Bellabeat Case Study

Oladipo Precious

2022-09-20

First we install the packages and load the libraries needed for data cleaning and analysis

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("devtools")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("rmarkdown")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
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(lubridate)
```

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

```
library(skimr)
library(janitor)
```

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

```
library(ggplot2)
```

Datasets

We will now import the datasets to be used for the analysis and help us answer our business task

```
daily_activity <- read_csv('/cloud/project/Capstone Project/Fitabase Data 4.12.16-5.12.16/dailyActivity_
```

```
## 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.
```

```
daily_step <- read_csv('/cloud/project/Capstone Project/Fitabase Data 4.12.16-5.12.16/dailySteps_merged
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, StepTotal
##
## 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.
```

```
daily_intensities <- read_csv('/cloud/project/Capstone Project/Fitabase Data 4.12.16-5.12.16/dailyInten
```

```
## Rows: 940 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (9): Id, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, Ve...
##
## 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.
```

```
daily_sleep <- read_csv('/cloud/project/Capstone Project/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.
```

```
## 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.
```

Previewing the datasets

Lets checkout the datasets

```
head(daily_activity)
```

```
## # A tibble: 6 x 15
##        Id Activ~1 Total~2 Total~3 Track~4 Logge~5 VeryA~6 Moder~7 Light~8 Seden~9
##     <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9 4/12/2~   13162    8.5     8.5        0    1.88   0.550    6.06       0
## 2 1.50e9 4/13/2~   10735    6.97    6.97       0    1.57   0.690    4.71       0
## 3 1.50e9 4/14/2~   10460    6.74    6.74       0    2.44   0.400    3.91       0
## 4 1.50e9 4/15/2~    9762    6.28    6.28       0    2.14   1.26     2.83       0
## 5 1.50e9 4/16/2~   12669    8.16    8.16       0    2.71   0.410    5.04       0
## 6 1.50e9 4/17/2~    9705    6.48    6.48       0    3.19   0.780    2.51       0
## # ... with 5 more variables: VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>, and abbreviated variable names
## #   1: ActivityDate, 2: TotalSteps, 3: TotalDistance, 4: TrackerDistance,
## #   5: LoggedActivitiesDistance, 6: VeryActiveDistance,
## #   7: ModeratelyActiveDistance, 8: LightActiveDistance,
## #   9: SedentaryActiveDistance
```

```
str(daily_activity)
```

```
## 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 ...
##  $ ActivityDate            : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance           : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   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>
```

```
head(daily_step)
```

```
## # A tibble: 6 x 3
##            Id ActivityDay StepTotal
##         <dbl> <chr>           <dbl>
## 1 1503960366 4/12/2016       13162
## 2 1503960366 4/13/2016       10735
## 3 1503960366 4/14/2016       10460
## 4 1503960366 4/15/2016        9762
## 5 1503960366 4/16/2016       12669
## 6 1503960366 4/17/2016        9705
```

```
str(daily_step)
```

```
## spec_tbl_df [940 x 3] (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 ...
##  $ ActivityDay: chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ StepTotal  : num [1:940] 13162 10735 10460 9762 12669 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDay = col_character(),
##   ..   StepTotal = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(daily_intensities)
```

```
## # A tibble: 6 x 10
##       Id Activ~1 Seden~2 Light~3 Fairl~4 VeryA~5 Seden~6 Light~7 Moder~8 VeryA~9
##    <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9 4/12/2~     728     328      13      25       0    6.06   0.550    1.88
## 2 1.50e9 4/13/2~     776     217      19      21       0    4.71   0.690    1.57
## 3 1.50e9 4/14/2~    1218     181      11      30       0    3.91   0.400    2.44
## 4 1.50e9 4/15/2~     726     209      34      29       0    2.83   1.26     2.14
## 5 1.50e9 4/16/2~     773     221      10      36       0    5.04   0.410    2.71
## 6 1.50e9 4/17/2~     539     164      20      38       0    2.51   0.780    3.19
## # ... with abbreviated variable names 1: ActivityDay, 2: SedentaryMinutes,
## #   3: LightlyActiveMinutes, 4: FairlyActiveMinutes, 5: VeryActiveMinutes,
## #   6: SedentaryActiveDistance, 7: LightActiveDistance,
## #   8: ModeratelyActiveDistance, 9: VeryActiveDistance
```

```
str(daily_intensities)
```

```
## spec_tbl_df [940 x 10] (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 ...
##  $ ActivityDay             : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ SedentaryMinutes        : num [1:940] 728 776 1218 726 773 ...
##  $ LightlyActiveMinutes    : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ FairlyActiveMinutes     : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ VeryActiveMinutes       : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ LightActiveDistance     : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ VeryActiveDistance      : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDay = col_character(),
##   ..   SedentaryMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   VeryActiveDistance = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(daily_sleep)
```

