Bellabeat

david

2022-05-31

Bellabeat: how can a wellness technology company play it smart?

This is an optional capstone project from the Google Data Analytics Course no. 8: Capstone Project. The analysis follows the 6 steps of Data Analysis taught in the Google course: Ask, Prepare, Process, Analyse, Share and Act.

STEP 1: ASK

Business Task:

Analyze FitBit fitness tracker data to gain insights into how consumers are using the FitBit app and discover trends for Bellabeat marketing strategy.

Business Objectives:

- What are the trends identified? -How could these trends apply to Bellabeat customers? -How could these trends help influence Bellabeat marketing strategy?

Deliverables:

-A clear summary of the business task -A description of all data sources used - Documentation of any cleaning or manipulation of data -A summary of analysis -Supporting visualizations and key findings -High-level content recommendations based on the analysis

Key Stakeholders:

- Urška Sršen: Bellabeat's cofounder and Chief Creative Officer -Sando Mur: Mathematician, Bellabeat's cofounder and key member of the Bellabeat executive team -Bellabeat marketing analytics team: A team of data analysts guiding Bellabeat's marketing strategy.

STEP 2: PREPARE

In the Prepare phase, we identify the data being used and its limitations.

Information on Data Source:

- Data is publicly available on Kaggle: FitBit Fitness Tracker Data and stored in 18 csv files. -Generated by respondents from a survey via Amazon Mechanical Turk between 12 March 2016 to 12 May 2016. -30 FitBit users consented to the submission of personal tracker data. -Data collected includes physical activity recorded in minutes, heart rate, sleep monitoring, daily activity and steps.

Limitations of Data Set:

-Data is collected 5 years ago in 2016. Users' daily activity, fitness and sleeping habits, diet and food consumption may have changed since then. Data may not be timely or relevant. -Sample size of 30 FitBit users is not representative of the entire fitness population. -As data is collected in a survey, we are unable to ascertain its integrity or accuracy.

STEP 3: PROCESS

We are using Rstudio for data cleaning, transformation and visualisation.

we import all the necesary libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
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(rmarkdown)
library(skimr)
```

we import our Data, we change the name of the date frames to make them easier to write

```
library(readr)
 WL <- read_csv("C:/Users/LENOVO/Desktop/PORTFOLIO/Fitabase Data 4.12.16-5.12.16/weightLogInfo_merged.cs
     col_types = cols(Id = col_number()))
 View(WL)
 library(readr)
 SD <- read_csv("C:/Users/LENOVO/Desktop/PORTFOLIO/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv",
     col_types = cols(Id = col_number()))
 View(SD)
 library(readr)
 DA <- read_csv("C:/Users/LENOVO/Desktop/PORTFOLIO/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.cs
     col_types = cols(Id = col_number()))
 View(DA)
```

we use the clean_names function to check that the column names are consistent

```
DA2 <- clean_names(DA)
 SD2 <- clean_names(SD)
 WL2 <- clean_names(WL)
 View(DA2)
 View(SD2)
 View(WL2)
```

We standardize date format

```
DA2$activity_date <- as.Date(DA2$activity_date, "%m/%d/%Y")
 WL2$date <- parse_date_time(WL2$date, orders = 'mdy HMS')
 WL2$date <- as.Date(WL2$date, "%m/%d/%y %h:%m:%s")
```

```
## Warning in as.POSIXlt.POSIXct(x, tz = tz): unknown timezone '%m/%d/%y %h:%m:%s'
```

```
 SD2$sleep_day <- parse_date_time(SD2$sleep_day, orders = 'mdy HMS')
 SD2$sleep_day <- as.Date(SD2$sleep_day, "%m/%d/%y %h:%m:%s")
```

```
## Warning in as.POSIXlt.POSIXct(x, tz = tz): unknown timezone '%m/%d/%y %h:%m:%s'
```

We examining the structure of data frames after formatting

```
str(SD2)
```

```
## spec_tbl_df [413 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id                 : num [1:413] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ sleep_day          : Date[1:413], format: "2016-04-12" "2016-04-13" ...
##  $ total_sleep_records : num [1:413] 1 2 1 2 1 1 1 1 1 1 ...
##  $ total_minutes_asleep: num [1:413] 327 384 412 340 700 304 360 325 361 430 ...
##  $ total_time_in_bed   : num [1:413] 346 407 442 367 712 320 377 364 384 449 ...
```

