# Case Study #2

Fitbit health tracker – a worldwide fitness company.

## About

Fitbit is a worldwide fitness company that uses sensors and wireless technology to bring better fitness and health experiences. This dataset focuses on data generated by 33 fitbit users who consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, weight, and sleep monitoring. The data was collected through fitbit health trackers between 12/04/2016-12/05/2016.

## Ask Phase

**Business task: analyze smart device data to gain insight into how consumers are using their smart devices.** The insights discovered will help guide marketing strategy for the company.

## Prepare Phase

**Checking for data integrity with ROCCC process:**

**The reliability of data** – which means accurate, complete, and unbiased data. As stated by the dataset distributor, this data was generated by respondents to a distributed survey via "Amazon Mechanical Turk". The data can be considered reliable only If there was no intervention of the respondent and the distributor during data transfer, collection, or any other modification during survey. As per bias in the data, the survey was accessible to all without any preference to a specific group (whoever wanted to share the data took the survey). However, the small number of respondents might cause some sort of misrepresentation.

**Original data** – second- or third-party data can be modified along the way or can be changed through problematic data migration/ transfer. To determine that the data is original the data should be validated with the original source. This data is second party, and it can't be validated because there is no access to the original. For the purpose of this analysis, it will be considered faithful to the original source.

**Comprehensive** – data that contains critical information needed to answer the business task. The datasets contain anything they claim to contain, personal tracker data, physical activity, heart rate, weight, and sleep monitoring.

**Current** – seemingly, this data is not current because it was collected in 2016, and it can be considered old, however, due to the nature of the information, this data is still relevant. It can be claimed that the tracked parameters are timeless. However, it can also be argued that the degree of activity is prone to different influences over the years. For the purpose of this analysis, the data will be considered current.

**Cited** – who created the dataset? Is it part of a credible organization?
It is known that the data was collected via “Amazon Mechanical Turk”. The company can be trusted; however, I am not familiar with this service and the way it operates. I believe that the survey takes its precautions to prevent possible problems such as multiple applications for a single user. For the purpose of this analysis, the data will be considered cited.

# Process Phase (Cleaning and Manipulating)

The cleaning and manipulation were done in Excel and SQL. In Excel, before the initial cleaning, a copy was created to keep the original at hand. The cleaning process includes removing duplicates, irrelevant data, extra spaces, blanks. Also, correcting inconsistencies, typos, misfielded values, and data types.

# Daily Activity Dataset

This dataset contains 940 records and 15 fields (Id, ActivityDay, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDistance, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, VeryActiveMinutes, SedentaryActiveDistance, LightActiveDistance, ModeratelyActiveDistance, VeryActiveDistance, Calories). The data covers a period of 30 days from 12/04/2016 to 12/05/2016.

1. Checked for duplicates (Data tab -> Data tools: remove duplicates). None discovered.
2. No irrelevant data was discovered through the Filter tool. There were no Null values in any of the fields or any numbers that stood out (e.g., negative values).
3. All IDs were checked for character length with the following formula =len(cell_ref) and no irregular lengths were detected. All lengths consisted of 10 characters.
4. The meta data about ‘TotalDistance’ and ‘TrackerDistance’ described the 2 fields as follows; total kilometers tracked, and total kilometers tracked by fitbit device respectively. Because the difference was not clear, these two fields were compared to one another. A logical test was created = cell_ref_field1 = cell_ref_field2 and replicated to all cells. Afterwards, a conditional formatting rule was created to mark all false values. No false values were discovered, which indicates that these two fields were identical. ‘TrackerDistance’ field was removed.
5. Redundancy check: are there any records with zero values in all fields?
   A logical test was created =AND(cell_ref1=0, cell_ref2=0,…) however records with TRUE outputs were not spotted. There was no need for any action to be taken.

# Other Datasets

‘Sleep day’, ‘hourly intensities’, and ‘weight’ datasets were checked for duplicates, character length in the Id field and for any irregular numbers (negative numbers, overly large numbers, or NULL values). Nothing irregular was discovered.

After the cleaning process, these datasets were uploaded to SQL Server.

The METs dataset couldn’t be cleaned in Excel due to its large number of records, so the cleaning was conducted via SQL Server.

## Are there any NULL values?

```
SELECT *
FROM [dbo].[METs]
WHERE [Id] IS NULL OR [ActivityMinute] IS NULL OR [METs] IS NULL
```

| Id | ActivityMinute | METs |
|---|---|---|

The output is an empty result, meaning that there are no NULL records.

## Checking for duplicates

1. **The total number of records:**

```
SELECT COUNT(*) total_rec_num
FROM [dbo].[METs]
```

| Total_rec_num |
|---|
| 1,325,580 |

The query above counts all records without considering duplicates.

2. **Checking for unique values:**

```
SELECT COUNT(count_id_act) unique_rec_count
FROM (
            SELECT COUNT(*) count_id_act
            FROM [dbo].[METs]
            GROUP BY [Id], [ActivityMinute]
      ) new_table
```

| unique_rec_count |
|---|
| 1,325,580 |

If there were any duplicates the unique_rec_count would have been smaller than total_rec_num. However, there is no discrepancy between the two results which indicates that there are no duplicate records.

## Id length

```
SELECT MIN(LEN([Id])) min_id_len, MAX(LEN([Id])) max_id_len
FROM [dbo].[METs]
```

First, the range of character length was examined.

| min_id_len | max_id_len |
|---|---|
| 11 | 12 |

This was an unexpected result. In all previous datasets the Ids exhibited a length of 10 characters.

**To verify, the following query was executed:**

```
SELECT COUNT([Id]) total_id_count
FROM (
SELECT [Id], LEN([Id]) len_id
FROM [dbo].[METs])new_t
WHERE len_id = 11 or len_id = 12
```

| total_id_count |
|---|
| 1,325,580 |

All records include extra spaces.

**Trimming extra spaces (leading and trailing):**

The id was initially stored as bigint datatype, however, to make the space trimming possible, the ids must be changed to nvarchar.

```
ALTER TABLE [dbo].[METs]
ALTER COLUMN [Id] nvarchar(50)
```

Removing trailing and leading spaces:

```
UPDATE [dbo].[METs]
SET [Id] = TRIM([Id])
```

To verify, the range of characters' length was checked once again:

```
SELECT MIN(LEN([Id])) min_id_len, MAX(LEN([Id])) max_id_len
FROM [dbo].[METs]
```

| min_id_len | max_id_len |
|---|---|
| 10 | 10 |

# Analyze Phase

1. **How many users tracked their parameters in each dataset?**

```
SELECT 'activity' tracked_parameter, COUNT(DISTINCT [Id]) users_count
FROM [dbo].[dailyActivity]
UNION ALL
SELECT 'intensities', COUNT(DISTINCT [Id])
FROM [dbo].[hourlyIntensities]
UNION ALL
SELECT 'METs', COUNT(DISTINCT [Id])
FROM [dbo].[METs]
UNION ALL
SELECT 'sleep', COUNT(DISTINCT [Id])
FROM [dbo].[sleepDay]
UNION ALL
SELECT 'weight', COUNT(DISTINCT [Id])
FROM [dbo].[weight]
```

| tracked_parameter | users_count |
|---|---|
| activity | 33 |
| intensities | 33 |
| METs | 33 |
| sleep | 24 |
| weight | 8 |

2. **The overall period tracked in the daily activity dataset:**

```
SELECT COUNT(DISTINCT [ActivityDate]) days_count
FROM [dbo].[dailyActivity]
```

| days_count |
|---|
| 31 |

Even though the overall period tracked is 31 days, it is not clear whether each user tracked activity for 31 days or less.