```
## # A tibble: 6 x 5
##           Id SleepDay              TotalSleepRecords TotalMinutesAsleep TotalT~1
##        <dbl> <chr>                             <dbl>              <dbl>    <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327      346
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384      407
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412      442
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340      367
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700      712
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304      320
## # ... with abbreviated variable name 1: TotalTimeInBed
```

```
str(daily_sleep)
```

```
## 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 ...
##  $ SleepDay          : chr [1:413] "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:0
##  $ TotalSleepRecords : num [1:413] 1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: num [1:413] 327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : num [1:413] 346 407 442 367 712 320 377 364 384 449 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   SleepDay = col_character(),
##   ..   TotalSleepRecords = col_double(),
##   ..   TotalMinutesAsleep = col_double(),
##   ..   TotalTimeInBed = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Cleaning

I noticed that the data type of the date column in all of the datasets is the character datatype and I want to change it to a datetime type also I changed the column name of all the date column to “date”

```
daily_activity <- daily_activity %>%
  rename(date = ActivityDate) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))

daily_step <- daily_step %>%
  rename(date = ActivityDay) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))

daily_intensities <- daily_intensities %>%
  rename(date = ActivityDay) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))

daily_sleep <- daily_sleep %>%
  rename(date = SleepDay) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))
```

Checking for Duplicates

I will now find out the number of unique users per dataframe and check if there's any duplicate. If there is, I will remove them to allow for accurate analysis

```
n_distinct(daily_activity$Id)
```

```
## [1] 33
```

```
n_distinct(daily_step$Id)
```

```
## [1] 33
```

```
n_distinct(daily_intensities$Id)
```

```
## [1] 33
```

```
n_distinct(daily_sleep$Id)
```

```
## [1] 24
```

```
sum(duplicated(daily_activity))
```

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

```
sum(duplicated(daily_step))
```

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

```
sum(duplicated(daily_intensities))
```

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

```
sum(duplicated(daily_sleep))
```

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

Now it's time to remove the null values with the drop_na() function

```
daily_activity <- daily_activity %>%
  distinct() %>%
  drop_na()

daily_step <- daily_step %>%
  distinct() %>%
  drop_na()

daily_intensities <- daily_intensities %>%
  distinct() %>%
  drop_na()

daily_sleep <- daily_sleep %>%
  distinct() %>%
  drop_na()
```

Since I've dropped the duplicates, let me chack again for duplicates if they exist

```
sum(duplicated(daily_activity))
```

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

```
sum(duplicated(daily_step))
```

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

```
sum(duplicated(daily_intensities))
```

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

```
sum(duplicated(daily_sleep))
```

