Bellabeat Fitness Analysis

Adrian Suarez

2023-01-04

About the Company

Bellabeat is a high-tech company that manufactures health-focused smart products. They use beautifully designed technology to inform and inspire women around the world by collecting data on activity, sleep, stress, and reproductive health. This has allowed Bellabeat to empower women with knowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women.

Business Task

Analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices. This insight will be used to help guide Bellabeat's marketing strategy for their products.

Loading Packages

```
install.packages('dplyr')
install.packages('tidyverse')
install.packages('ggplot2')
install.packages("stringr")
```

Importing Datasets

```
daily_activity <- read.csv("Fitbase_Data/dailyActivity_merged.csv")
sleep_day <- read.csv("Fitbase_Data/sleepDay_merged.csv")
weight_Info <- read.csv("Fitbase_Data/weightLogInfo_merged.csv")
```

```
head(daily_activity)
```

```
##           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
```

Merging Datasets

Combines the sleep and daily activity datasets.

```
combined_data <- merge(sleep_day, daily_activity, by="Id")
```

Data Cleaning and Data Prepping

Prepping Data for the Average Steps Taken graph

Groups the daily activity dataset by User Id, and then calculates the mean of the total steps taken for each user.

```
totals_steps <- daily_activity %>% group_by(Id) %>%
    summarise(average_steps = mean(TotalSteps))
head(totals_steps)
```

```
## # A tibble: 6 × 2
##           Id average_steps
##        <dbl>         <dbl>
## 1 1503960366        12117.
## 2 1624580081         5744.
## 3 1644430081         7283.
## 4 1844505072         2580.
## 5 1927972279          916.
## 6 2022484408        11371.
```

Arranges the average steps column in descending order, changes the datatype of the user Id to a character type and then preps the dataset for being plotted.

```
totals_steps <- totals_steps[order(-totals_steps$average_steps), ]
totals_steps$Id <- totals_steps$Id %>% as.character()
totals_steps$Id<-factor(totals_steps$Id, levels = totals_steps$Id)
```

Prepping Data for the Average Distance graph

Groups the combined dataset by User Id and then sums the total number of minutes for each user by activity category.

```
totals_Active <- combined_data %>% group_by(Id) %>%
    summarise(total_VeryActiveMinutes = sum(VeryActiveMinutes),
              total_FairlyActiveMinutes = sum(FairlyActiveMinutes),
              total_LightlyActiveMinutes = sum(LightlyActiveMinutes),
              total_SedentaryMinutes = sum(SedentaryMinutes))
head(totals_Active)
```

```
## # A tibble: 6 × 5
##           Id total_VeryActiveMinutes total_FairlyActiveMinutes total_L…¹ total…²
##        <dbl>                   <int>                     <int>     <int>   <int>
## 1 1503960366                   30000                     14850    170450  657325
## 2 1644430081                    1148                      2564     21416  139424
## 3 1844505072                      12                       120     10737  112215
## 4 1927972279                     205                       120      5980  204200
## 5 2026352035                      84                       224    222768  598416
## 6 2320127002                      42                        80      6144   37823
## # … with abbreviated variable names ¹total_LightlyActiveMinutes,
## #   ²total_SedentaryMinutes
```

Creates a temporary dataset with just the distances.

```
temp_distance_df <- combined_data %>% select(VeryActiveDistance, ModeratelyActiveDistance, LightActiveDistance, S
edentaryActiveDistance)
```

Using the previously created dataset, calculates the average distances for each category.

```
avg_very_distance = mean(temp_distance_df[["VeryActiveDistance"]])
avg_moderate_distance = mean(temp_distance_df[["ModeratelyActiveDistance"]])
avg_light_distance = mean(temp_distance_df[["LightActiveDistance"]])
avg_sedentary_distance = mean(temp_distance_df[["SedentaryActiveDistance"]])
avg_distances <- round(c(avg_sedentary_distance,avg_light_distance,avg_moderate_distance,avg_very_distance),2)
```

Creates an average distance dataframe of each category.

```
categories_distance <- c('Avg Sedentary Active Distance',
                  'Avg Lightly Active Distance',
                 'Avg Moderately Active Distance',
                 'Avg Very Active Hours')
df_averages_distance <- data.frame(categories_distance, avg_distances)
df_averages_distance
```

```
##              categories_distance avg_distances
## 1  Avg Sedentary Active Distance          0.00
## 2    Avg Lightly Active Distance          3.54
## 3 Avg Moderately Active Distance          0.73
## 4          Avg Very Active Hours          1.40
```

