How Can a Wellness Technology Company Play It Smart? - Bellabeat

Ask

Characters and products

● Stakeholders

○ Urška Sršen: Bellabeat's cofounder and Chief Creative Officer ○ Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team ○ Bellabeat marketing analytics team

● About Bellabeat Products

1. Bellabeat app: The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products.
2. Leaf: Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress.
3. Time: This wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress. The Time watch connects to the Bellabeat app to provide you with insights into your daily wellness.
4. Spring: This is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels.
5. Bellabeat membership: Bellabeat also offers a subscription-based membership program for users. Membership gives users 24/7 access to fully personalized guidance on nutrition, activity, sleep, health and beauty, and mindfulness based on their lifestyle and goals.

About the company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women.

By 2016, Bellabeat had opened offices around the world and launched multiple products. Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website. The company has invested in

traditional advertising media, such as radio, out-of-home billboards, print, and television, but focuses on digital marketing extensively. Bellabeat invests year-round in Google Search, maintaining active Facebook and Instagram pages, and consistently engages consumers on Twitter. Additionally, Bellabeat runs video ads on Youtube and display ads on the Google Display Network to support campaigns around key marketing dates.

Sršen knows that an analysis of Bellabeat's available consumer data would reveal more opportunities for growth. She has asked the marketing analytics team to focus on a Bellabeat product and analyze smart device usage data in order to gain insight into how people are already using their smart devices. Then, using this information, she would like high-level recommendations for how these trends can inform Bellabeat marketing strategy.

Business task

1. kaggle datasets download -d arashnic/fitbit

Process

Data source: \href{https://www.kaggle.com/arashnic/fitbit}

Installing and loading needed packages and libraries

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

Importing the datasets

```
## 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.
## Rows: 940 Columns: 3
## ── Column specification ──────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityDay
```

```
## dbl (2): Id, Calories
##
## 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.
## 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.
## 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.
## Rows: 2483658 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Time
## dbl (2): Id, Value
##
## 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.
## Rows: 22099 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, Calories
##
## 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.
## Rows: 22099 Columns: 4
## ── Column specification
──────────────────────────────────────────────────────────────────────────
## Delimiter: ","
```

```
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
##
## 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.
## Rows: 22099 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityHour
## 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.
## Rows: 1325580 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Calories
##
## 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.
## Rows: 1325580 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Intensity
##
## 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.
## Rows: 1325580 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Steps
##
## 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.
## Rows: 1325580 Columns: 3
## ── Column specification
──────────────────────────────────────────────────────────────────────────────
## Delimiter: ","
```

```
## chr (1): ActivityMinute
## dbl (2): Id, METs
##
## 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.
## Rows: 188521 Columns: 4
## ── Column specification
──────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): date
## dbl (3): Id, value, logId
##
## 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.
## 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.
## Rows: 67 Columns: 8
## ── Column specification
──────────────────────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Date
## dbl (6): Id, WeightKg, WeightPounds, Fat, BMI, LogId
## lgl (1): IsManualReport
##
## 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.

sleepDay <-
  read_csv("C:/Users/Koyenikan Arinola/Desktop/New
folder/Fitabase_data/sleepDay_merged.csv")

## 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.
```

Data summary

Name	hourlysteps
Number of rows	22099
Number of columns	3

_____

Column type frequency:	
character	1
numeric	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityHour	0	1	19	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.848235e+09	2.422e+09	15039	23201	44451	69621	88776
StepTotal	0	1	3.201700e+02	6.903e+02	60366	27002	14986	81067	89391
					0	0	40	357	10554

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

Data summary

Name	hourlyCalories
Number of rows	22099
Number of columns	3

_____

Column type frequency:	
character	1
numeric	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityHour	0	1	19	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.848235e+09	2.422e+09	15039	23201	44451	69621	88776
Calories	0	1	9.739000e+01	6.070e+01	42	63	83	108	948

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

Data summary

Name	hourlyintensities
Number of rows	22099
Number of columns	4

_____

Column type frequency:	
character	1
numeric	3

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityHour	0	1	16	18	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.848235e+09	2.422e+09	15039	23201	4.445115e+09	6.962181e+09	88776
TotalIntensity	0	1	1.204000e+01	2.113e+01	0	0	3.000000e+00	1.600000e+01	180
AverageIntensity	0	1	2.000000e-01	3.500e-01	0	0	5.000000e-02	2.700000e-01	3

