Case Study 2: How Can a Wellness Tech Co Play It Smart?

Antoinette de Janasz

2024-06-28

Bellabeat is a leading Silicon Valley fem-tech business targeting a global health-conscious community. Founded in 2014 by Urška Sršen and Sandro Mur, Bellabeat developed one of the first wearables specifically designed for women. Since then their products now include:

- Bellabeat app - provides health data related to activity, sleep, stress, menstruation, and mindfulness

- Leaf - a wellness tracker as a bracelet, necklace or clip
- Time - a wellness watch
- Spring - a water bottle to track hydration
- Bellabeat membership - a subscription-based membership

Objective

Analyze smart device data to gain insight into how consumers are using their smart devices. Then, using this information, provide recommendations on how these trends can inform Bellabeat's marketing strategy.

Data & Tools

The data used was "Fitbit Fitness Tracker Data" (CCO: Public Domain, made available through Mobius) and downloaded from Kaggle. The data contained two datasets of personal tracker data from 30+ Fitbit users who consented to the use of their information. The datasets include information about daily activity, steps, sleep monitoring, heart rates, and weight logs. Though the datasets were eight years old, they will serve for this project. The files are stored on my personal computer behind a firewall, inaccessible to all but myself. The Fitbit users' identities are protected by assigned numbers.

I merged the 3.12.16-4.11.16 and the 4.12.16-5.12.16 datasets to get a better representation of the data. As several of the datasets were very large, I used RStudio to process & prepare the data.

The RStudio packages I used were:

- tidyverse
- janitor
- lubridate
- ggplot2

I supplemented the data with Internet research. Sources include: BusinessofApps.com, NCHS Health E-Stats, the US Census Bureau, Fitbit.com, Bellabeat.com. Some of the information I obtained from these sites was used to create my own datasets using Excel to create spreadsheets and visuals.

Processing

Since there was limited data I felt it was necessary to merge the two datasets. I used these .csv datasets:

- "dailyActivity_merged"
- "dailyActivity_merged_2"
- "weightLogInfo_merged"
- "weightLogInfo_merged_2"
- "heartrate_seconds_merged"
- "heartrate_seconds_merged_2"
- "sleepDay_merged"

daily_activity_total

This Fitbit feature used the metrics: id#, date, daily number of steps, distance, tracker distance (seemed redundant), logged distance, very active distance, moderately active distance, light active distance, sedentary active distance (oxymoron?), very active minutes, fairly active minutes, lightly active minutes, sedentary minutes, and calories.

Reviewed 'dailyActivity_merged' & 'dailyActivity_merged_2' column names to verify matching names:

```r
head(dailyActivity_merged)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    3/25/2016      11004          7.11            7.11
## 2 1503960366    3/26/2016      17609         11.55           11.55
## 3 1503960366    3/27/2016      12736          8.53            8.53
## 4 1503960366    3/28/2016      13231          8.93            8.93
## 5 1503960366    3/29/2016      12041          7.85            7.85
## 6 1503960366    3/30/2016      10970          7.16            7.16
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               2.57                     0.46
## 2                        0               6.92                     0.73
## 3                        0               4.66                     0.16
## 4                        0               3.19                     0.79
## 5                        0               2.16                     1.09
## 6                        0               2.36                     0.51
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                4.07                       0                33
## 2                3.91                       0                89
## 3                3.71                       0                56
## 4                4.95                       0                39
## 5                4.61                       0                28
## 6                4.29                       0                30
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  12                  205              804     1819
## 2                  17                  274              588     2154
## 3                   5                  268              605     1944
## 4                  20                  224             1080     1932
## 5                  28                  243              763     1886
## 6                  13                  223             1174     1820
```

```r
head(dailyActivity_merged_2)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 2 1503960366    4/13/2016      10735          6.97            6.97
## 3 1503960366    4/14/2016      10460          6.74            6.74
## 4 1503960366    4/15/2016       9762          6.28            6.28
## 5 1503960366    4/16/2016      12669          8.16            8.16
## 6 1503960366    4/17/2016       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
```

```
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
```

