Case\_2\_Google\_Data\_Analytics

Yuliya Kucherenko

2023-05-21

## How Can a Wellness TechnologyCompany Play It Smart?

### Scenario:

* According to the scenario, I am a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market.
* Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. I have been asked 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 you discover will then help guide marketing strategy for the company.
* Sršen encourages me to use public data that explores smart device users’ daily habits. She points me to a specific data set: FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius).
* Sršen tells me that this data set might have some limitations, and encourages me to consider adding another data to help

## Smart devices data analysis.

**In the present study we analyze smart device fitness data to answer the following questions:**

* 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?

**We chose the Bellabeat’s Leaf smart device to apply the insight and provide recommendations based on our analysis.**

#### Bellabeat’s Leaf smart device.

* Bellabeat’s Leaf is a classic wellness tracker. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress. Since Bellabeat’s Leaf was designed to be worn as a bracelet, necklace, or clip it does not record heart rate.

#### Data source.

* The Kaggle data set (FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius)) contains personal fitness tracker from 30 fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users’ habits.
* The data set contains 18 csv files. For our analysis we used 3 data sets: dailyActivity\_merged.csv, sleepDay\_merged.csv, and weightLogInfo\_merged.csv

##### Data limitations.

* The Kaggle data set is the third party data. The data set is not completely reliable since it represents the tracking data of only 30 fitbit users dated from 4/12/2016 to 5/12/2016. Results could be different for other users or other period of time.

##### Data privacy.

* The data does not contain any personally identifing information. The user is represented with Id.

#### Data processing.

* The analysis was performed in R using RStudio. The report was created in RMarkdown using RStudio.

**Below we go step by step throughout the analysis.**

### Step 1. Importing dailyActivity\_merged dataset.

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggplot2)  
library(lubridate)  
library(dplyr)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

dailyActivity\_merged <- read\_csv("C:\\Users\\y.v.kucherenko\\Downloads\\archive(1)\\Fitabase Data 4.12.16-5.12.16\\dailyActivity\_merged.csv")

## Rows: 940 Columns: 15  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): ActivityDate  
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

glimpse(dailyActivity\_merged)

## Rows: 940  
## Columns: 15  
## $ Id <dbl> 1503960366, 1503960366, 1503960366, 150396036…  
## $ ActivityDate <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/…  
## $ TotalSteps <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019…  
## $ TotalDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8…  
## $ TrackerDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8…  
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ VeryActiveDistance <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5…  
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3…  
## $ LightActiveDistance <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0…  
## $ SedentaryActiveDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ VeryActiveMinutes <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4…  
## $ FairlyActiveMinutes <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21…  
## $ LightlyActiveMinutes <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, …  
## $ SedentaryMinutes <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818…  
## $ Calories <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203…

The dailyActivity\_merged data set consists of 940 rows and 15 columns.

### Step 2. Formating date in ActivityDate in dailyActivity\_merged.

dailyActivity\_merged$ActivityDate <- as.Date(dailyActivity\_merged$ActivityDate, "%m/%d/%Y")

After formating ActivityDate presents only date without time stamp.

### Step 3. Looking for duplicates in dailyActivity\_merged.

sum(duplicated(dailyActivity\_merged))

## [1] 0

There are no duplicated data in dailyActivity\_merged data set.

### Step 4. Looking for N/A data in dailyActivity\_merged.

sum(is.na(dailyActivity\_merged))

## [1] 0

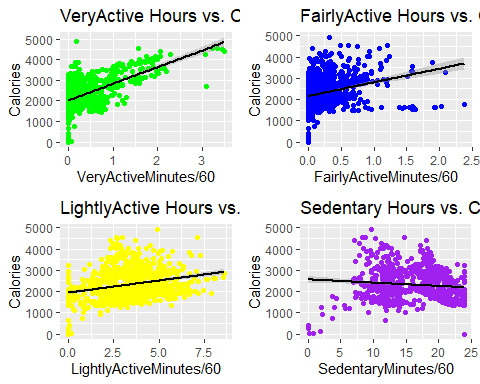
There are no N/A data in dailyActivity\_merged data set.

