~~~~
---
title: "Bellabeat case study"
author: "Charan"
date: "02-11-2022"
output:
  html_document: default
---

## Loading Packages

```{r eval=FALSE, include=FALSE}
install.packages("tidyverse")
install.packages("lubridate")
install.packages("skimr")
install.packages("janitor")
install.packages("dplyr")
install.packages("tidyr")
install.packages("ggalluvial")
install.packages("stringr")
install.packages("ggpubr")
```

```{r eval=FALSE, include=FALSE}
library(tidyverse)
library(lubridate)
library(skimr)
library(janitor)
library(dplyr)
library(tidyr)
library(ggalluvial)
library(stringr)
library(ggpubr)
```

## Importing Datasets

These three tables have compatible variables to identify trends in smart device usage.

```{r include=FALSE}
daily_activities <- read.csv("dailyActivity_merged.csv")
daily_sleeps <- read.csv("sleepDay_merged.csv")
hourly_steps <- read.csv("hourlySteps_merged.csv")
```

```{r include=FALSE}
head(daily_activities)
head(daily_sleeps)
head(hourly_steps)
```

## Cleaning Data

```{r eval=FALSE, include=FALSE}
daily_activities <- clean_names(daily_activities)
daily_sleeps <- clean_names(daily_sleeps)
hourly_steps <- clean_names(hourly_steps)
```

{r Verifying distinct values, include=FALSE} n_distinct(daily_activities\$id)
n_distinct(daily_sleeps\$id) n_distinct(daily_activities\$activity_date)
n_distinct(daily_sleeps\$sleep_day)

```{r Removing duplicates and nulls, include=FALSE}
sum(duplicated(daily_activities))
~~~~

~~~
sum(duplicated(daily_sleeps))
sum(duplicated(hourly_steps))
```

```{r include=FALSE}
colSums(is.na(daily_activities))
colSums(is.na(daily_sleeps))
colSums(is.na(hourly_steps))
```

```{r daily_sleeps needs further processing with duplicates, include=FALSE}
daily_sleeps <- daily_sleeps %>% distinct()
sum(duplicated(daily_sleeps))
```

```{r include=FALSE}
nrow(daily_sleeps)
```

```{r Changing the data formats, eval=FALSE, include=FALSE}
daily_sleeps <- daily_sleeps %>%
  mutate(id = as.character(id)) %>%
  rename(date = sleep_day) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))
```

```{r eval=FALSE, include=FALSE}
head(daily_sleeps)
```

```{r eval=FALSE, include=FALSE}
daily_activities <- daily_activities %>%
  mutate(id = as.character(id)) %>%
  rename(date = activity_date) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))
```

```{r eval=FALSE, include=FALSE}
hourly_steps <- hourly_steps %>%
  mutate(id = as.character(id)) %>%
  rename(date_time = activity_hour) %>%
  mutate(date_time = as.POSIXct(
    date_time,format ="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone()))
```

```{r include=FALSE}
head(hourly_steps)
```

## Analyzing

Users are grouped based on the total days they tracked their activities and recorded their sleep
in 31 days.

First, a new data frame activity_tracked is assigned using the aggregation data of
daily_activities where a new variable usage_freq is defined.

```{r eval=FALSE, include=FALSE}
activity_tracked <- daily_activities %>%
  group_by(id) %>%
  summarize(days_used = sum(n())) %>%
  mutate(usage_freq = case_when(
    days_used <= 10 ~ "Low",
    days_used <= 20 ~ "Moderate",
    TRUE ~ "High" ))
~~~

~~~
```

```{r include=FALSE}
head(activity_tracked)
```

Second, a new data frame sleep_recorded is assigned by aggregating daily_sleeps and defining a new
column record_freq.

```{r eval=FALSE, include=FALSE}
sleep_recorded <- daily_sleeps %>%
  group_by(id) %>%
  summarize(days_sleep = sum(n())) %>%
  mutate(record_freq = case_when(
    days_sleep <= 10 ~ "Rarely",
    days_sleep <= 20 ~ "Sometimes",
    TRUE ~ "Often"))
```

```{r include=FALSE}
head(sleep_recorded)
```

Third, both newly-made tables are joined by matching id on activity_tracked and sleep_recorded.

