BellaBeat Data Analytics Project

Ouatatchin Kone

2022-10-28

The company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health 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.

Stakeholders

- Urška Sršen: Bellabeat's cofounder and Chief Creative Officer
- Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team
- Bellabeat marketing analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy

Executive Summary

The aim of this analysis is to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights discovered will then help guide marketing strategy for the company. This report covers different phases in my analysis to help answer the business questions raised by the management. These phases include Ask questions, Prepare data, Process data, Analyze data, Share data, and Act.

Methodology

Before performing the analysis, the data was collected through a public domain, then wrangled to make sure it's cleaned, reliable and error-free by removing duplicates, finding and filling missing values and normalizing data. After that, I explored and found correlation between variables, proceeded to data visualization to better capture trends and insights and finally made highly recommendations to the executive team.

Ask phase

The executive team asked to analyze smart device fitness data as they could help unlock new growth opportunities for the company. More specifically, the following questions were raised:

- What are some trends in smart device usage?
- How could these trends apply to Bellabeat customers?

- How could these trends help influence Bellabeat marketing strategy?

Prepare phase

The data to explore and analyze was made available through FitBit Fitness Tracker Data: http://www.kaggle.com/arahnic/fitbit which is a Public Domain. I proceeded to the collection and the storage of data by making sure they meet the requirements in terms of integrity, reliability, credibility and security. However, going through my analysis I found that there are thirty-three (33) ID (users) instead of thirty (30) as mentioned in the business task. Therefore my analysis will focus on 33 users. I decided to work with the following four (4) data sets as for me they are the most relevant for this analysis task but also for the BellaBeat product I chose to apply my analysis on that is Bellabeat app: Provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. In addition, this product covers and is related to almost all of the other products of the company. These four data sets are:

- dailyActivity_merged
- heartrate_seconds_merged
- sleepDay_merged
- weightLogInfo_merged

For example, dailyActivity_merged contains other data sets such as dailyCalories_merged and dailySteps_merged which in turn are aggregates of smaller data sets including hourlyCalories_merged, hourlySteps_merged, etc.

Let's import the data sets

```
Activity <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\BellaBeat Data\\dailyActivity_merged.csv"
Heartrate <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\BellaBeat Data\\heartrate_seconds_merged
sleepDay <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\BellaBeat Data\\sleepDay_merged.csv")
weightLogInfo <- read.csv("C:\\Users\\BANKS\\Documents\\Mes cours\\BellaBeat Data\\weightLogInfo_merged
```

Process phase

Before analyzing data, let's load a number of packages.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
library(dplyr)
library(ggplot2)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(readr)
library(skimr)
library(tibble)
library(yaml)
library(gapminder)
library(ggpubr)
```

Now let's find and remove any duplicates in the data sets.

```
sum(duplicated(Activity))
```

```
## [1] 0
```

```
sum(duplicated(Heartrate))
```

```
## [1] 0
```

```
sum(duplicated(sleepDay))
```

```
## [1] 3
```

```
sum(duplicated(weightLogInfo))
```

```
## [1] 0
```

Three (3) duplicates were found in the data set "sleepDay". Let's handle them using the distinct function":

```
sleepDay <- distinct(sleepDay)
sum(duplicated(sleepDay))
```

```
## [1] 0
```

Here, I searched and found missing values in my data sets.

```
colSums(is.na(Activity))
```

```
##                       Id             ActivityDate               TotalSteps
##                        0                        0                        0
##            TotalDistance          TrackerDistance LoggedActivitiesDistance
##                        0                        0                        0
##       VeryActiveDistance ModeratelyActiveDistance      LightActiveDistance
##                        0                        0                        0
##  SedentaryActiveDistance        VeryActiveMinutes      FairlyActiveMinutes
##                        0                        0                        0
##     LightlyActiveMinutes         SedentaryMinutes                 Calories
##                        0                        0                        0
```

```
colSums(is.na(Heartrate))
```

```
##    Id  Time Value
##     0     0     0
```

```
colSums(is.na(sleepDay))
```

```
##                 Id           SleepDay  TotalSleepRecords TotalMinutesAsleep
##                  0                  0                  0                  0
##     TotalTimeInBed
##                  0
```

```
colSums(is.na(weightLogInfo))
```

```
##             Id           Date       WeightKg   WeightPounds            Fat
##              0              0              0              0             65
##            BMI IsManualReport          LogId
##              0              0              0
```

65 missing values (NA) were found in the data set "weightLogInfo", particularly in the variable "Fat". We can check whether our data set consists of variables with more than 30% of missing values. If yes, then we will just delete the whole variable.

```
colSums(is.na(weightLogInfo))/nrow(weightLogInfo)
```

```
##             Id           Date       WeightKg   WeightPounds            Fat
##      0.0000000      0.0000000      0.0000000      0.0000000      0.9701493
##            BMI IsManualReport          LogId
##      0.0000000      0.0000000      0.0000000
```

More than 97% missing values were found. Then, dealing with those by using the mean would bias our analysis as the total number of missing values is very high. We can use <.3 to create a logical comparison that can be used to delete column "Fat".

```
weightLogInfo <- weightLogInfo[colSums(is.na(weightLogInfo))/nrow(weightLogInfo) <.3]
```

Now that our data sets have been cleaned, let's make sure that they have similar date format to help us create tidy data and merge the data sets.

