The Bellabeat data analysis case study

Magdalena Malik

2023-06-17

Case Study Roadmap

ASK

BellaBeat is a successful small company with a potential to become a larger player in the global smart device market. Company was founded in 2013 by Urška Sršen and Sando Mur, is a high-tech company that manufactures health-focused smart products. Inspired by artistic approach of Urška Sršen, beautifully designed technology which is collecting data on activity, sleep, stress and reproductive health.

Bellabeat's cofounder and Chief Creative Officer, Urška Sršen, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. The task is to focus on one of the BellaBeat's products and analyze smart device data to gain insight into how consumers are using their smart devices.

PREPARE

The data set used for this analysis is public data set available on the Keggle website FitBit Fitness Tracker Data. According to the source of the data set, data are generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016. Data including information such as: daily activity, steps, calories and sleep habits.

Does my data ROCCC?

Reliable - Partly, data are collected from around 30 anonymous users, there is not much information about participants, also group seems to not be to big.

Original - No, data were collected by third party using Amazon Mechanical Turk.

Comprehensive - No, data sets are not complete, not all of them contain records from all participants.

Current - No, data was collected by period from 12/04/2016 to 12/05/201.

Cited - Yes, data are collected by credible organization.

Despite the fact that, our data are incomplete and there is no option to ask stockholders for upgrades, I will choose few data sets to prepare for analysis and get some insights into participants activity.

To get more information about available data sets it is time to load them. I am going to use RStudio to determinate in details what given data sets are representing, which of them can be use for analysis purpose and how.

Setting up my enviroments

Loading needed package:

```
library("here")
library("dplyr")
library("skimr")
```

```
library("janitor")
library("tidyr")
library("lubridate")
```

After setting up the environment,time to load all available data sets.

Loading all available data sets (18) and set the variable names accordingly.

```
activity <- read.csv("../project/dailyActivity_merged.csv")
calories <- read.csv("../project/dailyCalories_merged.csv")
intensities <- read.csv("../project/dailyIntensities_merged.csv")
steps <- read.csv("../project/dailySteps_merged.csv")
heart <- read.csv("../project/heartrate_seconds_merged.csv")
h_calories <- read.csv("../project/hourlyCalories_merged.csv")
h_intensities <- read.csv("../project/hourlyIntensities_merged.csv")
h_steps <- read.csv("../project/hourlySteps_merged.csv")
m_calories_n <- read.csv("../project/minuteCaloriesNarrow_merged.csv")
m_calories_w <- read.csv("../project/minuteCaloriesWide_merged.csv")
m_intensities_n <- read.csv("../project/minuteIntensitiesNarrow_merged.csv")
m_intensities_w <- read.csv("../project/minuteIntensitiesWide_merged.csv")
met <- read.csv("../project/minuteMETsNarrow_merged.csv")
sleep_m <- read.csv("../project/minuteSleep_merged.csv")
m_steps_n <- read.csv("../project/minuteStepsNarrow_merged.csv")
m_steps_w <- read.csv("../project/minuteStepsWide_merged.csv")
sleep_d <- read.csv("../project/sleepDay_merged.csv")
weight <- read.csv("../project/weightLogInfo_merged.csv")
```

Next step is checking which of loaded data sets can be used for our purpose. NOTE: according to the description, data sets should contain 30 records (by user ID).

```
# check the amount of records for all data sets (by Id)
n_distinct(activity$Id) # 33
```

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

```
n_distinct(calories$Id) # 33
```

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

```
n_distinct(intensities$Id) # 33
```

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

```
n_distinct(steps$Id) # 33
```

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

```
n_distinct(h_calories$Id) # 33
```

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

```
n_distinct(h_intensities$Id) # 33
```

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

```
n_distinct(h_steps$Id) # 33
```

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

```
n_distinct(m_calories_n$Id) # 33
```

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

```
n_distinct(m_calories_w$Id) # 33
```

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

```
n_distinct(m_intensities_n$Id) # 33
```

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

```
n_distinct(m_intensities_w$Id) # 33
```

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

```
n_distinct(met$Id) # 33
```

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

```
n_distinct(m_steps_n$Id) # 33
```

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

```
n_distinct(m_steps_w$Id) # 33
```

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

```
n_distinct(sleep_m$Id) # 24
```

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

```
n_distinct(sleep_d$Id) # 24
```

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

```
n_distinct(weight$Id) # 8
```

```
## [1] 8
```

```
n_distinct(heart$Id) # 14
```

```
## [1] 14
```

```
n_distinct(sleep_m$Id) # 24
```

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

Not all of data sets contains desirable amount of records. Since there is no chance to contact with stockholders to gain more accurate data sets, I will focus on few chosen one that, in my opinion, will be best fitting for further analysis.

Chosen data sets: - activity, - calories, - intensities, - steps.

