Bellabeat Product Analysis

Pritee Kadam

9/28/2021

# Bellabeat Products Analysis

## The problem statement

“Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat wants to expand their business for one of their products so they wants to analyse the usage of one of their products in order to gain insight into how people are already using their smart devices.Then, using this information,she(Urška Sršen, cofounder and Chief Creative Officer of Bellabeat) would like high-level recommendations for how these trends can inform Bellabeat marketing strategy”

### Key stakeholders:-

Urška Sršen: Bellabeat’s co-founder and Chief Creative Officer Sando Mur: Mathematician and Bellabeat’s co-founder Bellabeat marketing analytics team

### Introduction:-

This is a capstone project for my Google Data Analytics Capstone. In this case study, I will be analyzing one of Bellabeat’s smart device data to gain insight into how consumers are using their smart devices, Bellabeat is a high-tech manufacturer of health focused products for women. The insights I discover will then help guide to decide or enhance marketing strategy for the company

### 1) Ask:-

* Key Objectives:

1. What are some daily habit trends in smart devise usage?
2. How could these daily habit trends apply to Bellabeat customers?
3. How could these daily habit trends help influence the Bellabeat marketing strategy?

### 2) Prepare:-

Data preparation:  
  
 Source, Licensing, Privacy

Source: FitBit Fitness Tracker Data - (Dataset made available through Mobius)

License: CC0: Public Domain

Privacy: This Kaggle data set contains personal fitness tracker from 30  
Fitbit users who consented to the submission of personal tracker data,including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users’ habits.

### 3) Process:

#### importing necessory packages

if (!require('tidyverse'))  
 {  
 install.packages('tidyverse');  
 library(tidyverse);  
 }

## Loading required package: tidyverse

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

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.4 v dplyr 1.0.7  
## v tidyr 1.1.3 v stringr 1.4.0  
## v readr 2.0.1 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

if(!require('tidyverse'))  
{  
 install.packages('here');  
 library(here);  
}

if(!require('skimr'))  
{  
 install.packages('skimr');  
 library(skimr);  
}

## Loading required package: skimr

if(!require('janitor'))  
{  
 install.packages('janitor');  
 library(janitor)  
}

## Loading required package: janitor

##   
## Attaching package: 'janitor'

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

if(!require('lubridate'))  
{  
install.packages('lubridate');  
 library(lubridate)  
}

## Loading required package: lubridate

##   
## Attaching package: 'lubridate'

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

if(!require('dplyr'))  
{  
install.packages('dplyr');  
 library(dplyr);  
}

if(!require('ggplot2'))  
{  
install.packages('ggplot2');  
 library(ggplot2);  
}

if(!require('tidyr'))  
{  
 install.packages('tidyr');  
 library(tidyr);  
}

if(!require('corrplot'))  
{  
 install.packages('corrplot');  
 library(corrplot);  
}

## Loading required package: corrplot

## corrplot 0.90 loaded

if(!require('ggpubr'))  
{  
install.packages('ggpubr');  
 library(ggpubr);  
}

## Loading required package: ggpubr

if(!require('chron'))  
{  
 install.packages('chron');  
 library(chron);  
}

## Loading required package: chron

##   
## Attaching package: 'chron'

## The following objects are masked from 'package:lubridate':  
##   
## days, hours, minutes, seconds, years

if(!require('hms'))  
{  
 install.packages('hms');  
 library(hms);  
}

## Loading required package: hms

##   
## Attaching package: 'hms'

## The following object is masked from 'package:lubridate':  
##   
## hms

if(!require('kableExtra'))  
{  
 install.packages(kableExtra);  
 library(kableExtra);  
}

## Loading required package: kableExtra

##   
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':  
##   
## group\_rows

if(!require('magrittr'))  
{  
 install.packages('magrittr');  
 library(magrittr);  
}

## Loading required package: magrittr

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

if(!require('scales'))  
{  
 install.packages('scales');  
 library(scales);  
}

## Loading required package: scales

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

if(!require('devtools'))  
{  
install.packages('devtools');  
 library(devtools)  
}