## Daily Activity Dataset

### 3. The number of days tracked by each user:

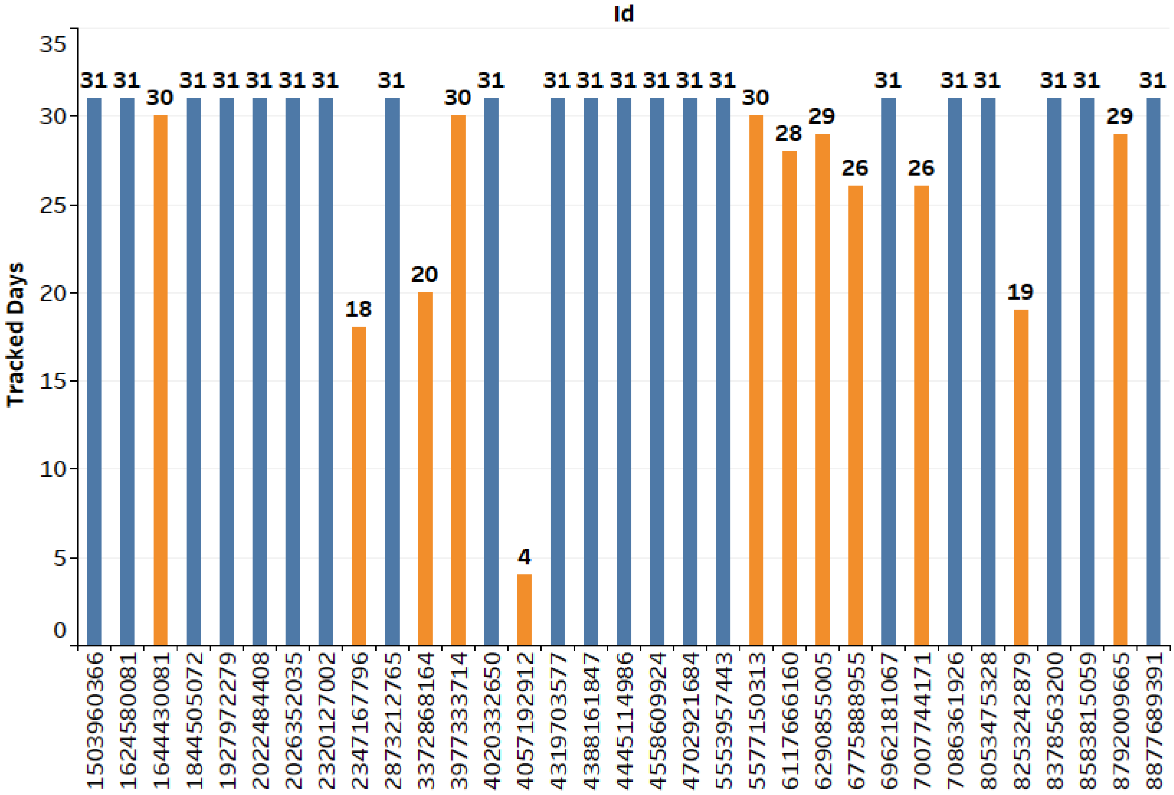
```
SELECT [Id], COUNT([ActivityDate]) tracked_days
FROM [dbo].[dailyActivity]
GROUP BY [Id]
ORDER BY tracked_days DESC
```

**Another option:**

```
SELECT
      DISTINCT [Id],
      COUNT([ActivityDate]) OVER(PARTITION BY [Id]) activity_day_count
FROM [dbo].[dailyActivity]
ORDER BY activity_day_count DESC
```

| Id | tracked_days |
|---|---|
| 4020332650 | 31 |
| 4702921684 | 31 |
| 4388161847 | 31 |
| 8583815059 | 31 |
| 4445114986 | 31 |
| 2873212765 | 31 |
| 2320127002 | 31 |
| 8877689391 | 31 |
| 1844505072 | 31 |
| 7086361926 | 31 |
| 2026352035 | 31 |
| 4558609924 | 31 |
| 8053475328 | 31 |
| 4319703577 | 31 |
| 1503960366 | 31 |
| 5553957443 | 31 |
| 1927972279 | 31 |
| 6962181067 | 31 |
| 1624580081 | 31 |
| 2022484408 | 31 |
| 8378563200 | 31 |
| 5577150313 | 30 |
| 1644430081 | 30 |
| 3977333714 | 30 |
| 8792009665 | 29 |
| 6290855005 | 29 |
| 6117666160 | 28 |
| 7007744171 | 26 |

| | |
|---|---|
| 6775888955 | 26 |
| 3372868164 | 20 |
| 8253242879 | 19 |
| 2347167796 | 18 |
| 4057192912 | 4 |



The highlighted bars (orange) are < 31 days.

As presented in the bar chart, most, however not all users, tracked their activity for 31 days.

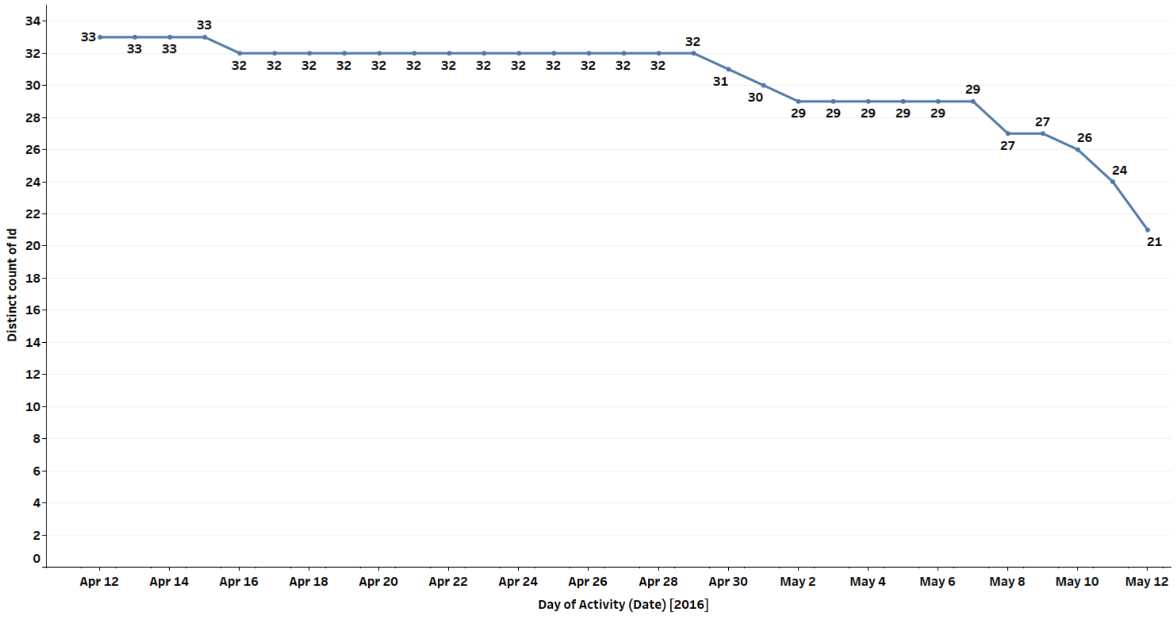
### 4. How many users tracked their activity each day?

**Daily Activity**

```
SELECT cast([ActivityDate] as DATE) activity_date, count(distinct [Id])
id_count_per_day
FROM [dbo].[dailyActivity]
GROUP BY cast([ActivityDate] as DATE)
ORDER BY cast([ActivityDate] as DATE)
```

| activity_date | id_count_per_day |
|---|---|
| 2016-04-12 | 33 |
| 2016-04-13 | 33 |
| 2016-04-14 | 33 |

| | |
|---|---|
| 2016-04-15 | 33 |
| 2016-04-16 | 32 |
| 2016-04-17 | 32 |
| 2016-04-18 | 32 |
| 2016-04-19 | 32 |
| 2016-04-20 | 32 |
| 2016-04-21 | 32 |
| 2016-04-22 | 32 |
| 2016-04-23 | 32 |
| 2016-04-24 | 32 |
| 2016-04-25 | 32 |
| 2016-04-26 | 32 |
| 2016-04-27 | 32 |
| 2016-04-28 | 32 |
| 2016-04-29 | 32 |
| 2016-04-30 | 31 |
| 2016-05-01 | 30 |
| 2016-05-02 | 29 |
| 2016-05-03 | 29 |
| 2016-05-04 | 29 |
| 2016-05-05 | 29 |
| 2016-05-06 | 29 |
| 2016-05-07 | 29 |
| 2016-05-08 | 27 |
| 2016-05-09 | 27 |
| 2016-05-10 | 26 |
| 2016-05-11 | 24 |
| 2016-05-12 | 21 |