PROCESS

Process of cleaning I will start from the data set 'activity', which contains information about daily activity, steps, distance and calories. I will use methods like:

```
skim_without_charts(activity)
```

Table 1: Data summary

Name	activity
Number of rows	940
Number of columns	15

Table 1: Data summary

Column type frequency:	
character	1
numeric	14
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDate	0	1	8	9	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.855407e+09	2.424805e+09	1503960366	2320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
TotalSteps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.072700e+04	3.601900e+04
TotalDistance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
TrackerDistance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
LoggedActivitiesDistance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00
VeryActiveDistance	0	1	1.500000e+00	2.060000e+00	0	0.000000e+00	1.000000e-01	2.050000e+00	2.192000e+01
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.000000e-01	8.000000e-01	6.480000e+00
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01
SedentaryActiveDistance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01
VeryActiveMinutes	0	1	2.116000e+01	2.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100000e+02
FairlyActiveMinutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.180000e+02
SedentaryMinutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.329500e+03	1.440000e+03
Calories	0	1	2.303610e+03	7.181700e+02	0	1.828500e+03	2.134000e+03	2.793250e+03	4.900000e+03

```
head(activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 2 1503960366    4/13/2016      10735          6.97            6.97
## 3 1503960366    4/14/2016      10460          6.74            6.74
## 4 1503960366    4/15/2016       9762          6.28            6.28
## 5 1503960366    4/16/2016      12669          8.16            8.16
## 6 1503960366    4/17/2016       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0              1.88                     0.55
## 2                        0              1.57                     0.69
## 3                        0              2.44                     0.40
## 4                        0              2.14                     1.26
```

```
## 5                        0              2.71                     0.41
## 6                        0              3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
```

```
glimpse(activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <int> 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        <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```
summary(activity)
```

```
##        Id            ActivityDate         TotalSteps    TotalDistance
##  Min.   :1.504e+09   Length:940         Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   Class :character   1st Qu.: 3790   1st Qu.: 2.620
##  Median :4.445e+09   Mode  :character   Median : 7406   Median : 5.245
##  Mean   :4.855e+09                      Mean   : 7638   Mean   : 5.490
##  3rd Qu.:6.962e+09                      3rd Qu.:10727   3rd Qu.: 7.713
##  Max.   :8.878e+09                      Max.   :36019   Max.   :28.030
##  TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   :0.0000          Min.   : 0.000
##  1st Qu.: 2.620   1st Qu.:0.0000          1st Qu.: 0.000
##  Median : 5.245   Median :0.0000          Median : 0.210
##  Mean   : 5.475   Mean   :0.1082          Mean   : 1.503
##  3rd Qu.: 7.710   3rd Qu.:0.0000          3rd Qu.: 2.053
##  Max.   :28.030   Max.   :4.9421          Max.   :21.920
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  Min.   :0.0000           Min.   : 0.000      Min.   :0.000000
```

```
##  1st Qu.:0.0000           1st Qu.: 1.945      1st Qu.:0.000000
##  Median :0.2400           Median : 3.365      Median :0.000000
##  Mean   :0.5675           Mean   : 3.341      Mean   :0.001606
##  3rd Qu.:0.8000           3rd Qu.: 4.782      3rd Qu.:0.000000
##  Max.   :6.4800           Max.   :10.710      Max.   :0.110000
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

To get the basic information about data set. Chosen data set has 15 columns and 940 rows. Data are ordered by 'Id' column and 'ActivityDate' column. I have spotted that 'ActivityDate' column is in not correct format , for my purpose I will change it into date format. I will rewrite it to new data set 'clean_activity' to keep original data separately.

```
clean_activity <- mutate(activity, ActivityDate=as.Date(ActivityDate, format = "%m/%d/%Y"))
glimpse(clean_activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <date> 2016-04-12, 2016-04-13, 2016-04-14, 2016-04-~
## $ TotalSteps               <int> 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        <int> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <int> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <int> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <int> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

Next step is to check column names by 'clean_names(clean_activity)':

Check for missing values:

```
sum(is.na(clean_activity))
```

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

Check for duplicates rows:

```
sum(duplicated(clean_activity))
```

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

To be ready to work with data and get wanted information, I will also add week day to the table.

Data set after cleaning will look like that:

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-04-12      13162          8.50            8.50
## 2 1503960366   2016-04-13      10735          6.97            6.97
## 3 1503960366   2016-04-14      10460          6.74            6.74
## 4 1503960366   2016-04-15       9762          6.28            6.28
## 5 1503960366   2016-04-16      12669          8.16            8.16
## 6 1503960366   2016-04-17       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0              1.88                     0.55
## 2                        0              1.57                     0.69
## 3                        0              2.44                     0.40
## 4                        0              2.14                     1.26
## 5                        0              2.71                     0.41
## 6                        0              3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories WeekDay
## 1                  13                  328              728     1985     Tue
## 2                  19                  217              776     1797     Wed
## 3                  11                  181             1218     1776     Thu
## 4                  34                  209              726     1745     Fri
## 5                  10                  221              773     1863     Sat
## 6                  20                  164              539     1728     Sun
```

Next data set, that I am planning to use, would be data set created from 3 data sets containing information about intensities, calories and steps.