```
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_number(),
##   ..   SleepDay = col_character(),
##   ..   TotalSleepRecords = col_double(),
##   ..   TotalMinutesAsleep = col_double(),
##   ..   TotalTimeInBed = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(WL2)
```

```
## spec_tbl_df [67 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id              : num [1:67] 1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ date            : Date[1:67], format: "2016-05-02" "2016-05-03" ...
##  $ weight_kg       : num [1:67] 52.6 52.6 133.5 56.7 57.3 ...
##  $ weight_pounds   : num [1:67] 116 116 294 125 126 ...
##  $ fat             : num [1:67] 22 NA NA NA NA 25 NA NA NA NA ...
##  $ bmi             : num [1:67] 22.6 22.6 47.5 21.5 21.7 ...
##  $ is_manual_report: logi [1:67] TRUE TRUE FALSE TRUE TRUE TRUE ...
##  $ log_id          : num [1:67] 1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_number(),
##   ..   Date = col_character(),
##   ..   WeightKg = col_double(),
##   ..   WeightPounds = col_double(),
##   ..   Fat = col_double(),
##   ..   BMI = col_double(),
##   ..   IsManualReport = col_logical(),
##   ..   LogId = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(DA2)
```

```
## spec_tbl_df [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id                         : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ activity_date              : Date[1:940], format: "2016-04-12" "2016-04-13" ...
##  $ total_steps                : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ total_distance             : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ tracker_distance           : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ logged_activities_distance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_distance       : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ moderately_active_distance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ light_active_distance      : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ sedentary_active_distance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ very_active_minutes        : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ fairly_active_minutes      : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ lightly_active_minutes     : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ sedentary_minutes          : num [1:940] 728 776 1218 726 773 ...
##  $ calories                   : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
```

```
##   .. cols(
##   ..   Id = col_number(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

we check for any null values

```
SD2%>%
    is.na() %>%
    sum()
```

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

```
DA2%>%
    is.na() %>%
    sum()
```

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

```
WL2%>%
    is.na() %>%
    sum()
```

```
## [1] 65
```

We notice there are no null values in DA2 and SD2, but there are 65 null values in WL2, witch indicates that the data frame might not be of our interest.

we count the distinct values in the Id column to see how many users are there in every table

```
distinct(WL2,id)
```

```
## # A tibble: 8 x 1
##           id
##        <dbl>
## 1 1503960366
## 2 1927972279
## 3 2873212765
```

```
## 4 4319703577
## 5 4558609924
## 6 5577150313
## 7 6962181067
## 8 8877689391
```

```
distinct(SD2,id)
```

```
## # A tibble: 24 x 1
##            id
##         <dbl>
##  1 1503960366
##  2 1644430081
##  3 1844505072
##  4 1927972279
##  5 2026352035
##  6 2320127002
##  7 2347167796
##  8 3977333714
##  9 4020332650
## 10 4319703577
## # ... with 14 more rows
```

```
distinct(DA2,id)
```

```
## # A tibble: 33 x 1
##            id
##         <dbl>
##  1 1503960366
##  2 1624580081
##  3 1644430081
##  4 1844505072
##  5 1927972279
##  6 2022484408
##  7 2026352035
##  8 2320127002
##  9 2347167796
## 10 2873212765
## # ... with 23 more rows
```

There are only 8 unique users id in the WL2 data frame, so there is not enough information for the table to be useful for our analysis,

Next we evaluate the duplicate rows in our data frames.

```
sum(duplicated(DA2))
```

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

```
sum(duplicated(SD2))
```

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

we remove the duplicate data in the SD2 table

```
SD3 <- SD2[!duplicated(SD2), ]
sum(duplicated(SD3))
```

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

we merge the two data frames, as we previously observed the DA2 table has 33 unique users while SD3 only has 24, so we will use left merge to include all the values from the DA2 table.

```
DA_SD <- merge(x= DA2, y= SD3,
                                by.x = c("id", "activity_date"), by.y = c("id", "sleep_day"), all.x = TRU
 DA_SD [is.na(DA_SD)] <- 0
 View(DA_SD)
```

We add a new colum for month and anothe for day of the week

```
DA_SD$month <- format(DA_SD$activity_date,"%B")
DA_SD$day_of_week <- format(DA_SD$activity_date,"%A")
```