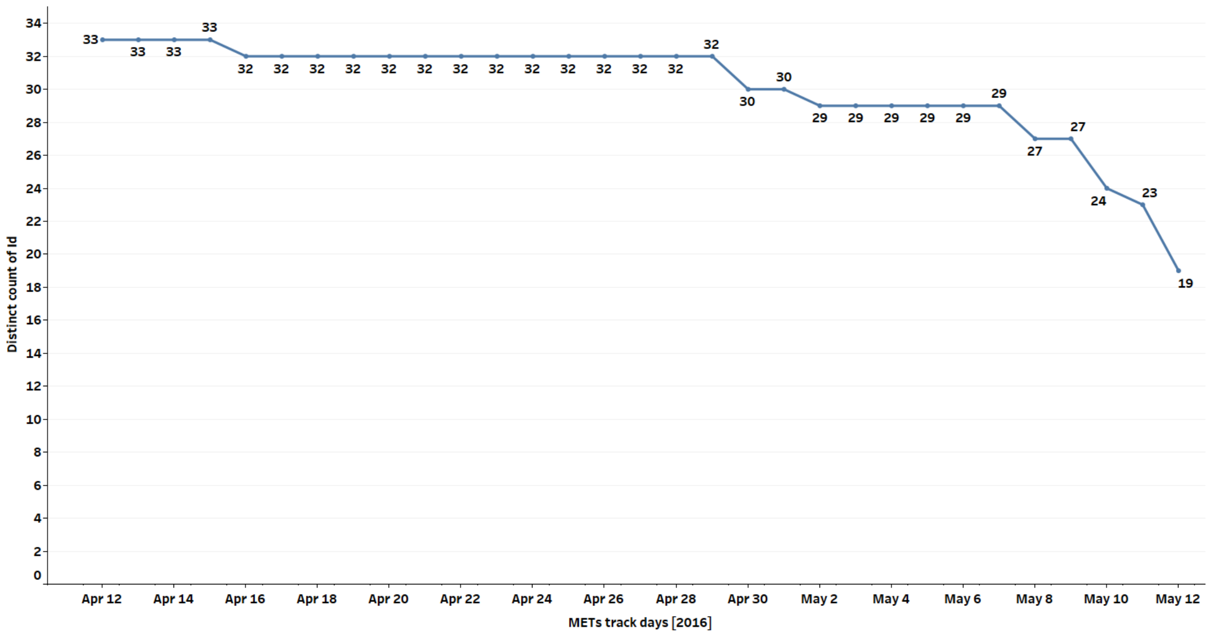


At the starting point of the tracking period, all 33 users tracked their activity. However, as time went by, user count decreased.

## METs

```sql
SELECT cast([ActivityMinute] as DATE) mets_date, count(distinct [Id])
id_count_per_day
FROM [dbo].[METs]
GROUP BY cast([ActivityMinute] as DATE)
ORDER BY cast([ActivityMinute] as DATE)
```

| mets_date | id_count_per_day |
|---|---|
| 2016-04-12 | 33 |
| 2016-04-13 | 33 |
| 2016-04-14 | 33 |
| 2016-04-15 | 33 |
| 2016-04-16 | 32 |
| 2016-04-17 | 32 |
| 2016-04-18 | 32 |
| 2016-04-19 | 32 |
| 2016-04-20 | 32 |
| 2016-04-21 | 32 |
| 2016-04-22 | 32 |
| 2016-04-23 | 32 |
| 2016-04-24 | 32 |
| 2016-04-25 | 32 |
| 2016-04-26 | 32 |
| 2016-04-27 | 32 |
| 2016-04-28 | 32 |
| 2016-04-29 | 32 |
| 2016-04-30 | 30 |
| 2016-05-01 | 30 |
| 2016-05-02 | 29 |
| 2016-05-03 | 29 |
| 2016-05-04 | 29 |
| 2016-05-05 | 29 |
| 2016-05-06 | 29 |
| 2016-05-07 | 29 |
| 2016-05-08 | 27 |
| 2016-05-09 | 27 |
| 2016-05-10 | 24 |
| 2016-05-11 | 23 |
| 2016-05-12 | 19 |



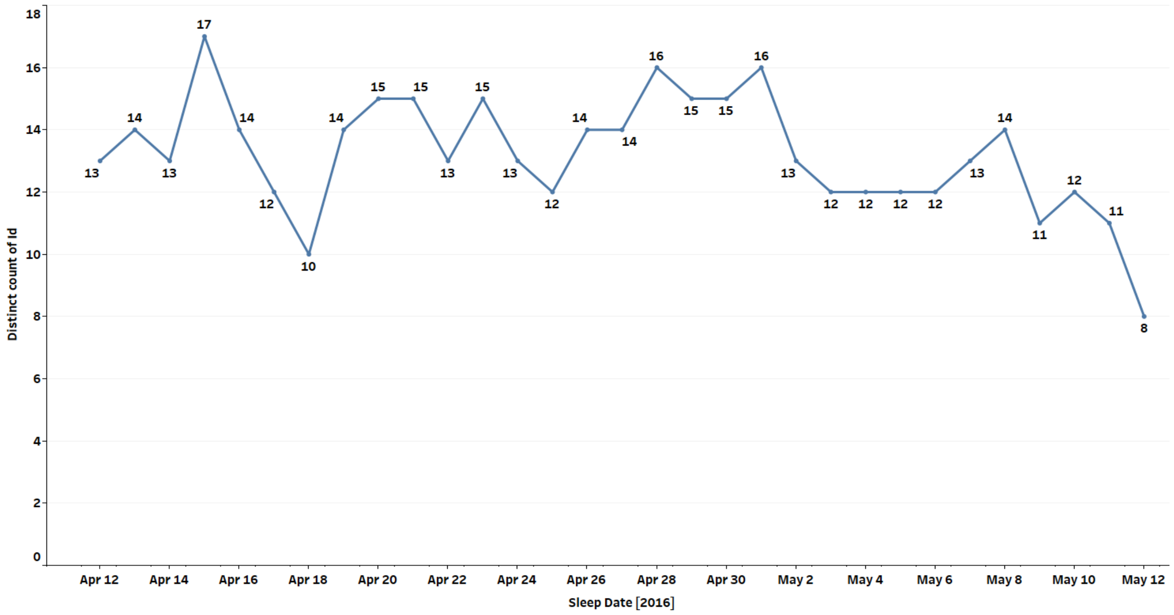
At the starting point of the tracking period, all 33 users tracked their METs. However, as time went by, user count decreased.

## Sleep

```
SELECT cast([SleepDay] as DATE) sleep_date, count(distinct [Id])
id_count_per_day
FROM [dbo].[sleepDay]
GROUP BY cast([SleepDay] as DATE)
ORDER BY cast([SleepDay] as DATE)
```

| sleep_date | id_count_per_day |
|---|---|
| 2016-04-12 | 13 |
| 2016-04-13 | 14 |
| 2016-04-14 | 13 |
| 2016-04-15 | 17 |
| 2016-04-16 | 14 |
| 2016-04-17 | 12 |
| 2016-04-18 | 10 |
| 2016-04-19 | 14 |
| 2016-04-20 | 15 |
| 2016-04-21 | 15 |
| 2016-04-22 | 13 |
| 2016-04-23 | 15 |
| 2016-04-24 | 13 |
| 2016-04-25 | 12 |
| 2016-04-26 | 14 |
| 2016-04-27 | 14 |
| 2016-04-28 | 16 |
| 2016-04-29 | 15 |

| | |
|---|---|
| 2016-04-30 | 15 |
| 2016-05-01 | 16 |
| 2016-05-02 | 13 |
| 2016-05-03 | 12 |
| 2016-05-04 | 12 |
| 2016-05-05 | 12 |
| 2016-05-06 | 12 |
| 2016-05-07 | 13 |
| 2016-05-08 | 14 |
| 2016-05-09 | 11 |
| 2016-05-10 | 12 |
| 2016-05-11 | 11 |
| 2016-05-12 | 8 |

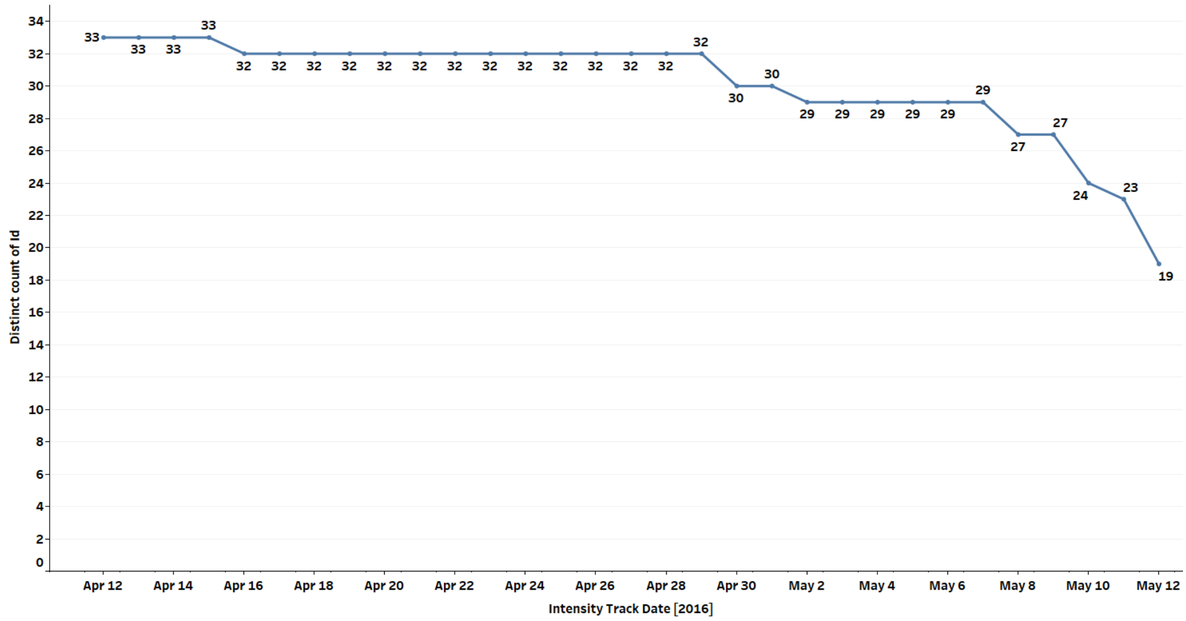


The total number of users who tracked their sleep is 24, however, it is evident that there is not even a single day where all users tracked their sleep. The tracking trend in sleep data is inconclusive.