```
head(h_intensities)
```

```
##           Id          ActivityHour TotalIntensity AverageIntensity
## 1 1503960366 4/12/2016 12:00:00 AM             20         0.333333
## 2 1503960366  4/12/2016 1:00:00 AM              8         0.133333
## 3 1503960366  4/12/2016 2:00:00 AM              7         0.116667
## 4 1503960366  4/12/2016 3:00:00 AM              0         0.000000
## 5 1503960366  4/12/2016 4:00:00 AM              0         0.000000
## 6 1503960366  4/12/2016 5:00:00 AM              0         0.000000
```

```
head(h_calories)
```

```
##           Id          ActivityHour Calories
## 1 1503960366 4/12/2016 12:00:00 AM       81
## 2 1503960366  4/12/2016 1:00:00 AM       61
## 3 1503960366  4/12/2016 2:00:00 AM       59
## 4 1503960366  4/12/2016 3:00:00 AM       47
## 5 1503960366  4/12/2016 4:00:00 AM       48
## 6 1503960366  4/12/2016 5:00:00 AM       48
```

```
head(h_steps)
```

```
##           Id          ActivityHour StepTotal
## 1 1503960366 4/12/2016 12:00:00 AM       373
## 2 1503960366  4/12/2016 1:00:00 AM       160
## 3 1503960366  4/12/2016 2:00:00 AM       151
## 4 1503960366  4/12/2016 3:00:00 AM         0
## 5 1503960366  4/12/2016 4:00:00 AM         0
## 6 1503960366  4/12/2016 5:00:00 AM         0
```

Now I can merge data into one data set 'h_activity'.

```
h_activity <- merge(h_calories, h_steps, by=c('Id', 'ActivityHour'))
h_activity <- merge(h_activity, h_intensities, by=c('Id', 'ActivityHour'))
head(h_activity)
```

```
##           Id          ActivityHour Calories StepTotal TotalIntensity
## 1 1503960366  4/12/2016 1:00:00 AM       61       160              8
## 2 1503960366  4/12/2016 1:00:00 PM       66       221              6
## 3 1503960366 4/12/2016 10:00:00 AM       99       676             29
## 4 1503960366 4/12/2016 10:00:00 PM       65        89              9
## 5 1503960366 4/12/2016 11:00:00 AM       76       360             12
## 6 1503960366 4/12/2016 11:00:00 PM       81       338             21
##   AverageIntensity
## 1         0.133333
## 2         0.100000
## 3         0.483333
## 4         0.150000
## 5         0.200000
## 6         0.350000
```

Here also I will do basic check of data set.

```
which(is.na(h_activity))
```

```
## integer(0)
```

```
sum(is.na(h_activity))
```

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

```
sum(duplicated(h_activity))
```

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

All data looks correct, so now I will also add week day to the data set:

Now I will split 'ActivityHour' column into columns 'ActivityDate' and 'ActivityTime':

Note that 'ActivityDate' and 'ActivityTime' changed again into type. Check the data set:

```
head(h_activity)
```

```
##           Id ActivityDate ActivityTime Calories StepTotal TotalIntensity
## 1 1503960366   2016-04-12     01:00:00       61       160              8
## 2 1503960366   2016-04-12     13:00:00       66       221              6
## 3 1503960366   2016-04-12     10:00:00       99       676             29
## 4 1503960366   2016-04-12     22:00:00       65        89              9
## 5 1503960366   2016-04-12     11:00:00       76       360             12
## 6 1503960366   2016-04-12     23:00:00       81       338             21
```