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

I want to ensure that the datasets columns are all in the correct syntax

```
clean_names(daily_activity)
```

```
## # A tibble: 940 x 15
##            id date       total~1 total~2 track~3 logge~4 very_~5 moder~6 light~7
##         <dbl> <date>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 1503960366 2016-04-12   13162    8.5     8.5        0    1.88   0.550    6.06
##  2 1503960366 2016-04-13   10735    6.97    6.97       0    1.57   0.690    4.71
##  3 1503960366 2016-04-14   10460    6.74    6.74       0    2.44   0.400    3.91
##  4 1503960366 2016-04-15    9762    6.28    6.28       0    2.14   1.26     2.83
##  5 1503960366 2016-04-16   12669    8.16    8.16       0    2.71   0.410    5.04
##  6 1503960366 2016-04-17    9705    6.48    6.48       0    3.19   0.780    2.51
##  7 1503960366 2016-04-18   13019    8.59    8.59       0    3.25   0.640    4.71
##  8 1503960366 2016-04-19   15506    9.88    9.88       0    3.53   1.32     5.03
##  9 1503960366 2016-04-20   10544    6.68    6.68       0    1.96   0.480    4.24
## 10 1503960366 2016-04-21    9819    6.34    6.34       0    1.34   0.350    4.65
## # ... with 930 more rows, 6 more variables: sedentary_active_distance <dbl>,
## #   very_active_minutes <dbl>, fairly_active_minutes <dbl>,
## #   lightly_active_minutes <dbl>, sedentary_minutes <dbl>, calories <dbl>, and
## #   abbreviated variable names 1: total_steps, 2: total_distance,
## #   3: tracker_distance, 4: logged_activities_distance,
## #   5: very_active_distance, 6: moderately_active_distance,
## #   7: light_active_distance
```

```
daily_activity<- rename_with(daily_activity, tolower)

clean_names(daily_step)
```

```
## # A tibble: 940 x 3
##            id date       step_total
##         <dbl> <date>          <dbl>
##  1 1503960366 2016-04-12      13162
##  2 1503960366 2016-04-13      10735
##  3 1503960366 2016-04-14      10460
##  4 1503960366 2016-04-15       9762
##  5 1503960366 2016-04-16      12669
##  6 1503960366 2016-04-17       9705
##  7 1503960366 2016-04-18      13019
##  8 1503960366 2016-04-19      15506
##  9 1503960366 2016-04-20      10544
## 10 1503960366 2016-04-21       9819
## # ... with 930 more rows
```

```
daily_step <- rename_with(daily_step, tolower)

clean_names(daily_intensities)
```

```
## # A tibble: 940 x 10
##            id date       seden~1 light~2 fairl~3 very_~4 seden~5 light~6 moder~7
##         <dbl> <date>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 1503960366 2016-04-12     728     328      13      25       0    6.06   0.550
##  2 1503960366 2016-04-13     776     217      19      21       0    4.71   0.690
##  3 1503960366 2016-04-14    1218     181      11      30       0    3.91   0.400
##  4 1503960366 2016-04-15     726     209      34      29       0    2.83   1.26 
##  5 1503960366 2016-04-16     773     221      10      36       0    5.04   0.410
##  6 1503960366 2016-04-17     539     164      20      38       0    2.51   0.780
##  7 1503960366 2016-04-18    1149     233      16      42       0    4.71   0.640
##  8 1503960366 2016-04-19     775     264      31      50       0    5.03   1.32 
##  9 1503960366 2016-04-20     818     205      12      28       0    4.24   0.480
## 10 1503960366 2016-04-21     838     211       8      19       0    4.65   0.350
## # ... with 930 more rows, 1 more variable: very_active_distance <dbl>, and
## #   abbreviated variable names 1: sedentary_minutes, 2: lightly_active_minutes,
## #   3: fairly_active_minutes, 4: very_active_minutes,
## #   5: sedentary_active_distance, 6: light_active_distance,
## #   7: moderately_active_distance
```

```
daily_intensities <- rename_with(daily_intensities, tolower)

clean_names(daily_sleep)
```

```
## # A tibble: 410 x 5
##            id date       total_sleep_records total_minutes_asleep total_time_i~1
##         <dbl> <date>                   <dbl>                <dbl>          <dbl>
##  1 1503960366 2016-04-12                   1                  327            346
##  2 1503960366 2016-04-13                   2                  384            407
##  3 1503960366 2016-04-15                   1                  412            442
##  4 1503960366 2016-04-16                   2                  340            367
##  5 1503960366 2016-04-17                   1                  700            712
##  6 1503960366 2016-04-19                   1                  304            320
##  7 1503960366 2016-04-20                   1                  360            377
##  8 1503960366 2016-04-21                   1                  325            364
##  9 1503960366 2016-04-23                   1                  361            384
## 10 1503960366 2016-04-24                   1                  430            449
## # ... with 400 more rows, and abbreviated variable name 1: total_time_in_bed
```

```
daily_sleep <- rename_with(daily_sleep, tolower)
```

Analyse Phase

Now that I have cleaned up all the datasets, they are now ready for analysis. First, I will merge the daily_activity and the daily_sleep datasets using id and date as our primary key. This is in order to see any correlation between variables

```
daily_activity_plus_sleep <- merge(daily_activity, daily_sleep, by = c("id", "date"))

glimpse(daily_activity_plus_sleep)
```

```
## Rows: 410
## Columns: 18
## $ id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ date                     <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04--~
## $ totalsteps               <dbl> 13162, 10735, 9762, 12669, 9705, 15506, 10544~
## $ totaldistance            <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3~
## $ trackerdistance          <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3~
## $ loggedactivitiesdistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ veryactivedistance       <dbl> 1.88, 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.3~
## $ moderatelyactivedistance <dbl> 0.55, 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.3~
## $ lightactivedistance      <dbl> 6.06, 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.6~
## $ sedentaryactivedistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ veryactiveminutes        <dbl> 25, 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 3~
## $ fairlyactiveminutes      <dbl> 13, 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23,~
## $ lightlyactiveminutes     <dbl> 328, 217, 209, 221, 164, 264, 205, 211, 262, ~
## $ sedentaryminutes         <dbl> 728, 776, 726, 773, 539, 775, 818, 838, 732, ~
## $ calories                 <dbl> 1985, 1797, 1745, 1863, 1728, 2035, 1786, 177~
## $ totalsleeprecords        <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ totalminutesasleep       <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, ~
## $ totaltimeinbed           <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, ~
```

Use of Smart Devices

I want to check how often our users use their phone in an interval of 31 days i.e a month. I will classify the samples into three categories - frequent user: number of days used between 21 and 31 days - moderate user: number of days used between 11 and 20 days - rare user: number of days used between 1 and 10 days

First we will create a new data frame grouping by id, calculating number of days used and creating a new column with the classification explained above.

```
daily_usage <- daily_activity_plus_sleep %>%
  group_by(id) %>%
  summarize(days_used=sum(n())) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 10 ~ "rare user",
    days_used >= 11 & days_used <= 20 ~ "moderate user",
    days_used >= 21 & days_used <= 31 ~ "frequent user",
  ))

head(daily_usage)
```

```
## # A tibble: 6 x 3
##           id days_used usage
##        <dbl>     <int> <chr>
## 1 1503960366        25 frequent user
## 2 1644430081         4 rare user
## 3 1844505072         3 rare user
## 4 1927972279         5 rare user
## 5 2026352035        28 frequent user
## 6 2320127002         1 rare user
```