## Intensity

```
SELECT cast([ActivityHour] as DATE) intensity_date, count(distinct [Id])
id_count_per_day
FROM [dbo].[hourlyIntensities]
GROUP BY cast([ActivityHour] as DATE)
ORDER BY cast([ActivityHour] as DATE)
```

| **intensity_date** | **id_count_per_day** |
|---|---|
| 2016-04-12 | 33 |
| 2016-04-13 | 33 |
| 2016-04-14 | 33 |
| 2016-04-15 | 33 |
| 2016-04-16 | 32 |
| 2016-04-17 | 32 |
| 2016-04-18 | 32 |
| 2016-04-19 | 32 |
| 2016-04-20 | 32 |
| 2016-04-21 | 32 |
| 2016-04-22 | 32 |
| 2016-04-23 | 32 |
| 2016-04-24 | 32 |
| 2016-04-25 | 32 |
| 2016-04-26 | 32 |
| 2016-04-27 | 32 |
| 2016-04-28 | 32 |
| 2016-04-29 | 32 |
| 2016-04-30 | 30 |
| 2016-05-01 | 30 |
| 2016-05-02 | 29 |
| 2016-05-03 | 29 |
| 2016-05-04 | 29 |
| 2016-05-05 | 29 |
| 2016-05-06 | 29 |
| 2016-05-07 | 29 |
| 2016-05-08 | 27 |
| 2016-05-09 | 27 |
| 2016-05-10 | 24 |
| 2016-05-11 | 23 |
| 2016-05-12 | 19 |



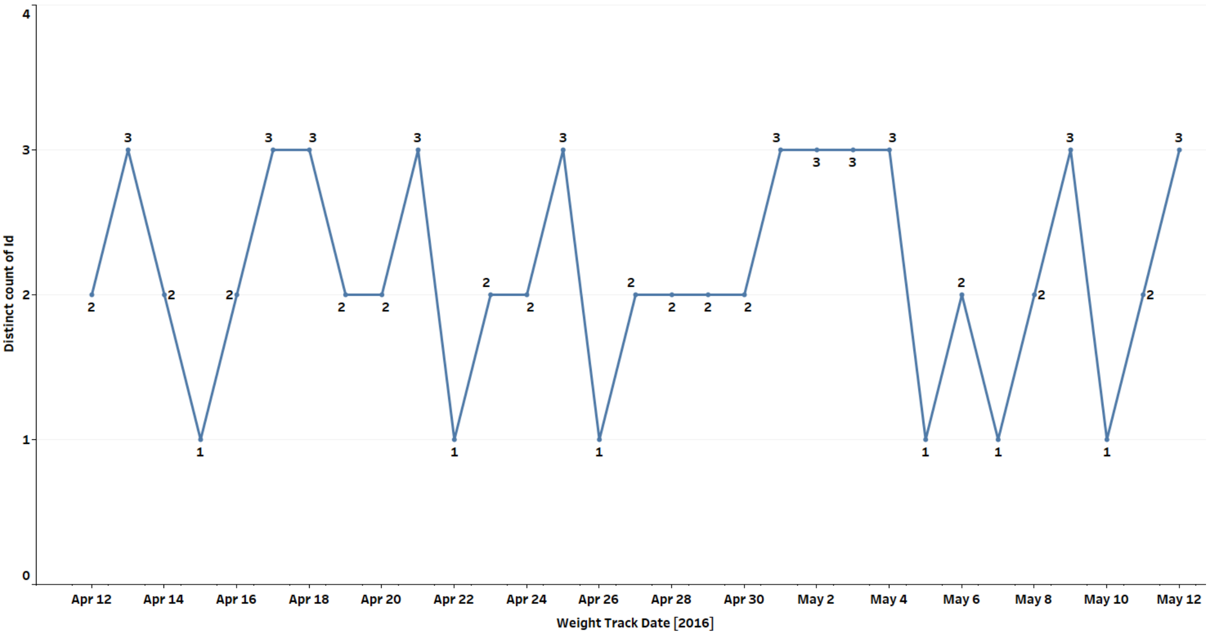
At the starting point of the tracking period, all 33 users tracked their Intensity. However, as time went by, user count decreased. It is worth noting that this trend is identical to the METs trend, which makes sense because METs are an indication of activity's intensity, which in turn indicates that METs were recorded at the same time as Intensity.

## Weight

```
SELECT cast([Date] as DATE) weight_date, count(distinct [Id]) id_count_per_day
FROM [dbo].[weight]
GROUP BY cast([Date] as DATE)
ORDET BY cast([Date] as DATE)
```

| weight_date | id_count_per_day |
|---|---|
| 2016-04-12 | 2 |
| 2016-04-13 | 3 |
| 2016-04-14 | 2 |
| 2016-04-15 | 1 |
| 2016-04-16 | 2 |
| 2016-04-17 | 3 |
| 2016-04-18 | 3 |
| 2016-04-19 | 2 |
| 2016-04-20 | 2 |
| 2016-04-21 | 3 |
| 2016-04-22 | 1 |
| 2016-04-23 | 2 |
| 2016-04-24 | 2 |
| 2016-04-25 | 3 |
| 2016-04-26 | 1 |
| 2016-04-27 | 2 |
| 2016-04-28 | 2 |

| | |
|---|---|
| 2016-04-29 | 2 |
| 2016-04-30 | 2 |
| 2016-05-01 | 3 |
| 2016-05-02 | 3 |
| 2016-05-03 | 3 |
| 2016-05-04 | 3 |
| 2016-05-05 | 1 |
| 2016-05-06 | 2 |
| 2016-05-07 | 1 |
| 2016-05-08 | 2 |
| 2016-05-09 | 3 |
| 2016-05-10 | 1 |
| 2016-05-11 | 2 |
| 2016-05-12 | 3 |



The total number of users who tracked their weight is 8, however, as displayed in the chart, the maximum count of users per day is 3. Also, it is evident that this line chart has no consistent trend.

### 5. The average number of calories burned by each user per day:

```
SELECT [Id], AVG([Calories]) avg_calories
FROM [dbo].[dailyActivity]
GROUP BY [Id]
ORDER BY avg_calories DESC
```

| Id | avg_calories |
|---|---|
| 8378563200 | 3436 |
| 8877689391 | 3420 |
| 5577150313 | 3359 |
| 4388161847 | 3093 |
| 4702921684 | 2965 |
| 8053475328 | 2945 |
| 1644430081 | 2811 |
| 8583815059 | 2732 |
| 6290855005 | 2599 |
| 7086361926 | 2566 |
| 7007744171 | 2544 |
| 2022484408 | 2509 |
| 4020332650 | 2385 |
| 6117666160 | 2261 |
| 4445114986 | 2186 |
| 1927972279 | 2172 |
| 6775888955 | 2131 |
| 2347167796 | 2043 |
| 4319703577 | 2037 |
| 4558609924 | 2033 |
| 6962181067 | 1982 |
| 4057192912 | 1973 |
| 8792009665 | 1962 |
| 3372868164 | 1933 |
| 2873212765 | 1916 |
| 5553957443 | 1875 |
| 1503960366 | 1816 |
| 8253242879 | 1788 |
| 2320127002 | 1724 |
| 1844505072 | 1573 |
| 2026352035 | 1540 |
| 3977333714 | 1513 |
| 1624580081 | 1483 |