```{r eval=FALSE, include=FALSE}
join_act_slp <- activity_tracked %>%
  left_join(sleep_recorded, by = "id")
```

```{r include=FALSE}
head(join_act_slp)
```

Few users never recorded their sleep in 31 days, so the NA in record_freq will be replaced by
'Never'.

```{r eval=FALSE, include=FALSE}
join_act_slp <- join_act_slp %>%
  mutate(days_sleep = replace(days_sleep, is.na(days_sleep), 0)) %>%
  mutate(record_freq = replace(record_freq, is.na(record_freq), "Never"))
```

```{r include=FALSE}
head(join_act_slp)
```

Lastly, the factors in usage_freq and record_freq are arranged from their highest level.

```{r eval=FALSE, include=FALSE}
join_act_slp$usage_freq <- ordered(
  join_act_slp$usage_freq,
  levels = c("High", "Moderate", "Low"))

join_act_slp$record_freq <- ordered(
  join_act_slp$record_freq,
  levels = c("Often", "Sometimes", "Rarely", "Never"))
```

The frequency of tracked activity and recorded sleep are shown on the left and right of the graph,
respectively.

```{r echo=TRUE, warning=FALSE}
join_act_slp %>%
  group_by(usage_freq, record_freq) %>%
~~~

```
  summarize(freq = n()) %>%
  ggplot(aes(axis1 = usage_freq, axis2 = record_freq, y = freq)) +
  geom_alluvium(aes(fill = record_freq), width = 1/12) +
  geom_stratum(width = 1/12, fill = "white", color = "grey") +
  geom_label(stat = "stratum", size = 4.5, aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("Activity\n tracking", "Sleep\n recording"),
                   expand = c(.05, .05)) +
  labs(title = "Frequency of device utilization", y = "") +
  scale_fill_brewer(type = "qual", palette = "Set1") +
  theme(legend.position = "none",
        panel.background = element_blank(),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14),
        plot.title = element_text(size=18, hjust = 0.5))
```

It can be confirmed from the graph that more than half of users (21 out of 33) did not use their smart devices to monitor their sleep regularly.

A new data frame daily_intensity is made by selecting a few variables from daily_activities and calculating the intensity_value.

```{r eval=FALSE, include=FALSE}
daily_intensity <- daily_activities %>%
  select(id, date,
         sedentary = sedentary_minutes,
         lightly = lightly_active_minutes,
         fairly = fairly_active_minutes,
         very = very_active_minutes,
         total_steps) %>%
  mutate(intensity_value = (sedentary*0 + lightly*1 + fairly*2 + very*3) / 1440)
```

```{r include=FALSE}
head(daily_intensity)
```

The daily_intensity is plotted as intensity value vs daily steps.

```{r echo=TRUE, warning=FALSE}
ggplot(daily_intensity, aes(x = total_steps, y = intensity_value))+
  geom_point() +
  geom_smooth(method = lm, formula = y~x, color = "red") +
  labs(title = "Intensity value vs Daily steps",
       x = "Daily steps", y= "Intensity value") +
  stat_cor(aes(label = paste(..rr.label.., sep = "~`,`~")),
           label.x = 23000, label.y = .7, size = 6) +
  theme(panel.background = element_blank(),
        axis.text.x = element_text(size = 14),
        axis.text.y = element_text(size = 14),
        axis.title.x = element_text(size = 14),
        axis.title.y = element_text(size = 14),
        plot.title = element_text(size=18, hjust = .5))
```

Daily step is positively correlated with the intensity value. It means that an activity with high active minutes is reflected in its total steps, and vice versa.

A new data frame agr_steps is created using the aggregated data from daily_activities and a new column active_lvl.

```{r eval=FALSE, include=FALSE}
agr_steps <- daily_activities %>%
  group_by(id) %>%
  summarize(avg_daily_steps = mean(total_steps)) %>%
```

~~~
  mutate(active_lvl = case_when(
    avg_daily_steps <= 5000 ~ "Sedentary",
    avg_daily_steps <= 7500 ~ "Lightly Active",
    avg_daily_steps <= 10000 ~ "Fairly Active",
    TRUE ~ "Very Active"))
```

```{r include=FALSE}
head(agr_steps)
```

Next, the active_lvl is arranged from the highest to the lowest level.