## Loading required package: devtools

## Loading required package: usethis

if(!require('readxl'))  
{  
 install.packages('readxl');  
 library(readxl);  
}

## Loading required package: readxl

if(!require('tinytex'))  
{  
 install.packages('tinytex');  
 library(tinytex)  
}

## Loading required package: tinytex

As I explore through all the files present I found that much files's columns are already present in the dailyActivity\_merged file so we are here taking only those files which we have to analyze so those files are:  
 dailyActivity\_merged  
 hourlyIntensities\_merged  
 heartrate\_seconds\_merged  
 sleepDay\_merged  
 weightLogInfo\_merged

###- importing files:

Activity <- read.csv("dailyActivity\_merged.csv")

Hourly\_Calories <- read.csv("hourlyCalories\_merged.csv")

Hourly\_Intensities <- read.csv("hourlyIntensities\_merged.csv")

Hourly\_Steps <- read.csv("hourlySteps\_merged.csv")

Heartrate\_Seconds <- read.csv("heartrate\_seconds\_merged.csv")

Sleep\_Day <- read.csv("sleepDay\_merged.csv")

WeightLogInfo <- read.csv("weightLogInfo\_merged.csv")

### Exploring files-

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

head(Hourly\_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(Hourly\_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(Hourly\_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

head(Heartrate\_Seconds)

## Id Time Value  
## 1 2022484408 4/12/2016 7:21 97  
## 2 2022484408 4/12/2016 7:21 102  
## 3 2022484408 4/12/2016 7:21 105  
## 4 2022484408 4/12/2016 7:21 103  
## 5 2022484408 4/12/2016 7:21 101  
## 6 2022484408 4/12/2016 7:22 95

head(Sleep\_Day)

## Id SleepDay TotalSleepRecords TotalMinutesAsleep TotalTimeInBed  
## 1 1503960366 4/12/2016 0:00 1 327 346  
## 2 1503960366 4/13/2016 0:00 2 384 407  
## 3 1503960366 4/15/2016 0:00 1 412 442  
## 4 1503960366 4/16/2016 0:00 2 340 367  
## 5 1503960366 4/17/2016 0:00 1 700 712  
## 6 1503960366 4/19/2016 0:00 1 304 320

head(WeightLogInfo)

## Id Date WeightKg WeightPounds BMI  
## 1 1503960366 5/2/2016 23:59 52.6 115.9631 22.65  
## 2 1503960366 5/3/2016 23:59 52.6 115.9631 22.65  
## 3 1927972279 4/13/2016 1:08 133.5 294.3171 47.54  
## 4 2873212765 4/21/2016 23:59 56.7 125.0021 21.45  
## 5 2873212765 5/12/2016 23:59 57.3 126.3249 21.69  
## 6 4319703577 4/17/2016 23:59 72.4 159.6147 27.45

### Checking the length of Id column

length(unique(Activity$Id))

## [1] 33

length(unique(Hourly\_Intensities$Id))

## [1] 33

length(unique(Heartrate\_Seconds$Id))

## [1] 7

length(unique(Sleep\_Day$Id))

## [1] 24

length(unique(WeightLogInfo$Id))

## [1] 8

length(unique(Hourly\_Calories$Id))

## [1] 33

length(unique(Hourly\_Steps$Id))

## [1] 33

### Check for duplicates

any(duplicated(Activity))

## [1] FALSE

any(duplicated(Hourly\_Calories))

## [1] FALSE

any(duplicated(Hourly\_Intensities))

## [1] FALSE

any(duplicated(Hourly\_Steps))

## [1] FALSE

any(duplicated(Heartrate\_Seconds))

## [1] TRUE

any(duplicated(Sleep\_Day))

## [1] TRUE

any(duplicated(WeightLogInfo))

## [1] FALSE

### remove duplicate from Sleep\_Day

Sleep\_Day <- distinct(Sleep\_Day)  
any(duplicated(Sleep\_Day))

## [1] FALSE

here in most of files we have unnecessory columns are present which we don’t have to use so we are going to delete them first