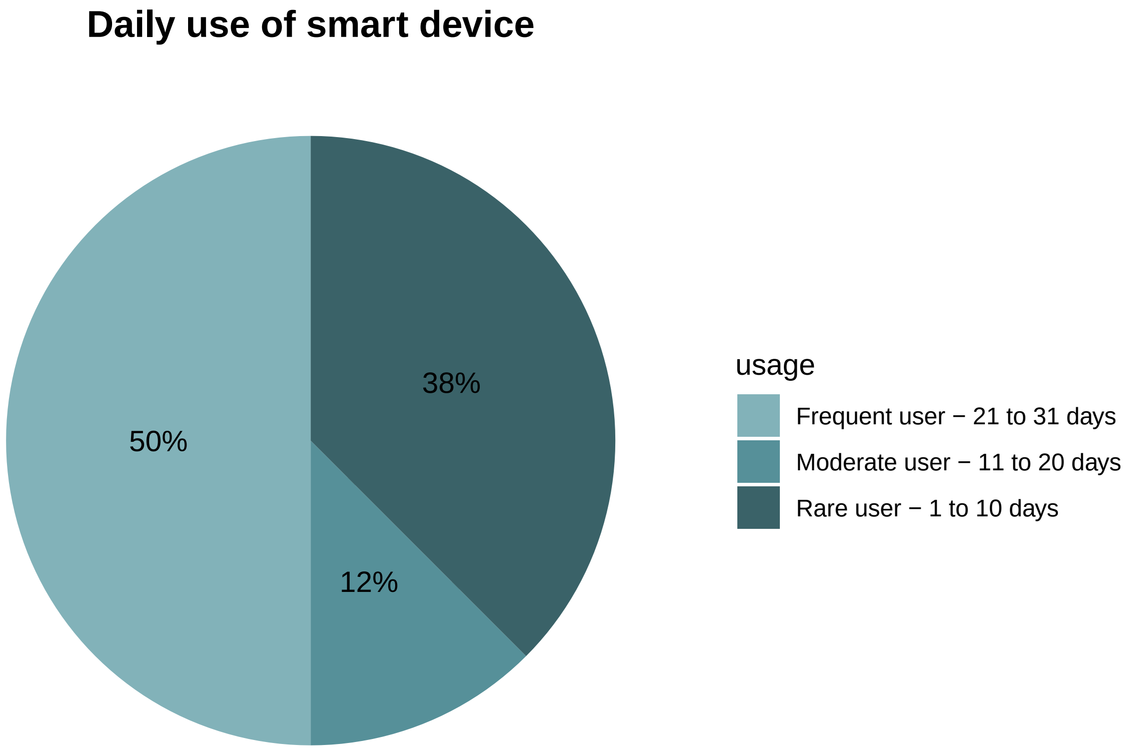
To better visualize the result, we should create our data set in percentage

```
daily_use_percentage <- daily_usage %>%
  group_by(usage) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(usage) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

daily_use_percentage$usage <- factor(daily_use_percentage$usage, levels = c("frequent user", "moderate

head(daily_use_percentage)
```

```
## # A tibble: 3 x 3
##   usage         total_percent labels
##   <fct>                 <dbl> <chr>
## 1 frequent user         0.5   50%
## 2 moderate user         0.125 12%
## 3 rare user             0.375 38%
```

Now we can visualize the data on device use as a pie chart

```
daily_use_percentage %>%
  ggplot(aes(x="",y=total_percent, fill=usage)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  scale_fill_manual(values = c("#82b2b9","#569099","#3a6268"),
                    labels = c("Frequent user - 21 to 31 days",
                               "Moderate user - 11 to 20 days",
                               "Rare user - 1 to 10 days"))+
  labs(title="Daily use of smart device")
```



From our pie chart, we can see that - 50% of our users use their devices frequently i.e from 21 - 21 days - 12% of the users use their devices moderately i.e from 11 - 20 days - 38% of the users rarely user their devices i.e about 1 - 10 days