### Step 6. Sorting data in dailyActivity\_merged by Date in ascending order.

dailyActivity\_time\_sorted <- dailyActivity\_merged %>%  
 arrange(ActivityDate)

### Step 7. Plotting 4 types of activity (VeryActive, FairlyActive, LightlyActive and Sedentary) vs. Calories. Black line represents linear regression.

plt\_1 <- ggplot(data = dailyActivity\_time\_sorted) + geom\_point(mapping = aes(x= SedentaryMinutes/60, y = Calories), color = "purple") + stat\_smooth(mapping = aes(x= SedentaryMinutes/60, y = Calories, color = "black"), data = dailyActivity\_time\_sorted, geom = "smooth", position = "identity",method = "lm", formula = y ~ x) + labs(title = "Sedentary Hours vs. Calories") + scale\_colour\_identity()  
plt\_2 <- ggplot(data = dailyActivity\_time\_sorted) + geom\_point(mapping = aes(x= VeryActiveMinutes/60, y = Calories), color = "green") + stat\_smooth(mapping = aes(x= VeryActiveMinutes/60, y = Calories, color = "black"), data = dailyActivity\_time\_sorted, geom = "smooth", position = "identity",method = "lm", formula = y ~ x) + labs(title = "VeryActive Hours vs. Calories") + scale\_colour\_identity()  
plt\_3 <- ggplot(data = dailyActivity\_time\_sorted) + geom\_point(mapping = aes(x= FairlyActiveMinutes/60, y = Calories), color = "blue") + stat\_smooth(mapping = aes(x= FairlyActiveMinutes/60, y = Calories, color = "black"), data = dailyActivity\_time\_sorted, geom = "smooth", position = "identity",method = "lm", formula = y ~ x) + labs(title = "FairlyActive Hours vs. Calories") + scale\_colour\_identity()  
plt\_4 <- ggplot(data = dailyActivity\_time\_sorted) + geom\_point(mapping = aes(x= LightlyActiveMinutes/60, y = Calories), color = "yellow") + stat\_smooth(mapping = aes(x= LightlyActiveMinutes/60, y = Calories, color = "black"), data = dailyActivity\_time\_sorted, geom = "smooth", position = "identity",method = "lm", formula = y ~ x) + labs(title = "LightlyActive Hours vs. Calories") + scale\_colour\_identity()  
grid.arrange(plt\_2, plt\_3, ncol=2, widths=c(1,1), plt\_4, plt\_1, nrow=2)



The data, presented in the dashboard, show that the basic metabolic level is around 2000 kCal, which is average for women. High intense activity (VeryActiveMinutes/60) results in burning more calories (the highest slope of the regression line) in comparison with moderate and light activities (FairlyActiveMinutes/60 and LightlyActiveMinutes/60, respectively). The individuals who showed high level of activity for more than 3 hours a day were able to burn additionally up to 2000 kCal. In opposite, sedentary activity does not results in additional calories burn.

### Step 8. Calculating mean and sd in VeryActiveDistance, ModeratelyActiveDistance, LightActiveDistance and SedentaryActiveDistance columns.

veryActiveDistance <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_VeryActiveDistance = mean(VeryActiveDistance), sd\_VeryActiveDistance = sd(VeryActiveDistance))  
  
moderatelyActiveDistance <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_ModeratelyActiveDistance = mean(ModeratelyActiveDistance), sd\_ModeratelyActiveDistance = sd(ModeratelyActiveDistance))  
  
lightActiveDistance <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_LightActiveDistance = mean(LightActiveDistance), sd\_LightActiveDistance = sd(LightActiveDistance))  
  
sedentaryActiveDistance <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_SedentaryActiveDistance = mean(SedentaryActiveDistance), sd\_SedentaryActiveDistance = sd(SedentaryActiveDistance))

### Spep 9. Creating a dataframe for 4 types of activity\_Distance.

activity\_Dictance = data.frame(c(veryActiveDistance[1], moderatelyActiveDistance[1], lightActiveDistance[1], sedentaryActiveDistance[1]), c(veryActiveDistance[2], moderatelyActiveDistance[2], lightActiveDistance[2], sedentaryActiveDistance[2]))  
  
activity\_Distance\_df = data.frame(distance = c("Very Active", "Fairly Active", "Lightly Active", "Sedentary Active"),  
 mean = c(activity\_Dictance$mean\_VeryActiveDistance, activity\_Dictance$mean\_ModeratelyActiveDistance, activity\_Dictance$mean\_LightActiveDistance, activity\_Dictance$mean\_SedentaryActiveDistance),  
 sd = c(activity\_Dictance$sd\_VeryActiveDistance, activity\_Dictance$sd\_ModeratelyActiveDistance, activity\_Dictance$sd\_LightActiveDistance, activity\_Dictance$sd\_SedentaryActiveDistance))  
  
#View(activity\_Distance\_df)

### Step 10. Calculating mean and sd in VeryActiveMinutes, ModeratelyActiveMinutes, LightActiveMinutes and SedentaryMinutes.