Merged datasets together using merge() and saved as a data frame.

```
daily_activity_total <- merge(dailyActivity_merged,dailyActivity_merged_2, all=TRUE)
```

```
str(daily_activity_total)
```

```
## 'data.frame':    1397 obs. of  15 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr  "3/25/2016" "3/26/2016" "3/27/2016" "3/28/2016" ...
##  $ TotalSteps              : int  11004 17609 12736 13231 12041 10970 12256 12262 10057 10990 ...
##  $ TotalDistance           : num  7.11 11.55 8.53 8.93 7.85 ...
##  $ TrackerDistance         : num  7.11 11.55 8.53 8.93 7.85 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  2.57 6.92 4.66 3.19 2.16 ...
##  $ ModeratelyActiveDistance: num  0.46 0.73 0.16 0.79 1.09 ...
##  $ LightActiveDistance     : num  4.07 3.91 3.71 4.95 4.61 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  33 89 56 39 28 30 33 47 44 26 ...
##  $ FairlyActiveMinutes     : int  12 17 5 20 28 13 12 21 13 14 ...
##  $ LightlyActiveMinutes    : int  205 274 268 224 243 223 239 200 168 216 ...
##  $ SedentaryMinutes        : int  804 588 605 1080 763 1174 820 866 737 855 ...
##  $ Calories                : int  1819 2154 1944 1932 1886 1820 1889 1868 1755 1811 ...
```

Calculate total number of days in the new data frame.

```
difftime("2016-3-12", "2016-5-9", units = "days")
```

```
## Time difference of -58 days
```

The 'LoggedActivitiesDistance' and 'SedentaryActiveDistance' columns were mostly 0 values and therefore removed.

```
no_date <- daily_activity_total$LoggedActivitiesDistance <- NULL
no_date <- daily_activity_total$SedentaryActiveDistance <- NULL
```

Count the unique users.

```
daily_activity_total %>%
  count(Id, sort=TRUE)
```

```
##            Id  n
## 1  4020332650 63
## 2  1503960366 50
## 3  1624580081 50
## 4  4445114986 46
## 5  4702921684 46
## 6  6962181067 45
## 7  1844505072 43
## 8  1927972279 43
## 9  2022484408 43
## 10 2026352035 43
```

```
## 11 2320127002 43
## 12 2873212765 43
## 13 4319703577 43
## 14 4558609924 43
## 15 5553957443 43
## 16 7086361926 43
## 17 8378563200 43
## 18 8877689391 43
## 19 3977333714 42
## 20 8053475328 42
## 21 5577150313 41
## 22 8792009665 41
## 23 1644430081 40
## 24 4388161847 39
## 25 6290855005 39
## 26 8583815059 39
## 27 6117666160 38
## 28 7007744171 38
## 29 4057192912 36
## 30 6775888955 35
## 31 2347167796 33
## 32 8253242879 31
## 33 3372868164 30
## 34 6391747486  9
## 35 2891001357  8
```

Created a data frame to get the sums of each activity grouped by the Id.

```
daily_activity_total %>%
   group_by(Id) %>%
  summarise(across(where(is.numeric), sum))
```

```
## # A tibble: 35 x 12
##            Id TotalSteps TotalDistance TrackerDistance VeryActiveDistance
##         <dbl>      <int>         <dbl>           <dbl>              <dbl>
##  1 1503960366     596789         387.            387.             142.
##  2 1624580081     258360         174.            174.              30.1
##  3 1644430081     311237         226.            226.              33.2
##  4 1844505072     123669          81.8            81.8              0.870
##  5 1927972279      54570          37.8            37.8              2.97
##  6 2022484408     498589         356.            356.             112.
##  7 2026352035     213286         132.            132.               0.190
##  8 2320127002     183884         124.            124.               4.26
##  9 2347167796     318355         212.            212.              32.0
## 10 2873212765     313868         212.            212.              24.9
## # i 25 more rows
## # i 7 more variables: ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, VeryActiveMinutes <int>,
## #   FairlyActiveMinutes <int>, LightlyActiveMinutes <int>,
## #   SedentaryMinutes <int>, Calories <int>
```

A new data frame was created to get the average of the various activities grouped by user, then round the numbers.

```
average_by_user <- daily_activity_total %>%
  group_by(Id) %>%
```

```
  summarise_at(.vars = c("TotalSteps", "TrackerDistance", "VeryActiveMinutes", "FairlyActiveMinutes", "I
mutate(across(where(is.numeric), ~round(., 0)))
```