STEP 4: ANALYZE

We start analizing the data to gain insights

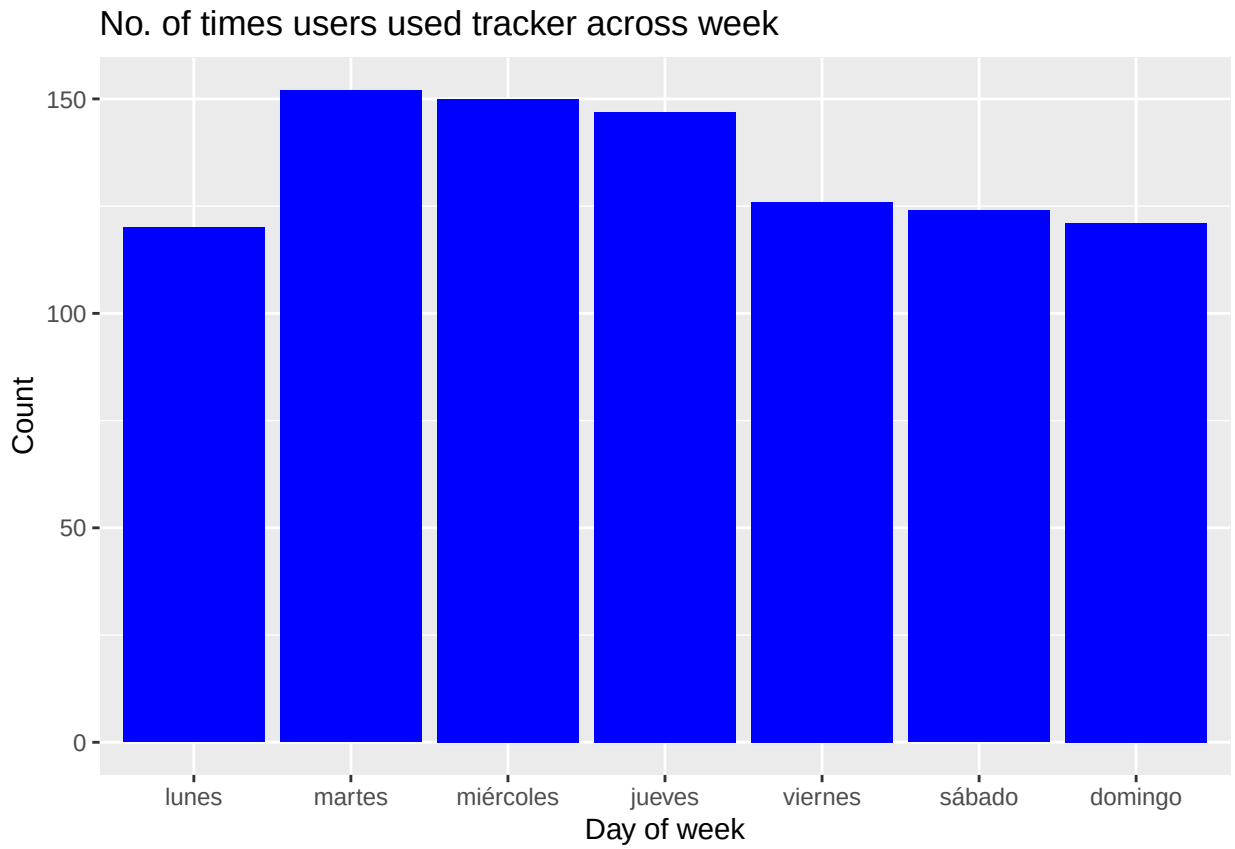
```
DA_SD%>%  select(total_steps,total_distance,sedentary_minutes,very_active_minutes) %>% summary()
```

```
##   total_steps    total_distance   sedentary_minutes very_active_minutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0    Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8    1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :1057.5    Median :  4.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2    Mean   : 21.16
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5    3rd Qu.: 32.00
##  Max.   :36019   Max.   :28.030   Max.   :1440.0    Max.   :210.00
```

The average count of recorded steps is 7638 which is less than recommended 10000 steps and average of total distance covered is 5.490 km which is also less than recommended 8 km mark. The average sedentary minutes is 991.2 minutes or 16.52 hours which is very high as it should be at most 7 hours.Even if you are doing enough physical activity, sitting for more than 7 to 10 hours a day is bad for your health. (source: CDC). This indicates that users are using the FitBit app to log daily activities such as daily commute, inactive movements (moving from one spot to another) or running errands. The average of very active minutes is 21.16 which is less than target of 30 minutes per day.