Time spent on device per day

I want to see how long the users wear their device for per day. To guage this, I will merge the daily_usage and the daily_activity dataframe.

```
daily_usage_plus_activity <- merge(daily_usage, daily_activity, by = c("id"))
head(daily_usage_plus_activity)
```

```
##           id days_used         usage       date totalsteps totaldistance
## 1 1503960366        25 frequent user 2016-05-07      11992          7.71
## 2 1503960366        25 frequent user 2016-05-06      12159          8.03
## 3 1503960366        25 frequent user 2016-05-01      10602          6.81
## 4 1503960366        25 frequent user 2016-04-30      14673          9.25
## 5 1503960366        25 frequent user 2016-04-12      13162          8.50
## 6 1503960366        25 frequent user 2016-04-13      10735          6.97
##   trackerdistance loggedactivitiesdistance veryactivedistance
## 1            7.71                        0               2.46
## 2            8.03                        0               1.97
## 3            6.81                        0               2.29
## 4            9.25                        0               3.56
## 5            8.50                        0               1.88
## 6            6.97                        0               1.57
##   moderatelyactivedistance lightactivedistance sedentaryactivedistance
## 1                     2.12                3.13                       0
## 2                     0.25                5.81                       0
## 3                     1.60                2.92                       0
## 4                     1.42                4.27                       0
## 5                     0.55                6.06                       0
## 6                     0.69                4.71                       0
##   veryactiveminutes fairlyactiveminutes lightlyactiveminutes sedentaryminutes
## 1                37                  46                  175              833
## 2                24                   6                  289              754
## 3                33                  35                  246              730
## 4                52                  34                  217              712
## 5                25                  13                  328              728
## 6                21                  19                  217              776
##   calories
## 1     1821
## 2     1896
## 3     1820
## 4     1947
## 5     1985
## 6     1797
```

I need to create a new data frame calculating the total amount of minutes users wore the device every day and creating three different categories:

- All day - device was worn all day.
- More than half day - device was worn more than half of the day.
- Less than half day - device was worn less than half of the day.

```
minutes_worn <- daily_usage_plus_activity %>%
  mutate(total_minutes_worn = veryactiveminutes+fairlyactiveminutes+lightlyactiveminutes+sedentaryminute
  mutate (percent_minutes_worn = (total_minutes_worn/1440)*100) %>%
  mutate (worn = case_when(
    percent_minutes_worn == 100 ~ "All day",
    percent_minutes_worn < 100 & percent_minutes_worn >= 50~ "More than half day",
    percent_minutes_worn < 50 & percent_minutes_worn > 0 ~ "Less than half day"
  ))

head(minutes_worn)
```

```
##           id days_used         usage       date totalsteps totaldistance
## 1 1503960366        25 frequent user 2016-05-07      11992          7.71
## 2 1503960366        25 frequent user 2016-05-06      12159          8.03
## 3 1503960366        25 frequent user 2016-05-01      10602          6.81
## 4 1503960366        25 frequent user 2016-04-30      14673          9.25
## 5 1503960366        25 frequent user 2016-04-12      13162          8.50
## 6 1503960366        25 frequent user 2016-04-13      10735          6.97
##   trackerdistance loggedactivitiesdistance veryactivedistance
## 1            7.71                        0               2.46
## 2            8.03                        0               1.97
## 3            6.81                        0               2.29
## 4            9.25                        0               3.56
## 5            8.50                        0               1.88
## 6            6.97                        0               1.57
##   moderatelyactivedistance lightactivedistance sedentaryactivedistance
## 1                     2.12                3.13                       0
## 2                     0.25                5.81                       0
## 3                     1.60                2.92                       0
## 4                     1.42                4.27                       0
## 5                     0.55                6.06                       0
## 6                     0.69                4.71                       0
##   veryactiveminutes fairlyactiveminutes lightlyactiveminutes sedentaryminutes
## 1                37                  46                  175              833
## 2                24                   6                  289              754
## 3                33                  35                  246              730
## 4                52                  34                  217              712
## 5                25                  13                  328              728
## 6                21                  19                  217              776
##   calories total_minutes_worn percent_minutes_worn               worn
## 1     1821               1091             75.76389 More than half day
## 2     1896               1073             74.51389 More than half day
## 3     1820               1044             72.50000 More than half day
## 4     1947               1015             70.48611 More than half day
## 5     1985               1094             75.97222 More than half day
## 6     1797               1033             71.73611 More than half day
```