Activity <- subset(Activity,select=-c(TrackerDistance,LoggedActivitiesDistance,SedentaryActiveDistance))  
  
head(Activity)

## Id ActivityDate TotalSteps TotalDistance VeryActiveDistance  
## 1 1503960366 4/12/2016 13162 8.50 1.88  
## 2 1503960366 4/13/2016 10735 6.97 1.57  
## 3 1503960366 4/14/2016 10460 6.74 2.44  
## 4 1503960366 4/15/2016 9762 6.28 2.14  
## 5 1503960366 4/16/2016 12669 8.16 2.71  
## 6 1503960366 4/17/2016 9705 6.48 3.19  
## ModeratelyActiveDistance LightActiveDistance VeryActiveMinutes  
## 1 0.55 6.06 25  
## 2 0.69 4.71 21  
## 3 0.40 3.91 30  
## 4 1.26 2.83 29  
## 5 0.41 5.04 36  
## 6 0.78 2.51 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

WeightLogInfo <- subset(WeightLogInfo,select=-c(WeightPounds,Id))  
head(WeightLogInfo)

## Date WeightKg BMI  
## 1 5/2/2016 23:59 52.6 22.65  
## 2 5/3/2016 23:59 52.6 22.65  
## 3 4/13/2016 1:08 133.5 47.54  
## 4 4/21/2016 23:59 56.7 21.45  
## 5 5/12/2016 23:59 57.3 21.69  
## 6 4/17/2016 23:59 72.4 27.45

### 1. change the data type from char to date of ActivityDate and Adding Day\_of\_Week column to Activity :

Activity <- Activity %>%  
mutate(ActivityDate = mdy(ActivityDate), Day\_of\_Week = weekdays(ActivityDate))

### let’s check the added column in Activity dataframe

head(Activity)

## Id ActivityDate TotalSteps TotalDistance VeryActiveDistance  
## 1 1503960366 2016-04-12 13162 8.50 1.88  
## 2 1503960366 2016-04-13 10735 6.97 1.57  
## 3 1503960366 2016-04-14 10460 6.74 2.44  
## 4 1503960366 2016-04-15 9762 6.28 2.14  
## 5 1503960366 2016-04-16 12669 8.16 2.71  
## 6 1503960366 2016-04-17 9705 6.48 3.19  
## ModeratelyActiveDistance LightActiveDistance VeryActiveMinutes  
## 1 0.55 6.06 25  
## 2 0.69 4.71 21  
## 3 0.40 3.91 30  
## 4 1.26 2.83 29  
## 5 0.41 5.04 36  
## 6 0.78 2.51 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  
## Day\_of\_Week  
## 1 Tuesday  
## 2 Wednesday  
## 3 Thursday  
## 4 Friday  
## 5 Saturday  
## 6 Sunday

### add new columnSleep\_Amount to Sleep\_Day dataframe

Sleep\_Day = Sleep\_Day %>%   
mutate(Sleep\_Amount = case\_when(TotalMinutesAsleep/60>=6.0 & TotalMinutesAsleep/60<=9.0 ~ "Good Sleep",  
 TotalMinutesAsleep/60<6.0 ~ "Under Sleep",   
 TotalMinutesAsleep/60>9.0 ~ "Over sleep"))

### let’s see the data

head(Sleep\_Day)

## Id SleepDay TotalSleepRecords TotalMinutesAsleep TotalTimeInBed  
## 1 1503960366 4/12/2016 0:00 1 327 346  
## 2 1503960366 4/13/2016 0:00 2 384 407  
## 3 1503960366 4/15/2016 0:00 1 412 442  
## 4 1503960366 4/16/2016 0:00 2 340 367  
## 5 1503960366 4/17/2016 0:00 1 700 712  
## 6 1503960366 4/19/2016 0:00 1 304 320  
## Sleep\_Amount  
## 1 Under Sleep  
## 2 Good Sleep  
## 3 Good Sleep  
## 4 Under Sleep  
## 5 Over sleep  
## 6 Under Sleep

