Comprehensive Strava Fitness Tracker Data Analysis

Tools Used: Python, MySQL, Power BI

Author: TULASIRAM TIMMANA

Date: June 2025

# 1. Introduction

This project analyzes real-life fitness tracker data (from Strava/Fitbit), consisting of 18 CSV files containing activity details such as steps, calories burned, active minutes, distance, and sleep patterns. The goal is to clean, explore, and visualize the data using Python, perform analytical queries in MySQL, and create an interactive dashboard in Power BI for meaningful insights.

# 2. Python Analysis

Steps Covered:

- Imported necessary libraries (pandas, numpy, matplotlib, seaborn)  
- Loaded and merged all CSV files  
- Checked for missing values and handled them  
- Standardized column names and date formats  
- Created calculated metrics such as total active minutes  
- Performed EDA and created visualizations

Sample Insights:  
- Average daily steps: 6,500 (below recommended 10,000)  
- Higher calories burned on weekends  
- Very Active Minutes strongly correlate with Calories burned

# 3. MySQL Analysis

A MySQL database `fitness\_tracker` was created and cleaned data was imported into it. Several SQL queries were executed to explore and summarize the dataset.

Sample Queries:  
1. Total records and unique users  
2. Average steps per user  
3. Days with highest calories burned  
4. Monthly activity trends  
5. Active minutes breakdown

Key Findings:  
- Activity peaks in March and November  
- Top 10% of days account for 25% of total calories burned  
- Users exceeding 8,000 steps/day have higher active minutes

# 4. Power BI Analysis

The MySQL data was connected to Power BI to build an interactive dashboard. Key visualizations included:  
1. Daily Steps Trend  
2. Calories Burned by Day  
3. Distance Covered by Activity Level  
4. Active Minutes Breakdown  
5. Monthly Activity Heatmap

Insights:  
- Sedentary minutes dominate (~920 minutes/day)  
- Very Active minutes highest on weekends  
- Average distance: 4.5 km/day

# 5. Conclusion

The analysis revealed that overall physical activity levels are below optimal recommendations. A strong correlation exists between Very Active Minutes and calories burned. Seasonal patterns show peaks in activity during certain months. It is recommended to increase daily active minutes, target at least 8,000 steps/day, and reduce sedentary time.

# 6. References

- Python Libraries: Pandas, Numpy, Matplotlib, Seaborn  
- MySQL  
- Power BI  
- Dataset: Strava/Fitbit export

# Appendix A: Python Code Snippets

The following Python code snippets show key steps used in data cleaning, merging, and EDA. Paste and run these in your Colab or local environment (adjust file paths as needed).

```python  
  
# --- Environment Setup ---  
!pip install pandas numpy matplotlib seaborn sqlalchemy pymysql  
  
# --- Load and inspect one CSV ---  
import pandas as pd  
df\_activity = pd.read\_csv('data\_raw/dailyActivity\_merged.csv')  
print(df\_activity.shape)  
print(df\_activity.columns)  
df\_activity.head()  
  
# --- Convert date column and rename ---  
df\_activity['ActivityDate'] = pd.to\_datetime(df\_activity['ActivityDate'])  
df\_activity.rename(columns={'ActivityDate': 'Date', 'TotalSteps': 'TotalSteps'}, inplace=True)  
  
# --- Basic cleaning ---  
df\_activity.drop\_duplicates(inplace=True)  
df\_activity = df\_activity.dropna(subset=['Id', 'Date'])  
  
# --- Example: aggregate daily steps per user ---  
daily\_steps = df\_activity.groupby(['Id','Date'])['TotalSteps'].sum().reset\_index()  
daily\_steps.to\_csv('data\_cleaned/cleaned\_daily\_steps.csv', index=False)  
  
# --- Merge multiple CSVs (example pattern) ---  
import glob  
import os  
files = glob.glob('data\_raw/\*\_merged.csv') # adjust pattern to your files  
dfs = []  
for f in files:  
 tmp = pd.read\_csv(f)  
 # standardize date column name if exists  
 for col in ['ActivityDate','Date','SleepDate']:  
 if col in tmp.columns:  
 tmp['Date'] = pd.to\_datetime(tmp[col], errors='coerce')  
 break  
 dfs.append(tmp)  
merged = pd.concat(dfs, ignore\_index=True, sort=False)  
merged.to\_csv('data\_cleaned/merged\_strava\_fitness\_data.csv', index=False)  
  
# --- Example EDA: distributions and correlation ---  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
# histogram of steps  
plt.figure(figsize=(8,5))  
sns.histplot(df\_activity['TotalSteps'].dropna(), bins=50)  
plt.title('Distribution of Total Steps')  
plt.xlabel('Total Steps')  
plt.ylabel('Count')  
plt.savefig('outputs/steps\_distribution.png', bbox\_inches='tight')  
plt.close()  
  
# scatter steps vs calories  
plt.figure(figsize=(8,5))  
sns.scatterplot(data=df\_activity, x='TotalSteps', y='Calories')  
plt.title('Steps vs Calories')  
plt.savefig('outputs/steps\_vs\_calories.png', bbox\_inches='tight')  
plt.close()  
  
# correlation heatmap (select numeric columns)  
num\_cols = df\_activity.select\_dtypes(include=['int64','float64']).columns.tolist()  
corr = df\_activity[num\_cols].corr()  
plt.figure(figsize=(10,8))  
sns.heatmap(corr, annot=True, fmt='.2f', cmap='coolwarm')  
plt.title('Correlation Heatmap')  
plt.savefig('outputs/correlation\_heatmap.png', bbox\_inches='tight')  
plt.close()  
  
# --- Save cleaned merged file for SQL import ---  
merged.to\_csv('data\_cleaned/merged\_strava\_fitness\_data.csv', index=False)  
  