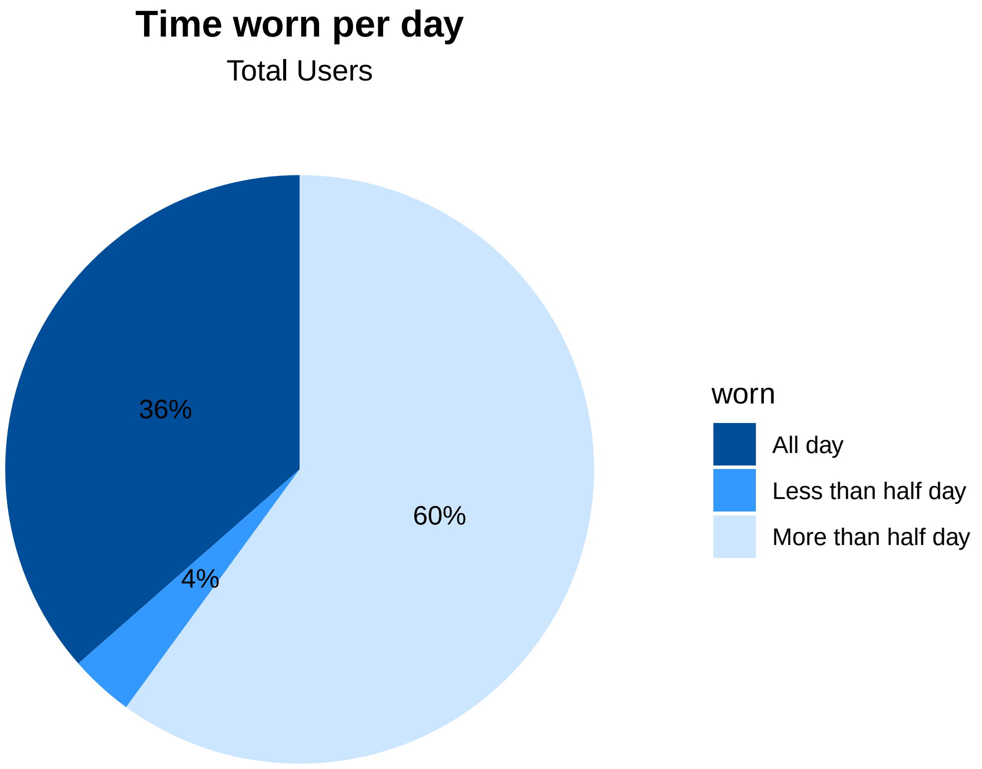
I will now create a dataframe that will show the total of users and will calculate percentage of minutes worn the device taking into consideration the three categories created.

```
minutes_worn_percent<- minutes_worn%>%
  group_by(worn) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(worn) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

head(minutes_worn_percent)
```

```
## # A tibble: 3 x 3
##   worn               total_percent labels
##   <chr>                      <dbl> <chr>
## 1 All day                   0.365  36%
## 2 Less than half day        0.0351 4%
## 3 More than half day        0.600  60%
```

We can see that only 4% of the tested users use their devices for less than half a day, and the rest use it for more than half a ay or the whole day.

```
  ggplot(minutes_worn_percent, aes(x="",y=total_percent, fill=worn)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold"),
        plot.subtitle = element_text(hjust = 0.5)) +
    scale_fill_manual(values = c("#004d99", "#3399ff", "#cce6ff"))+
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5), size = 3.5)+
  labs(title="Time worn per day", subtitle = "Total Users")
```



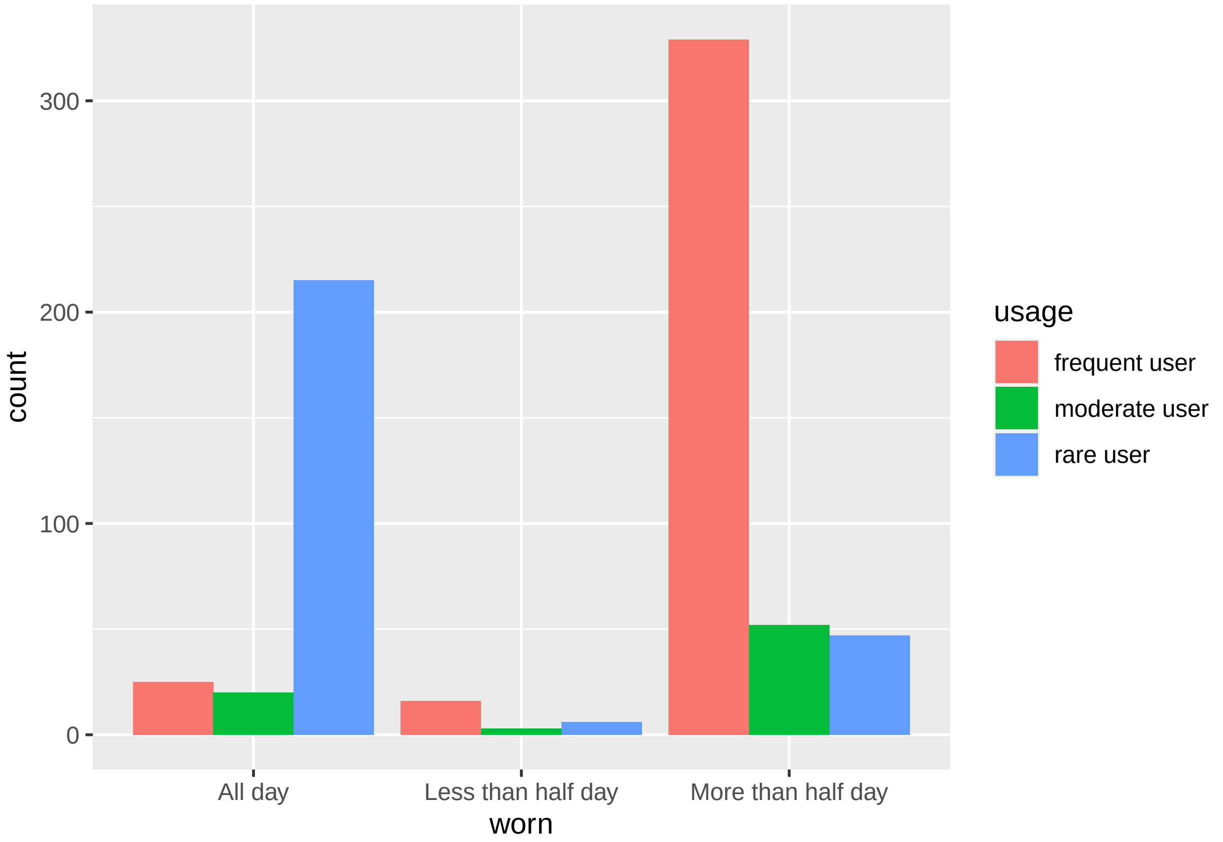
Relationship between Categories of Daily users and the Time used Daily