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

Data summary

Name	hourlyCalories
Number of rows	22099
Number of columns	3

_____

Column type frequency:	
character	1
numeric	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityHour	0	1	19	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.848235e+09	2.422e+09	15039	23201	44451	69621	88776
Calories	0	1	9.739000e+01	6.070e+01	42	63	83	108	948

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

Data summary

Name	heartrate_seconds
Number of rows	2483658
Number of columns	3

_____

Column type frequency:	
character	1
numeric	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Time	0	1	19	21	0	961274	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	5.513765e+09	3761.0	20224844	43881618	55539574	69621810	8877689391
Value	0	1	7.733000e+01	19.4	36	63	73	88	203

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

Data summary

Name	dailyActivity
Number of rows	940
Number of columns	15

_____

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	2.320127e+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

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
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	2.660000e+00	0	0.000000e+00	2.100000e-01	2.050000e+00	2.192000e+01
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.400000e-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	3.284000e+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	1.229500e+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

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

Data summary

Name	dailyCalories
Number of rows	940
Number of columns	3

Column type frequency:

character	1
numeric	2

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDay	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	2320127002.0	4445114986	6.962181e+09	8877689391
Calories	0	1	2.303610e+03	7.181700e+02	0	1828.5	2134	2.793250e+03	4900

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

Data summary

Name	dailyIntensities
Number of rows	940
Number of columns	10

Column type frequency:

character	1
numeric	9

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDay	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	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
SedentaryMinutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.229500e+03	1.440000e+03
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.180000e+02
FairlyActiveMinutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02
VeryActiveMinutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100000e+02
SedentaryActiveDistance	0	1	0.000000e+00	1.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.400000e-01	8.000000e-01	6.480000e+00
VeryActiveDistance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	2.100000e-01	2.050000e+00	2.192000e+01

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

Data summary

Name	minuteCaloriesNarrow
Number of rows	1325580
Number of columns	3

Column type frequency:

character	1
numeric	2

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityMinute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.847898e+09	2.422313e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
Calories	0	1	1.620000e+00	1.410000e+00	0	9.400000e-01	1.220000e+00	1.430000e+00	1.975000e+01

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

Data summary

Name	minuteIntensitiesNarrow
Number of rows	1325580
Number of columns	3

_____

Column type frequency:	
character	1
numeric	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityMinute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4847897691.9	2.422313e+09	1503960366	2320127002	4451149861	6962181067	8877689391
Intensity	0	1	0.2	5.200000e-01	0	0	0	0	3

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

Data summary

Name	minuteStepsNarrow
Number of rows	1325580
Number of columns	3

_____

Column type frequency:	
character	1
numeric	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityMinute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.847898e+09	2.422313e+09	15039	23201	44451	69621	88776
Steps	0	1	5.340000e+00	1.813000e+01	0	0	0	0	220

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

Data summary

Name	minuteMETsNarrow
Number of rows	1325580
Number of columns	3

_____

Column type frequency:	
character	1
numeric	2

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityMinute	0	1	19	21	0	44160	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.847898e+09	2.422313e+09	1503960366	2320127002	4445114986	6962181067	8877689391
METs	0	1	1.469000e+01	1.206000e+01	0	10	10	11	157

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

Data summary

Name	sleepDay
Number of rows	413
Number of columns	5

_____

Column type frequency:

character	1
numeric	4

_____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
SleepDay	0	1	20	21	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	5.000979e+09	2.06036e+09	1503960366	3977333714	4702921684	6962181067	8792009665
TotalSleepRecords	0	1	1.120000e+00	3.50000e-01	1	1	1	1	3
TotalMinutesAsleep	0	1	4.194700e+02	1.18340e+02	58	361	433	490	796
TotalTimeInBed	0	1	4.586400e+02	1.27100e+02	61	403	463	526	961

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

Data summary

Name	weightLoginfo

Number of rows	67
Number of columns	8

Column type frequency:	
character	1
logical	1
numeric	6

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Date	0	1	19	21	0	56	0

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
IsManualReport	0	1	0.61	TRU: 41, FAL: 26