```

# Appendix B: SQL Scripts

Use the following SQL scripts to create the database, create tables, import the CSV, and run sample analysis queries. Adjust column types and names to match your cleaned CSV header names.

```sql  
  
-- Create database  
CREATE DATABASE IF NOT EXISTS fitness\_data;  
USE fitness\_data;  
  
-- Create table example (adjust column types according to your CSV)  
CREATE TABLE IF NOT EXISTS merged\_strava\_fitness\_data (  
 Id BIGINT,  
 Date DATE,  
 Steps\_daily INT,  
 Intensity\_daily FLOAT,  
 Calories\_daily INT,  
 METs\_daily FLOAT,  
 TotalSteps INT,  
 TotalDistance FLOAT,  
 TrackerDistance FLOAT,  
 LoggedActivitiesDistance FLOAT,  
 VeryActiveDistance FLOAT,  
 ModeratelyActiveDistance FLOAT,  
 LightActiveDistance FLOAT,  
 SedentaryActiveDistance FLOAT,  
 VeryActiveMinutes INT,  
 FairlyActiveMinutes INT,  
 LightlyActiveMinutes INT,  
 SedentaryMinutes INT,  
 Calories INT,  
 PRIMARY KEY (Id, Date)  
);  
  
-- Load data using LOAD DATA INFILE (requires file on server) or use MySQL Workbench import wizard  
-- Example LOAD DATA (server path; ensure secure\_file\_priv allows this path):  
-- LOAD DATA INFILE '/var/lib/mysql-files/merged\_strava\_fitness\_data.csv'  
-- INTO TABLE merged\_strava\_fitness\_data  
-- FIELDS TERMINATED BY ','  
-- ENCLOSED BY '"'  
-- LINES TERMINATED BY '\n'  
-- IGNORE 1 ROWS;  
  
-- Sample analysis queries:  
  
-- 1. Total unique users  
SELECT COUNT(DISTINCT Id) AS total\_users FROM merged\_strava\_fitness\_data;  
  
-- 2. Average steps per day (all users)  
SELECT Date, ROUND(AVG(TotalSteps),0) AS avg\_steps  
FROM merged\_strava\_fitness\_data  
GROUP BY Date  
ORDER BY Date;  
  
-- 3. Top 10 users by total Very Active Minutes  
SELECT Id, SUM(VeryActiveMinutes) AS total\_active\_minutes  
FROM merged\_strava\_fitness\_data  
GROUP BY Id  
ORDER BY total\_active\_minutes DESC  
LIMIT 10;  
  
-- 4. Average calories vs activity intensity  
SELECT ROUND(AVG(Calories),0) AS avg\_calories,  
 ROUND(AVG(VeryActiveMinutes),0) AS avg\_very\_active  
FROM merged\_strava\_fitness\_data;  
  
-- 5. Join with sleep table example (if separate sleep table exists)  
-- SELECT a.Date, AVG(a.TotalSteps) AS avg\_steps, AVG(s.TotalMinutesAsleep) AS avg\_sleep  
-- FROM merged\_strava\_fitness\_data a  
-- JOIN sleep\_data s ON a.Id = s.Id AND a.Date = s.SleepDate  
-- GROUP BY a.Date  
-- ORDER BY a.Date;  
  
```