I filtered the minutes_worn dataframe grouping it by the usage and worn columns in order to get a clearer picture of the relationships between the type of users and the minutes they use

```
minutes_worn_pivot <- minutes_worn %>%
  group_by(usage, worn) %>%
  summarise(count = n(), .groups = "drop_last")

head(minutes_worn_pivot)
```

```
## # A tibble: 6 x 3
## # Groups:   usage [2]
##   usage         worn               count
##   <chr>         <chr>              <int>
## 1 frequent user All day               25
## 2 frequent user Less than half day    16
## 3 frequent user More than half day   329
## 4 moderate user All day               20
## 5 moderate user Less than half day     3
## 6 moderate user More than half day    52
```

```
ggplot(minutes_worn_pivot, aes(fill = usage, x = worn, y = count)) +
  geom_bar(position = "dodge", stat = "identity")
```



From the chart above, we can see that: - The users that put on their device all day have the highest category of rare users - The users that wear their devices for less than half a day, have the lowest number of use across all categories. - The users that have their devices on for more than half a day, has the highest category of frequent users.

Analysis on Sleep

```
daily_sleep <- daily_sleep %>%
  mutate(difference = totaltimeinbed - totalminutesasleep )

head(daily_sleep)
```

```
## # A tibble: 6 x 6
##           id date       totalsleeprecords totalminutesasleep totaltime~1 diffe~2
##        <dbl> <date>                 <dbl>              <dbl>       <dbl>   <dbl>
## 1 1503960366 2016-04-12                 1                327         346      19
## 2 1503960366 2016-04-13                 2                384         407      23
## 3 1503960366 2016-04-15                 1                412         442      30
## 4 1503960366 2016-04-16                 2                340         367      27
## 5 1503960366 2016-04-17                 1                700         712      12
## 6 1503960366 2016-04-19                 1                304         320      16
## # ... with abbreviated variable names 1: totaltimeinbed, 2: difference
```

```
sleep_table <- daily_sleep %>%
  group_by(id) %>%
  summarise(diff = sum(difference))

head(sleep_table)
```

```
## # A tibble: 6 x 2
##           id  diff
##        <dbl> <dbl>
## 1 1503960366   573
## 2 1644430081   208
## 3 1844505072   927
## 4 1927972279   104
## 5 2026352035   881
## 6 2320127002     8
```

It is seen that the cummulated difference in time spent on bed and the actual time spent sleeping is a lot, meaning that users do not go to sleep for a while even after they've gone to bed.

Type of user per activity level

Since we don't have any demographic variables from our sample we want to determine the type of users with the data we have. We can classify the users by activity considering the daily amount of steps. We can categorize users as follows:

- Sedentary - Less than 5000 steps a day.
- Lightly active - Between 5000 and 7499 steps a day.
- Fairly active - Between 7500 and 9999 steps a day.
- Very active - More than 10000 steps a day.

Classification has been made per the following article https://www.10000steps.org.au/articles/counting-steps/

First we will calculate the daily steps average by user.

```
daily_average <- daily_activity_plus_sleep %>%
  group_by(id) %>%
  summarise (mean_daily_steps = mean(totalsteps), mean_daily_calories = mean(calories), mean_daily_sleep

head(daily_average)
```

```
## # A tibble: 6 x 4
##           id mean_daily_steps mean_daily_calories mean_daily_sleep
##        <dbl>            <dbl>               <dbl>            <dbl>
## 1 1503960366           12406.               1872.             360.
## 2 1644430081            7968.               2978.             294
## 3 1844505072            3477                1676.             652
## 4 1927972279            1490                2316.             417
## 5 2026352035            5619.               1541.             506.
## 6 2320127002            5079                1804               61
```