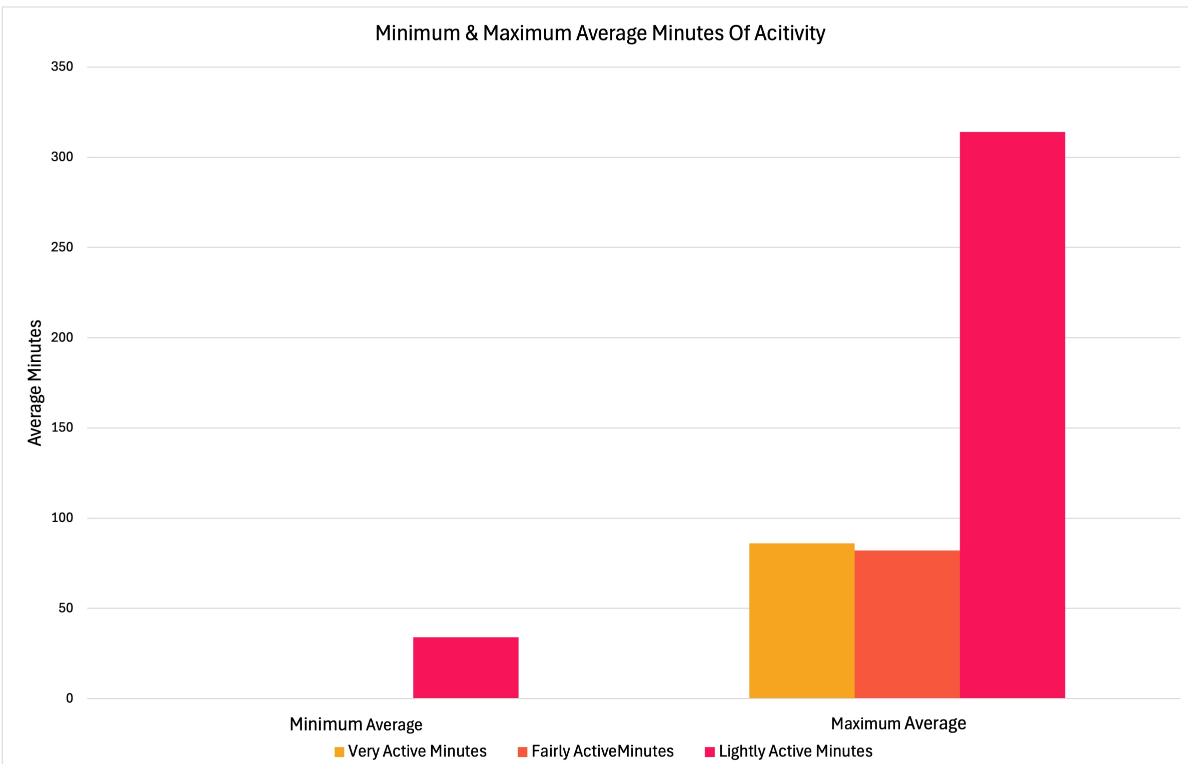
Determine minimums and maximums.

```
user_min <- sapply(average_by_user,min)
```

```
user_max <- sapply(average_by_user,max)
```

	Id #	Total Steps	Tracker Distance	V. Active Minutes	F. Active Minutes	L. Active Minutes	Calories
Min. Avg.	1503960366	774	0	0	0	34	1434
Max. Avg.	8877689391	16424	13	86	82	314	3429

I used the time metric to determine how long the users spent on physical activity. I also reduced it to the average amount of time spent: 'VeryActiveMinutes', 'FairlyActiveMinutes', and 'LightlyActiveMinutes', then took the maximum and minimum times of the metrics. Considering that this is measured in *minutes,* there is not a vast difference between the average maximum times and the average minimum times. This is based on 35 Fitbit users over a span of 58 days.



weight_log

31.43% of users tracked their weight. The percentage is low possibly due to the majority of data had to be put in manually. Metrics tracked were: id#, date & time, weight in kilograms and in pounds, fat, body mass index (BMI), manually reported, and a log id. The log id was different from the user id and changed with every entry of the other metrics. It's not clear what the log id was tracking.

Given that obesity is a serious health concern in the United States, I am surprised this is not more of a focus for Fitbit. And, as of 2024, Bellabeat has not incorporated weight tracking into their products.