```
##   AverageIntensity WeekDay
## 1         0.133333     Tue
## 2         0.100000     Tue
## 3         0.483333     Tue
## 4         0.150000     Tue
## 5         0.200000     Tue
## 6         0.350000     Tue
```

Now I can start processing plots and analysis on prepared data sets.

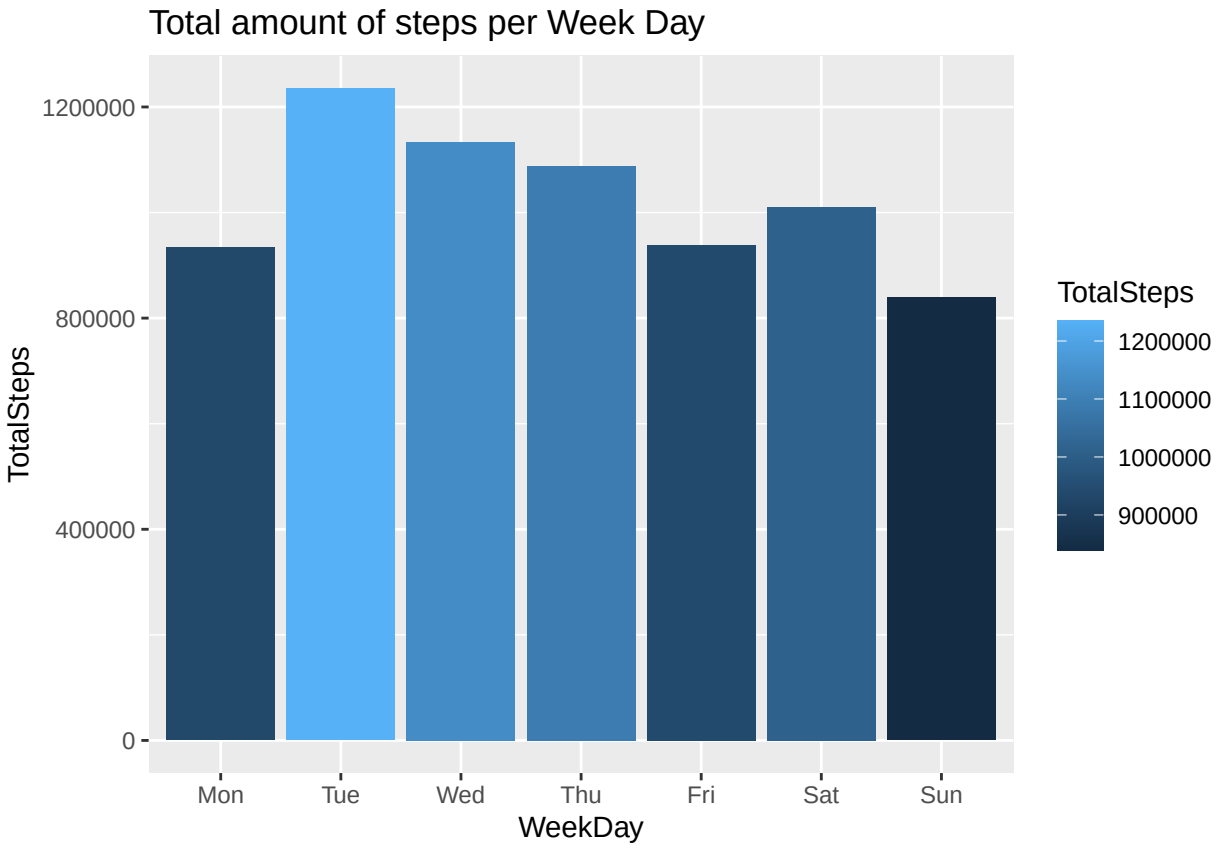
ANALYZE AND SHARE

Install ggplot for plotting data

Lets start from the first data set 'clean_activity', I will group and plot data to see how activity looks like in each day of the week. First lets check steps, data will be grouped by week day and summed:

```
week_day_activity <- aggregate(clean_activity$TotalSteps, by=list(WeekDay=clean_activity$WeekDay), FUN=s
colnames(week_day_activity)[2] ="TotalSteps"
```

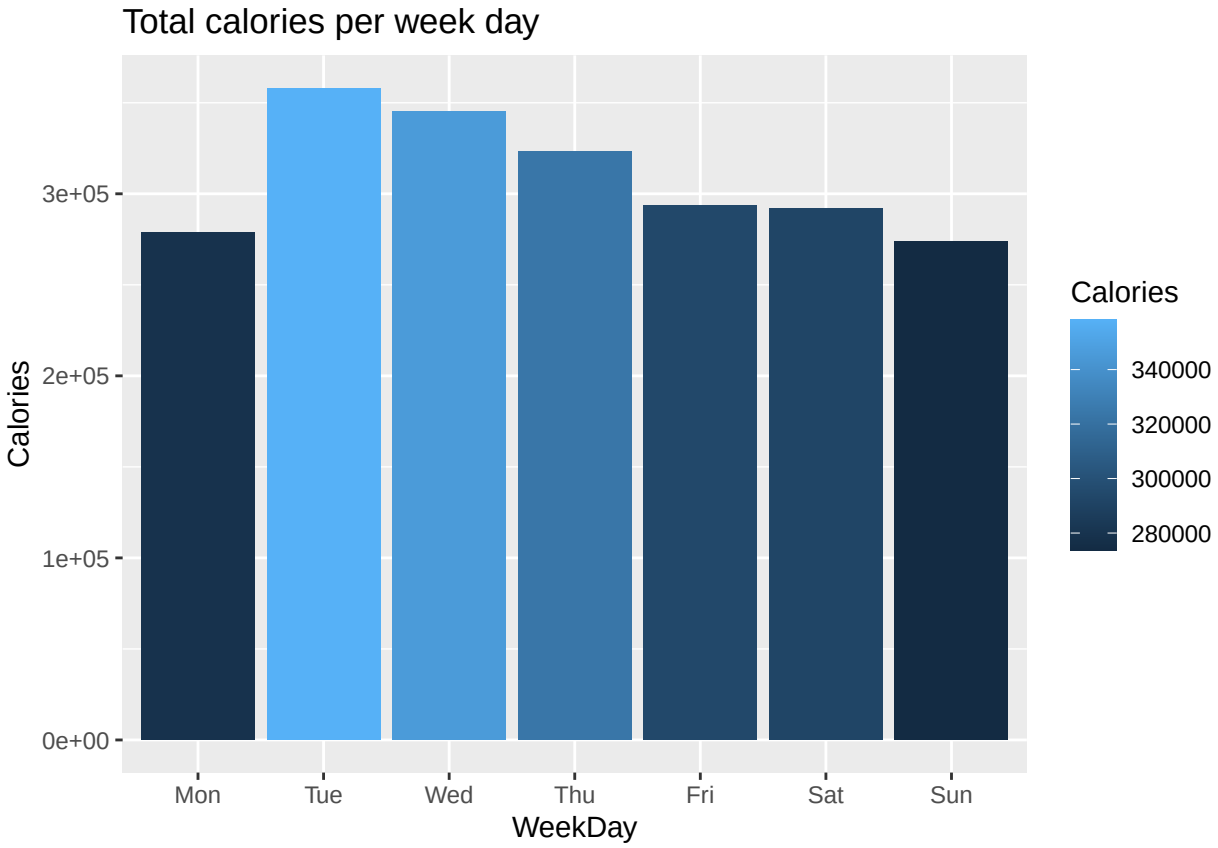
And now it can be plotted :



Clearly we can see that the most active day is Tuesday, and the least active day is Sunday. But also we can see that activity is not spread evenly thought out other week days.

Lets take a look on calories burn by week days:

```
week_day_calories <- aggregate(clean_activity$Calories, by=list(WeekDay=clean_activity$WeekDay), FUN=su
colnames(week_day_calories)[2] ="Calories"
```

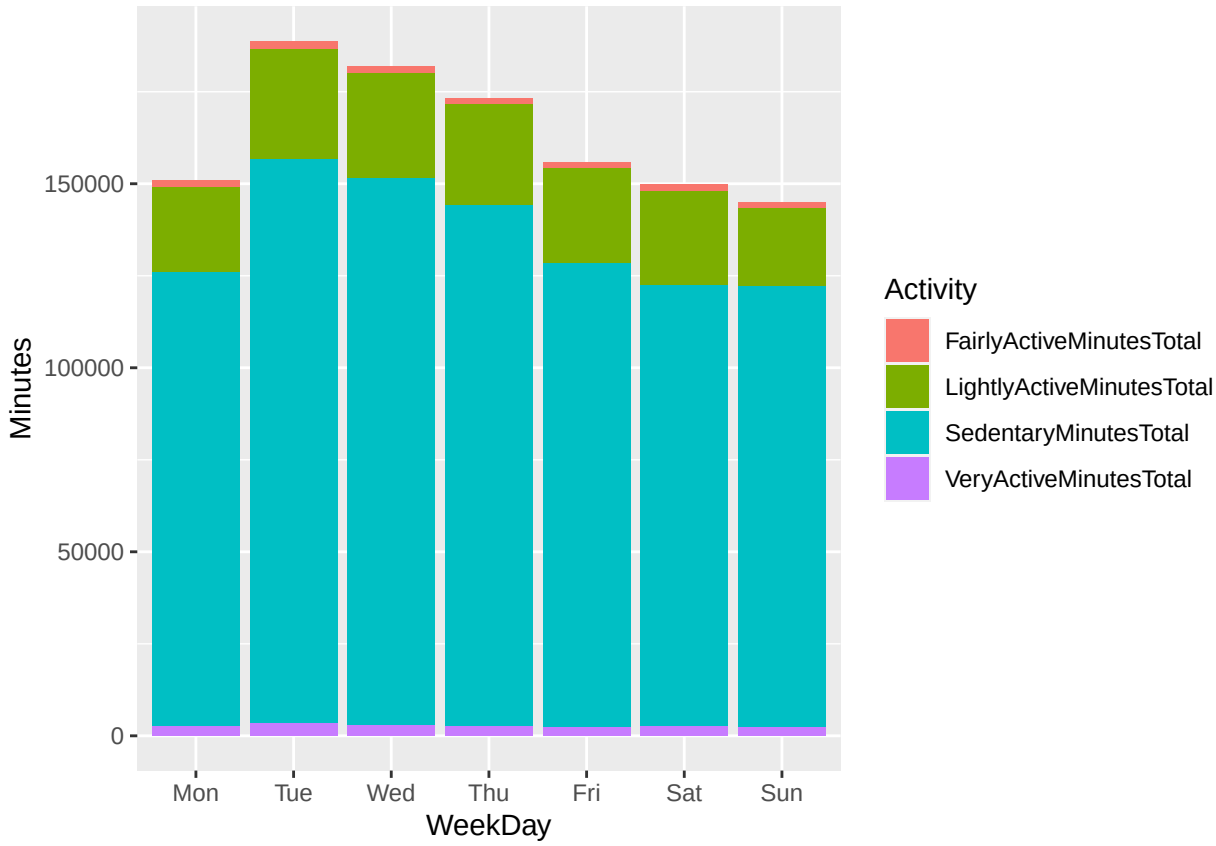


As in the case of steps, also calories show similar pattern. The highest burn is in Tuesday and the lowest in Sunday. It seems that the Tuesday is the most active day in the week.

To see more related patterns, I will also plot the data, which storing information about activity by amount of minutes relatively to the week days. Here we have four groups: - Very Active - Moderate - Light - Sedentary.

I will sum activity according to the day of the week to see how different level of activity are spread during the week, and also particular days. To get to that point I have to group data by week day, sum them up and change format of data to long.