I will now classify the users by the daily average steps.

```
user_type <- daily_average %>%
  mutate(user_type = case_when(
    mean_daily_steps < 5000 ~ "sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7499 ~ "lightly active",
    mean_daily_steps >= 7500 & mean_daily_steps < 9999 ~ "fairly active",
    mean_daily_steps >= 10000 ~ "very active"
  ))

head(user_type)
```

```
## # A tibble: 6 x 5
##           id mean_daily_steps mean_daily_calories mean_daily_sleep user_type
##         <dbl>            <dbl>                <dbl>            <dbl> <chr>
## 1 1503960366           12406.                1872.             360. very active
## 2 1644430081            7968.                2978.             294  fairly active
## 3 1844505072            3477                 1676.             652  sedentary
## 4 1927972279            1490                 2316.             417  sedentary
## 5 2026352035            5619.                1541.             506. lightly acti~
## 6 2320127002            5079                 1804               61  lightly acti~
```

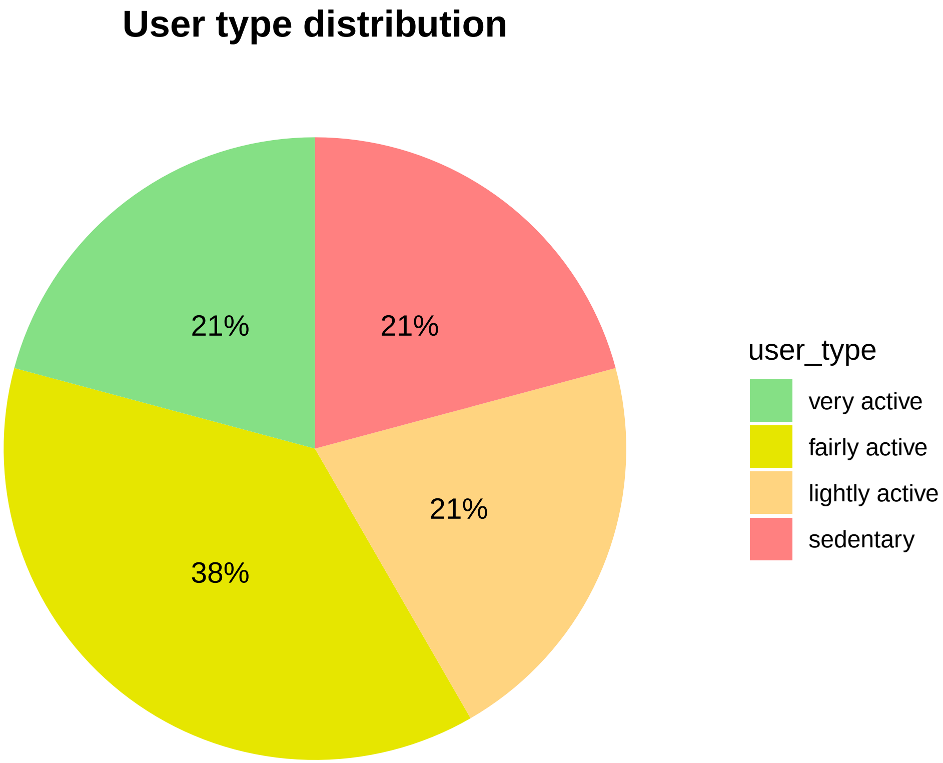
I will create a data frame with the percentage of each user type to better visualize them on a graph.

```
user_type_percent <- user_type %>%
  group_by(user_type) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(user_type) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

user_type_percent$user_type <- factor(user_type_percent$user_type , levels = c("very active", "fairly ac


head(user_type_percent)
```

```
## # A tibble: 4 x 3
##   user_type      total_percent labels
##   <fct>                  <dbl> <chr>
## 1 fairly active          0.375 38%
## 2 lightly active         0.208 21%
## 3 sedentary              0.208 21%
## 4 very active            0.208 21%
```

A visualization would help

```
user_type_percent %>%
  ggplot(aes(x="",y=total_percent, fill=user_type)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  scale_fill_manual(values = c("#85e085","#e6e600", "#ffd480", "#ff8080")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  labs(title="User type distribution")
```



The visualization shows that the percentage of users who are very active, lightly active and sedentary are the same, whiles users who are fairly active made up a higher percent of the users

Insights and Recommendations

1. Over 60% of users are frequent or moderate users of smart devices. It is advisable to make the Bellabeat app available for download on smart devices apart from phones
2. Only a small percentage, 4% of users already use their fitbit tracker for less than half of the day. This is a promising trend for Bellabeat as it shows a ready market for the Bellabeat app.
3. It shows that the number of rare users is greater in those that wears their tracer all day. Periodic notifications from the Bellabeat app to check into the app is advised.
4. The fitbit tracker data shows a disparity in time spent in bed and actual time spent sleeping, the Bellabeat app could include in their marketing, a features that plays soothing sounds to enable users to sleep faster
5. Really active users should be rewarded to motivate others