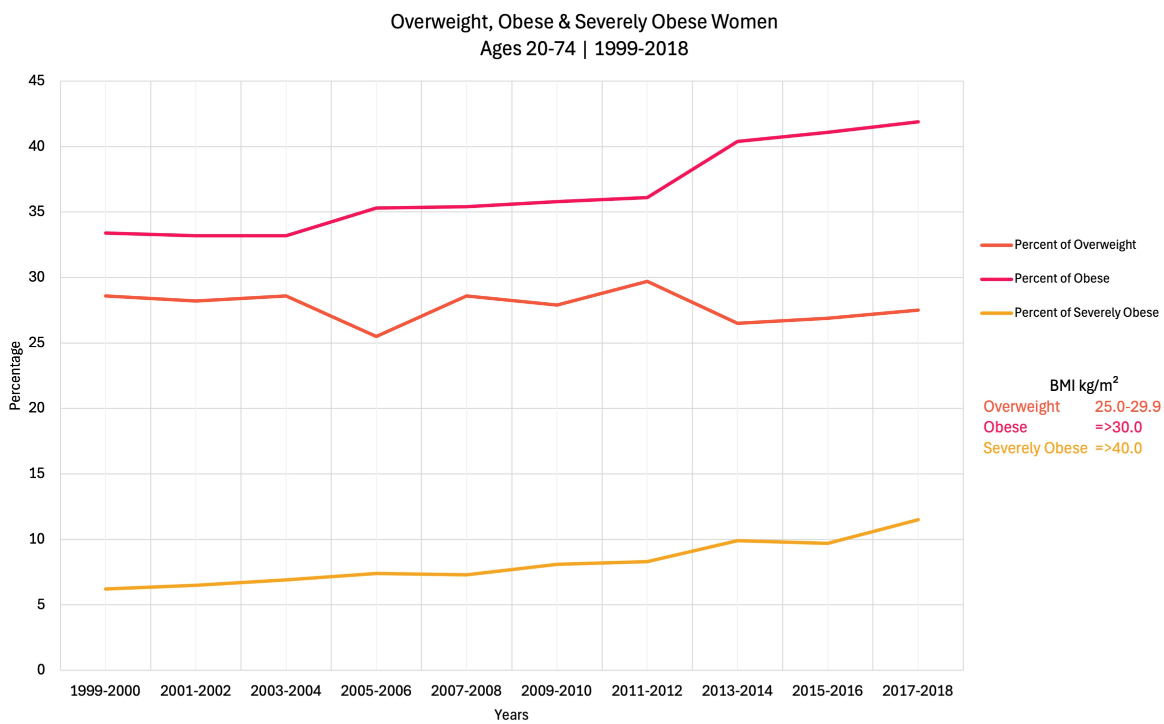


Figure 1: *Fryar CD, Carroll MD, Afful J. Prevelance of overweight, obesity, and severe obesity among adults aged 20 and over.United States, 1960-*

Reviewed 'weightLogInfo_merged' & 'weightLogInfo_merged_2' column names to verify matching names:

```
colnames(weightLogInfo_merged)
```

```
## [1] "Id"             "Date"            "WeightKg"        "WeightPounds"
## [5] "Fat"            "BMI"             "IsManualReport" "LogId"
```

```
colnames(weightLogInfo_merged_2)
```

```
## [1] "Id"             "Date"            "WeightKg"        "WeightPounds"
## [5] "Fat"            "BMI"             "IsManualReport" "LogId"
```

Merged these datasets together using merge() and saved as a data frame.

```
weight_log <- merge(weightLogInfo_merged,weightLogInfo_merged_2, all=TRUE)
```

```
str(weight_log)
```

```
## 'data.frame':    33 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.93e+09 2.35e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "4/5/2016 11:59:59 PM" "4/10/2016 6:33:26 PM" "4/3/2016 11:59:59 PM" "4/6/20
##  $ WeightKg      : num  53.3 129.6 63.4 56.7 57.2 ...
##  $ WeightPounds  : num  118 286 140 125 126 ...
##  $ Fat           : int  22 NA 10 NA NA NA NA NA NA NA ...
##  $ BMI           : num  23 46.2 24.8 21.5 21.6 ...
##  $ IsManualReport: chr  "True" "False" "True" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

The original column, 'IsManualReport', was in a logical format. I changed the format from logical to numerical: 1=TRUE, 0=FALSE to make it easier to calculate.

```
weight_log$IsManualReport <- as.integer(as.logical(weight_log$IsManualReport))
```

```
distinct_users_by_reporttype <- weight_log %>%
count(Id, IsManualReport, sort = TRUE)
```

Create a table to reflect how many total uses and whether or not they were manually loaded into their Fitbit.

```
attr(distinct_users_by_reporttype$Id, "label") <- "Fitbit user ID#"
attr(distinct_users_by_reporttype$IsManualReport, "label") <- "info manually loaded, 1=YES, 0=NO"
attr(distinct_users_by_reporttype$n, "label") <- "number of uses"
view(distinct_users_by_reporttype)
```

#Weight Log Users over 58 Days

User Id	Total Number of Uses	Manually Loaded
6962181067	14	Yes
8877689391	9	No
2873212765	2	Yes
1503960366	1	Yes
1927972279	1	No
2347167796	1	Yes
2891001357	1	Yes
4445114986	1	Yes
4558609924	1	Yes
4702921684	1	Yes
8253242879	1	Yes

heartrate_seconds

The heartrate feature was used by less than half of the Fitbit users. It measured three variables: id, date & time, and the heartrate. The heartrate was measured every 5 seconds.

Reviewed 'heartrate_seconds_merged' & 'heartrate_seconds_merged_2' column names to verify matching names:

```
colnames(heartrate_seconds_merged)
```

```
## [1] "Id"    "Time"  "Value"
```

```
colnames(heartrate_seconds_merged_2)
```

```
## [1] "Id"    "Time"  "Value"
```

Count the number of distinct id#s for each dataset.