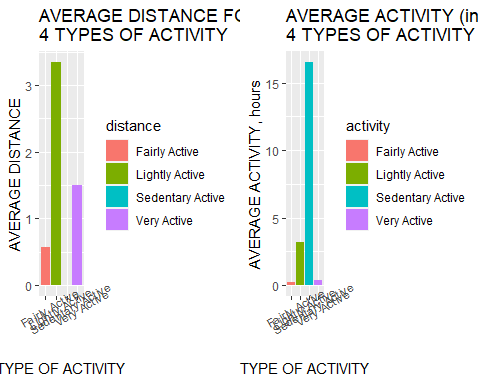
veryActiveMinutes <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_VeryActiveMinutes = mean(VeryActiveMinutes), sd\_VeryActiveMinutes = sd(VeryActiveMinutes))  
  
fairlyActiveMinutes <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_FairlyActiveMinutes = mean(FairlyActiveMinutes), sd\_FairlyActiveMinutes = sd(FairlyActiveMinutes))  
  
lightlyActiveMinutes <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_LightlyActiveMinutes = mean(LightlyActiveMinutes), sd\_LightlyActiveMinutes = sd(LightlyActiveMinutes))  
  
sedentaryMinutes <- dailyActivity\_time\_sorted %>%  
 summarize(mean\_SedentaryMinutes = mean(SedentaryMinutes), sd\_SedentaryMinutes = sd(SedentaryMinutes))

### Step 11. Creating a dataframe for 4 types of activity\_Minutes.

activity\_Minutes = data.frame(c(veryActiveMinutes[1], fairlyActiveMinutes[1], lightlyActiveMinutes[1], sedentaryMinutes[1]), c(veryActiveMinutes[2], fairlyActiveMinutes[2], lightlyActiveMinutes[2], sedentaryMinutes[2]))  
  
activity\_Minutes\_df = data.frame(activity = c("Very Active", "Fairly Active", "Lightly Active", "Sedentary Active"),  
 mean = c(activity\_Minutes$mean\_VeryActiveMinutes/60, activity\_Minutes$mean\_FairlyActiveMinutes/60, activity\_Minutes$mean\_LightlyActiveMinutes/60, activity\_Minutes$mean\_SedentaryMinutes/60),  
 sd = c(activity\_Minutes$sd\_VeryActiveMinutes/60, activity\_Minutes$sd\_FairlyActiveMinutes/60, activity\_Minutes$sd\_LightlyActiveMinutes/60, activity\_Minutes$sd\_SedentaryMinutes/60))  
  
#View(activity\_Minutes\_df)

### Step 12. Creating a bar plot for activity\_Distance and activity\_Minutes.

plot\_8 <- ggplot(data = activity\_Distance\_df) +  
 geom\_bar( aes(x = distance, y = mean, fill = distance), stat = "identity") +  
 labs(title = "AVERAGE DISTANCE FOR \n4 TYPES OF ACTIVITY", x = "TYPE OF ACTIVITY", y = "AVERAGE DISTANCE") +  
 theme(axis.text.x = element\_text(angle = 30))  
plot\_9 <- ggplot(data = activity\_Minutes\_df) +  
 geom\_bar( aes(x=activity, y=mean, fill=activity), stat="identity") +  
 labs(title = "AVERAGE ACTIVITY (in hours) FOR \n4 TYPES OF ACTIVITY", x = "TYPE OF ACTIVITY", y = "AVERAGE ACTIVITY, hours") +  
 theme(axis.text.x = element\_text(angle = 30))  
grid.arrange(plot\_8, plot\_9, ncol=2, widths=c(1,1), nrow=1)



#Calculating mean values of TotalDistance and Total Steps  
total\_distance\_1 <- mean(dailyActivity\_merged$TotalDistance)  
print(total\_distance\_1)

## [1] 5.489702

total\_steps <- mean(dailyActivity\_merged$TotalSteps)  
print(total\_steps)

## [1] 7637.911

* The results, shown in the left panel, show that the Sedentary activity is the dominant type of activity of the individuals, whose data were presented in dailyActivity\_merged data set. An average amount of time spent for sedentary activity was 16.52 hours a day, followed by 3.21 hours for light activity. Individuals were very active for 0.35 hours and fairly active for 0.23 hours a day respectively.
* The right panel shows the average distance the individuals, whose data were presented in dailyActivity\_merged data set, make during a day. The average Lightly active distance is 3.34 (the units are not defined, supposedly, miles), followed by 1.50 for Very active distance and 0.57 for Fairaly active distance, respectively.\*\*
* The average Total Distance for the individuals, whose data were presented in dailyActivity\_merged data set, is 5.49 and average Total Steps is 7638, respectively.