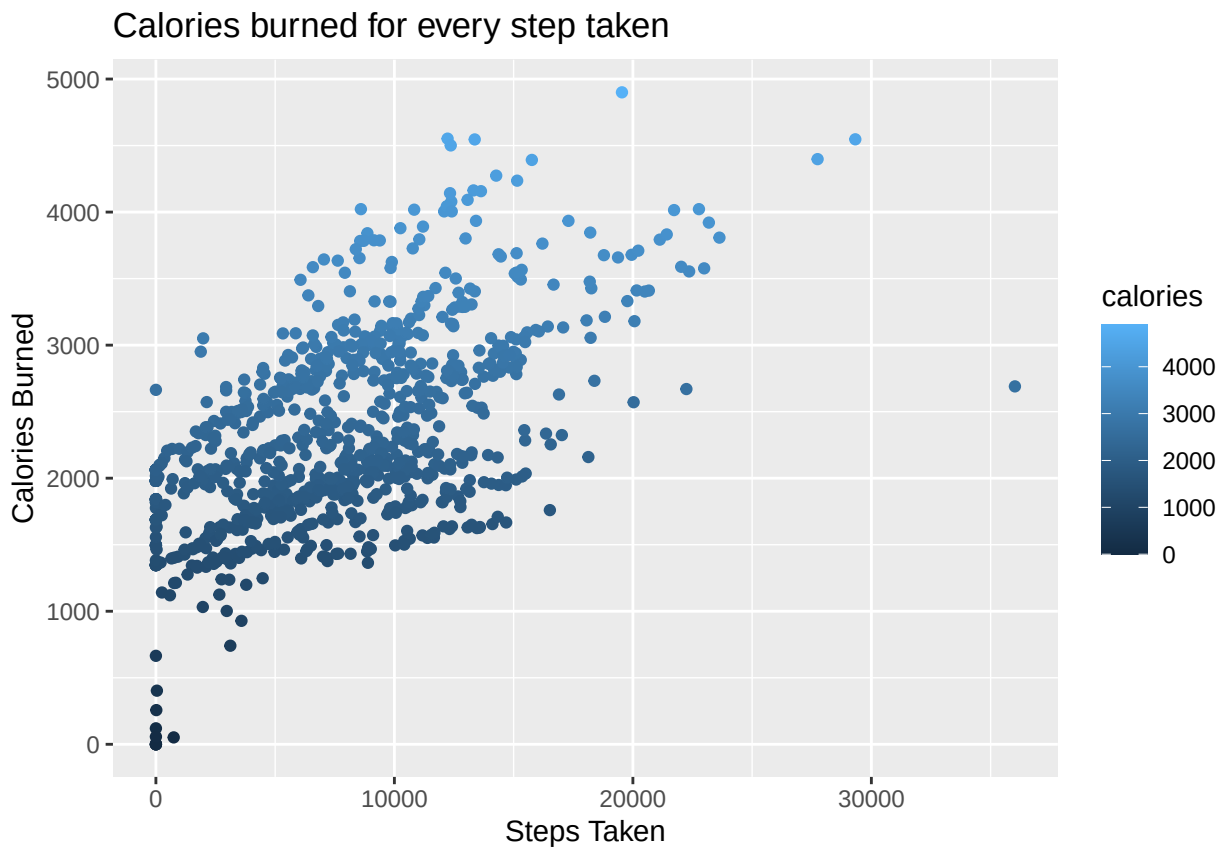
STEP 5: SHARE

```
DA_SD$day_of_week <- ordered(DA_SD$day_of_week,levels=c("lunes","martes","miércoles","jueves","viernes"
ggplot(data=DA_SD) + geom_bar(mapping = aes(x=day_of_week),fill="blue") +
         labs(x="Day of week",y="Count",title="No. of times users used tracker across week")
```



The plot shows that the frequency of usage of FitBit fitness tracker application is high on TuesdayS, WednesdayS and ThursdayS , this could be due to the fact people might get distracted on weekends, or may be too busy on Mondays.