```
heartrate_by_user1 <- heartrate_seconds_merged %>%
  count(Id, sort = TRUE)
```

```
heartrate_by_user2 <- heartrate_seconds_merged_2 %>%
  count(Id, sort = TRUE)
```

Merge the two into one data frame.

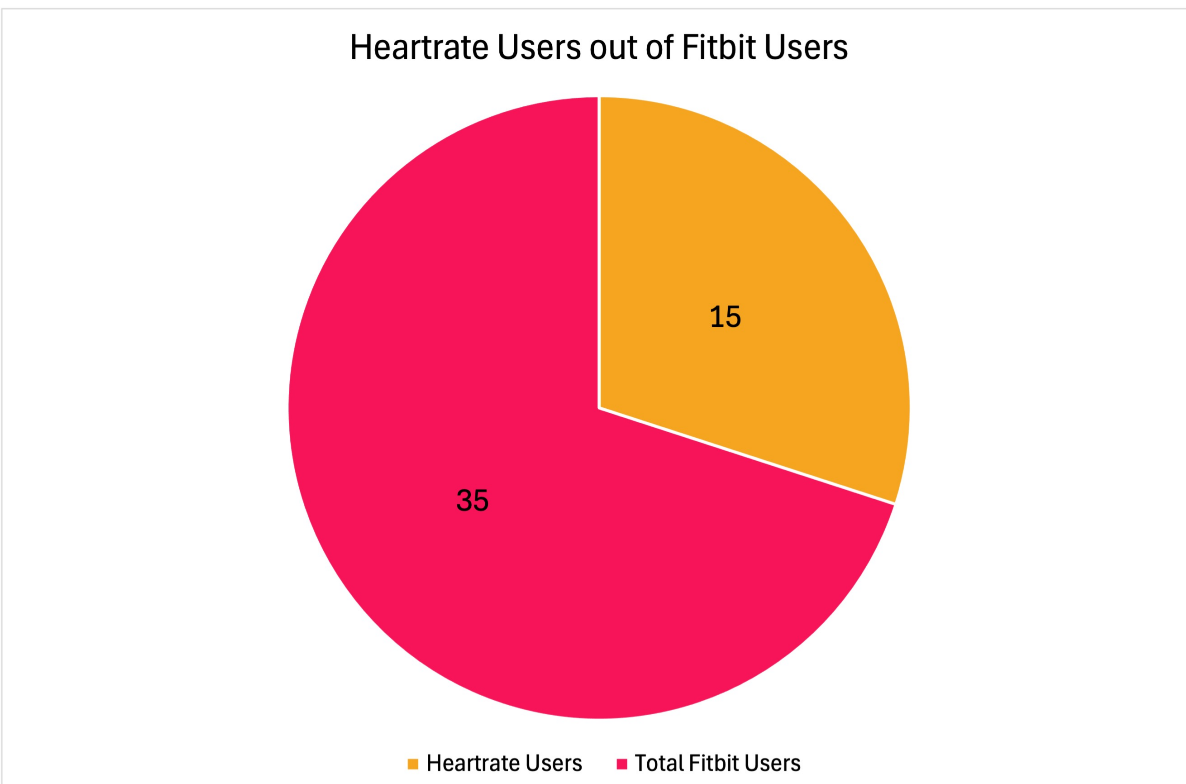
```
heartrate_by_user_total <- merge(heartrate_by_user1,heartrate_by_user2, all=TRUE)
view(heartrate_by_user_total)
```

Count unique ids.

```r
heartrate_by_user_total %>%
  count(Id, sort = TRUE)
```

```
##             Id n
## 1  2022484408 2
## 2  2026352035 2
## 3  2347167796 2
## 4  4020332650 2
## 5  4558609924 2
## 6  5553957443 2
## 7  5577150313 2
## 8  6117666160 2
## 9  6775888955 2
## 10 6962181067 2
## 11 7007744171 2
## 12 8792009665 2
## 13 8877689391 2
## 14 4388161847 1
## 15 6391747486 1
```

In this dataset, only the unique ids were of consequence to determine what percentage all the Fitbit users measured their heartrate. Surprisingly, only 42.86% used this feature although it didn't require any manual input.



sleepDay

I used only one dataset, "sleepDay_merged", because the first dataset measured sleep metrics on a per-minute basis. I was only interested in determining how many and how often users were using the feature but not the other metrics.

68.57% of subscribers used the sleep monitoring feature. The data was defined by the user's Id and recorded their daily 'total sleep records', 'total minutes asleep', and 'total minutes in bed'.

Obtained the sum of the 'TotalSleepRecords' column organized by 'Id'.