### Step 13. Importing sleepDay\_merged dataset.

sleepDay\_merged <- read\_csv("C:\\Users\\y.v.kucherenko\\Downloads\\archive(1)\\Fitabase Data 4.12.16-5.12.16\\sleepDay\_merged.csv")

## Rows: 413 Columns: 5  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): SleepDay  
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

glimpse(sleepDay\_merged)

## Rows: 413  
## Columns: 5  
## $ Id <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150…  
## $ SleepDay <chr> "4/12/2016 12:00:00 AM", "4/13/2016 12:00:00 AM", "…  
## $ TotalSleepRecords <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2…  
## $ TotalTimeInBed <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3…

The data set consists of 413 rows and 5 columns.

### Step 14. Formating date in sleepDay\_merged data set.

sleepDay\_merged$SleepDay <- as.Date(sleepDay\_merged$SleepDay, "%m/%d/%Y")

### Step 15. Looking for duplicated records in sleepDay\_merged data set.

sum(duplicated(sleepDay\_merged))

## [1] 3

There are 3 duplicated records in the data set.

### Step 16. Removing duplicated records and assigning the unique records to in sleepDay data set.

sleepDay <- unique(sleepDay\_merged)  
sum(duplicated(sleepDay))

## [1] 0

There no duplicated records in sleepDay data set.

### Step 17. Looking for N/A values in sleepDay data set.

sum(is.na(sleepDay))

## [1] 0

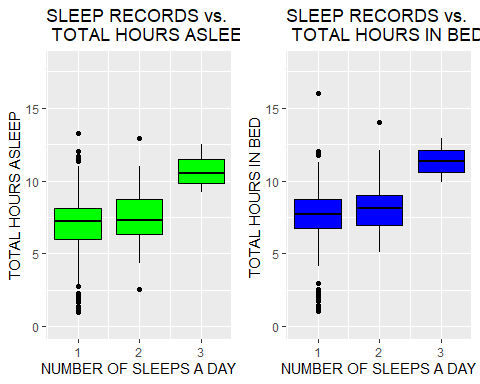
glimpse(sleepDay)

## Rows: 410  
## Columns: 5  
## $ Id <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150…  
## $ SleepDay <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-16, 20…  
## $ TotalSleepRecords <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2…  
## $ TotalTimeInBed <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3…

There are no N/A values in sleepDay data set. The data set consists of 410 rows and 5 columns.

### Step 18. Creating a pointplot and boxplot for TotalTimeInBed and TotalTimeAsleep using sleepDay dataframe.

plot\_13 <- ggplot(data = sleepDay) + geom\_boxplot(mapping = aes(x = TotalSleepRecords, y = TotalMinutesAsleep/60, group = TotalSleepRecords), color = "black", fill = "green") + ylim(0, 18) + labs(title = "SLEEP RECORDS vs.\n TOTAL HOURS ASLEEP", x = "NUMBER OF SLEEPS A DAY", y = "TOTAL HOURS ASLEEP")  
plot\_14 <- ggplot(data = sleepDay) + geom\_boxplot(mapping = aes(x = TotalSleepRecords, y = TotalTimeInBed/60, group = TotalSleepRecords), color = "black", fill = "blue") + ylim(0, 18) + labs(title = "SLEEP RECORDS vs.\n TOTAL HOURS IN BED", x = "NUMBER OF SLEEPS A DAY", y = "TOTAL HOURS IN BED ")  
grid.arrange(plot\_13, plot\_14, ncol=2, widths=c(1,1), nrow=1)



As it is shown in the left panel of the dashboard, most individuals, whose data are presented in sleepyDay\_merged data set, have either 1 or 2 sleeps a day. The average length of sleep varies from 6.89 hours for 1 sleep a day to 7.55 hours for 2 sleeps a day. In rare cases the individuals had 3 sleeps a days of a total average length of 10.73 hours a day. The right panel of the dashboard shows that additionally to a sleap the individuals laid in bed to up to 0.8 hours.