Prepping Data for the Average Active Hours graph

Selects the total minutes from each category in the totals dataframe. Then takes the average of each and converting minutes to hours. Lastly creates a dataframe of the average active hours of all users for each category.

```
temp_df <- totals_Active %>% select(total_VeryActiveMinutes, total_FairlyActiveMinutes, total_LightlyActiveMinute
s, total_SedentaryMinutes)
```

```
avg_very_active = mean(temp_df[["total_VeryActiveMinutes"]])
avg_fairly_active = mean(temp_df[["total_FairlyActiveMinutes"]])
avg_Lightly_active = mean(temp_df[["total_LightlyActiveMinutes"]])
avg_Sedentary_active = mean(temp_df[["total_SedentaryMinutes"]])
avg_minutes <- c(avg_Sedentary_active,avg_Lightly_active,avg_fairly_active,avg_very_active)
avg_hours <- round(avg_minutes/60,2)
```

```
categories <- c('Avg Sedentary Hours',
                  'Avg Lightly Active Hours',
                 'Avg Fairly Active Hours',
                 'Avg Very Active Hours')
df_averages <- data.frame(categories, avg_hours)
df_averages
```

```
##                categories avg_hours
## 1      Avg Sedentary Hours   6904.71
## 2 Avg Lightly Active Hours   1727.11
## 3  Avg Fairly Active Hours    149.92
## 4    Avg Very Active Hours    207.12
```

Prepping Data for the Weight Info Graph

Separates the Date column in the weight info dataframe to two columns, then converts the datatype of the Date column to datetime, and converts the datatype of the Id column to a character type.

```
weight_Info %>%
  separate(Date, c("Date", "Time"), " ") %>% head()
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 67 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
##           Id      Date     Time WeightKg WeightPounds Fat   BMI IsManualReport
## 1 1503960366  5/2/2016 11:59:59     52.6     115.9631  22 22.65           True
## 2 1503960366  5/3/2016 11:59:59     52.6     115.9631  NA 22.65           True
## 3 1927972279 4/13/2016  1:08:52    133.5     294.3171  NA 47.54          False
## 4 2873212765 4/21/2016 11:59:59     56.7     125.0021  NA 21.45           True
## 5 2873212765 5/12/2016 11:59:59     57.3     126.3249  NA 21.69           True
## 6 4319703577 4/17/2016 11:59:59     72.4     159.6147  25 27.45           True
##          LogId
## 1 1.462234e+12
## 2 1.462320e+12
## 3 1.460510e+12
## 4 1.461283e+12
## 5 1.463098e+12
## 6 1.460938e+12
```

```
weight_Info$Date <- as.POSIXct(weight_Info$Date, format="%m/%d/%Y")
head(weight_Info)
```

```
##           Id       Date WeightKg WeightPounds Fat   BMI IsManualReport
## 1 1503960366 2016-05-02     52.6     115.9631  22 22.65           True
## 2 1503960366 2016-05-03     52.6     115.9631  NA 22.65           True
## 3 1927972279 2016-04-13    133.5     294.3171  NA 47.54          False
## 4 2873212765 2016-04-21     56.7     125.0021  NA 21.45           True
## 5 2873212765 2016-05-12     57.3     126.3249  NA 21.69           True
## 6 4319703577 2016-04-17     72.4     159.6147  25 27.45           True
##          LogId
## 1 1.462234e+12
## 2 1.462320e+12
## 3 1.460510e+12
## 4 1.461283e+12
## 5 1.463098e+12
## 6 1.460938e+12
```

```
weight_Info$Id <- weight_Info$Id %>% as.character()
```