```
sleep_day <- sleepDay_merged %>%
 group_by(Id) %>%
 summarise(TotalSleepRecords = sum(TotalSleepRecords))
```

Calculated the number of unique users.

```
sleepDay_merged %>%
  count(Id, sort = TRUE)
```

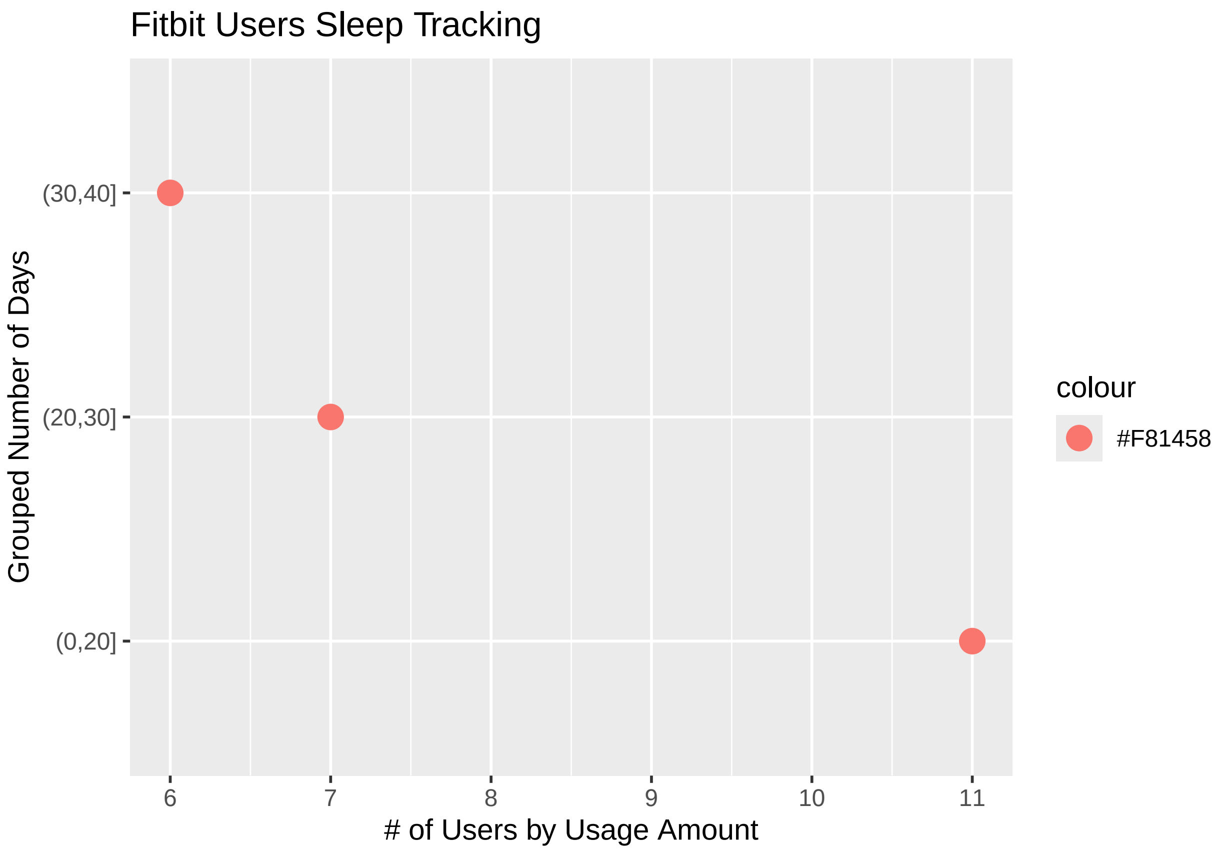
```
##            Id  n
## 1  8378563200 32
## 2  5553957443 31
## 3  6962181067 31
## 4  2026352035 28
## 5  3977333714 28
## 6  4445114986 28
## 7  4702921684 28
## 8  4319703577 26
## 9  5577150313 26
## 10 1503960366 25
## 11 4388161847 24
## 12 7086361926 24
## 13 6117666160 18
## 14 2347167796 15
## 15 8792009665 15
## 16 4020332650  8
## 17 1927972279  5
## 18 4558609924  5
## 19 1644430081  4
## 20 1844505072  3
## 21 6775888955  3
## 22 8053475328  3
## 23 7007744171  2
## 24 2320127002  1
```

The users were divided into groups: 0-20, 20-30, and 30-40. These groups reflected how many days a unique user tracked their sleep.