```
Activity <- Activity %>%
  rename("Date" = "ActivityDate")
```

```
 weightLogInfo <- weightLogInfo %>%
  mutate(Date = as.POSIXct(Date, format = "%m/%d/%Y %H:%M" , TZ=Sys.timezone())) %>%
  separate(Date, into = c('Date', 'Time'), sep = ' ', remove = TRUE)
```

```
sleepDay  <- sleepDay  %>%
  mutate(SleepDay = as.POSIXct(SleepDay, format = "%m/%d/%Y %H:%M" , tz=Sys.timezone())) %>%
  separate(SleepDay, into = c('Date', 'Time'), sep = ' ', remove = TRUE)
```

```
Heartrate <- Heartrate %>%
  mutate(Time = as.POSIXct(Time, format = "%m/%d/%Y %H:%M" , tz=Sys.timezone())) %>%
  separate(Time, into = c('Date', 'Time'), sep = ' ', remove = TRUE)
```

Checking our new columns

```
colnames(Activity)
```

```
##  [1] "Id"                       "Date"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
colnames(Heartrate)
```

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

```
colnames(sleepDay)
```

```
## [1] "Id"                 "Date"               "Time"
## [4] "TotalSleepRecords"  "TotalMinutesAsleep" "TotalTimeInBed"
```

```
colnames(weightLogInfo)
```

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

Let's add a new column called "Total_Minutes_Asleep_in_Hours" to the data set "sleepDay". "Total_Minutes_Asleep_in_Hours" is the converted values in hours of the total minutes asleep for each user which can help find insights about which users sleep the most or the least and make some recommendations.

```
sleepDay <- sleepDay  %>%
  mutate(Total_Minutes_Asleep_in_Hours = TotalMinutesAsleep/60)
```

```
colnames(sleepDay)
```

```
## [1] "Id"                              "Date"
## [3] "Time"                            "TotalSleepRecords"
## [5] "TotalMinutesAsleep"              "TotalTimeInBed"
## [7] "Total_Minutes_Asleep_in_Hours"
```

Analysis Phase

After our data sets have been cleaned and prepared, they are ready for the analysis phase. Firstly, let's create summaries for each data set prepared:

```
summary(Activity)
```

```
##        Id                 Date              TotalSteps    TotalDistance
##  Min.   :1.504e+09   Length:940         Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   Class :character   1st Qu.: 3790   1st Qu.: 2.620
##  Median :4.445e+09   Mode  :character   Median : 7406   Median : 5.245
##  Mean   :4.855e+09                      Mean   : 7638   Mean   : 5.490
##  3rd Qu.:6.962e+09                      3rd Qu.:10727   3rd Qu.: 7.713
##  Max.   :8.878e+09                      Max.   :36019   Max.   :28.030
##  TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   :0.0000           Min.   : 0.000
##  1st Qu.: 2.620   1st Qu.:0.0000           1st Qu.: 0.000
##  Median : 5.245   Median :0.0000           Median : 0.210
##  Mean   : 5.475   Mean   :0.1082           Mean   : 1.503
##  3rd Qu.: 7.710   3rd Qu.:0.0000           3rd Qu.: 2.053
##  Max.   :28.030   Max.   :4.9421           Max.   :21.920
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  Min.   :0.0000           Min.   : 0.000      Min.   :0.000000
##  1st Qu.:0.0000           1st Qu.: 1.945      1st Qu.:0.000000
##  Median :0.2400           Median : 3.365      Median :0.000000
##  Mean   :0.5675           Mean   : 3.341      Mean   :0.001606
##  3rd Qu.:0.8000           3rd Qu.: 4.782      3rd Qu.:0.000000
##  Max.   :6.4800           Max.   :10.710      Max.   :0.110000
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

```
summary(sleepDay)
```

```
##        Id                 Date               Time           TotalSleepRecords
##  Min.   :1.504e+09   Length:410         Length:410         Min.   :1.00
##  1st Qu.:3.977e+09   Class :character   Class :character   1st Qu.:1.00
##  Median :4.703e+09   Mode  :character   Mode  :character   Median :1.00
##  Mean   :4.995e+09                                         Mean   :1.12
##  3rd Qu.:6.962e+09                                         3rd Qu.:1.00
##  Max.   :8.792e+09                                         Max.   :3.00
##  TotalMinutesAsleep TotalTimeInBed  Total_Minutes_Asleep_in_Hours
##  Min.   : 58.0      Min.   : 61.0   Min.   : 0.9667
##  1st Qu.:361.0      1st Qu.:403.8   1st Qu.: 6.0167
##  Median :432.5      Median :463.0   Median : 7.2083
##  Mean   :419.2      Mean   :458.5   Mean   : 6.9862
##  3rd Qu.:490.0      3rd Qu.:526.0   3rd Qu.: 8.1667
##  Max.   :796.0      Max.   :961.0   Max.   :13.2667
```

```
summary(weightLogInfo)
```

```
##        Id                 Date               Time              WeightKg
##  Min.   :1.504e+09   Length:67          Length:67          Min.   : 52.60
##  1st Qu.:6.962e+09   Class :character   Class :character   1st Qu.: 61.40
##  Median :6.962e+09   Mode  :character   Mode  :character   Median : 62.50
##  Mean   :7.009e+09                                         Mean   : 72.04
##  3rd Qu.:8.878e+09                                         3rd Qu.: 85.05
##  Max.   :8.878e+09                                         Max.   :133.50
##   WeightPounds        BMI        IsManualReport         LogId
##  Min.   :116.0   Min.   :21.45   Length:67          Min.   :1.460e+12
##  1st Qu.:135.4   1st Qu.:23.96   Class :character   1st Qu.:1.461e+12
##  Median :137.8   Median :24.39   Mode  :character   Median :1.462e+12
##  Mean   :158.8   Mean   :25.19                      Mean   :1.462e+12
##  3rd Qu.:187.5   3rd Qu.:25.56                      3rd Qu.:1.462e+12
##  Max.   :294.3   Max.   :47.54                      Max.   :1.463e+12
```

```
summary(Heartrate)
```

```
##        Id                 Date               Time               Value
##  Min.   :2.022e+09   Length:2483658     Length:2483658     Min.   : 36.00
##  1st Qu.:4.388e+09   Class :character   Class :character   1st Qu.: 63.00
##  Median :5.554e+09   Mode  :character   Mode  :character   Median : 73.00
##  Mean   :5.514e+09                                         Mean   : 77.33
##  3rd Qu.:6.962e+09                                         3rd Qu.: 88.00
##  Max.   :8.878e+09                                         Max.   :203.00
```