### we are going to delete some columns from Sleep\_Day dataframe because we do’t need it

Sleep\_Day <- subset(Sleep\_Day,select=-c(SleepDay,TotalSleepRecords,TotalTimeInBed))

### Let’s check the dataframe

head(Sleep\_Day)

## Id TotalMinutesAsleep Sleep\_Amount  
## 1 1503960366 327 Under Sleep  
## 2 1503960366 384 Good Sleep  
## 3 1503960366 412 Good Sleep  
## 4 1503960366 340 Under Sleep  
## 5 1503960366 700 Over sleep  
## 6 1503960366 304 Under Sleep

### since Id is in dbl data type we have to convert it into character type of all dataframes

Activity <- mutate(Activity, Id=as.character('Id'))  
Hourly\_Intensities <-mutate(Hourly\_Intensities,Id=as.character('Id'))  
Hourly\_Calories <- mutate(Hourly\_Calories,Id=as.character('Id'))  
Hourly\_Steps <- mutate(Hourly\_Steps,Id=as.character('Id'))  
Heartrate\_Seconds <- mutate(Heartrate\_Seconds,Id=as.character('Id'))  
Sleep\_Day <- mutate(Sleep\_Day,Id=as.character('Id'))  
WeightLogInfo <-mutate(WeightLogInfo,Id=as.character('Id'))

### check it

summary(Activity)

## Id ActivityDate TotalSteps TotalDistance   
## Length:940 Min. :2016-04-12 Min. : 0 Min. : 0.000   
## Class :character 1st Qu.:2016-04-19 1st Qu.: 3790 1st Qu.: 2.620   
## Mode :character Median :2016-04-26 Median : 7406 Median : 5.245   
## Mean :2016-04-26 Mean : 7638 Mean : 5.490   
## 3rd Qu.:2016-05-04 3rd Qu.:10727 3rd Qu.: 7.713   
## Max. :2016-05-12 Max. :36019 Max. :28.030   
## VeryActiveDistance ModeratelyActiveDistance LightActiveDistance  
## Min. : 0.000 Min. :0.0000 Min. : 0.000   
## 1st Qu.: 0.000 1st Qu.:0.0000 1st Qu.: 1.945   
## Median : 0.210 Median :0.2400 Median : 3.365   
## Mean : 1.503 Mean :0.5675 Mean : 3.341   
## 3rd Qu.: 2.053 3rd Qu.:0.8000 3rd Qu.: 4.782   
## Max. :21.920 Max. :6.4800 Max. :10.710   
## 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 Day\_of\_Week   
## Min. : 0 Length:940   
## 1st Qu.:1828 Class :character   
## Median :2134 Mode :character   
## Mean :2304   
## 3rd Qu.:2793   
## Max. :4900

### changing data type of date column

#Hourly\_Calories  
Hourly\_Calories$ActivityHour=as.POSIXct(Hourly\_Calories$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
Hourly\_Calories$ActivityHour <- format(Hourly\_Calories$ActivityHour, format = "%m/%d/%y")  
  
#Heartrate\_Seconds  
Heartrate\_Seconds$Time=as.POSIXct(Heartrate\_Seconds$Time, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
Heartrate\_Seconds$Time <- format(Heartrate\_Seconds$Time, format = "%m/%d/%y")  
  
# WeightLogInfo  
WeightLogInfo$Date=as.POSIXct(WeightLogInfo$Date, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
WeightLogInfo$Date <- format(WeightLogInfo$Date, format = "%m/%d/%y")  
  
#hourly\_Intensities  
Hourly\_Intensities$ActivityHour=as.POSIXct(Hourly\_Intensities$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
Hourly\_Intensities$ActivityHour <- format(Hourly\_Intensities$ActivityHour, format = "%H:%M:%S")  
Hourly\_Intensities$ActivityHour <- format(Hourly\_Intensities$ActivityHour, format = "%m/%d/%y")

WeightLogInfo$Date <- as.Date(WeightLogInfo$Date)

head(WeightLogInfo)