```
library(dplyr)
sleep_day_range <- sleep_day %>% mutate(cuts = cut(TotalSleepRecords, c(0, 20, 30, 40))) %>%
  group_by(cuts) %>%
  summarize(day_range=n())
```

Created a plot showing the number of users in each group.

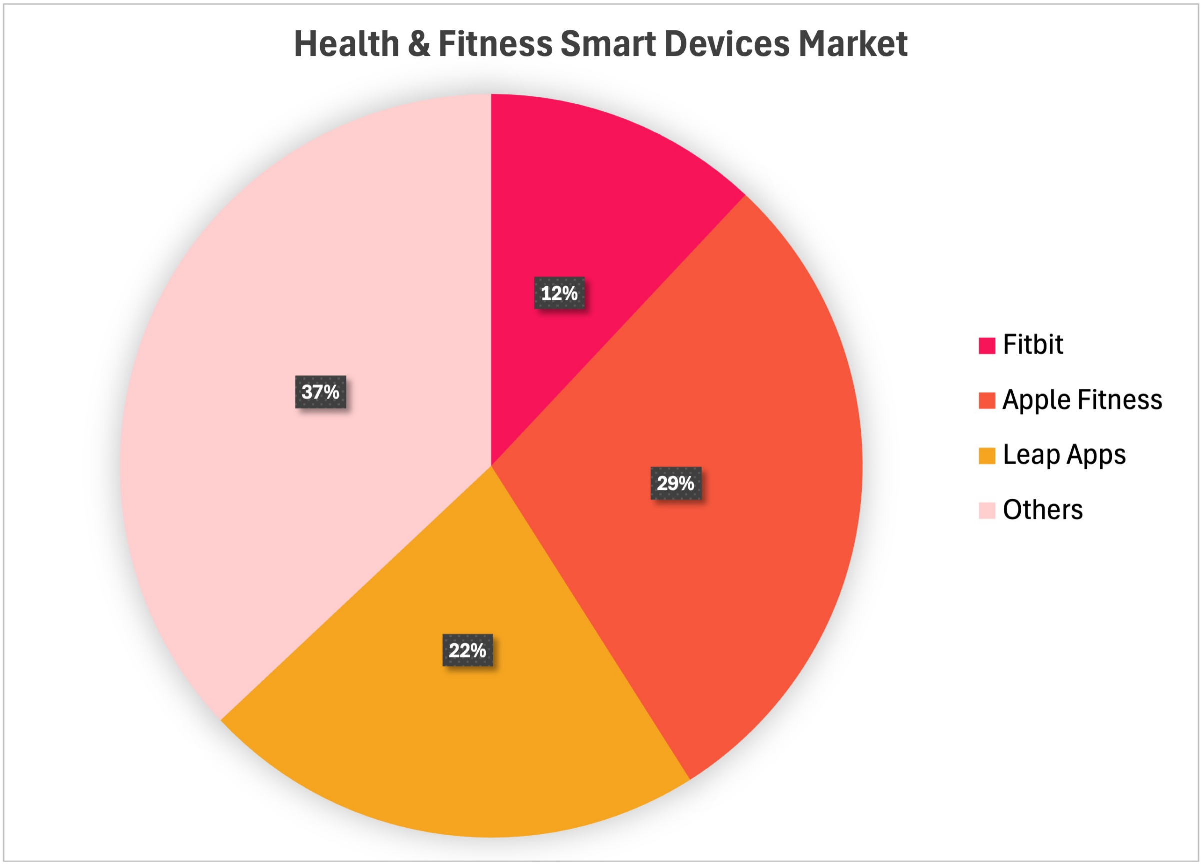
```
ggplot(data=sleep_day_range) +
  geom_point(mapping=aes(x=day_range, y=cuts, color = "#F81458"),size=4) +
  labs(title = "Fitbit Users Sleep Tracking",
       x = "# of Users by Usage Amount",
       y = "Grouped Number of Days")
```



24 users tracked their sleep over 58 days. Out of those 24, only 6 users tracked for 30-40 days, 7 users tracked for 20-30 days, and 11 users tracked their sleep for 0-20 days.

The Market

In 2023, Fitbit had 128M registered users, according to company data which represents 12% of the market share. According to businessofapps.com all those users brought in approximately **$1 Billion** in revenue to the company.



In addition to selling through their own website, Fitbits can be purchased through major retailers, both in store and online.

#Analysis & Recommendations

Bellabeat is at the forefront of the market by targeting women and addressing their specific health and wellness needs. Their products cover many of the same features as Fitbit with the addition of some women-centric features.

Weight Tracking