These different summaries from our cleaned data give us an overview of the trends. For example, we can see that users sleep on average 6.98 hours per day which is below 8 hours recommended.

Now let's merge some data sets

```
Activity_sleepDay_merged <- full_join(Activity, sleepDay, by=c("Id", "Date")) %>%
  select(-LoggedActivitiesDistance, -SedentaryActiveDistance)
  Activity_sleepDay_merged[is.na(Activity_sleepDay_merged)] <- 0
```

Merging “Activity” and “sleepDay” having “Id” and “Date” in common will facilitate our analysis and allow us to find trends among variables.

```
Activity_weightLogInfo_merged <- full_join(Activity, weightLogInfo, by=c("Id", "Date")) %>%
  select(-LoggedActivitiesDistance, -SedentaryActiveDistance)
  Activity_weightLogInfo_merged[is.na(Activity_weightLogInfo_merged)] <- 0
```

Merging “Activity” and ” weightLogInfo” having “Id” and “Date” in common helps us compute the number of users that tracked their respective weights.

Next let’s find percentage of users that monitored their weights and those that did not

```
Percentage_weight_monitored <- Activity_weightLogInfo_merged %>%
  group_by(Id) %>%
  summarise(weight_per_user = sum(WeightKg)) %>%
  count(Total_Users_monitored_weight = sum(weight_per_user > 0),
        Total_Users_didnot_monitor_weight = sum(weight_per_user < 1)) %>%
  mutate(percent_Users_monitored_weight = round(Total_Users_monitored_weight/33*100, digits = 2),
         percent_Users_didnot_monitor_weight = round(Total_Users_didnot_monitor_weight/33*100, digits =
```

Here, I found that only 8 users (24.24%) of 33 tracked their weights regularly while 25 (75.76%) did not. This shows a weak interest from users in this activity and some suggestions will be formulated in the recommendation phase.

Now let’s calculate activity records per user

```
Activity_Records_per_User <- Activity_sleepDay_merged %>%
        group_by(Id) %>%
        summarise(TotalSteps = sum(TotalSteps), TotalDistance =
                                    sum(TotalDistance), TotalCalories = sum(Calories))
```

This allows us to gain insights on users’ records in terms of total steps, total distance and total calories burnt in 31 days of activity. This gives us an overview of most and least active users.

Here, I track the most active users in terms of number of days and nights:

```
Users_frequency <- Activity_sleepDay_merged %>%
  group_by(Id) %>%
  summarise(Number_of_days = sum(TotalSteps > 0), Number_of_nights = sum(TotalMinutesAsleep > 0))
```

This computes the number of days and nights users have been active and using BellaBeat devices. The results show that users do not use the smart devices the same way and are not equally active all along the 31 days.

Next, I calculate average heart rate per user

```
Average_heart_rate_per_user <- Heartrate %>%
  group_by(Id) %>%
  summarise(Average_heart_rate = mean(Value)) %>%
  mutate(Average_heart_rate = round(Average_heart_rate, digits = 2))
```

Calculating the average heart rate per user gives us different values for each user and helps us understand which user has a normal heart rate and which ones go beyond the threshold of 80. For example, Id 2026352035 has an average heart rate of 93.77 which is not far from tachycardia.

Finally, I calculate average sleep hours per user

```
Average_sleep_hours_per_user <- sleepDay %>%
  group_by(Id) %>%
  summarise(Average_sleep_hours = mean(Total_Minutes_Asleep_in_Hours))
```