**Minimum and maximum average calories burned per day:**

**Option 1:**

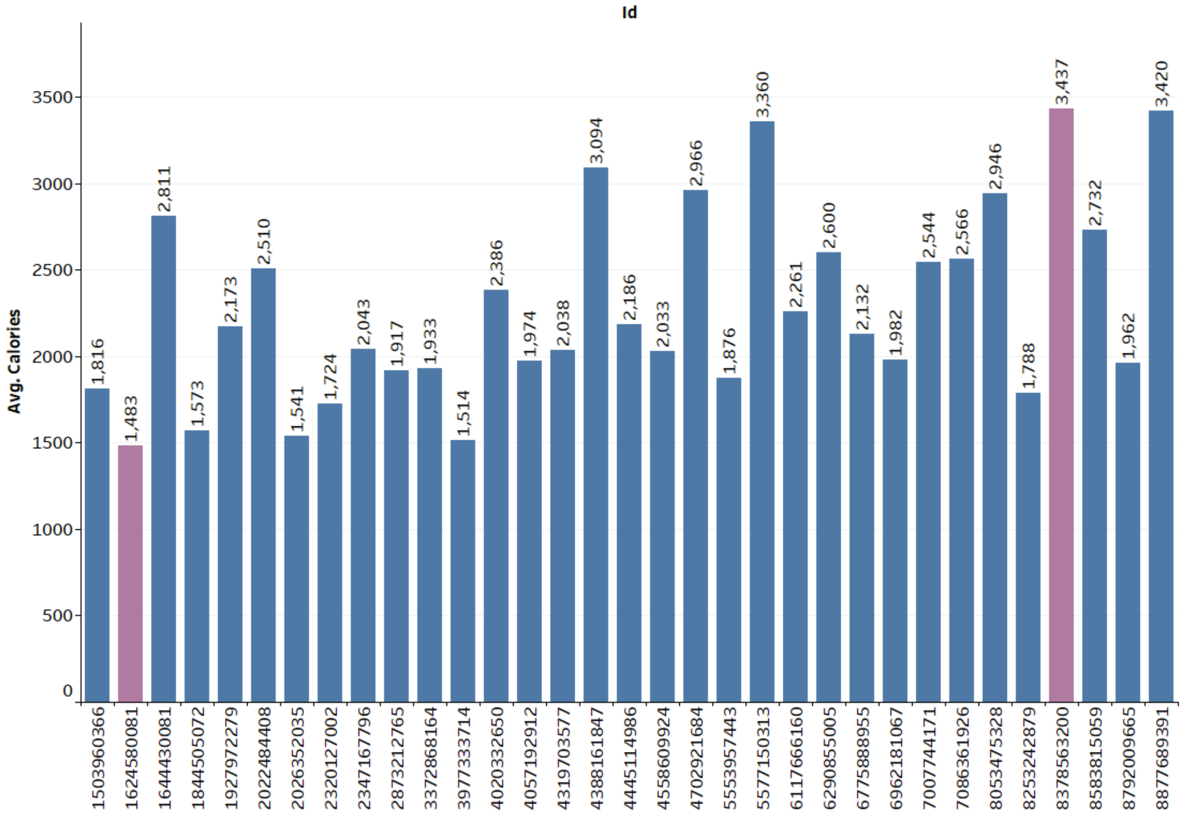
```
SELECT MIN(avg_calories) min_avg_cal, MAX(avg_calories) max_avg_cal
FROM
      (SELECT [Id], AVG([Calories]) avg_calories
      FROM [dbo].[dailyActivity]
      GROUP BY [Id]) avg_cal_t
```

**Option 2:**

```
WITH avg_cal_t
AS
(
      SELECT [Id], AVG([Calories]) avg_calories
      FROM [dbo].[dailyActivity]
      GROUP BY [Id])

SELECT MIN(avg_calories) min_avg_cal, MAX(avg_calories) max_avg_cal
FROM avg_cal_t
```

| min_avg_cal | max_avg_cal |
|---|---|
| 1,483 | 3,436 |

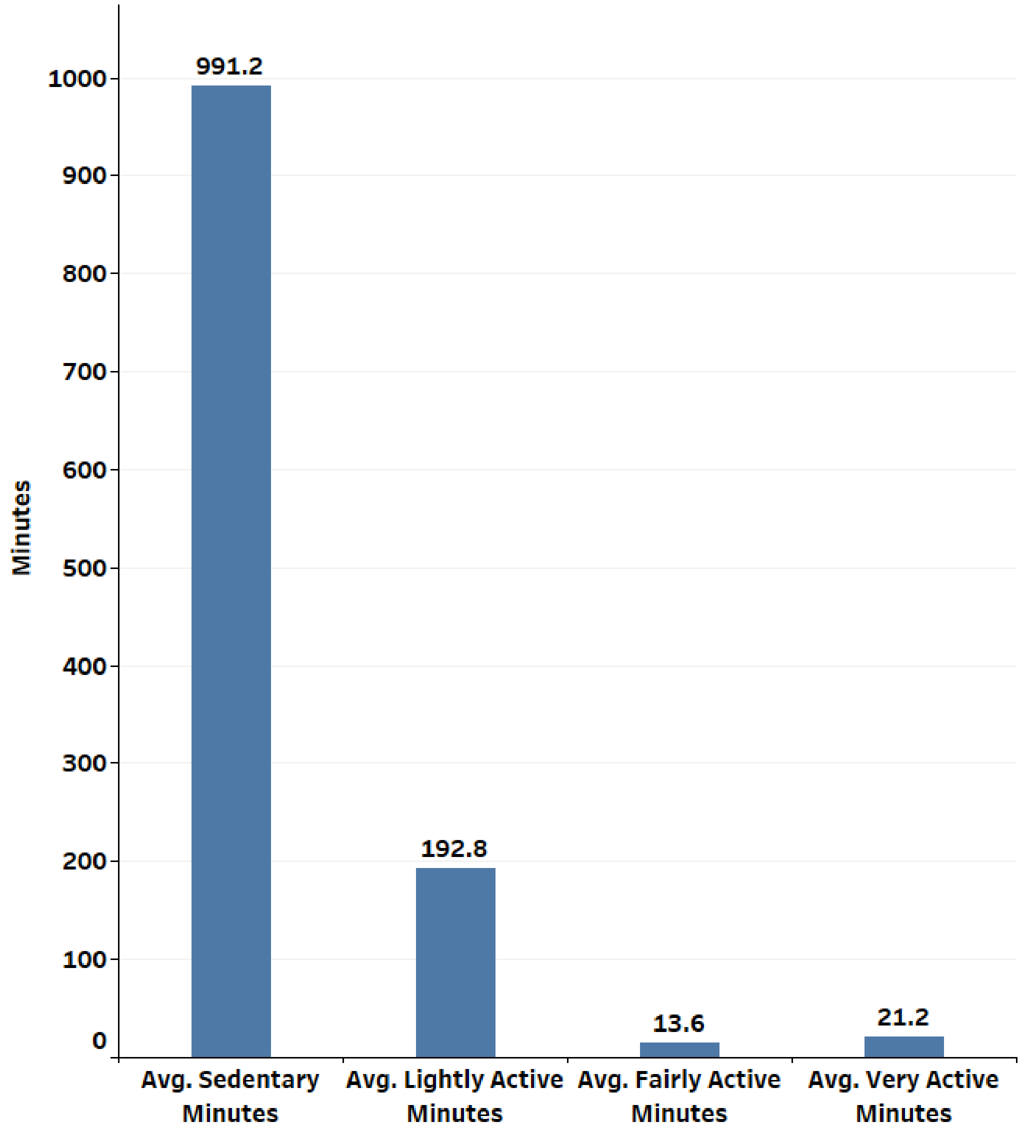


This calorie burn bar chart includes not only the activity, but also calories burned for basic bodily functions, such as breathing, circulating blood, and more.