Obesity is creating a health crisis in the United States. in 2018, over 25% of women were overweight, over 40% of women were obese, and over 10% of women were morbidly obese. This can lead to diabetes, heart disease, high blood pressure, stroke, etc. Bellabeat can easily add weight tracking by developing a way for the Bellabeat app to sync with a smart scale. Fitbit developed their own smart scale that syncs with their app.

Expand Age Demographics

Although aiming for inclusivity, the age demographic of Bellabeat appears to focus on the 21-49 group judging by the photos on the website and the menstruation/ovulation tracking features. Many of the metrics already being tracked can be incorporated to track health issues that relate to middle-aged+ women, such symptoms of perimenopause, menopause, and post menopause. Recent studies, such as one by the National Library of Medicine, have found that mindfulness can help alleviate the psycho-physiological effects in women experiencing these biological changes.

The number of women in their 50's and 60's have more spending power than women in their 20's and 30's and are nearly as large as those age groups! Developing digital products for tracking and improving the health of middle aged women will give Bellabeat another **8 million** potential customers. In addition, Bellabeat will need to add more photos of older women to their marketing.

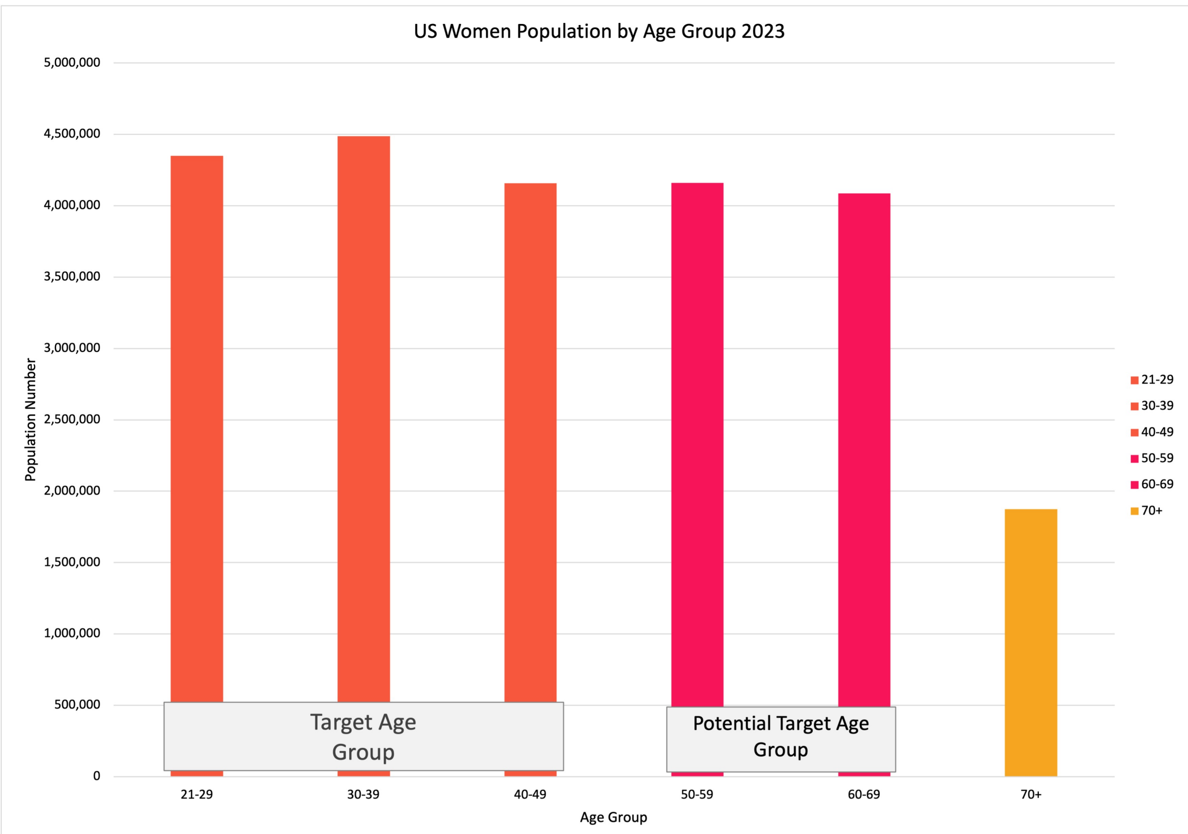


Figure 2: *United States Census Bureau, "Annual Estimates of the Resident Population by Single Year of Age and Sex for the United States: April 1,*