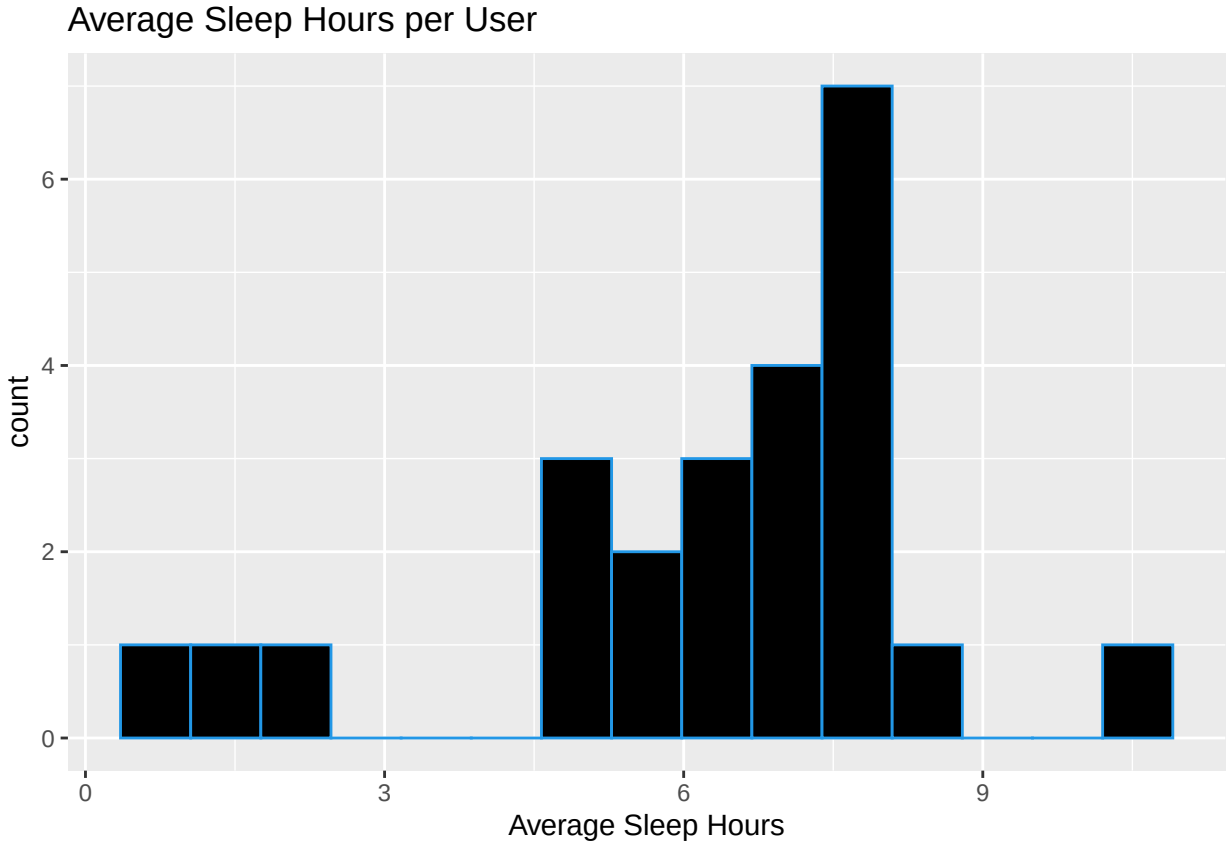
This computation allows us to know that 24 users of 33 (about 73%) use their devices once in bed at night. It also gives us insights on who has the highest sleep hours per night.

Visualization Phase

Here, we will go through a series of visualizations which come from previous data sets created in the analysis. For each visualization, I will go deeper in my analysis and explain probable causes for insights.

For this graph, I plot the frequency distribution of average sleep hours among users.

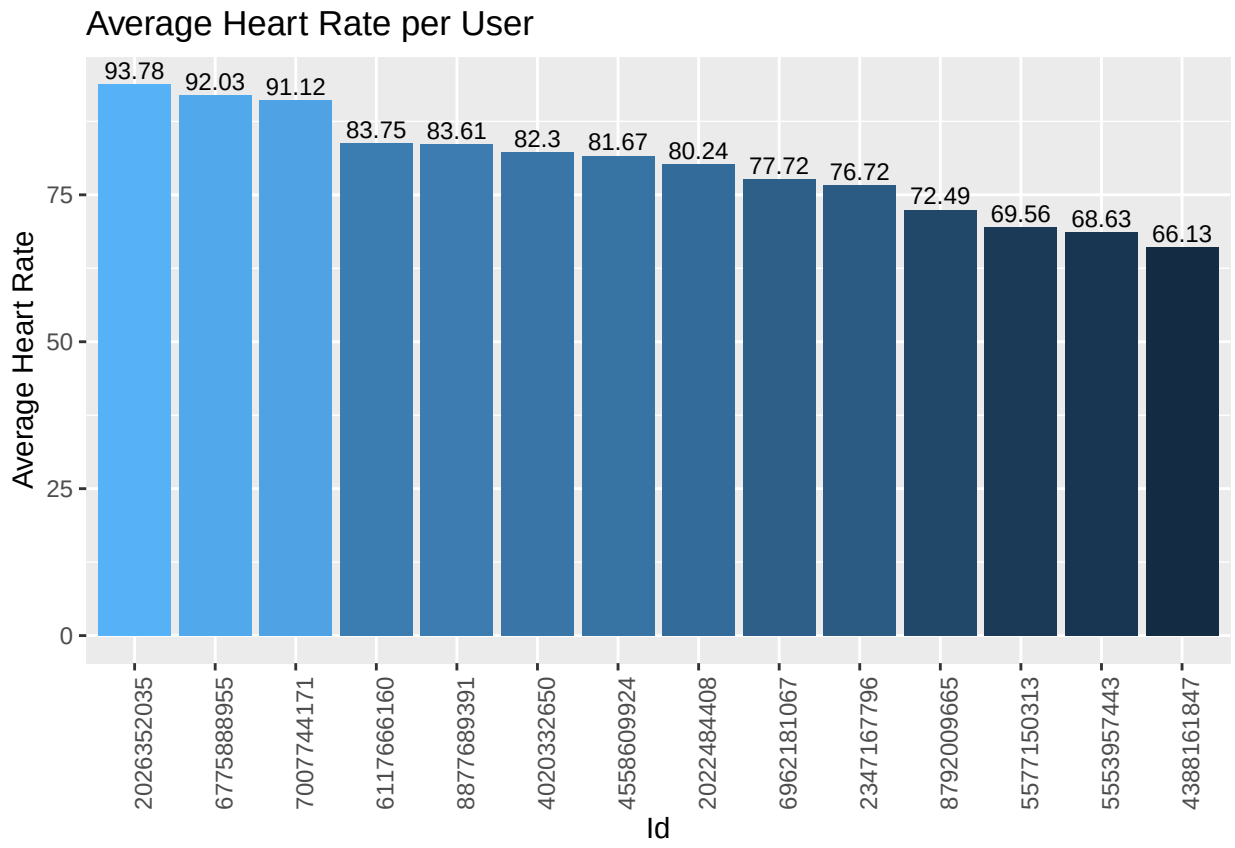
```
ggplot(data=Average_sleep_hours_per_user,aes(x = Average_sleep_hours))+
  geom_histogram(colour = 4, fill = "black", bins = 15)+
  labs(title = "Average Sleep Hours per User", x = "Average Sleep Hours" )
```