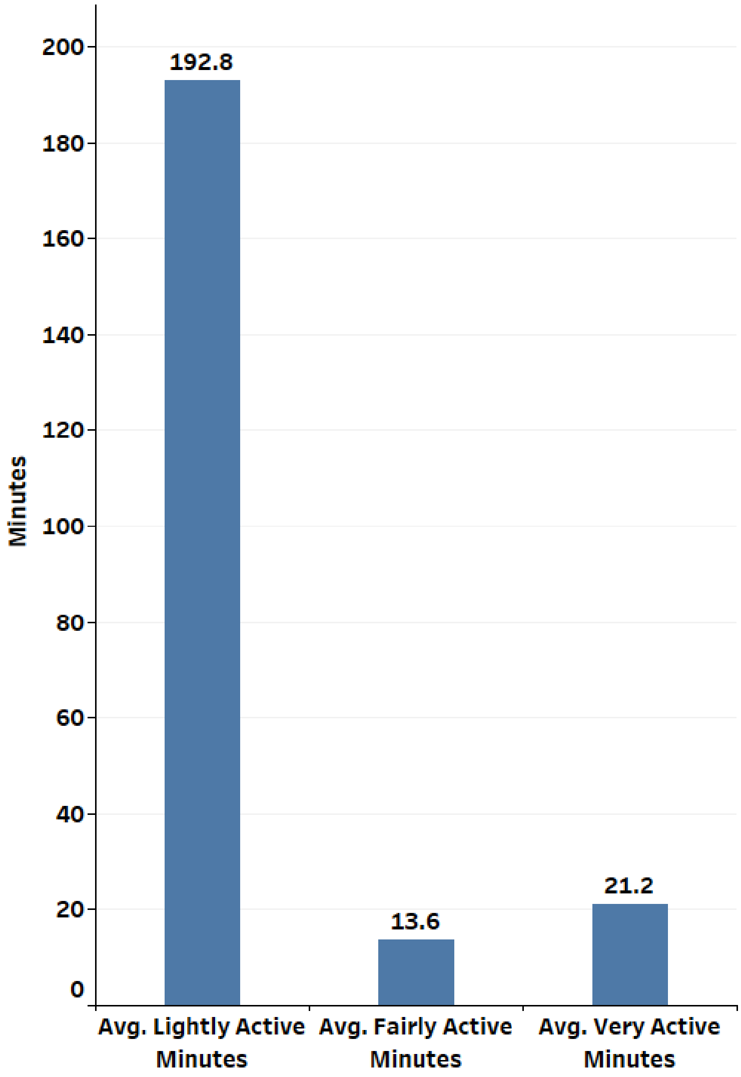
### 6. The average time users spend on each activity (very active, fairly active, lightly active, and sedentary activity)

```
SELECT
      AVG([VeryActiveMinutes]) avg_very_act_min,
      AVG([FairlyActiveMinutes]) avg_fairly_act_min,
      AVG([LightlyActiveMinutes])avg_light_act_min,
      AVG([SedentaryMinutes]) avg_sed_act_min
FROM [dbo].[dailyActivity]
```

| avg_very_act_min | avg_fairly_act_min | avg_light_act_min | avg_sed_act_min |
|---|---|---|---|
| 21.2 | 13.6 | 192.8 | 991.2 |



Overall, users spend more time on sedentary activities. This result is expected since fitbit, as long as being on, always tracking user's activity, even if the user is asleep. More interesting is to examine the other groups (lightly active, fairly active, and very active).



On average users spend more time on light activities, then on very active activities, and only lastly on fairly active activities.

(The range of intensities: very active activity > fairly active activity > light activity > sedentary activity)

### 7. Average steps, distance, and calories

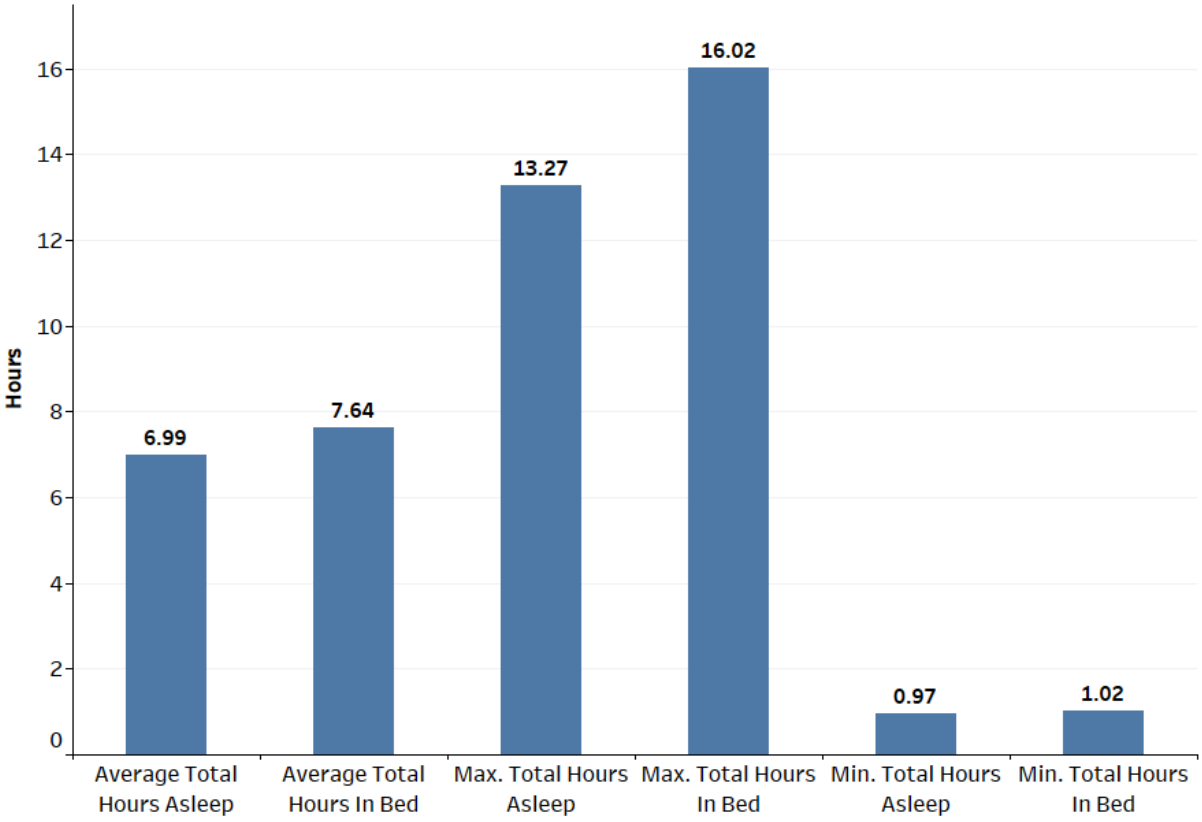
```
SELECT
	AVG([TotalSteps]) avg_total_steps,
	ROUND(AVG([TotalDistance]),2) avg_total_dis,
	AVG([Calories]) avg_clories
FROM [dbo].[dailyActivity]
```

| avg_total_steps | avg_total_dis [km] | avg_clories |
|---|---|---|
| 7637 | 5.49 | 2303 |

### 8. Analyzing sleeping data

```
SELECT
	COUNT (DISTINCT [Id]) sleep_user_count,
	AVG([TotalMinutesAsleep])/60.0 avg_sleep_hour,
	MIN([TotalMinutesAsleep]) min_sleep_minutes,
	MAX([TotalMinutesAsleep])/60.0 max_sleep_hour,
	AVG([TotalTimeInBed])/60.0 avg_in_bed_hour,
	MIN([TotalTimeInBed]) min_in_bed_minutes,
	MAX([TotalTimeInBed])/60.0 max_in_bed_hour
FROM [dbo].[sleepDay]
```

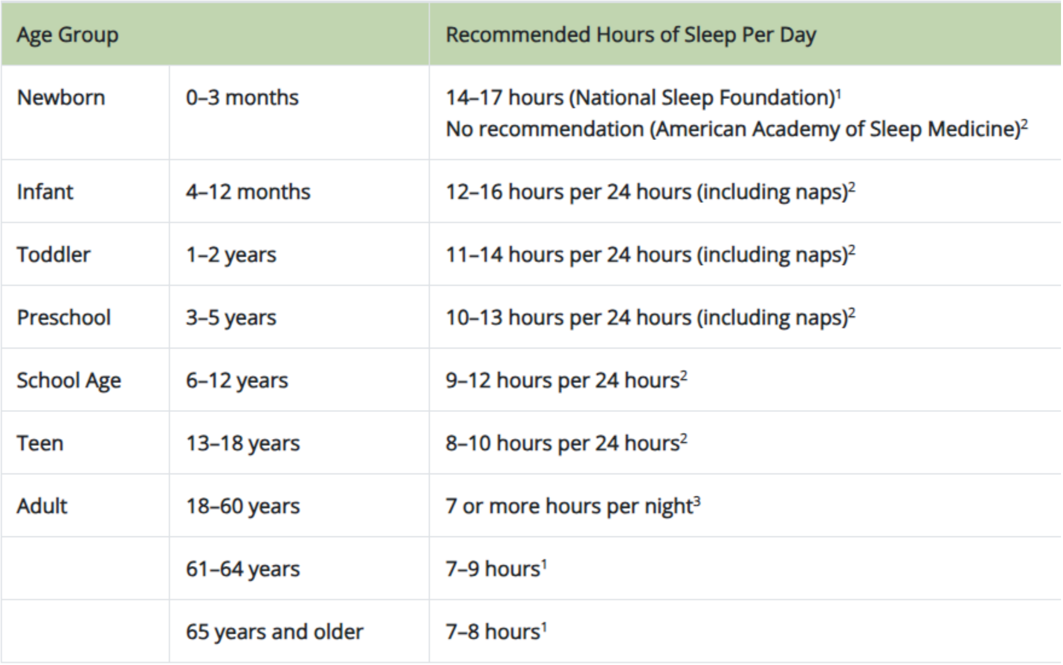
| sleep_user_count | avg_sleep_hour | min_sleep_minutes | max_sleep_hour | avg_in_bed_hour | min_in_bed_minutes | max_in_bed_hour |
|---|---|---|---|---|---|---|
| 24 | 6.983333 | 58 | 13.266666 | 7.633333 | 61 | 16.016666 |



Note that total hours in bed include the total hours asleep.