## Date WeightKg BMI Id  
## 1 <NA> 52.6 22.65 Id  
## 2 <NA> 52.6 22.65 Id  
## 3 <NA> 133.5 47.54 Id  
## 4 <NA> 56.7 21.45 Id  
## 5 <NA> 57.3 21.69 Id  
## 6 <NA> 72.4 27.45 Id

Heartrate\_Seconds$Time <-as.Date(Heartrate\_Seconds$Time)

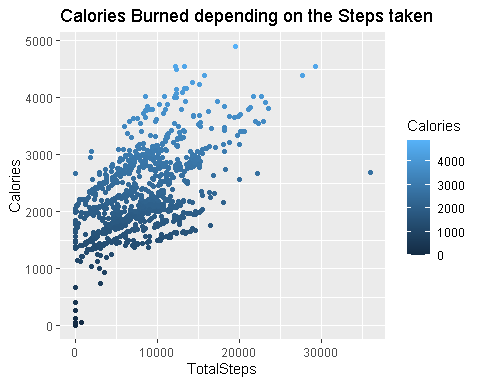
Hourly\_Calories$ActivityHour <-as.Date(Hourly\_Calories$ActivityHour)

head(Hourly\_Calories)

## Id ActivityHour Calories  
## 1 Id 0004-12-16 81  
## 2 Id 0004-12-16 61  
## 3 Id 0004-12-16 59  
## 4 Id 0004-12-16 47  
## 5 Id 0004-12-16 48  
## 6 Id 0004-12-16 48

## Share

ggplot(data=Activity,aes(x=TotalSteps,y=Calories,color=Calories))+geom\_point()+ ggtitle("Calories Burned depending on the Steps taken")



## Observation

we can say smart devices works very good on total steps taken and calories burned

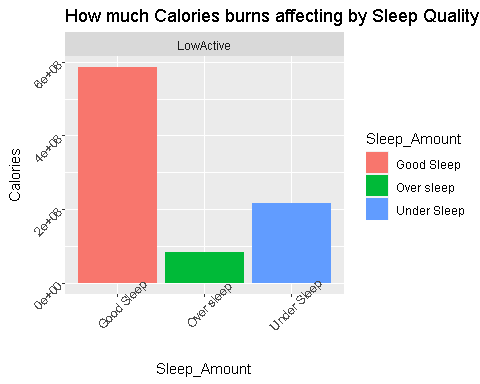
Activity <- merge(Activity,Sleep\_Day,by="Id")

colnames(Activity)

## [1] "Id" "ActivityDate"   
## [3] "TotalSteps" "TotalDistance"   
## [5] "VeryActiveDistance" "ModeratelyActiveDistance"  
## [7] "LightActiveDistance" "VeryActiveMinutes"   
## [9] "FairlyActiveMinutes" "LightlyActiveMinutes"   
## [11] "SedentaryMinutes" "Calories"   
## [13] "Day\_of\_Week" "TotalMinutesAsleep"   
## [15] "Sleep\_Amount"

Activity <- Activity %>%  
 mutate(DistanceCategory= case\_when(  
 Activity$TotalDistance < 4500 ~ "LowActive",  
 Activity$TotalDistance >= 5000 & .$TotalDistance < 6000 ~ "AverageActive",  
 Activity$TotalDistance >= 6500 & .$TotalDistance < 7000 ~ "MorethanAverageActive",  
 Activity$TotalDistance > 7500 ~ "VeryActive"  
 ))

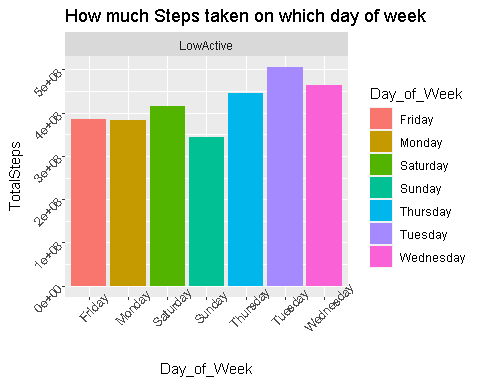
ggplot(data = Activity)+  
 geom\_col(mapping = aes(Sleep\_Amount, y= Calories, fill =Sleep\_Amount))+  
 facet\_grid(~DistanceCategory)+  
 labs(title = "How much Calories burns affecting by Sleep Quality By Distance Categories")+  
 theme(axis.text = element\_text(angle = 45))