### Step 19. Importing weightLogInfo\_merged.csv.

weightLogInfo\_merged <- read\_csv("C:\\Users\\y.v.kucherenko\\Downloads\\archive(1)\\Fitabase Data 4.12.16-5.12.16\\weightLogInfo\_merged.csv")

## Rows: 67 Columns: 8  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): Date  
## dbl (6): Id, WeightKg, WeightPounds, Fat, BMI, LogId  
## lgl (1): IsManualReport  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

glimpse(weightLogInfo\_merged)

## Rows: 67  
## Columns: 8  
## $ Id <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 2873212…  
## $ Date <chr> "5/2/2016 11:59:59 PM", "5/3/2016 11:59:59 PM", "4/13/2…  
## $ WeightKg <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70.3, …  
## $ WeightPounds <dbl> 115.9631, 115.9631, 294.3171, 125.0021, 126.3249, 159.6…  
## $ Fat <dbl> 22, NA, NA, NA, NA, 25, NA, NA, NA, NA, NA, NA, NA, NA,…  
## $ BMI <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27.25,…  
## $ IsManualReport <lgl> TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, …  
## $ LogId <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.461283e+12,…

The data set consists of 67 rows and 8 columns.

### Step 20. Formating date in weightLogInfo\_merged Date column.

weightLogInfo\_merged$Date <- as.Date(weightLogInfo\_merged$Date, "%m/%d/%Y")

### Step 21. Looking for duplicates in weightLogInfo\_merged data set.

sum(duplicated(weightLogInfo\_merged))

## [1] 0

There are 3 records in weightLogInfo\_merged data set.

### Step 22. removing duplicates and assigning the data set to weightLogInfo.

weightLogInfo <- unique(weightLogInfo\_merged)  
sum(duplicated(weightLogInfo))

## [1] 0

### Step 23. Looking for N/A data in weightLogInfo data set.

sum(is.na(weightLogInfo))

## [1] 65

N/A values are present in Fat column of weightLogInfo\_merged data set.

### Step 24. Removing Fat, IsManualReport and LogId from weightLogInfo data set.

weightLog <- select(weightLogInfo, -"Fat", -"IsManualReport", -"LogId")

### Step 25. Filtering BMI column for normal (BMI <= 25), overweight (BMI > 25) and obese (BMI > 30) groups.

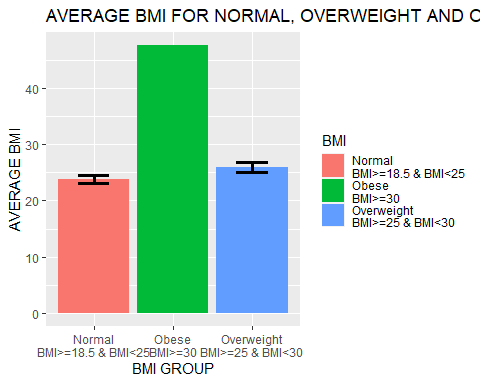
normal\_group <- weightLog %>%   
 filter(BMI >= 18.5 & BMI < 25)  
mean\_normal <- mean(normal\_group$BMI)  
sd\_normal <- sd(normal\_group$BMI)  
  
overweight\_group <- weightLog %>%   
 filter(BMI >= 25 & BMI < 30)  
mean\_overweight <- mean(overweight\_group$BMI)  
sd\_overweight <- sd(overweight\_group$BMI)  
  
obese\_group <- weightLog %>%   
 filter(BMI >= 30)  
mean\_obese <- mean(obese\_group$BMI)  
sd\_obese <- sd(obese\_group$BMI)

### Step 26. Creating dataframe of BMI for 3 different groups.

data\_BMI <- data.frame(BMI = c("Normal\nBMI>=18.5 & BMI<25", "Overweight\nBMI>=25 & BMI<30", "Obese\nBMI>=30"),  
 mean = c(mean\_normal, mean\_overweight, mean\_obese),  
 sd = c(sd\_normal, sd\_overweight, sd\_obese))

### Step 27. Plotting boxplots for normal, overweight and obese groups.

ggplot(data = data\_BMI) +  
 geom\_bar( aes(x=BMI, y=mean, fill= BMI), stat="identity") +  
 geom\_errorbar( aes(x=BMI, ymin = mean-sd, ymax=mean+sd), width=0.4, colour="black", linewidth=1.3) +  
 labs(title = "AVERAGE BMI FOR NORMAL, OVERWEIGHT AND OBESE GROUPS", x = "BMI GROUP", y = "AVERAGE BMI")



As it is shown in the plot, the majority of the individuals, whose data are presented in weightLogInfo\_merged data set, are either of normal or overweight group. Only one individual was in the obese group (BMI = 47.54). It is important to notice. that the individuals in the normal weight group (BMI <= 25) have an average BMI value of 23.8 that is close to the upper high level of normal BMI.