The following table* summarized the recommended hours of sleep per day:

*National Center for Chronic Disease Prevention and Health Promotion, Division of Population Health, September 14, 2022, accessed 14 September 2023, <https://www.cdc.gov/sleep/about_sleep/how_much_sleep.html>



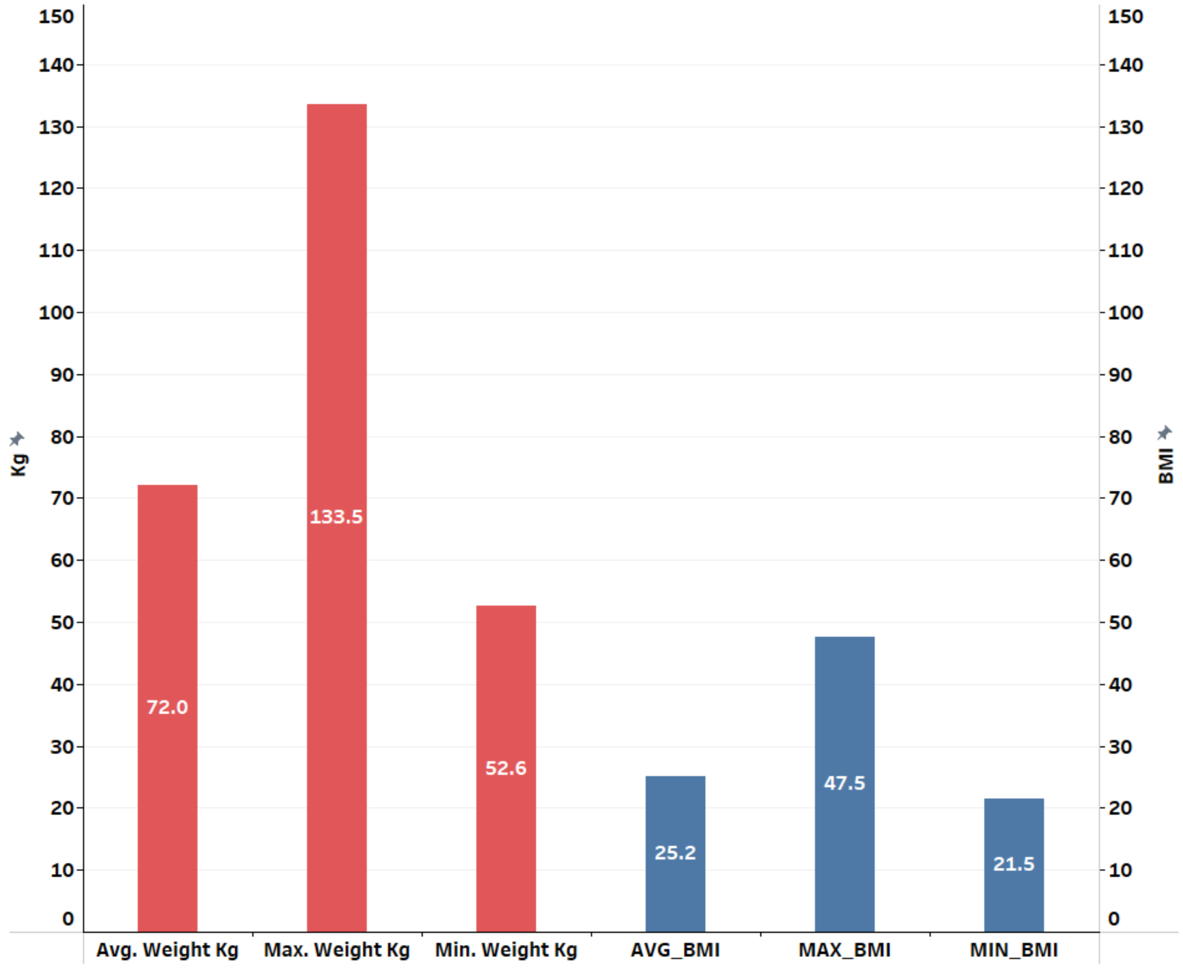
| Age Group | | Recommended Hours of Sleep Per Day |
|---|---|---|
| Newborn | 0–3 months | 14–17 hours (National Sleep Foundation)[1]<br>No recommendation (American Academy of Sleep Medicine)[2] |
| Infant | 4–12 months | 12–16 hours per 24 hours (including naps)[2] |
| Toddler | 1–2 years | 11–14 hours per 24 hours (including naps)[2] |
| Preschool | 3–5 years | 10–13 hours per 24 hours (including naps)[2] |
| School Age | 6–12 years | 9–12 hours per 24 hours[2] |
| Teen | 13–18 years | 8–10 hours per 24 hours[2] |
| Adult | 18–60 years | 7 or more hours per night[3] |
| | 61–64 years | 7–9 hours[1] |
| | 65 years and older | 7–8 hours[1] |

The average total hours asleep is approximately as recommended for an adult group. Total hours in bed include the total hours asleep, therefore, it is always bypassing the latter. The minimum and maximum total hours asleep/in bed present the extreme cases which do not project on other users' behavior.

## 9. Analyzing weight data

```
SELECT
	COUNT(DISTINCT [Id]) weight_user_count,
	ROUND(AVG([WeightKg]),2) avg_weight_kg,
	ROUND(AVG([BMI]),2) avg_bmi,
	ROUND(MIN([WeightKg]),2) min_weight_kg,
	ROUND(MAX([WeightKg]),2) max_weight_kg,
	ROUND(MIN([BMI]),2) min_bmi,
	ROUND(MAX([BMI]),2)	max_bmi
FROM [dbo].[weight]
```

| weight_user_count | avg_weight_kg | avg_bmi | min_weight_kg | max_weight_kg | min_bmi | max_bmi |
|---|---|---|---|---|---|---|
| 8 | 72.04 | 25.19 | 52.6 | 133.5 | 21.45 | 47.54 |



## BMI ranges

- **Underweight:** BMI is less than 18.5
- **Normal weight:** BMI is 18.5 to 24.9
- **Overweight:** BMI is 25 to 29.9
- **Obese:** BMI is 30 or more

## BMI

The maximum value is of an obese user (47.54 BMI) and the minimum value (21.45 BMI) is within the range of normal weight, however, the average BMI (25.19 BMI) is within the range of the overweight values. It is important to note that averages are greatly influenced by extreme values.

## Weight in Kg

The calculated weights are in accordance with the BMI values.

## Examination of each user contribution to the average BMI and weight

```
SELECT
	[Id],
	ROUND(AVG([BMI]),2) avg_bmi,
	ROUND(AVG([WeightKg]),2) avg_weight_kg
FROM [dbo].[weight]
GROUP BY [Id]
```

| Id | avg_bmi | avg_weight_kg |
|---|---|---|
| 1503960366 | 22.65 | 52.6 |
| 1927972279 | 47.54 | 133.5 |
| 2873212765 | 21.57 | 57 |
| 4319703577 | 27.41 | 72.35 |
| 4558609924 | 27.21 | 69.64 |
| 5577150313 | 28 | 90.7 |
| 6962181067 | 24.03 | 61.55 |
| 8877689391 | 25.49 | 85.15 |

It is evident that most users are in the overweight range, however, this data is not enough to make a general conclusion on Fitbit users due to its small number of users.