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1.00	7.009282e+09	1.950322e+09	1.503960e+09	6.962181e+09	6.962181e+09	8.877689e+09	8.877689e+09
WeightKg	0	1.00	7.204000e+01	1.392000e+01	5.260000e+01	6.140000e+01	6.250000e+01	8.505000e+01	1.335000e+02
WeightPounds	0	1.00	1.588100e+02	3.070000e+01	1.159600e+02	1.353600e+02	1.377900e+02	1.875000e+02	2.943200e+02
Fat	65	0.03	2.350000e+01	2.120000e+00	2.200000e+01	2.275000e+01	2.350000e+01	2.425000e+01	2.500000e+01
BMI	0	1.00	2.519000e+01	3.070000e+00	2.145000e+01	2.396000e+01	2.439000e+01	2.556000e+01	4.754000e+01
LogId	0	1.00	1.461772e+12	7.829948e+08	1.460444e+12	1.461079e+12	1.461802e+12	1.462375e+12	1.463098e+12

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

Datasets needed for the analysis are:

hourlySteps, hourlyCalories, hourlyIntensities, Dailyactivity, sleepDay. This is because there are 33 and 24 distinct user's activities to analyze and the goal of this analysis to check for activities by hour and day. weightloginfo and heartrate datasets have 8 and 14 distinct users inputs which is not enough for inference

Preparing the datasets and creating data frames

```
hourlycalories <- hourlyCalories%>%arrange(Id)#For uniform order

hourlysteps
```

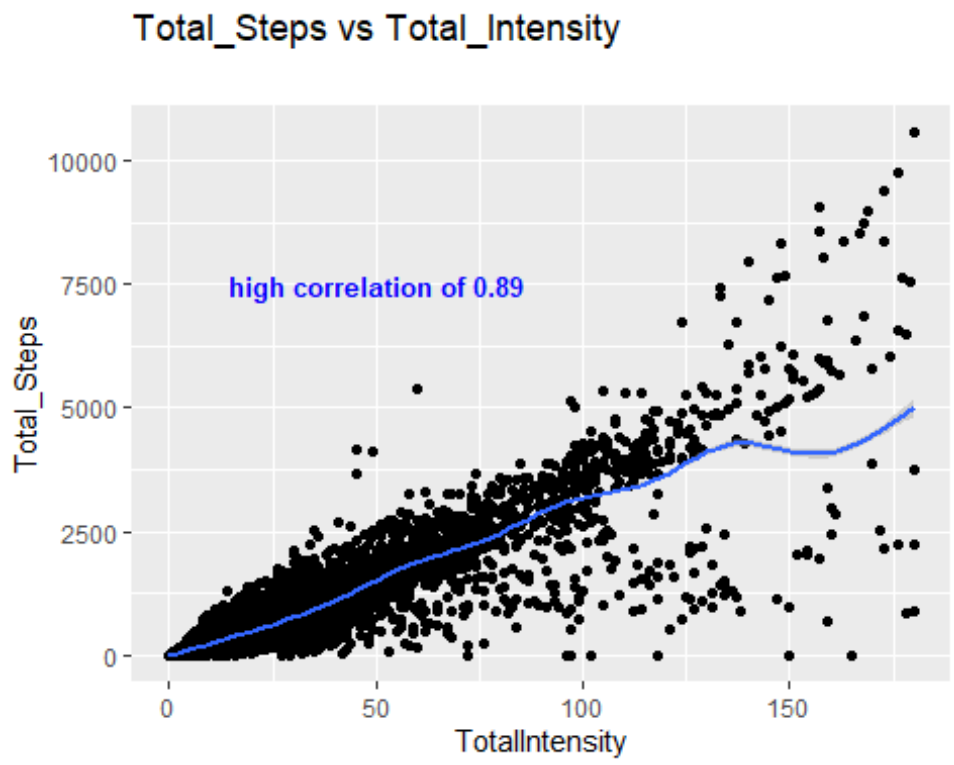
```
## # A tibble: 22,099 × 3
##            Id ActivityHour          StepTotal
##         <dbl> <chr>                     <dbl>
##  1 1503960366 4/12/2016 12:00:00 AM       373
##  2 1503960366 4/12/2016 1:00:00 AM        160
##  3 1503960366 4/12/2016 2:00:00 AM        151
##  4 1503960366 4/12/2016 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
##  7 1503960366 4/12/2016 6:00:00 AM          0
##  8 1503960366 4/12/2016 7:00:00 AM          0
##  9 1503960366 4/12/2016 8:00:00 AM        250
## 10 1503960366 4/12/2016 9:00:00 AM       1864
## # … with 22,089 more rows
```

```
hourlyActivity_df <-
data.frame(hourlyintensities,total_calories=hourlycalories$Calories,
                                Total_Steps= hourlysteps$StepTotal)

hourlyActivity_df_sep<-hourlyActivity_df%>%
  separate(col=ActivityHour,into = c('the_date', 'the_time'),sep=" ")
```