```{r eval=FALSE, include=FALSE}
agr_steps$active_lvl <- ordered(
  agr_steps$active_lvl,
  levels = c("Very Active", "Fairly Active",
             "Lightly Active", "Sedentary"))
```

```{r include=FALSE}
agr_steps <- arrange(agr_steps, active_lvl)
head(agr_steps)
```

The following code chunk created another table named user_types. it is simply a count table of
users based on their activity level.

```{r eval=FALSE, include=FALSE}
user_type <- agr_steps %>%
  group_by(active_lvl) %>%
  summarize(count = n())
```

```{r include=FALSE}
head(user_type)
```

A doughnut chart is constructed from user_types to visualize the distribution of users' activity
levels.

```{r echo=TRUE}
user_type <- user_type %>%
  mutate(fraction = count / sum(count)) %>%
  mutate(ymax = cumsum(fraction)) %>%
  mutate(ymin = c(0, head(ymax, n=-1))) %>%
  mutate(label_position = (ymax + ymin) / 2) %>%
  mutate(label = paste0(
      active_lvl, "\n", round(fraction*100, digits = 1),"%"))

ggplot(user_type, aes(
    ymax=ymax, ymin=ymin, xmax=9, xmin=3, fill=active_lvl)) +
  geom_rect() +
  geom_label(x = 6, aes(y = label_position, label = label),
             fill = "white", inherit.aes = FALSE,
             alpha = .5, size = 6, label.size = 0) +
  ggtitle("Users' activity levels based on average daily steps") +
  scale_color_brewer(palette = 4) +
  coord_polar(theta = "y") +
  theme_void() +
  annotate("text", x = 0, y = 0, label = "33\nUsers", size = 7) +
  theme(legend.position = "none",
        plot.title = element_text(size = 18, hjust = 0.5))
```
~~~

The amount of users' at each activity level is almost equally the same. However, there are only 33 users assessed in this study.

The table hourly_steps has a column named date_time. Separate the information of date and time into two different columns of date and time to ease later data transformations.

~~~
```{r eval=FALSE, include=FALSE}
hourly_steps <- hourly_steps %>%
  separate(date_time, into = c("date", "time"), sep= " ") %>%
  mutate(date = ymd(date)) %>%
  mutate(time = str_sub(time, 1, 5))
```

```{r include=FALSE}
head(hourly_steps)
```
~~~

Mutate agr_steps into hourly_steps by matching id in both tables so that the new table has the variable active_lvl in it.

~~~
```{r eval=FALSE, include=FALSE}
join_steps_utype <- hourly_steps %>%
  left_join(agr_steps, by = "id")
```

```{r include=FALSE}
head(join_steps_utype)
```
~~~

Below is a code chunk to generate a line graph of steps in time series of hours. Different colors represent the different user types; very active to sedentary users. For an approximation, 1000 steps are about 700 m.

~~~
```{r eval=FALSE, include=FALSE}
lg <- join_steps_utype %>%
  group_by(active_lvl, time) %>%
  summarize(avg_hourly_steps = mean(step_total)) %>%
  ggplot(aes(x=time, y=avg_hourly_steps,
             group=active_lvl, color=active_lvl)) +
  geom_line(size = .8) +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 1250)) +
  labs(title = "Steps throughout the day", x="", y="") +
  theme(axis.text.x = element_text(size = 12, angle = 90, vjust = 0.3),
        axis.text.y = element_text(size = 14),
        panel.border = element_rect(
            fill = "transparent", color = "black", size = .5),
        panel.background = element_blank(),
        plot.title = element_text(size=18, hjust = .5),
        legend.title = element_blank(),
        legend.text = element_text(size=12))
```

```{r echo=TRUE}
lg +
  annotate("rect", xmin = "16:00", xmax = "20:00",
           ymin = 0, ymax = 1250, alpha = .1) +
  annotate("rect", xmin = "11:00", xmax = "15:00",
           ymin = 0, ymax = 1250, alpha = .1) +
  annotate("rect", xmin = "07:00", xmax = "10:00",
           ymin = 0, ymax = 1250, alpha = .1) +
  annotate("text", x = "18:00", y = 1210,
           label = "Evening", hjust = "center", size = 4) +
  annotate("text", x = "13:00", y = 1210,
           label = "Lunch", hjust = "center", size = 4) +
  annotate("text", x = 9.5, y = 1210,
~~~