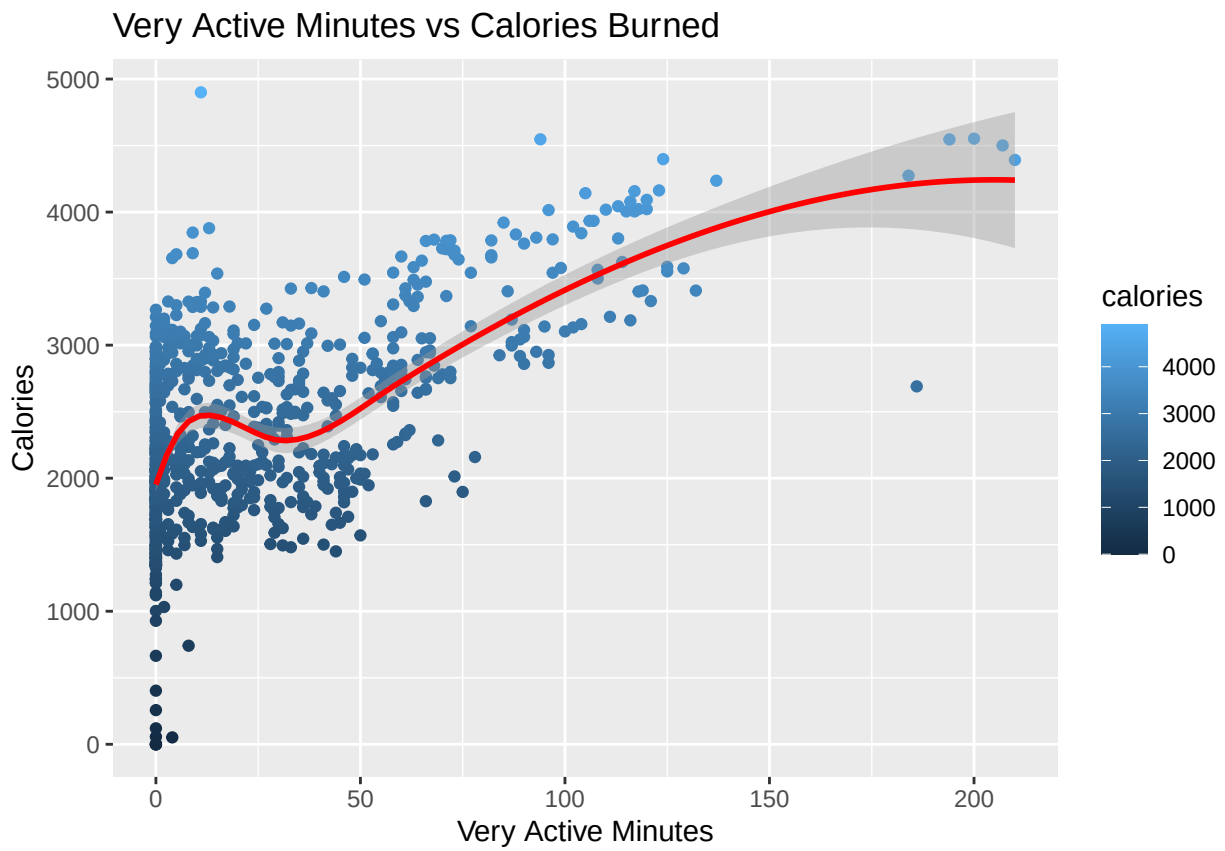
```
ggplot(data=DA_SD) + geom_point(mapping=aes(x=total_steps, y=calories, color=calories)) +
      labs(x="Steps Taken",y="Calories Burned",title = "Calories burned for every step taken")
```



We can se a positive correlation between calories burn and steps taken.

```
ggplot(data=DA_SD,aes(x = very_active_minutes, y = calories, color = calories)) + geom_point() +
        geom_smooth(method = "loess",color="red") +
        labs(x="Very Active Minutes",y="Calories",title = "Very Active Minutes vs Calories Burned")
```