We can see from this histogram that the distribution of average sleep hours is concentrated within the 6-hour region. This means that BellaBeat users sleep on average 6 – 7 hours per day which is below the 8 hours recommended by doctors. We also note that a small, but not inconsiderable, proportion sleeps only 1 – 3 hours per day which is very critical.

Here, I plot Id against Average heart rate and look at heart rate variation per user.

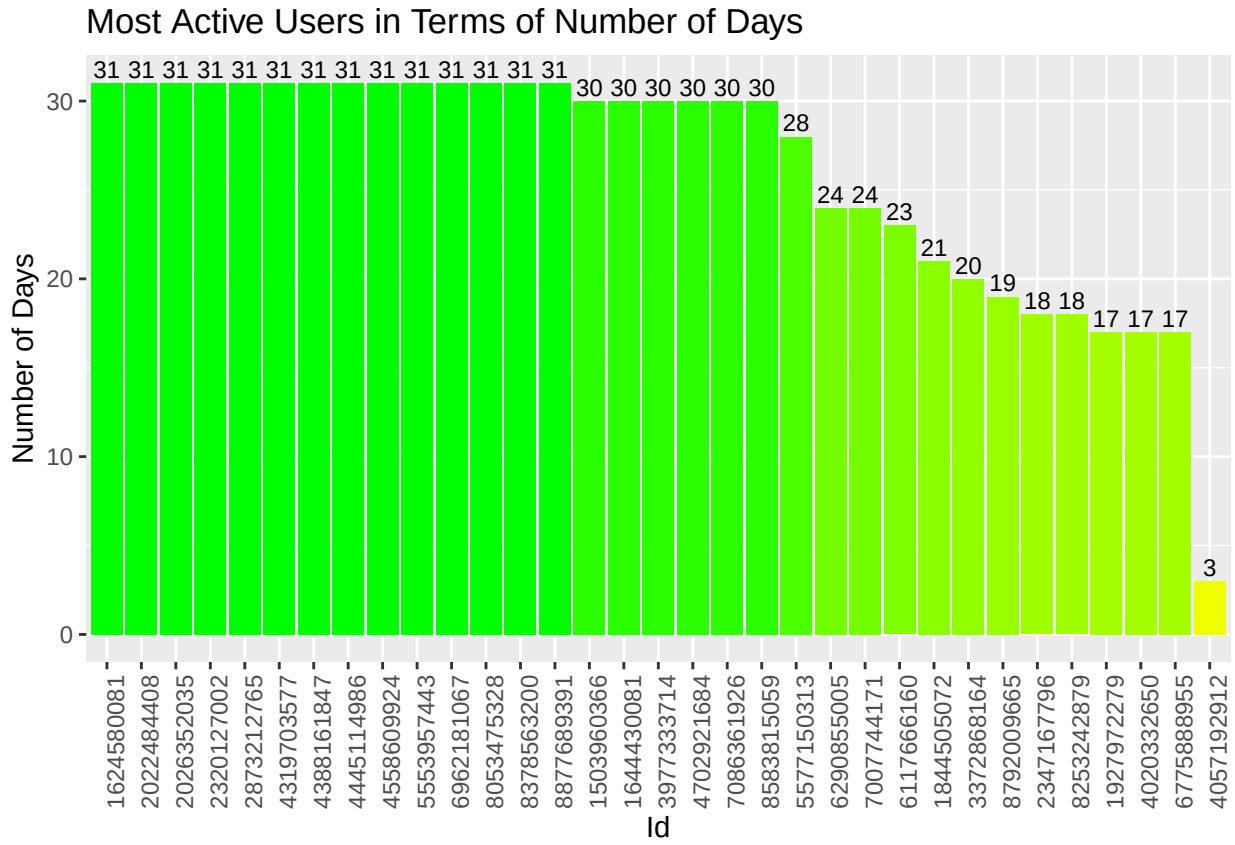
```
ggplot(data = Average_heart_rate_per_user, aes(x = reorder(Id, -Average_heart_rate), y = Average_heart_
  geom_bar(stat = "identity")+
  geom_text(aes(label = Average_heart_rate), size = 3, vjust = -0.3)+
  theme(legend.position = "none")+
  theme(axis.text.x = element_text(angle = 90))+
  labs(title = "Average Heart Rate per User", x = "Id", y = "Average Heart Rate")
```



This bar plot indicates that BellaBeat users have different heart rates and as it is displayed in descending order, we clearly see that 7 users have heart rates higher than 80 per minute (normal heart rate). Id 2026352035 has the highest heart rate, 93.78 per minute. High heart rates may be caused by stress, fear or other medical reasons. Then, a psychological or medical approach should be considered.

In this section, I plot Id against Number of days to effectively capture most active users during the day.