Analyse

Checking for correlation between User's Intensity, Steps, Calories

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

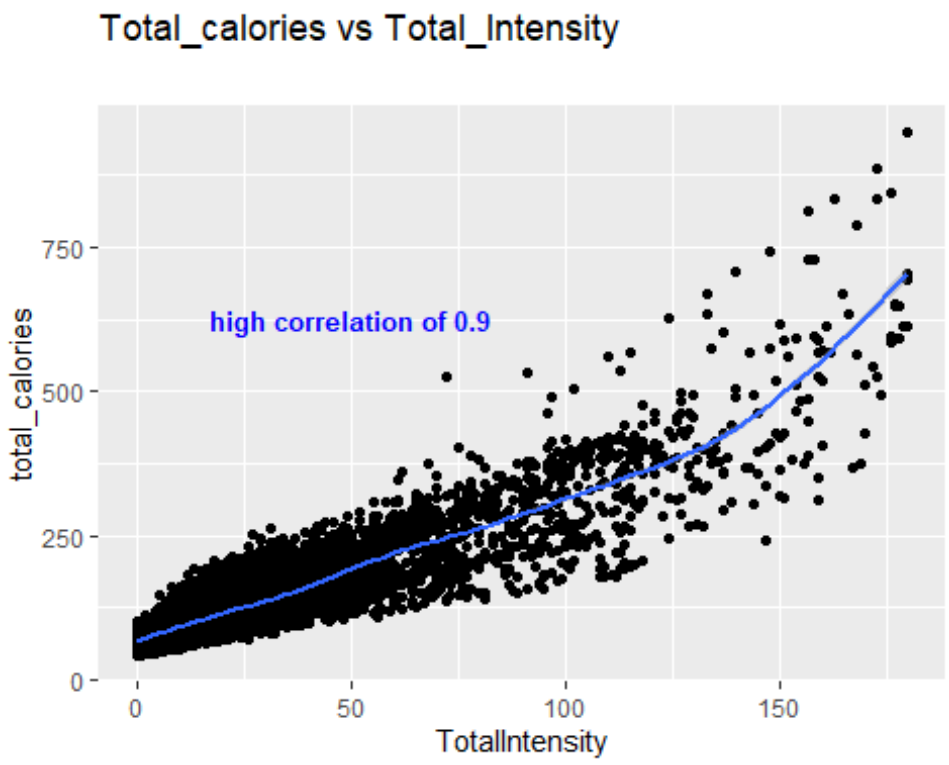



```
##
##  Pearson's product-moment correlation
##
## data:  hourlyActivity_df_sep$TotalIntensity and
hourlyActivity_df_sep$Total_Steps
## t = 299.91, df = 22097, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8933518 0.8985529
## sample estimates:
##      cor
## 0.895983

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