Plot data:



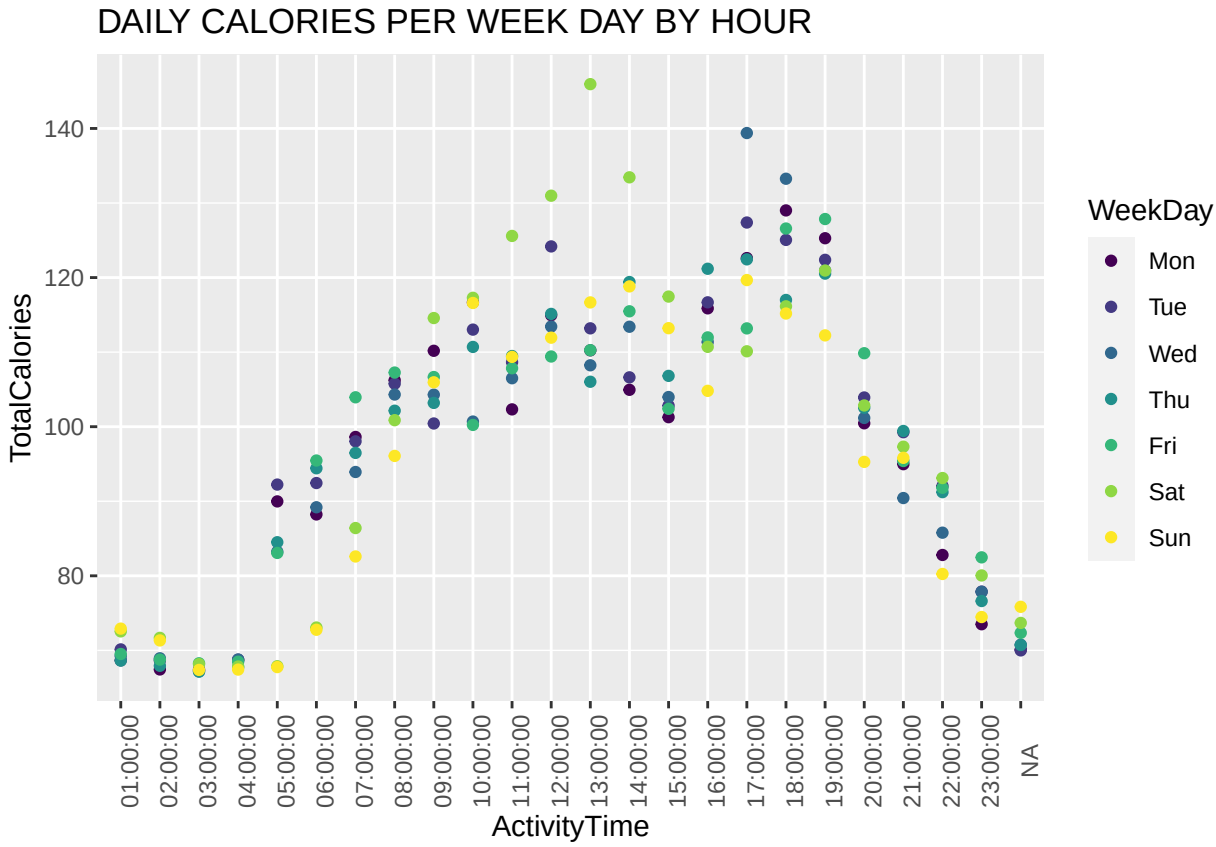
Full day has 1440 minutes, so as we can see on the plot activity time is way shorter than sedentary time. Mean value of total active minutes is 227.6342 and for sedentary time is 991.6607. On average daily active time is around 18% of day.

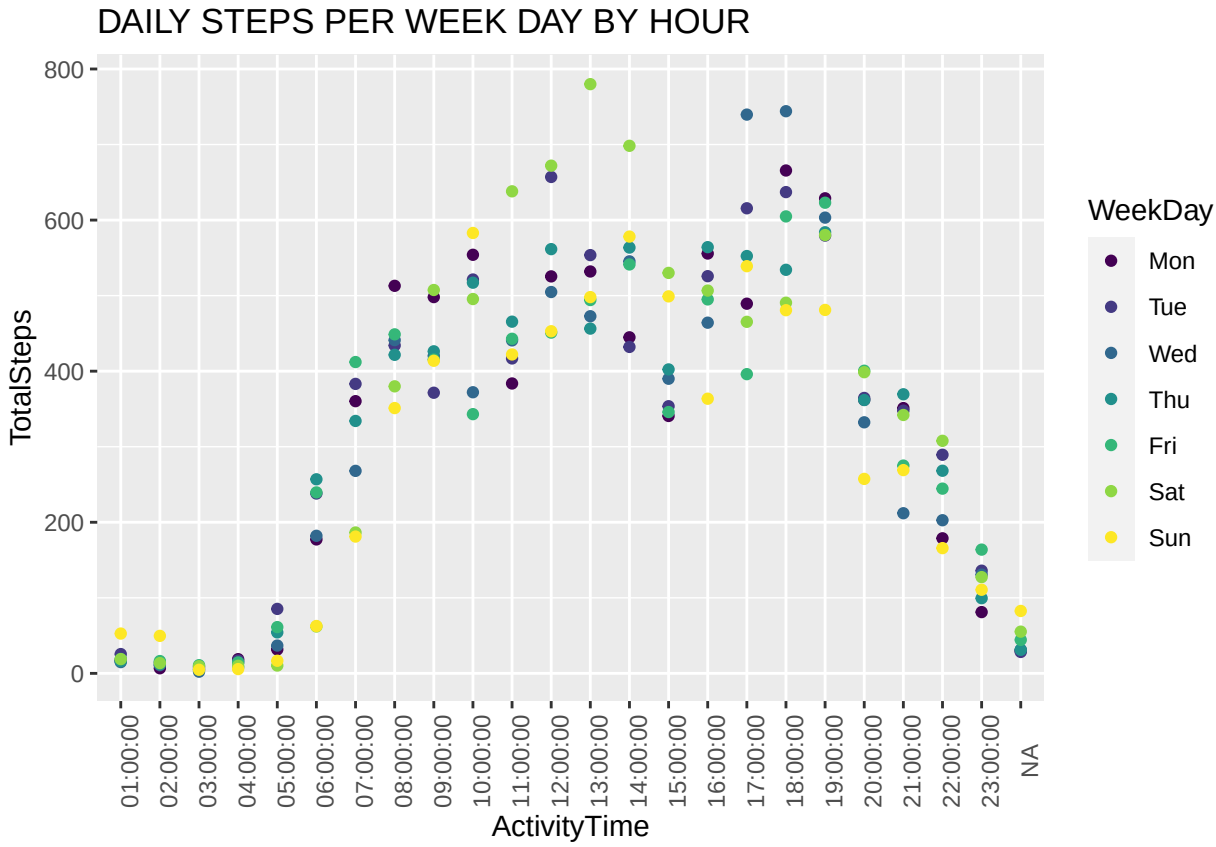
To check how activity looks like during each week by hour, I will use second prepared data set 'h_activity'. Firstly I will group data by week day and sort them by hour:

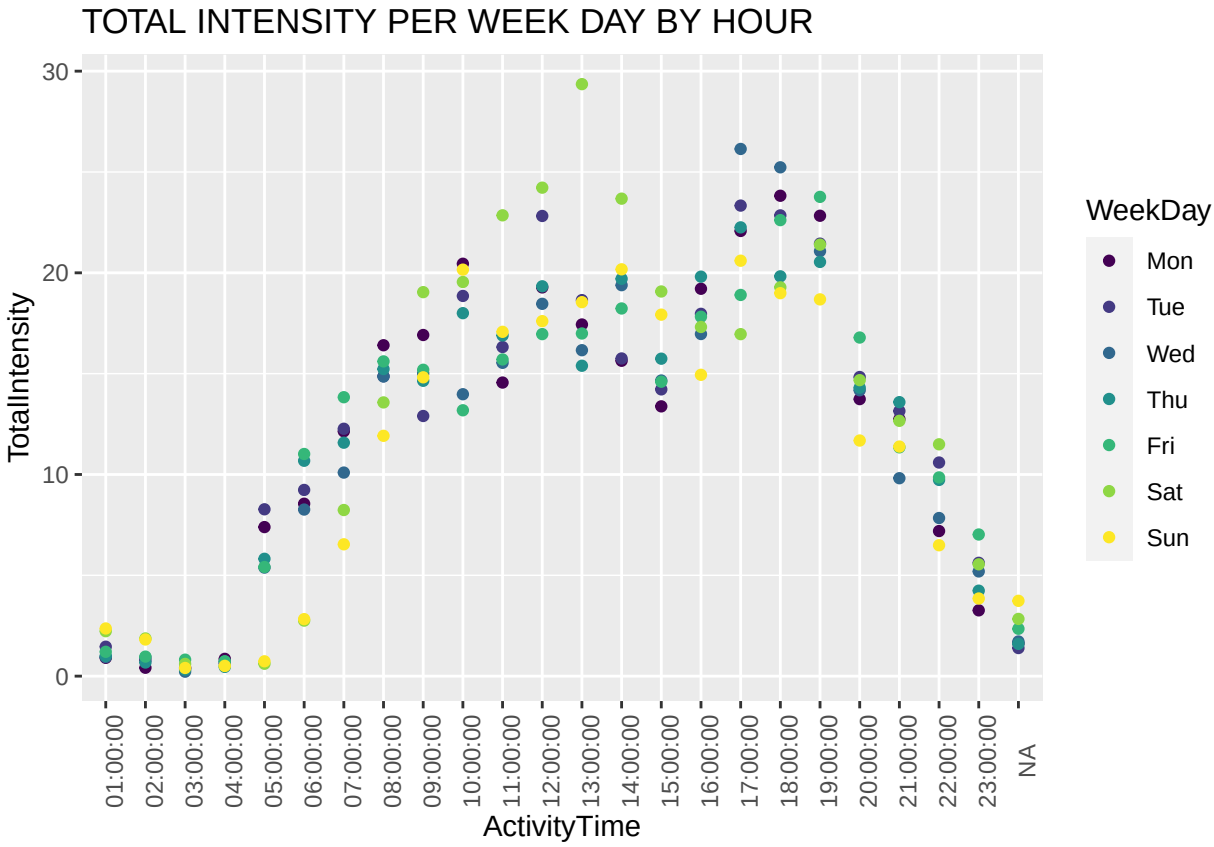
```
h_activity_grouped <- h_activity %>% group_by(WeekDay, ActivityTime) %>% arrange(ActivityTime) %>%
  summarise(TotalCalories=mean(Calories),
            TotalSteps=mean(StepTotal),
            TotalIntensity=mean(TotalIntensity))
```