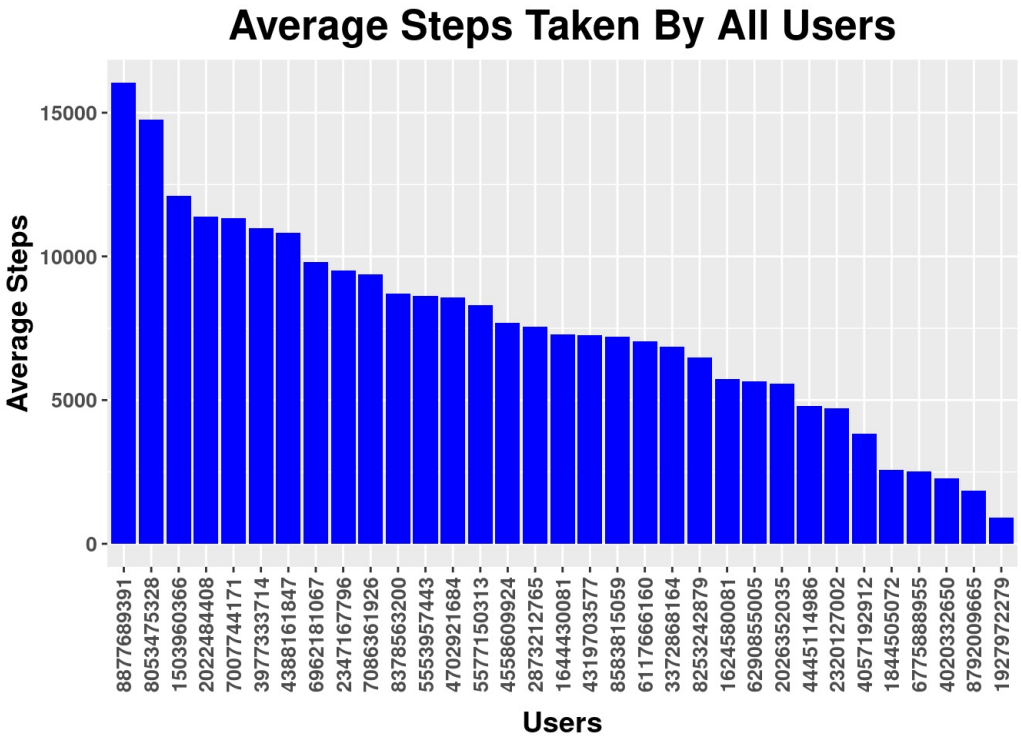
Groups the combined dataset by User Id, and then creates a dataframe called Average Calories that takes the average of the calories burned by each user.

```
Average_calories_df <- combined_data %>% group_by(Id) %>%
    summarise(Average_calories = round(mean(Calories),2))
Average_calories_df$Id <- Average_calories_df$Id %>% as.character()
Average_calories_df <- Average_calories_df[order(-Average_calories_df$Average_calories), ]
Average_calories_df$Id<-factor(Average_calories_df$Id, levels = Average_calories_df$Id)
head(Average_calories_df)
```

```
## # A tibble: 6 × 2
##   Id         Average_calories
##   <fct>                 <dbl>
## 1 8378563200            3437.
## 2 5577150313            3360.
## 3 4388161847            3094.
## 4 4702921684            2966.
## 5 8053475328            2946.
## 6 1644430081            2811.
```