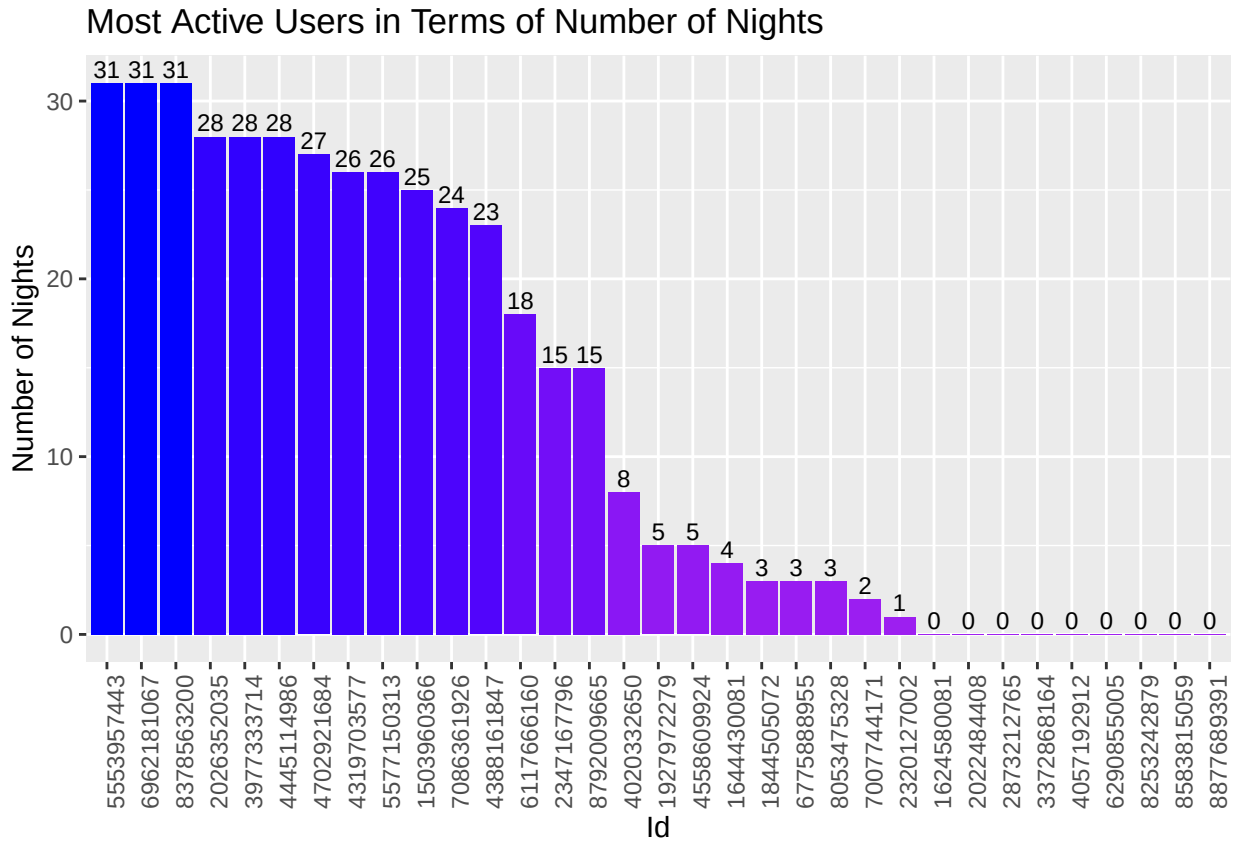
```
ggplot(data = Users_frequency,aes(x = reorder(Id, -Number_of_days), y = Number_of_days, fill = Number_o
  geom_bar(stat = "identity")+
  geom_text(aes(label = Number_of_days), size = 3, vjust = -0.3)+
  theme(legend.position = "none")+
  theme(axis.text.x = element_text(angle = 90))+
  scale_fill_gradient2(low = "red", mid = "yellow", high = "green")+
  labs(title = "Most Active Users in Terms of Number of Days", x = "Id", y = "Number of Days")
```



This bar plot depicts the number of days that users wear their smart devices and perform required activities. We note that 20 of 33 users are active all month long (considering 30 days or 31 days in the month) while user with Id 4057192912 is active for only 3 days.

In this section, I plot Id against number of nights to effectively capture most active users during the night.