```
## `summarise()` has grouped output by 'WeekDay'. You can override using the
## `.groups` argument.
```

Sorted data are stored in new data set 'h_activity_grouped', so time to plot them and see how they are spread.


DAILY CALORIES PER WEEK DAY BY HOUR
TotalCalories
140
120
100
80
01:00:00
02:00:00
03:00:00
04:00:00
05:00:00
06:00:00
07:00:00
08:00:00
09:00:00
10:00:00
11:00:00
12:00:00
13:00:00
14:00:00
15:00:00
16:00:00
17:00:00
18:00:00
19:00:00
20:00:00
21:00:00
22:00:00
23:00:00
NA
ActivityTime
WeekDay
Mon
Tue
Wed
Thu
Fri
Sat
Sun


DAILY STEPS PER WEEK DAY BY HOUR
TotalSteps
800
600
400
200
0
01:00:00
02:00:00
03:00:00
04:00:00
05:00:00
06:00:00
07:00:00
08:00:00
09:00:00
10:00:00
11:00:00
12:00:00
13:00:00
14:00:00
15:00:00
16:00:00
17:00:00
18:00:00
19:00:00
20:00:00
21:00:00
22:00:00
23:00:00
NA
ActivityTime
WeekDay
Mon
Tue
Wed
Thu
Fri
Sat
Sun



According to the daily activity, clearly we can see that the most active day is Tuesday. As to the hourly activity, hours between 8 am and 8 pm are the most active. It is worth to notice that values way above average are related to steps, calories and also intensity in Saturdays, but only in morning hours till noon.

ACT

The BellaBeat is an application who can change the way people think about their activity. As we can clearly see from above analysis, BellaBeat's users are not active as much as it is required, on top of that we can see that activity is not regular or constant through period of time. Key take would be to focus marketing campaign on adding option to application which will encourage users to take activity more often. Encouraging users will also be a key aspect in case of collecting data, since it is clear that users related to above case did not share their information about calories, heart rate and sleep, which are crucial information for further analysis.