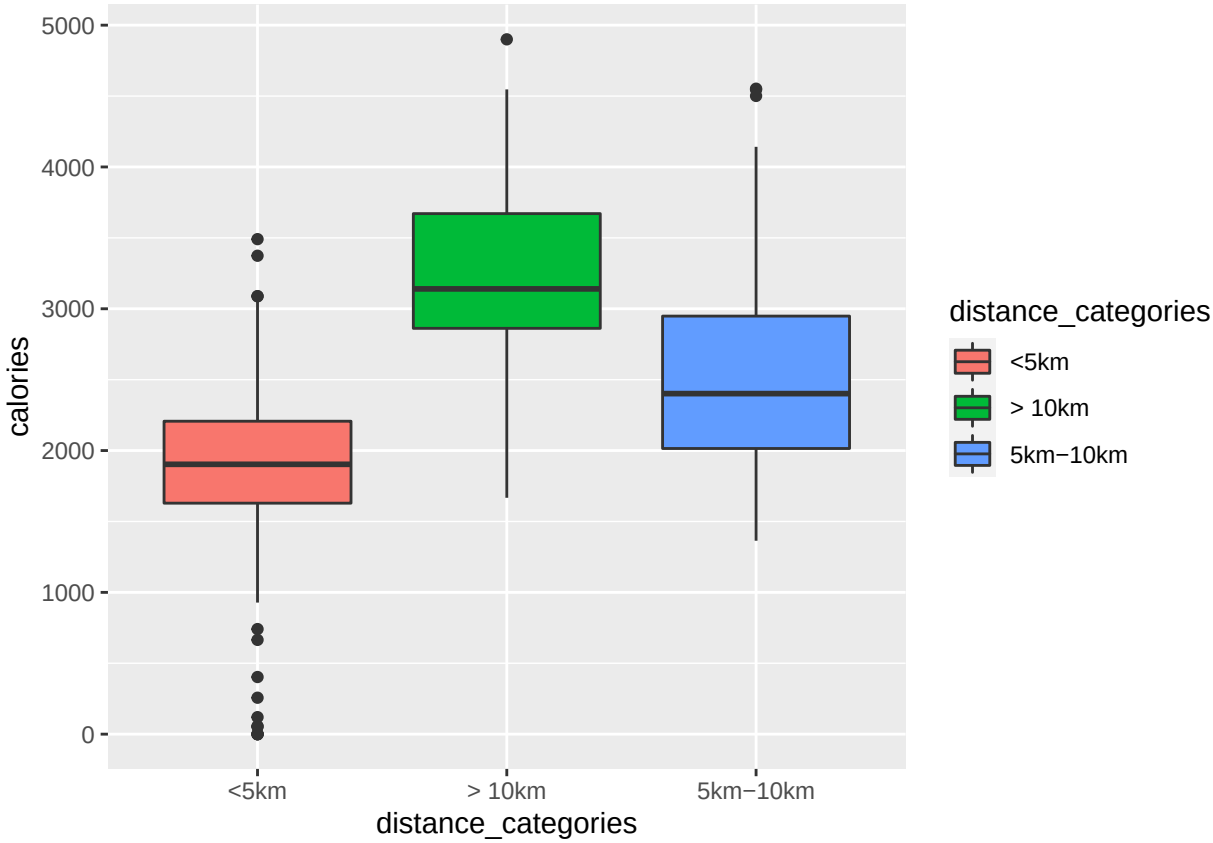
```
## `geom_smooth()` using formula 'y ~ x'
```



As we can see there is a positive correlation between very active minutes and calories burned