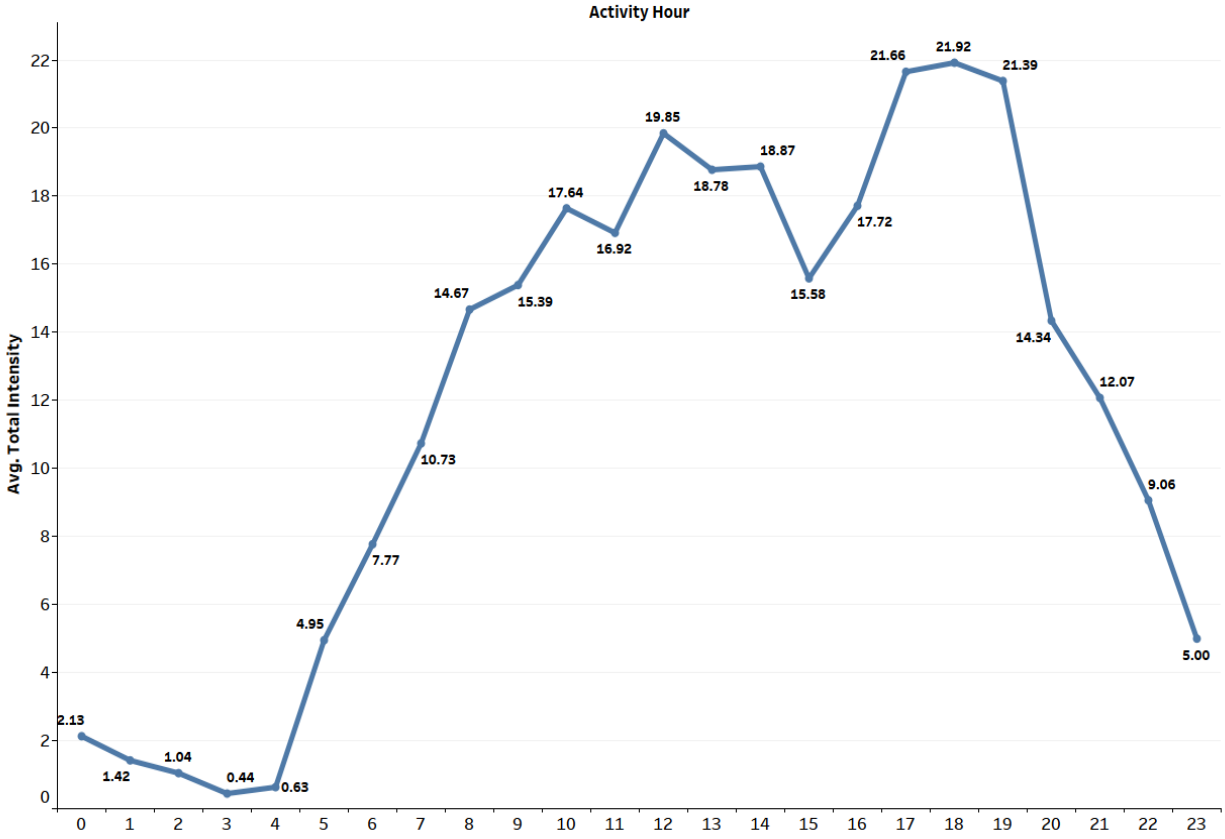
### 10. Time when users are most active.

**Option 1**

```
WITH total_avg_int as(
SELECT
	distinct CAST([ActivityHour] AS TIME) act_time,
	AVG([TotalIntensity]) over(partition by
DATEPART(hour,[ActivityHour])) avg_hour_act
FROM [dbo].[hourlyIntensities])

SELECT
	distinct top 1 act_time,
	MAX(avg_hour_act) over(partition by act_time) as max_act
FROM total_avg_int
ORDER BY max_act DESC
```

| act_time | max_act |
|---|---|
| 18:00:00.0000000 | 21.9216335540839 |



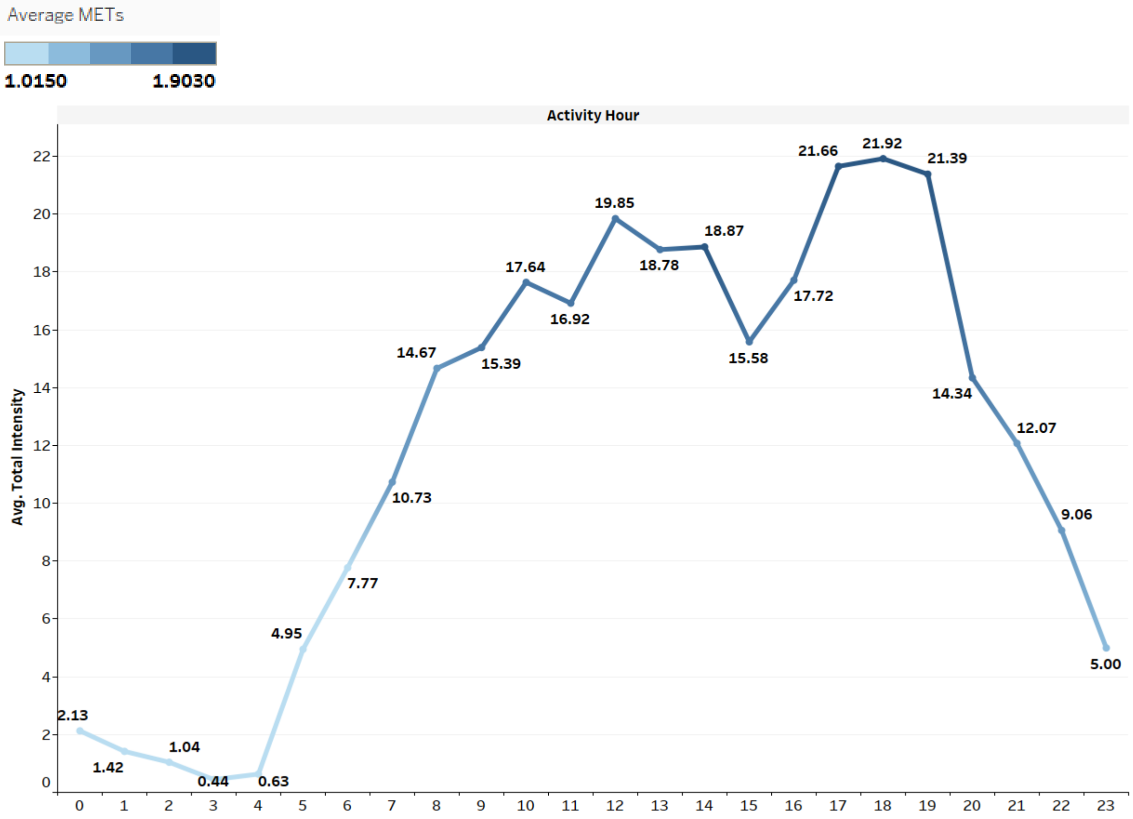
The line chart above shows average intensities during the day over a month period. The highest average intensity was found to be at 18 o'clock, however 12 o'clock is also worth mentioning. It can be assumed that 18 o'clock is the time when most users engage in non-sedentary activities.

## Joined METs Dataset with Intensity Dataset

In this option, METs and Intensity were joined together to understand the relation of METs (metabolic equivalent of task) to the recorded intensities. METs can be useful as rating physical activity to indicate its intensity, therefore the representation of both on the same line chart can visualy show how these two paremeters relate to one another.

```
SELECT
      DISTINCT TOP 1 CAST([ActivityHour] AS TIME) act_time,
      AVG([TotalIntensity]) OVER(PARTITION BY
      DATEPART(hour,[ActivityHour])) avg_act,
      AVG([METs]/10.0) OVER(PARTITION BY DATEPART(hour,[ActivityHour]))
avg_mets
FROM  [dbo].[hourlyIntensities] AS hour_act
JOIN [dbo].[METs] as met
ON hour_act.[Id] = met.[Id] AND hour_act.[ActivityHour] =
met.[ActivityMinute]
ORDER BY avg_act DESC
```

| act_time | avg_act | avg_mets |
|---|---|---|
| 18:00:00.0000000 | 21.9216335540839 | 1.858498 |



## Summary

Firstly, it is worth mentioning that the following summary refers to only 33 users. Extra data might reveal different insights.

- Sleep and Weight seems to be the least preferable parameters to be tracked by Fitbit users.
- In the beginning of the tracking period, all users tracked Activity, Intensity, and METs, however user count kept on decreasing. The reason for this decrease is not clear.
- Sleep and Weight were tracked by fewer users. There was not a single day where all users tracked their parameters.
- Fitbit users walk an average distance of 5.49 km, lose 2,303 calories, and walk 7,637 steps per day.
- Most Fitbit users engage in light activities.
- Sleep data revealed that Fitbit users sleep a recommended number of hours.
- Fitbit users who track their weight are mostly overweight.
- Fitbit's user preferable times to engage with non-sedentary activities are at 12 and 18 o'clock on average.

## Further Investigation

- Check how many users consistently tracked their parameters during the given period.
- Which activity is better for burning calories.