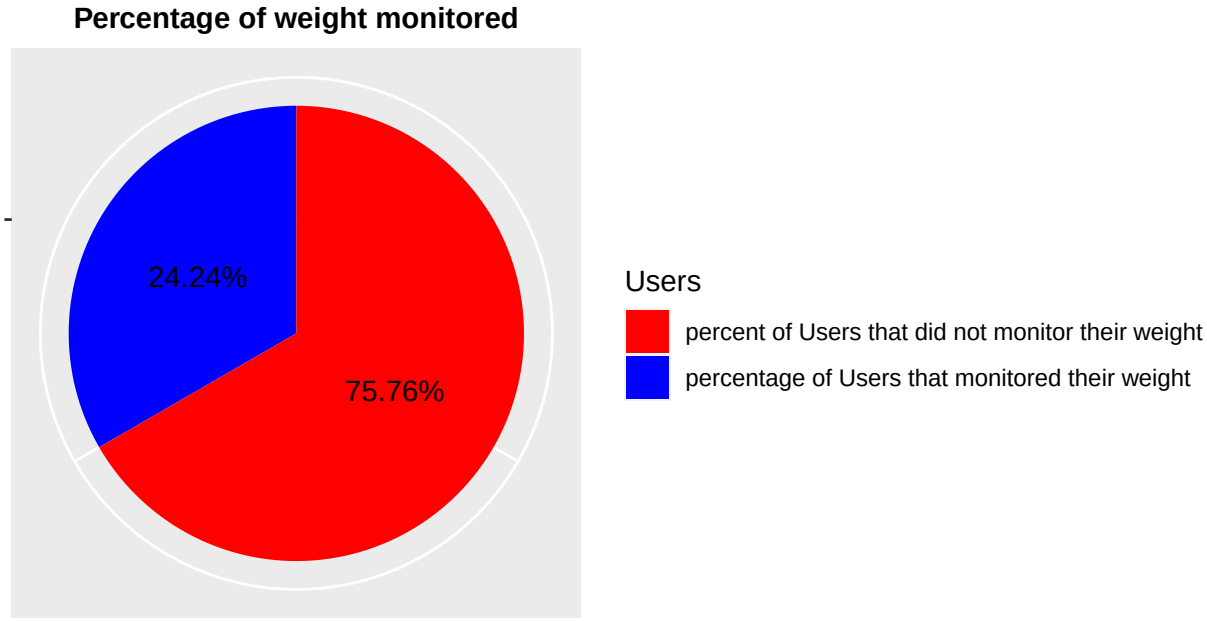
```
ggplot(data = Users_frequency,aes(x = reorder(Id, -Number_of_nights), y = Number_of_nights, fill = Numb
  geom_bar(stat = "identity")+
  geom_text(aes(label = Number_of_nights), size = 3, vjust = -0.3)+
  theme(legend.position = "none")+
  theme(axis.text.x = element_text(angle = 90))+
  scale_fill_gradient2(low = "red", mid = "purple", high = "blue")+
  labs(title = "Most Active Users in Terms of Number of Nights", x = "Id", y = "Number of Nights")
```



This bar plot depicts the number of nights that users wear their smart devices once in bed. Contrary to the previous number of days, we note that 24 of 33 users wear their smart devices once in bed which correspond exactly to the same users that tracked their sleep hours that I mentioned earlier in the average sleep hours section. This means that both sections are correlated as when they wear their smart devices during night this automatically tracks their sleep hours. On the other hand, 9 users do not wear their smart devices in the night.

Here, I plot the number of users that tracked their weight in terms of percentage.

```
Percentage_weight_monitored <- data.frame(
  Users = c("percentage of Users that monitored their weight", "percent of Users that did not monitor t
  value = c("24.24%", "75.76%")
)
ggplot(Percentage_weight_monitored, aes(x = "", y = value, fill = Users)) +
  geom_col(width = 1) +
  scale_fill_manual(values = c("red", "blue")) +
  coord_polar("y", start = 0) +
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=11, face = "bold")) +
  geom_text(aes(label = value),
            position = position_stack(vjust = 0.5))+
  labs(title="Percentage of weight monitored")
```