```
##
##  Pearson's product-moment correlation
##
## data:  hourlyActivity_df_sep$Total_Steps and
hourlyActivity_df_sep$total_calories
## t = 209.05, df = 22097, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8104921 0.8193486
## sample estimates:
##      cor
## 0.814968
```

The plot shows a linear relationship between Total_steps and Total_calories burnt. Alerting Users on their total steps taken at intervals and Introducing challenges that helps in taking more steps would help users interested in burning more calories, which will result to happier users.

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

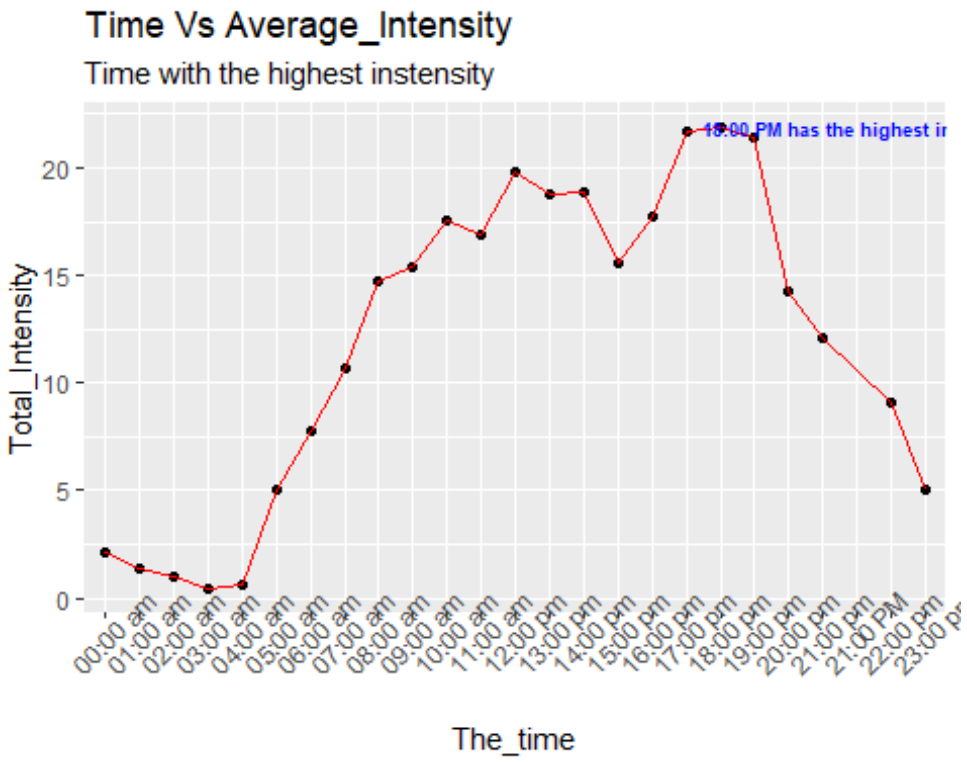


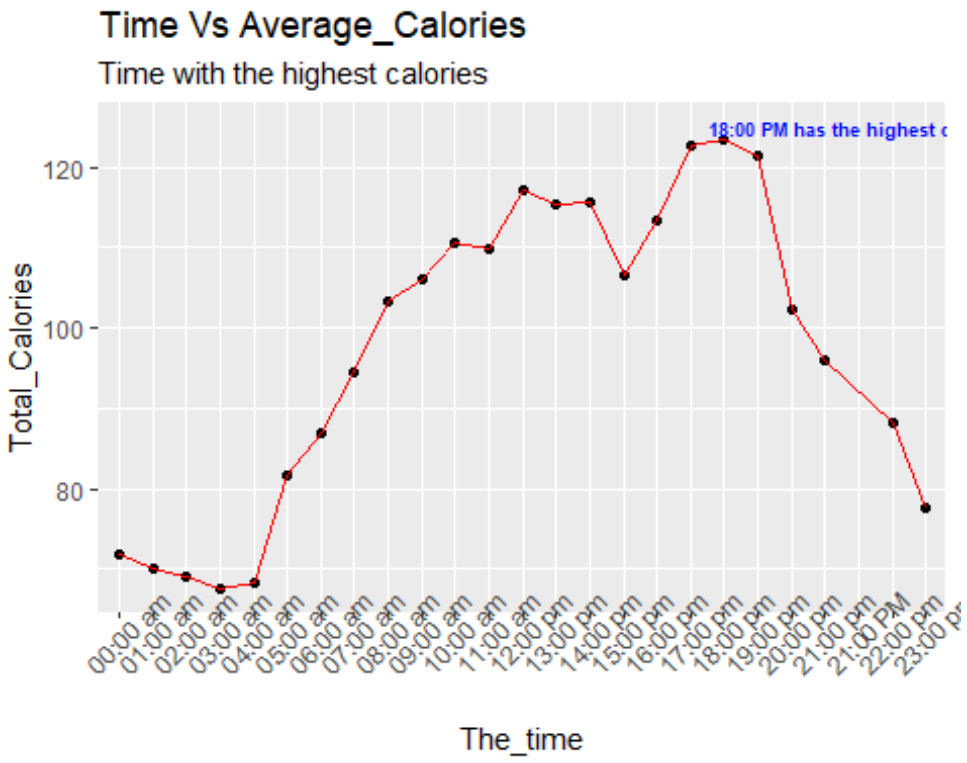