## Observations

we can see the smart device works very well on sleep data and calories burned

ggplot(data = Activity)+  
 geom\_col(mapping = aes(Day\_of\_Week, y=TotalSteps, fill =Day\_of\_Week))+  
 facet\_grid(~DistanceCategory)+  
 labs(title = "How much Steps taken on which day of week")+  
 theme(axis.text = element\_text(angle = 45))



Observations\_ by data viz we can say on sunday there are less data for totalsteps taken due to holiday on tuesday again back to routine work so good steps are taken and on other days average steps are taken

## Recommendations:

The app which is made must have to update to track all the data which is needed from the smart devices.

So I would like to suggest the company must have to work on the app to get these information consistently and accurately.

Because the smart devices are rooted to the mobile to track the information from the consumers through the smart devices.

The age of the consumer not tracked so I would like to add the age data to this app to get accurate results.

### 2. Add new column TotalActiveMinutes to Activity which is the sum of all three ActiveMinutes columns:

Activity <- Activity %>%  
 rowwise() %>%  
 mutate(TotalActiveMinutes = sum(c(VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes)))

### Next, I created new columns with categories in datasets

### 1. Sleep\_Day

Sleep\_Day = Sleep\_Day %>% mutate(Sleep\_Amount = case\_when(TotalMinutesAsleep/60>=6.0 & TotalMinutesAsleep/60<=9.0 ~ "Good Sleep",  
 TotalMinutesAsleep/60<6.0 ~ "Under Sleep",   
 TotalMinutesAsleep/60>9.0 ~ "Over sleep"))  
   
 New\_Weight <- WeightLogInfo %>%  
 select(Id,Date,WeightKg,BMI)

### since Id is in dbl data type we have to convert it into character type of all dataframes

Activity <- mutate(Activity, Id=as.character('Id'))  
Hourly\_Intensities <-mutate(Hourly\_Intensities,Id=as.character('Id'))  
Hourly\_Calories <- mutate(Hourly\_Calories,Id=as.character('Id'))  
Hourly\_Steps <- mutate(Hourly\_Steps,Id=as.character('Id'))  
Heartrate\_Seconds <- mutate(Heartrate\_Seconds,Id=as.character('Id'))  
Sleep\_Day <- mutate(Sleep\_Day,Id=as.character('Id'))  
WeightLogInfo <-mutate(WeightLogInfo,Id=as.character('Id'))

### check it

summary(Activity)

## Id ActivityDate TotalSteps TotalDistance   
## Length:385400 Min. :2016-04-12 Min. : 0 Min. : 0.000   
## Class :character 1st Qu.:2016-04-19 1st Qu.: 3790 1st Qu.: 2.620   
## Mode :character Median :2016-04-26 Median : 7406 Median : 5.245   
## Mean :2016-04-26 Mean : 7638 Mean : 5.490   
## 3rd Qu.:2016-05-04 3rd Qu.:10727 3rd Qu.: 7.713   
## Max. :2016-05-12 Max. :36019 Max. :28.030   
## VeryActiveDistance ModeratelyActiveDistance LightActiveDistance  
## Min. : 0.000 Min. :0.0000 Min. : 0.000   
## 1st Qu.: 0.000 1st Qu.:0.0000 1st Qu.: 1.945   
## Median : 0.210 Median :0.2400 Median : 3.365   
## Mean : 1.503 Mean :0.5675 Mean : 3.341   
## 3rd Qu.: 2.053 3rd Qu.:0.8000 3rd Qu.: 4.782   
## Max. :21.920 Max. :6.4800 Max. :10.710   
## 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 Day\_of\_Week TotalMinutesAsleep Sleep\_Amount   
## Min. : 0 Length:385400 Min. : 58.0 Length:385400   
## 1st Qu.:1828 Class :character 1st Qu.:361.0 Class :character   
## Median :2134 Mode :character Median :432.5 Mode :character   
## Mean :2304 Mean :419.2   
## 3rd Qu.:2793 3rd Qu.:490.0   
## Max. :4900 Max. :796.0   
## DistanceCategory TotalActiveMinutes  
## Length:385400 Min. : 0.0   
## Class :character 1st Qu.:146.8   
## Mode :character Median :247.0   
## Mean :227.5   
## 3rd Qu.:317.2   
## Max. :552.0