Since BMI, sleep, TotalDistance and Calories columns are located in different data sets, we combined 3 described above data sets(dailyActivity\_merged, sleepDay\_merged, and weightLogInfo\_merged) into one data set using inner\_joint.

### Step 28. Merging the data (inner\_join) by Id, ActivityDate, SleepDay and Date columns.

knitr::opts\_chunk$set(error = TRUE)  
two\_df\_merged <- dailyActivity\_merged %>% inner\_join(sleepDay\_merged, by = c("ActivityDate" = "SleepDay", "Id" = "Id"))  
   
all\_data\_merged <- two\_df\_merged %>% inner\_join(weightLogInfo\_merged, by = c("Id" = "Id", "ActivityDate" = "Date"))  
  
glimpse(all\_data\_merged)

## Rows: 35  
## Columns: 24  
## $ Id <dbl> 1503960366, 1503960366, 1927972279, 455860992…  
## $ ActivityDate <date> 2016-05-02, 2016-05-03, 2016-04-13, 2016-05-…  
## $ TotalSteps <dbl> 14727, 15103, 356, 3428, 12231, 10199, 5652, …  
## $ TotalDistance <dbl> 9.71, 9.66, 0.25, 2.27, 9.14, 6.74, 3.74, 1.0…  
## $ TrackerDistance <dbl> 9.71, 9.66, 0.25, 2.27, 9.14, 6.74, 3.74, 1.0…  
## $ LoggedActivitiesDistance <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000…  
## $ VeryActiveDistance <dbl> 3.21, 3.73, 0.00, 0.00, 5.98, 3.40, 0.57, 0.0…  
## $ ModeratelyActiveDistance <dbl> 0.57, 1.05, 0.00, 0.00, 0.83, 0.83, 1.21, 0.0…  
## $ LightActiveDistance <dbl> 5.92, 4.88, 0.25, 2.27, 2.32, 2.51, 1.96, 1.0…  
## $ SedentaryActiveDistance <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0…  
## $ VeryActiveMinutes <dbl> 41, 50, 0, 0, 200, 50, 8, 0, 0, 50, 5, 13, 35…  
## $ FairlyActiveMinutes <dbl> 15, 24, 0, 0, 37, 14, 24, 0, 0, 3, 13, 42, 41…  
## $ LightlyActiveMinutes <dbl> 277, 254, 32, 190, 159, 189, 142, 86, 217, 28…  
## $ SedentaryMinutes <dbl> 798, 816, 986, 1121, 525, 796, 548, 862, 837,…  
## $ Calories <dbl> 2004, 1990, 2151, 1692, 4552, 1994, 1718, 146…  
## $ TotalSleepRecords <dbl> 1, 1, 1, 1, 1, 1, 3, 2, 1, 1, 1, 1, 1, 1, 1, …  
## $ TotalMinutesAsleep <dbl> 277, 273, 398, 115, 549, 366, 630, 508, 370, …  
## $ TotalTimeInBed <dbl> 309, 296, 422, 129, 583, 387, 679, 535, 386, …  
## $ WeightKg <dbl> 52.6, 52.6, 133.5, 69.9, 90.7, 62.5, 62.1, 61…  
## $ WeightPounds <dbl> 115.9631, 115.9631, 294.3171, 154.1031, 199.9…  
## $ Fat <dbl> 22, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  
## $ BMI <dbl> 22.65, 22.65, 47.54, 27.32, 28.00, 24.39, 24.…  
## $ IsManualReport <lgl> TRUE, TRUE, FALSE, TRUE, FALSE, TRUE, TRUE, T…  
## $ LogId <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.4…

The new data set all\_data\_merged consists of only 35 rows and 24 columns.

### Step 29. Looking for duplicates and N/A vales.

sum(duplicated(all\_data\_merged))

## [1] 0

sum(is.na(all\_data\_merged))

## [1] 34

### Step 30. Removing “Fat”, “IsManual” and LogId columns from all\_data\_merged and assigning to all\_data data set; sorting ActivityDate in all\_data in ascending order.

all\_data <- select(all\_data\_merged, -"Fat", -"IsManualReport", -"LogId")  
all\_data <- arrange(all\_data, ActivityDate)  
sum(is.na(all\_data))

## [1] 0

sum(duplicated(all\_data))

## [1] 0

glimpse(all\_data)