```
##
##  Pearson's product-moment correlation
##
## data:  hourlyActivity_df_sep$TotalIntensity and
hourlyActivity_df_sep$total_calories
## t = 300.99, df = 22097, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8939999 0.8991711
## sample estimates:
##       cor
## 0.8966161
```

The plot shows a positive linear relationship between Total_intensity and Total_calories burnt. Introducing workout challenges would help users interested in burning more calories. ### Checking the for the most active hour of the day

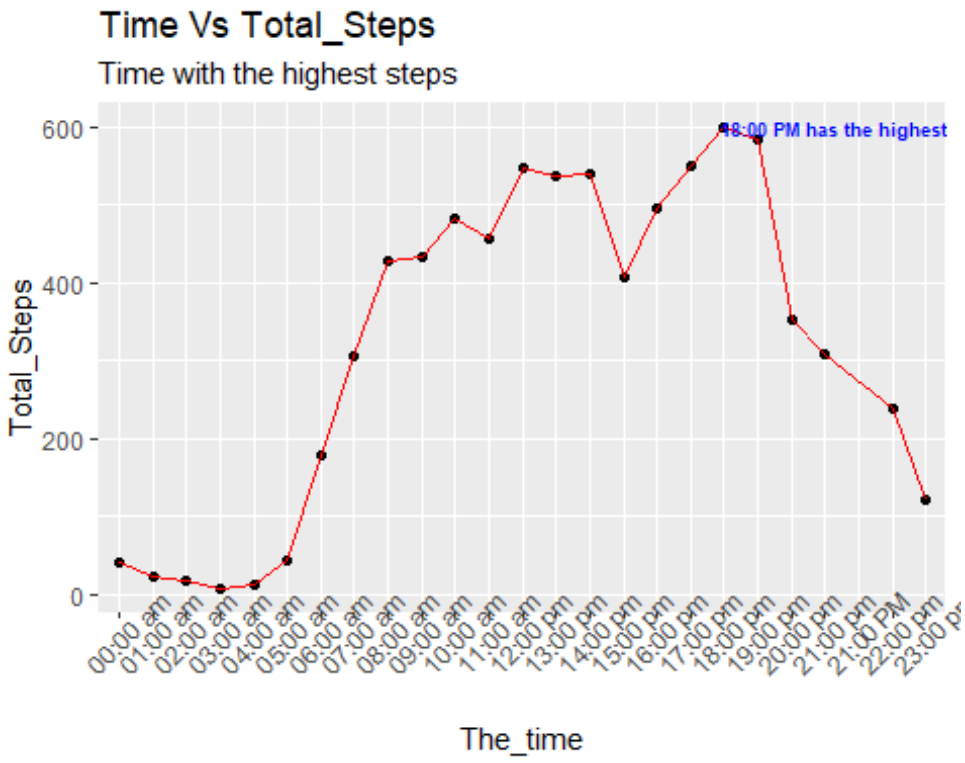
```
## `summarise()` has grouped output by 'The_time'. You can override using the
## `.groups` argument.
```

Checking for the time with the highest intensities and calories burnt





Checking for the time with the most steps



The graph shows that there are more Activities going at 18:00 PM, 17:00 PM and 17:00 PM respectively and little or no activity at 2:00 AM, 3:00 AM and 1:00 AM. Creating a schedule reminder or alerting Users in the evening on steps or intensity would help create a routine which results to healthy habits.