~~~
           label = "Morning", hjust = "center", size = 4) +
  annotate("segment", x = .5, xend = 24.5, y = 1000, yend = 1000,
           colour = "black", linetype = 2) +
  annotate("text", x = 25, y = 1000, size = 4,
           label = "700 m", hjust = "left") +
  coord_cartesian(xlim = c(1.5, 23.5), clip = "off")
```
~~~

The graph visualizes the average step counts of different user types per hour. It can be expected that very active users mostly have the highest hourly steps within a day, followed by fairly active, lightly, and then sedentary. Regarding the time, the peak step counts happened in the morning (8:00-10:00), lunchtime (12:00-14:00), and evening (17:00-19:00). The trends are most likely influenced by users' daily jobs and activities, by which the data is not available.

A new data frame weekday_steps is created from daily_activities by merging it with agr_steps. In this table, a new column weekday will tell which day of the week each activity occurred. The average step counts for each day of the week are calculated.

~~~
```{r eval=FALSE, include=FALSE}
weekday_steps <- daily_activities %>%
  mutate(weekday = weekdays(date)) %>%
  left_join(agr_steps, by = "id") %>%
  group_by(active_lvl, weekday) %>%
  summarize (daily_steps = mean(total_steps), sd = sd(total_steps))
```

```{r eval=FALSE, include=FALSE}
weekday_steps$weekday <- ordered(
  weekday_steps$weekday,
  levels=c("Monday", "Tuesday", "Wednesday", "Thursday",
           "Friday", "Saturday", "Sunday"))
```

```{r include=FALSE}
head(weekday_steps)
```
~~~

The code chunk below created a bar graph of average step counts within days of the week, across different user types. The black whiskers show standard deviation of each value.

~~~
```{r echo=TRUE}
ggplot(weekday_steps, aes(
    x=weekday, y=daily_steps, group=active_lvl, fill=active_lvl)) +
  geom_bar(stat='identity') +
  scale_y_continuous(expand = c(0, 0), limits = c(0, 20500)) +
  labs(title = "Average daily steps in days of the week", x="", y="") +
  facet_wrap(vars(active_lvl)) +
  geom_errorbar(aes(
      x=weekday, ymin=ifelse(daily_steps-sd < 0, 0, daily_steps-sd),
      ymax=daily_steps+sd), width=0.4, colour = "black", alpha=0.9, size=.5) +
  guides(x =  guide_axis(angle = 45)) +
  theme(axis.text.x = element_text(size = 14, vjust = 0.3),
        axis.text.y = element_text(size = 14),
        panel.border = element_rect(
            fill = "transparent", color = "black", size = .5),
        panel.background = element_blank(),
        strip.text.x = element_text(size = 16),
        plot.title = element_text(size = 18, hjust = .5),
        legend.position = "none")
```
~~~

The charts show no patterns in step counts on days of the week within the same user types. The standard deviation for each value is quite large, meaning that the data varies quite a lot around the mean. It is potentially caused by too few data points for each calculation in this analysis. Additional data could reveal more insightful trends.

Recommendations

The goal of the entire analysis is to take insights into smart device usage into marketing strategies for one of the Bellabeat products. For this purpose, the Leaf wellness tracker is chosen because its features are relevant to the usage trends. The following recommendations focus on the marketing campaign of the Leaf.

-   Account for users' reluctances to monitor their sleep. Promote the versatilities of Leaf wellness-tracker that can be simply clipped on a piece of clothing while sleeping.
-   Encourage women with different activity levels to stay informed about their health. Whether they are active or less active, Leaf's smart technology can help with every woman's wellness goals and mindful practices.
-   Highlight Leaf's stylish feature: the combination of fashion and function. Leaf can be worn in three different ways, with five colour options. Women can track daily activity while still being stylish anytime and anywhere during the day.

Marketing strategies resulting from this trend analysis are somewhat limited to designing marketing campaigns because of the lack of user demographic data such as age, location, and occupation. For example, Bellabeat can target its advertising based on the age segmentation of consumers, make a promotional campaign to prospective consumers in the retailers' proximity, and develop a product based on consumers' purchasing power.