Activity = Activity %>%   
 mutate(Steps\_Amount = case\_when(TotalSteps<=4500 ~ "Less Walker",  
 TotalSteps>4000 & TotalSteps <=9000 ~ "Good Walker", TotalSteps>9000 & TotalSteps<=12000 ~ "Better Walker",  
 TotalSteps>12000 ~ "Best Walker"))

New\_Weight = New\_Weight %>%   
mutate(Weight\_Amount = case\_when(BMI <= 18.5 ~ "UnderWeight", BMI >= 18.6 & BMI <= 24.9 ~ "NormalWeight", BMI >= 25 & BMI <= 29.9 ~ "OverWeight", BMI >= 30 ~ "Obesity"))

Heartrate\_Seconds = Heartrate\_Seconds %>%  
mutate(Heartrate\_Amount = case\_when(Value <= 80 ~ "Normal B.P.",   
Value >= 81 ~ "High B.P."))

Activity = Activity %>% mutate(Burned\_Calories = case\_when(Calories<=1800 ~ "Low", Calories>1800 & Calories<=2200 ~ "Medium",  
 Calories>2200 & Calories<=2600 ~ "High", Calories>2600 ~ "Very High"))

### 4. Sedentary Time

Activity = Activity %>%   
 mutate(Sedentary\_Time = case\_when(SedentaryMinutes > 626 ~ "Good",  
 SedentaryMinutes < 627 & SedentaryMinutes ~ "Good", SedentaryMinutes/60>10 & SedentaryMinutes/60<=12 ~ "Bad",  
 SedentaryMinutes/60>12 ~ "Very Bad"))

### changing data type of date column

### Hourly\_Calories

Hourly\_Calories$ActivityHour=as.POSIXct(Hourly\_Calories$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
Hourly\_Calories$ActivityHour <- format(Hourly\_Calories$ActivityHour, format = "%m/%d/%y")

### Heartrate\_Seconds

Heartrate\_Seconds$Time=as.POSIXct(Heartrate\_Seconds$Time, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
Heartrate\_Seconds$Time <- format(Heartrate\_Seconds$Time, format = "%m/%d/%y")

### WeightLogInfo

WeightLogInfo$Date=as.POSIXct(WeightLogInfo$Date, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
WeightLogInfo$Date <- format(WeightLogInfo$Date, format = "%m/%d/%y")

### hourly\_Intensities

Hourly\_Intensities$ActivityHour=as.POSIXct(Hourly\_Intensities$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())  
Hourly\_Intensities$ActivityHour <- format(Hourly\_Intensities$ActivityHour, format = "%H:%M:%S")  
Hourly\_Intensities$ActivityHour <- format(Hourly\_Intensities$ActivityHour, format = "%m/%d/%y")  
head(WeightLogInfo)

## Date WeightKg BMI Id  
## 1 <NA> 52.6 22.65 Id  
## 2 <NA> 52.6 22.65 Id  
## 3 <NA> 133.5 47.54 Id  
## 4 <NA> 56.7 21.45 Id  
## 5 <NA> 57.3 21.69 Id  
## 6 <NA> 72.4 27.45 Id

### Recommendations:

The app which is made must have to update to track all the data which is needed from the smart devices.

So I would like to suggest the company must have to work on the app to get these information consistently and accurately.

Because the smart devices are rooted to the mobile to track the information from the consumers through the smart devices.

The age of the consumer not tracked so I would like to add the age data to this app to get accurate results.