Checking the for the daily trend

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 940 rows
[1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

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

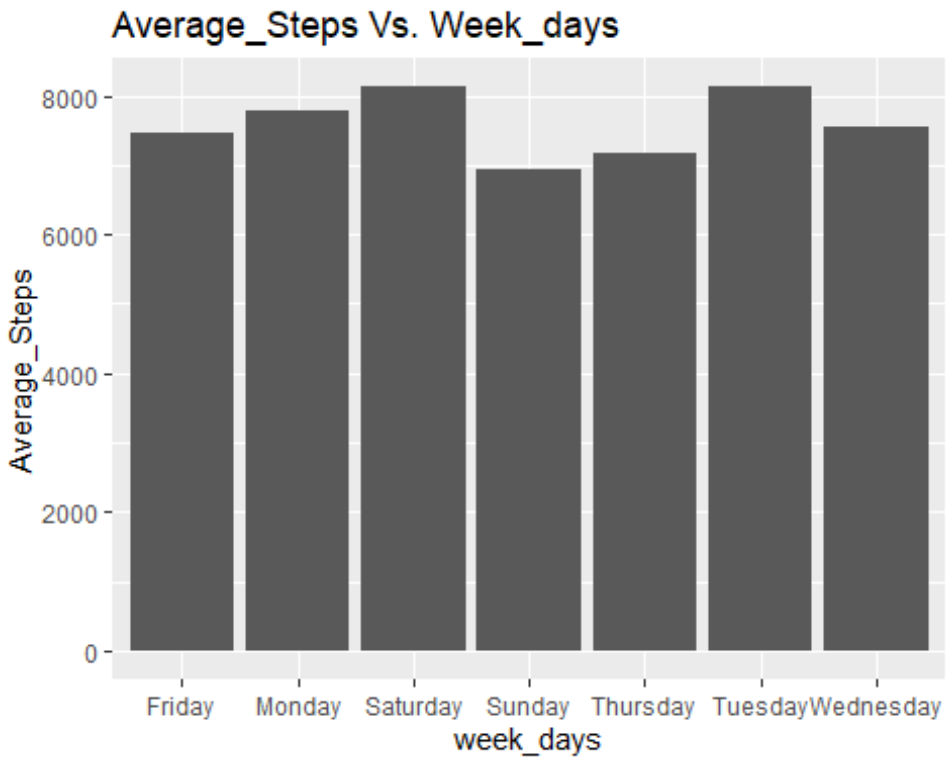
The plots above shows a slight decrease in users activities, creating reminders and challenges would help Users stay consistent.

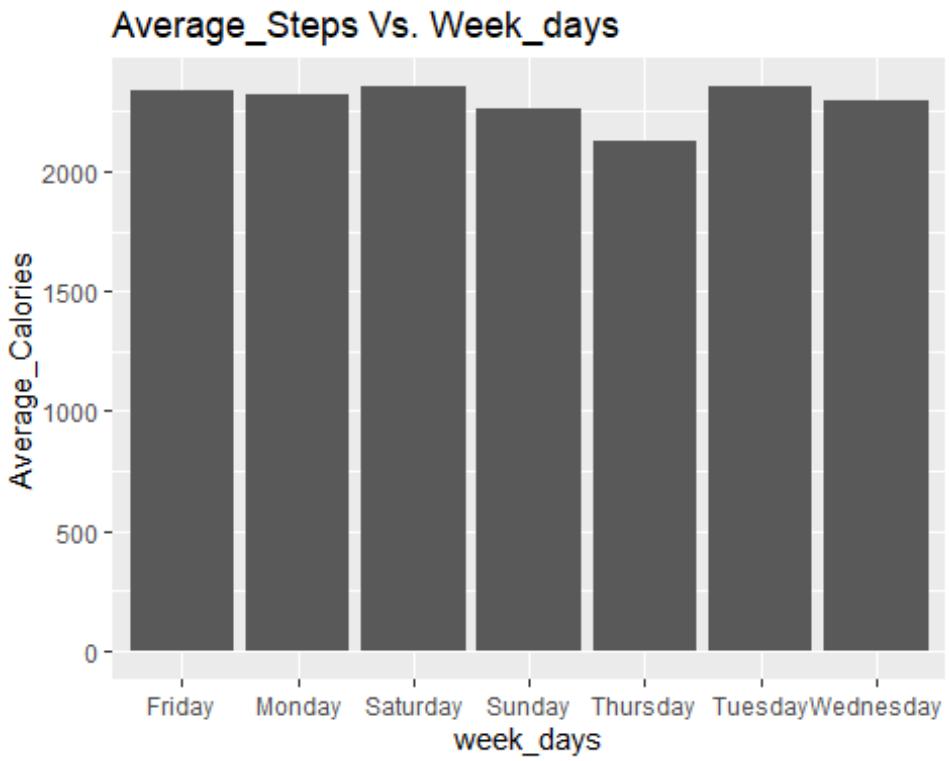
Checking for activites by days of the week

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

## # A tibble: 7 × 3
## # Groups:   week_days [7]
##   week_days Average_Calories Average_Steps
##   <chr>                <dbl>         <dbl>
```

```
## 1 Saturday             2353.         8131
## 2 Tuesday              2353.         8126.
## 3 Monday               2324.         7782
## 4 Wednesday            2297.         7545.
## 5 Friday               2333.         7449.
## 6 Thursday             2130.         7159.
## 7 Sunday               2264.         6937.
```