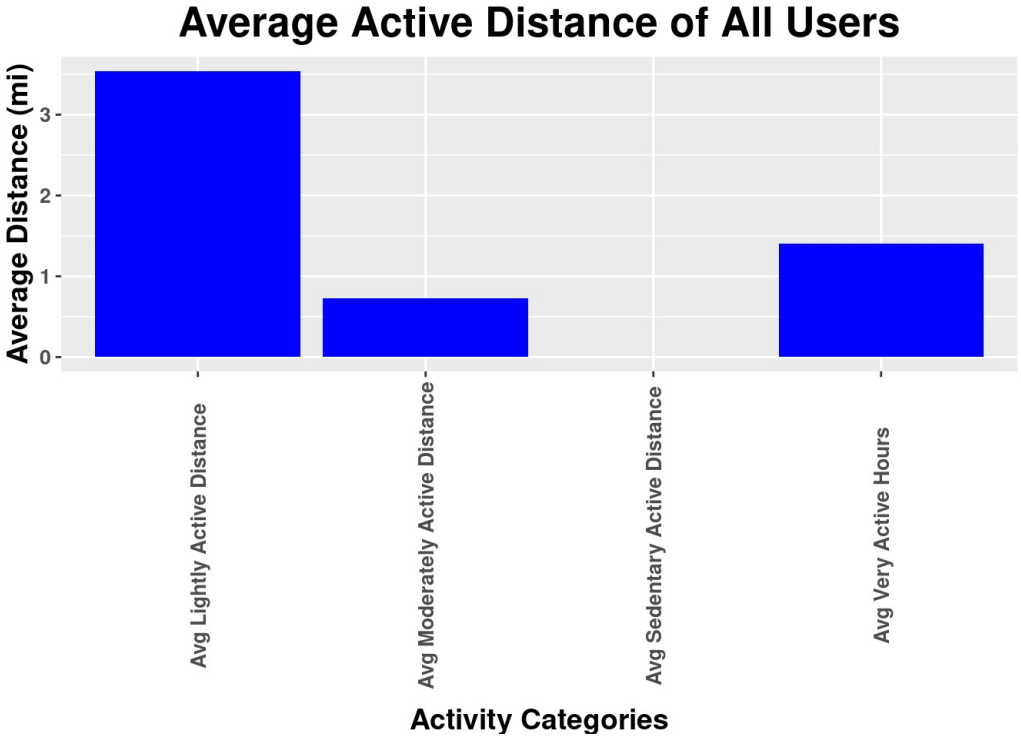
Visualizations

```
ggplot(data=totals_steps, aes(x=Id, y=average_steps)) + geom_bar(stat="identity", fill="blue") +
labs(titles="Average Steps Taken By All Users",x="Users",y="Average Steps") +
theme(plot.title = element_text(hjust=0.5, size=20, face="bold"),
      axis.title.x = element_text(vjust=-1,size=14, face="bold"),
      axis.title.y = element_text(size=14, face="bold"),
      axis.text.x = element_text(vjust=0.5,face="bold", size=10, angle=90),
      axis.text.y = element_text(face="bold", size=10))
```



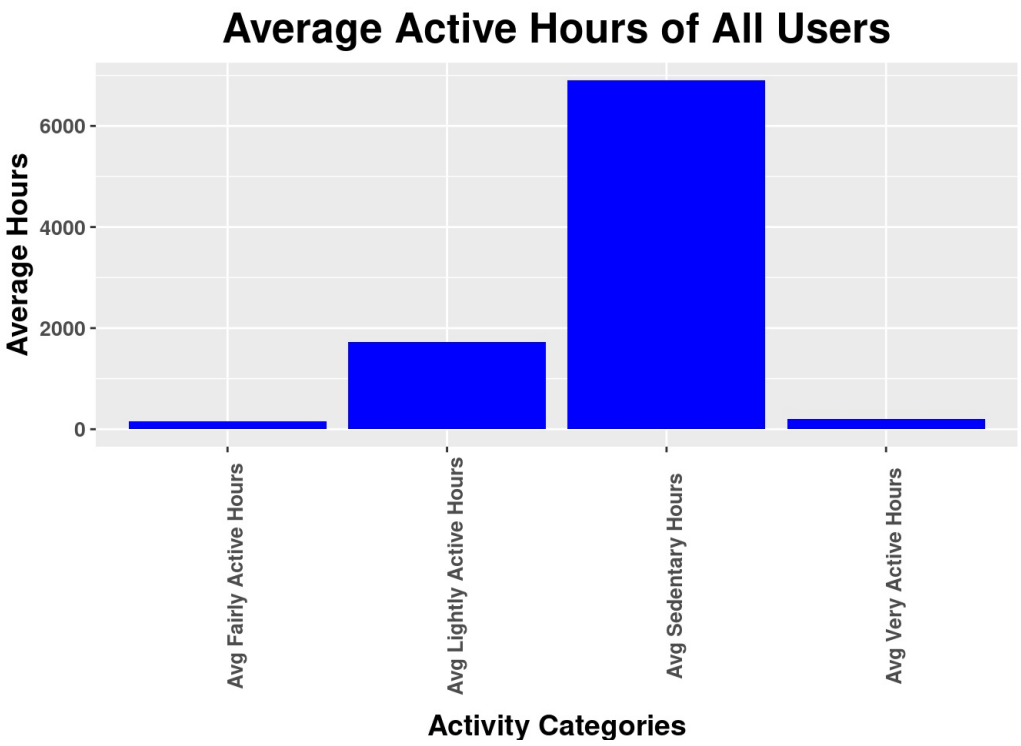
All 33 participants kept track of their steps taken, showing that this would be a good feature for a fitness app to have.

```
ggplot(data=df_averages_distance, aes(x=categories_distance, y=avg_distances)) + geom_bar(stat="identity", fill="
blue") +
labs(titles="Average Active Distance of All Users",x="Activity Categories",y="Average Distance (mi)") +
theme(plot.title = element_text(hjust=0.5, size=20, face="bold"),
      axis.title.x = element_text(vjust=-1,size=14, face="bold"),
      axis.title.y = element_text(size=14, face="bold"),
      axis.text.x = element_text(vjust=0.5,face="bold", size=10, angle=90),
      axis.text.y = element_text(face="bold", size=10))
```