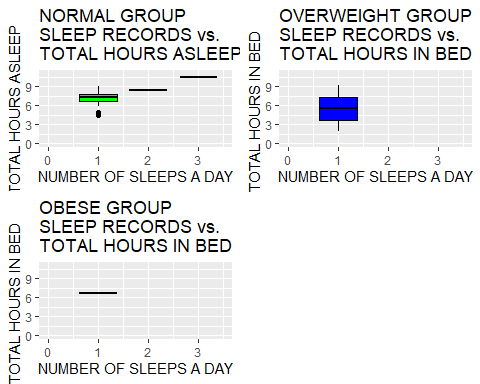
## Rows: 35  
## Columns: 21  
## $ Id <dbl> 6962181067, 1927972279, 6962181067, 696218106…  
## $ ActivityDate <date> 2016-04-12, 2016-04-13, 2016-04-13, 2016-04-…  
## $ TotalSteps <dbl> 10199, 356, 5652, 1551, 5563, 13217, 12231, 1…  
## $ TotalDistance <dbl> 6.74, 0.25, 3.74, 1.03, 3.68, 8.74, 9.14, 6.7…  
## $ TrackerDistance <dbl> 6.74, 0.25, 3.74, 1.03, 3.68, 8.74, 9.14, 6.7…  
## $ LoggedActivitiesDistance <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000…  
## $ VeryActiveDistance <dbl> 3.40, 0.00, 0.57, 0.00, 0.00, 3.66, 5.98, 0.3…  
## $ ModeratelyActiveDistance <dbl> 0.83, 0.00, 1.21, 0.00, 0.00, 0.19, 0.83, 0.6…  
## $ LightActiveDistance <dbl> 2.51, 0.25, 1.96, 1.03, 3.68, 4.88, 2.32, 5.6…  
## $ SedentaryActiveDistance <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0…  
## $ VeryActiveMinutes <dbl> 50, 0, 8, 0, 0, 50, 200, 5, 13, 35, 48, 53, 3…  
## $ FairlyActiveMinutes <dbl> 14, 0, 24, 0, 0, 3, 37, 13, 42, 41, 4, 27, 33…  
## $ LightlyActiveMinutes <dbl> 189, 32, 142, 86, 217, 280, 159, 295, 238, 19…  
## $ SedentaryMinutes <dbl> 796, 986, 548, 862, 837, 741, 525, 634, 689, …  
## $ Calories <dbl> 1994, 2151, 1718, 1466, 1756, 2173, 4552, 202…  
## $ TotalSleepRecords <dbl> 1, 1, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ TotalMinutesAsleep <dbl> 366, 398, 630, 508, 370, 357, 549, 427, 442, …  
## $ TotalTimeInBed <dbl> 387, 422, 679, 535, 386, 366, 583, 446, 458, …  
## $ WeightKg <dbl> 62.5, 133.5, 62.1, 61.7, 61.5, 62.0, 90.7, 61…  
## $ WeightPounds <dbl> 137.7889, 294.3171, 136.9071, 136.0252, 135.5…  
## $ BMI <dbl> 24.39, 47.54, 24.24, 24.10, 24.00, 24.21, 28.…

### Step 31. Filtering all\_data for 3 types of BMI.

obese\_data <- all\_data %>%  
 filter(BMI >=30)  
  
overweight\_data <- all\_data %>%  
 filter(BMI >= 25 & BMI < 30)  
  
normal\_data <- all\_data %>%   
 filter(BMI >= 18.5 & BMI < 25)

### Step 32. Plotting normal\_data, overweight\_data and obese\_data as TotalSleepRecords vs. TotalMinutesAsleep/60.

plot\_19 <- ggplot(data = normal\_data) + geom\_boxplot(mapping = aes(x = TotalSleepRecords, y = TotalMinutesAsleep/60, group = TotalSleepRecords), color = "black", fill = "green") + ylim(0, 11) + xlim(0, 3.5) + labs(title = "NORMAL GROUP\nSLEEP RECORDS vs.\nTOTAL HOURS ASLEEP", x = "NUMBER OF SLEEPS A DAY", y = "TOTAL HOURS ASLEEP")  
plot\_20 <- ggplot(data = overweight\_data) + geom\_boxplot(mapping = aes(x = TotalSleepRecords, y = TotalMinutesAsleep/60, group = TotalSleepRecords), color = "black", fill = "blue") + ylim(0, 11) + xlim(0, 3.5) +labs(title = "OVERWEIGHT GROUP\nSLEEP RECORDS vs.\nTOTAL HOURS IN BED", x = "NUMBER OF SLEEPS A DAY", y = "TOTAL HOURS IN BED ")  
plot\_21 <- ggplot(data = obese\_data) + geom\_boxplot(mapping = aes(x = TotalSleepRecords, y = TotalMinutesAsleep/60, group = TotalSleepRecords), color = "black", fill = "yellow") + ylim(0, 11) + xlim(0, 3.5) +labs(title = "OBESE GROUP\nSLEEP RECORDS vs.\nTOTAL HOURS IN BED", x = "NUMBER OF SLEEPS A DAY", y = "TOTAL HOURS IN BED ")  
grid.arrange(plot\_19, plot\_20, ncol=2, widths=c(1,1), plot\_21, nrow=2)



