1. How many unique users’ data was collected for daily activity and daily sleep data sets respectively?

SELECT COUNT(DISTINCT Id)

FROM `fitbit\_data.dailyactivty`

33 users were returned for daily activity.

For daily sleep

SELECT COUNT(DISTINCT Id)

FROM `fitbit\_data.sleepday`

24 users were returned for daily sleep.

1. What are the average activity levels and sleep of the entire dataset?

SELECT AVG(TotalSteps) AS Steps,

AVG(TotalDistance) AS Distance,

AVG(VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes)/60 AS ActivityHours,

AVG(SedentaryMinutes)/60 AS SedentaryHours,

AVG(Calories) as Calories

FROM `fitbit\_data.dailyactivty`

We see from the dataset, the average number of steps is 7637.91, 5.49 miles, 3.79 hours of activity, 16.52 hours of being sedentary, and 2303.6 calories burnt.

SELECT AVG(TotalMinutesAsleep)/60 as SleepHours,

AVG(TotalTimeInBed)/60 as HoursInBed

FROM `bellabeat-450515.fitbit\_data.sleepday`

Users on average sleep for 7 hours and spend 7.64 hours in bed.

1. Now create a table showing each individual user’s average activity levels filtered by least to most steps.

SELECT Id,

AVG(TotalSteps) AS Steps,

AVG(TotalDistance) AS Distance,

AVG(VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes)/60 AS ActivityHours,

AVG(SedentaryMinutes)/60 AS SedentaryHours,

AVG(Calories) as Calories

FROM `fitbit\_data.dailyactivty`

GROUP BY Id

ORDER BY Steps

1. Create a column in the table that shows total active hours.

ALTER TABLE dailyactivity

ADD COLUMN ActiveHours varchar(255);

UPDATE dailyactivity

SET ActiveHours AS (VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes)/60

1. Now, going back to the data from number 3, how can we adjust the query to show each unique user’s ID and assign a status to see if they are meeting the recommended number of steps per day and hours active per the Mayo Clinic and US. Department of Health and Human Services?

SELECT Id,

AVG(TotalSteps) AS Steps,

AVG(TotalDistance) AS Distance,

AVG(VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes)/60 AS ActivityHours,

AVG(SedentaryMinutes)/60 AS SedentaryHours,

AVG(Calories) as Calories,

CASE

  WHEN AVG(TotalSteps) < 3000 THEN 'Not Active'

  WHEN AVG(TotalSteps) BETWEEN 3000 AND 4000 THEN 'Sufficiently Active'

  WHEN AVG(TotalSteps) BETWEEN 4000 AND 9000 THEN 'Very Active'

  ELSE 'Extremely Active'

  END AS StepStatus,

CASE

  WHEN ActiveHours < 2.5 THEN 'Noncompliant'

  WHEN ActiveHours > 2.5 THEN 'Compliant'

END AS ComplianceStatus

FROM `fitbit\_data.dailyactivty`

GROUP BY Id

The table shows most users actually surpasses the number of steps taken by the average American and the US Department of Health and Human Service’s recommendation with only about 6 not being compliant or not active.

To look at specific users who are noncompliant, add the “having’ clause underneath the group by query will filter the data further down.

HAVING ComplianceStatus = 'Noncompliant'

1. Creating a new table called “mostactive” using the previous query’s structure but for compliant users, how can we check which users traveled the most distance?

SELECT \*,

DENSE\_RANK() OVER(ORDER BY Distance DESC) AS RANKNUM

FROM `fitbit\_data.mostactive`

ORDER BY RANKNUM

The window function query shows user 8877689391 traveled the most distance of 13.21 miles.

How would the query be adjusted to compare extremely active users to extremely active users and very active users to their own group for something like sedentary hours?

SELECT \*,

DENSE\_RANK() OVER(PARTITION BY StepStatus ORDER BY SedentaryHours DESC) AS RANKNUM

FROM `fitbit\_data.mostactive`

Adding a partition allows us to rank users within their subcategory. We see user 8053475328 is the most sedentary of extremely active users while 8253242879 is the most sedentary for very active users.

1. How can we combine activity and sleep data together?

SELECT [dailyactivity].Id,

[dailyactivity].ActivityDate,

[dailyactivity].TotalSteps,

[dailyactivity].TotalDistance,

[dailyactivity].(VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes) AS ActiveMin,

[dailyactivity].Calories,

[sleepday].TotalMinutesAsleep,

[sleepday].TotalTimeInBed

FROM `fitbit\_data.dailyactivty`

LEFT JOIN `fitbit\_data.sleepday`

ON 'fitbit\_data.dailyactivity.Id'='fitbit\_data.sleepday.Id'

AND 'fitbit\_datadailyactivity.activitydate'='fitbit\_data.sleepday.sleepday'

1. Create a table categorizing the amount of sleep users’ had and aggregate counts of each category.

SELECT

CASE

  WHEN TotalMinutesAsleep/60 < 5 THEN 'Sleep Deprived'

  WHEN TotalMinutesAsleep/60 BETWEEN 5 and 7 THEN 'Insufficient Sleep'

  WHEN TotalMinutesAsleep/60 BETWEEN 7 and 9 THEN 'Sufficient Sleep'

  WHEN TotalMinutesAsleep/60 > 9 THEN 'Overslept'

  END AS SleepStatus,

  COUNT (\*) AS StatusCount

FROM fitbit\_data.sleepday

GROUP BY SleepStatus

ORDER BY StatusCount DESC

Users who submitted their sleep data mostly get sufficient sleep with some occasions where the sleep is not sufficient. It’s quite rare that they oversleep or are too sleep deprived.

1. Calculate the correlation between hours of sleep and hours spent in bed.

SELECT CORR(TotalTimeInBed,TotalMinutesAsleep) AS Correlation

FROM `fitbit\_data.sleepday`

A positive correlation of 0.9304 is reported.

1. Create a table with with R values for various correlations of activity factors.

SELECT CORR(TotalSteps,Calories) AS StepsVsCalories,

CORR(TotalSteps,TotalDistance) AS StepsVsDistance,

CORR(ActiveHours,TotalDistance) AS ActiveHoursVsDistance,

CORR(SedentaryMinutes,Calories) AS SedVsCalories,

CORR(SedentaryMinutes,TotalSteps) AS SedVsSteps

FROM `fitbit\_data.dailyactivty`