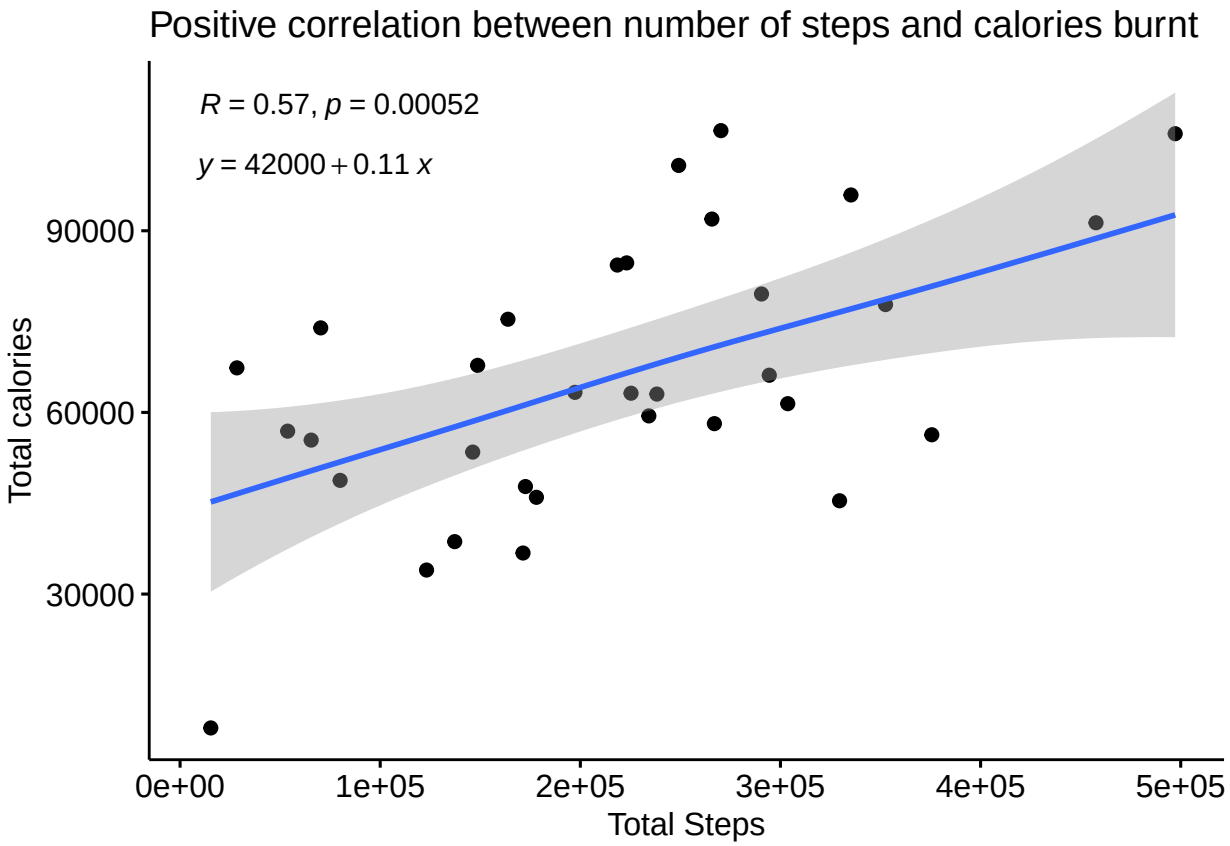


As shown on this pie chart, only 24.24% users tracked their weight which is very low. Indeed, more specifically this corresponds to 8 of 33 users that tracked their weight which demonstrates a lack of interest in this activity from 75.76% users.

Here, I plot total steps against total calories to try to find any correlation between both variables.

```
correlation <- ggscatter(
  data = Activity_Records_per_User, x = "TotalSteps", y = "TotalCalories",
) +
  geom_point(mapping = aes(x = TotalSteps, y = TotalCalories,))+
  geom_smooth(method="gam",mapping = aes(x = TotalSteps, y = TotalCalories,
                                         ))+
  stat_cor(label.x = 10000, label.y = 110500) +
  stat_regline_equation(label.x = 9000, label.y = 100500)
  ggpar(correlation,
        main ="Positive correlation between number of steps and calories burnt",
        xlab ="Total Steps", ylab="Total calories")
```

```
## `geom_smooth()` using formula 'y ~ s(x, bs = "cs")'
```



This scatter shows positive correlation between total steps and total calories burnt by users. More specifically, the higher the steps the higher the calories burnt.

I can go deeper in my analysis to explain this graph and the equation and values above in order to better understand the correlation between these variables. To do that, let's create a regression model.

```
model <- lm(TotalCalories ~ TotalSteps, data = Activity_Records_per_User)
summary(model)
```

```
##
## Call:
## lm(formula = TotalCalories ~ TotalSteps, data = Activity_Records_per_User)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -35648 -12946  -1821  15636  35165
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.187e+04  6.929e+03   6.043 1.09e-06 ***
## TotalSteps  1.092e-01  2.817e-02   3.876 0.000515 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18570 on 31 degrees of freedom
## Multiple R-squared:  0.3264, Adjusted R-squared:  0.3047
## F-statistic: 15.02 on 1 and 31 DF,  p-value: 0.0005152
```

Interpretation of the results We can see from the output that a change in one unit in Total Steps will bring 0.1092 units to change in Total Calories. Then we have the following linear regression equation:

Total Calories = 42000 + 0.11*Total Steps

Most important, the p-value is 0.0005152, almost zero, (p-value < 0.001) which indicates a significant positive correlation between Total Steps and Total Calories.

Analysis Summary

This analysis summary helps answer the first two business questions of this case study: What are some trends in smart device usage? How could these trends apply to Bellabeat customers?

- BellaBeat users sleep on average 6 hours per day which is below the 8 hours recommended by doctors.
- 50% of the users that monitored their heart rates have abnormal values that go beyond 80 heart rates per minute with 93.78 as the highest rate.
- 20 of 33 users (over 61%) wear their smart devices during the day for 30-31 days in the month while 39% wear their devices for less than 30 days.
- 24 of 33 users (over 73%) wear their smart devices in the night with only 3 users wearing for 31 days in the month. 27% users do not wear their devices at all in the night.
- 100% users wear their smart devices during the day while 27% of that group do not wear in the night.
- 24.24% users track their weight while 75.76% do not.
- There is positive correlation between total steps and total calories burnt by users. The higher the steps the higher the calories burnt.

Act phase

High-level recommendations to the executive team

This section allows to answer the third business question: How could these trends help influence Bellabeat marketing strategy? The management should apply the following recommendations to the Bellabeat app product:

- Most users have sleep hours below 8 hours which may have some consequences on their productivity and wellness during the day. Then, it is crucial to send them scheduled notifications to their smart devices to go to bed at appropriate time. They should also receive guidance (on food for example, as food plays a key role in our sleep quality) on how to have a good sleep.
- Include functionalities to monitor in which context or situation (physical activity, stress, fear, etc.) a high or critical heart rate occurred.
- Send messages to users that do not wear their smart devices on a regular basis to do so during each day and night in the month as this will help the technical team to collect reliable and update data, and consequently provide them with customized and effective guidance. To help achieve that, the technical team should add a functionality in each device (with the choice to disable at any moment by the user herself) that will alert users when it remains unused for a certain duration (1 hour for example).
- Send notifications to users to regularly track their weights as only 8 of 33 users track their weights. This is important as this helps keep an eye on their BMI and avoid obesity which is a sign of diabetes. In addition, user 1927972279 has an average weight of 133.5 kg, total steps of 28700 for only 19.67 km as total distance. This is low compared to other users. She should be advised to practice more physical activities and provided with useful guidance to reduce her weight and make healthier choices.
- Create a wellness and psychological department with specialists within BellaBeat or in partnership with a specialized firm to effectively provide customized guidance to users that need it. The Bellabeat membership product does it but partially and online. For example, the users that have 93-90 heart rate per minute must have some problems probably related to stress or fear, and this face-to-face guidance can be useful for them.


Personalize your
health & wellness
bellabeat