The data, presented in the dashbord, show that individuals with normal BMI value have longer average value of sleep (7.29 hours) than those in overweight and obese groups (5.53 hours and 6.63 hours, respectively). Only the individuals with normal BMI value had more than 1 sleep a day. However, it is important to notice that number of individuals in the normal group was much higher (33 persons) than in obese group and overweight group (1 and 2 individuals, respectively).

## Summary of the analysis findings.

* In the present analysis we have used the Kaggle data set (FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius)) to get useful information for Bellabeat’s Leaf users. The data obtained from the three FitBit’s data sets (dailyActivity\_merged, sleepDay\_merged, and weightLogInfo\_merged) provide the data for daily activity such as total steps, total distance, calories, activity type, activity duration, sleep duration, time in bed, BMI etc.
* According to the information, given on Bellabeat website [link](https://bellabeat.com/), The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress. Our analysis using the FitBit Fitness Tracker Data provides the Bellabeat’s Leaf customers with the data insights obtained from the similar to Bellabeat Leaf devices.

### Our data analysis finding suggest the following:

* The group of individuals, whose data are included in the dailyActivity\_merged data set, are more likely women, since the basic daily calories level is approximately 2000 kCal. There is a wide distribution in daily activity within the group. However, the analysis suggests that the sedentarily and light intense activities are the dominant ones within the group. An average amount of time spent for sedentary activity was 16.52 hours a day, followed by 3.21 hours for light activity. Individuals were very active for 0.35 hours and fairly active for 0.23 hours a day respectively.
* The average daily total distance value for the individuals, whose data were presented in dailyActivity\_merged data set, is 5.49 and average value for total steps is 7638, respectively.
* Data, presented in sleepyDay\_merged data set, show that that the most of individuals have either 1 or 2 sleeps a day. The average length of sleep varies from 6.89 hours for 1 sleep a day to 7.55 hours for 2 sleeps a day. In rare cases the individuals had 3 sleeps a days of a total average duration of 10.73 hours. Additionally, the individuals laid in bed without a sleep up to 0.8 hours.
* The majority of the individuals, whose data are presented in weightLogInfo\_merged data set, are either of normal or overweight group. Only one individual was in the obese group (BMI = 47.54). However, the individuals in the normal weight group (BMI <= 25) have an average BMI value of 23.8 that is close to the upper high level of normal BMI.
* The results, shown in all\_data\_merged data set (which includes inner joined data for the three mentioned above data sets (dailyActivity\_merged, sleepyDay\_merged and weightLogInfo\_merged)), suggest that the individuals with normal BMI have slightly longer sleep time duration than overweight or obese individuals. However, the conclusion has a certain limitation due to small number of individuals in overweight and obese groups.

## Bellabeat business strategy recommendations.

According to the **Business Insider** review [link](https://www.businessinsider.com/guides/tech/best-smart-jewelry), Bellabeat Leaf is **The best piece of smart jewelry overall**.

* **Pros:** Versatile design, eco-friendly materials, great mindfulness features, fitness tracking, and you never have to charge it.
* **Cons:** No notifications and replacing the battery after six months requires removing small screws.
* The picture of Bellabeat’s Leaf wellness tracker  
  

1. Since Bellabeat’s Leaf was designed to be worn as a bracelet, necklace, or clip it does not record heart rate. This is the main disadvantage of Bellabeat’s Leaf.
2. Another problem of the Bellabeat Leaf smart device is a lack of notification. The competitive smart devices have a notification option, which allows to inform the customer if there is a necessity of more intense daily physical activity or there is an indication that the customer’s sleep pattern shows a stress sign.
3. We suggest that the Bellabeat Leaf design could be slightly improved to meet the customers expectations and the Bellabeat App should provide more features for the customers to help them to maintain healthy lifestyle. Since the BellaBeat Leaf customers are mostly women, the results of our analysis could give some insights about similar smart devices users and their lifestyle.