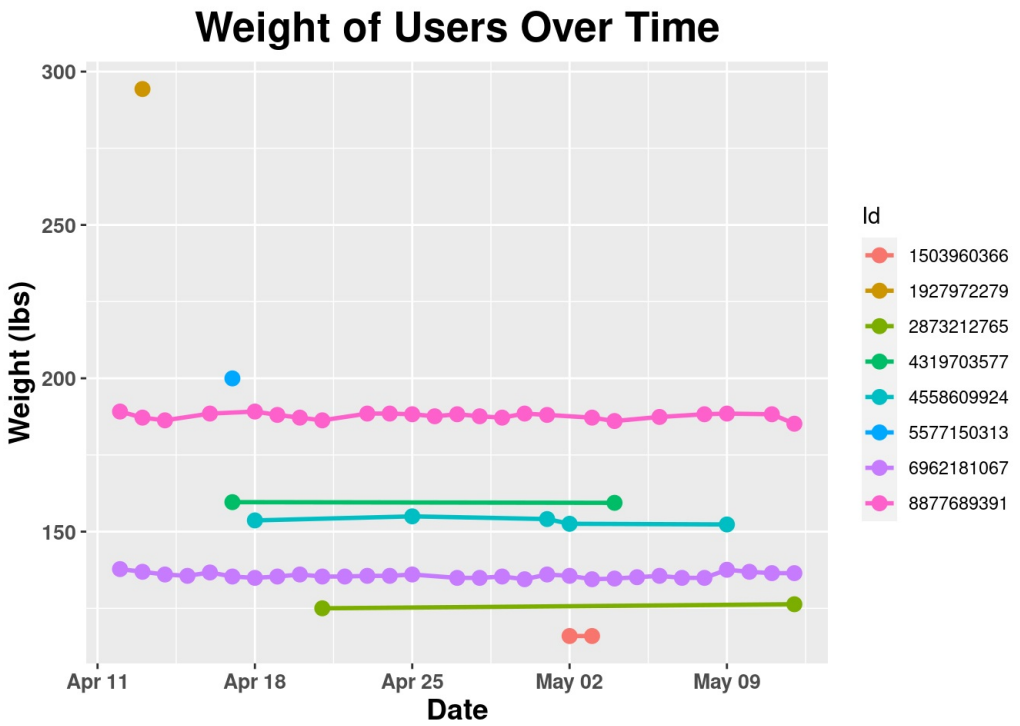
The highest distances tracked by users were in the Lightly Active category, showing that the majority of people are using the app for lightly active activities.

```
ggplot(data=df_averages, aes(x=categories, y=avg_hours)) + geom_bar(stat="identity", fill="blue") +
labs(titles="Average Active Hours of All Users",x="Activity Categories",y="Average Hours") +
theme(plot.title = element_text(hjust=0.5, size=20, face="bold"),
      axis.title.x = element_text(vjust=-1,size=14, face="bold"),
      axis.title.y = element_text(size=14, face="bold"),
      axis.text.x = element_text(face="bold", size=10, angle=90),
      axis.text.y = element_text(face="bold", size=10))
```



