_Fitness_Data
 - ▶ master-fitness-activity
- ▼ O2_Data
 - ▶ O2_Kaggle_data
 - ▶ O2_Malaga_Data
- ▶ Sales_Device_Data

master-fitness-activity ? X +

master-fitness-activity QUERY SHARE COPY SNAPSHOT

SCHEMA DETAILS PREVIEW

Table info

Table ID redbackoperationsdataai.Master_Fitness_Data.master-fitness-activity

Created Nov 27, 2022, 1:55:14 PM UTC+11

Last modified Nov 27, 2022, 8:44:59 PM UTC+11

Table expiration NEVER

Data location US

Default collation

Description

Storage info ?

Number of rows	15,740,893
Total logical bytes	2.08 GB
Active logical bytes	2.08 GB
Long term logical bytes	0 B
Total physical bytes	215.17 MB
Active physical bytes	215.17 MB
Long term physical bytes	0 B
Time travel physical	67.28 MB

PERSONAL HISTORY PROJECT HISTORY