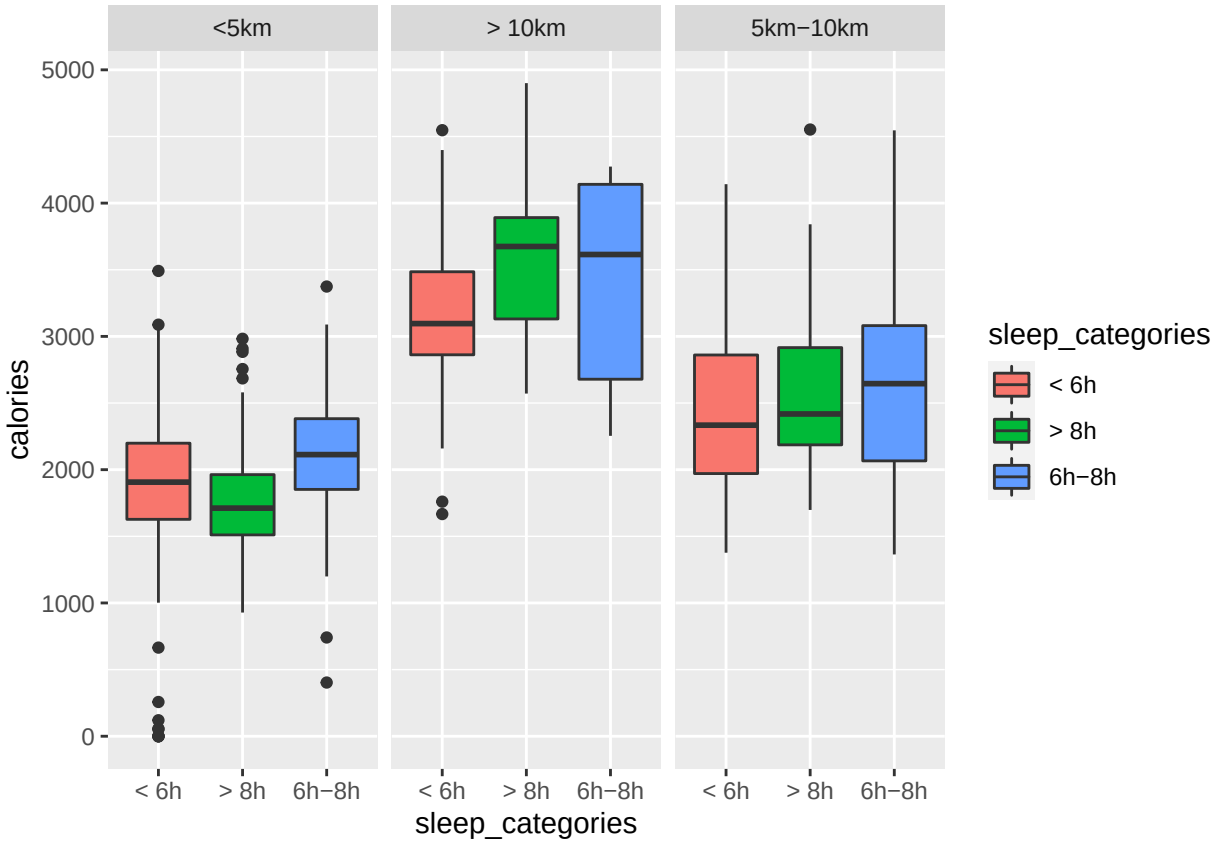
```
DA_SD <- DA_SD %>%
     mutate(sleep_categories = case_when(
         total_minutes_asleep >360 & total_minutes_asleep <= 480 ~ "6h-8h",
         total_minutes_asleep > 480 ~ "> 8h",
         TRUE ~ "< 6h"
     )) %>%
     mutate(calorie_categories = case_when(
         calories > 1500 & calories <= 2500 ~ "1.5k-2.5k",
         calories > 2500 ~ "> 2.5k",
         TRUE ~ "< 1.5k"
     )) %>%
     mutate(distance_categories = case_when(
         total_distance > 5 & total_distance <= 10 ~ "5km-10km",
         total_distance > 10 ~ "> 10km",
         TRUE ~ "<5km"
    ))

ggplot(data= DA_SD) +
     geom_boxplot(mapping= aes(x=distance_categories, y= calories, fill= distance_categories))
```



This plot shows the correlation between distance & calories burnt

```
ggplot(data= DA_SD) +
     geom_boxplot(mapping= aes(x=sleep_categories, y= calories, fill= sleep_categories))+facet_wrap("di
```



This plot shows the Correlation between sleep & calories burnt

STEP 6: ACT

What are the trends identified?

The Majority of users (81.3%) are using the FitBit app to track sedentary activities. the frequency of usage of FitBit fitness tracker application is high on tusday, wednesday and Thursday this could be due to the fact people might get distracted on weekends, or may be too busy on mondays. People who tend to sleep <6h a day & people tend to sleep >8h a day burn fewer calories as compared to people with 6h-8h sleep while covering similar distance.

How could these trends apply to Bellabeat customers?

Even though Bellabeat costumers are exclusively women, they share similarities with FitBit users, like the desire to improve their overall health, and to track their habits.

How could these trends help influence Bellabeat marketing strategy?

Bellabeat marketing team can take advantage on the fact that users tends to be more sedentary on weekends, to articulate an ad campaign that encourage people to exercise more on weekends, in addition a marketing strategy can be implemented to tell about sufficient sleep required by body, how it be achieved and how bellabeat can help them keep track of it and improve it.