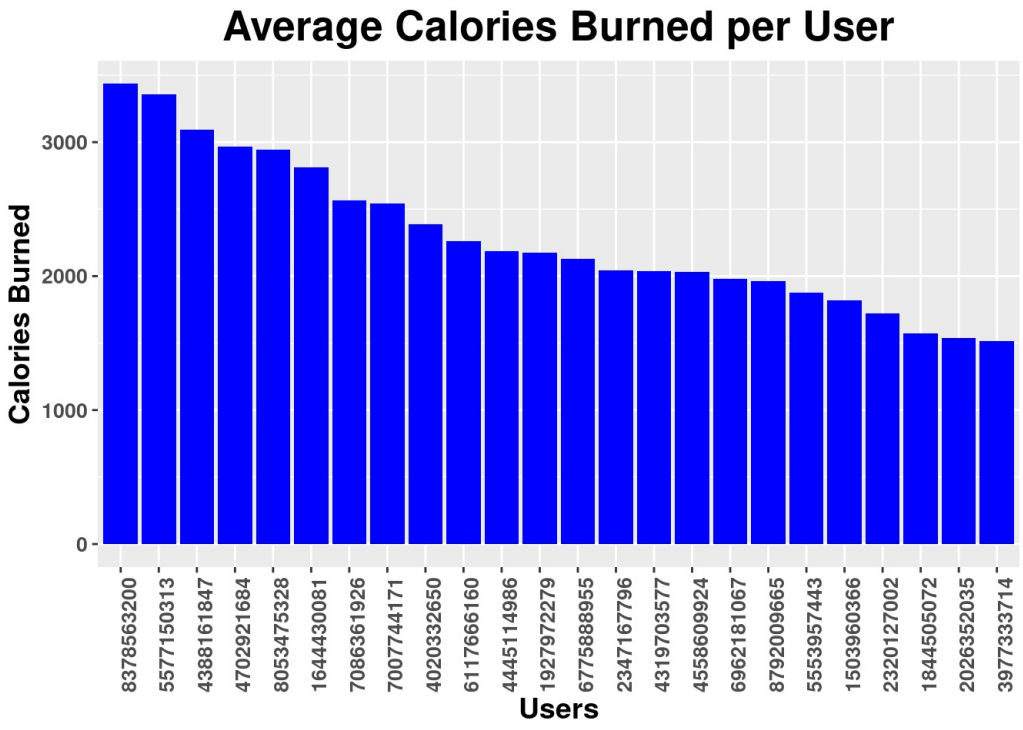
The majority of active hours logged by users are in the Sedentary category followed by the Lightly Active category. This shows that the majority of users are using the app for calmer activities, not necessarily for vigorous exercise.

```
ggplot(data=weight_Info, aes(x=Date, y=WeightPounds, color=Id)) + geom_point(size=3) + geom_line(linewidth=1) +
labs(titles="Weight of Users Over Time",x="Date",y="Weight (lbs)") +
theme(plot.title = element_text(hjust=0.5, size=20, face="bold"),
      axis.title.x = element_text(size=14, face="bold"),
      axis.title.y = element_text(size=14, face="bold"),
      axis.text.x = element_text(face="bold", size=10),
      axis.text.y = element_text(face="bold", size=10))
```



Only 8 of the 33 possible participants chose to keep track of their weight, and the weight of these participants stay pretty constant. This shows that people aren't using this app for weight-loss.

```
ggplot(data=Average_calories_df, aes(x=Id, y=Average_calories)) + geom_bar(stat="identity", fill="blue") +
labs(titles="Average Calories Burned per User",x="Users",y="Calories Burned") +
theme(plot.title = element_text(hjust=0.5, size=20, face="bold"),
      axis.title.x = element_text(size=14, face="bold"),
      axis.title.y = element_text(size=14, face="bold"),
      axis.text.x = element_text(face="bold", size=10, angle=90),
      axis.text.y = element_text(face="bold", size=10))
```



Out of the 33 possible participants 24 decided to keep track of calories burned. This would be a good feature to have in a fitness app.

Conclusion and Business Recommendations

Through collecting data on activity, sleep, and reproductive health Bellabeat has been able to empower women with knowledge about their own health. This company has been growing rapidly and is positioning itself as a top tech wellness company for women. Through analyzing the FitBit tracker data I have found some insights that can be used to help Bellabeat's marketing strategy:

The majority of users who are using the app are using it for **sedentary and light activity purposes**. Therefore focusing on products, for example that **count the number of steps taken a day** or that **track sleeping habits** would beneficial the company.

For example, focusing specifically on **the Leaf**, one of Bellabeats' products that is a wellness tracker that can be worn as a bracelet, necklace, or clip. **This product can be leveraged to track the number of steps taken throughout the day as well as the number of hours slept a night.** This would be beneficial since the majority of users who are using other fitness apps are looking to track these features.

Although a lot of the users are tracking number of calories burned a day, this data is not being used as a way to lose weight. This is because only 6 out of the total 33 participants chose to track their weight, furthermore their respective weights remained mostly constant.

Thank you for taking the time to go through my markdown regarding using Fitbit data to make recommendations for the Bellabeat marketing strategy.