Users on an average are more active on Saturdays and tuesdays and least active of Sundays



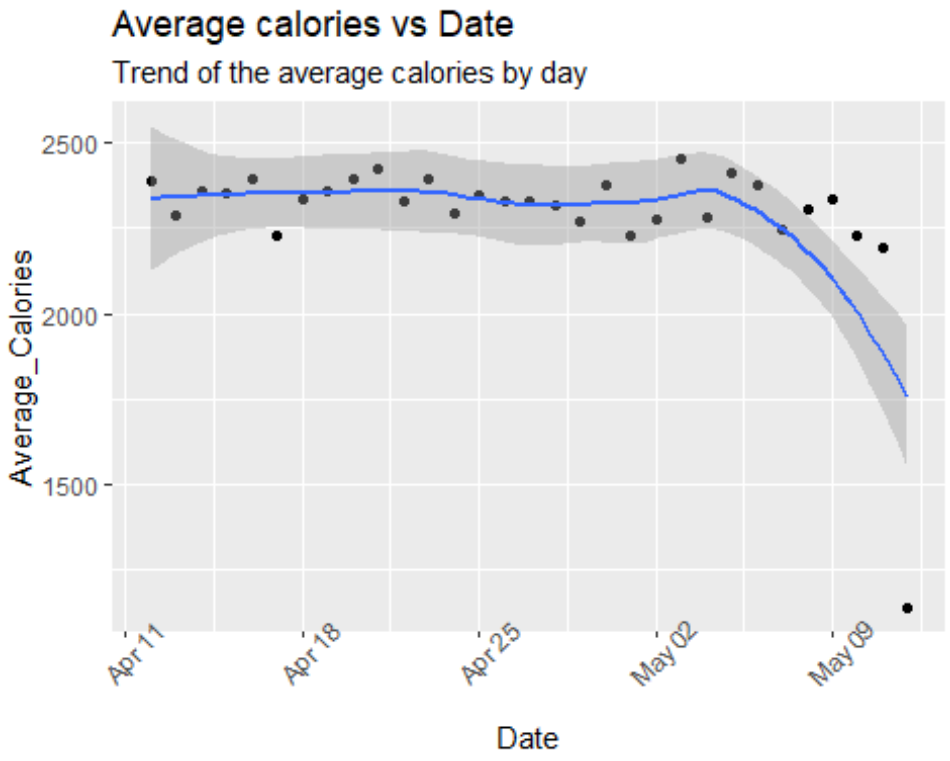


The plots above shows the highest Average steps and Calories in days of the week. Creating a schedule

reminder or alerting Users on the Saturdays or Tuesdays or days the would be available would help create a weekly routine which results to healthy habits.

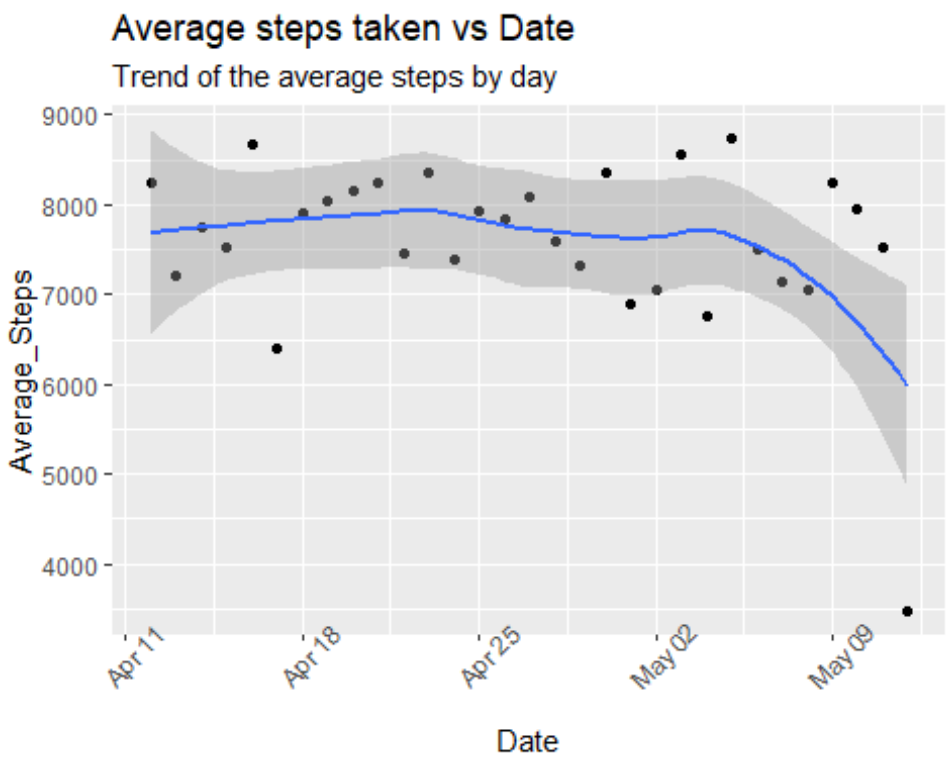
there's a Slight decrease in average calories taken by day

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

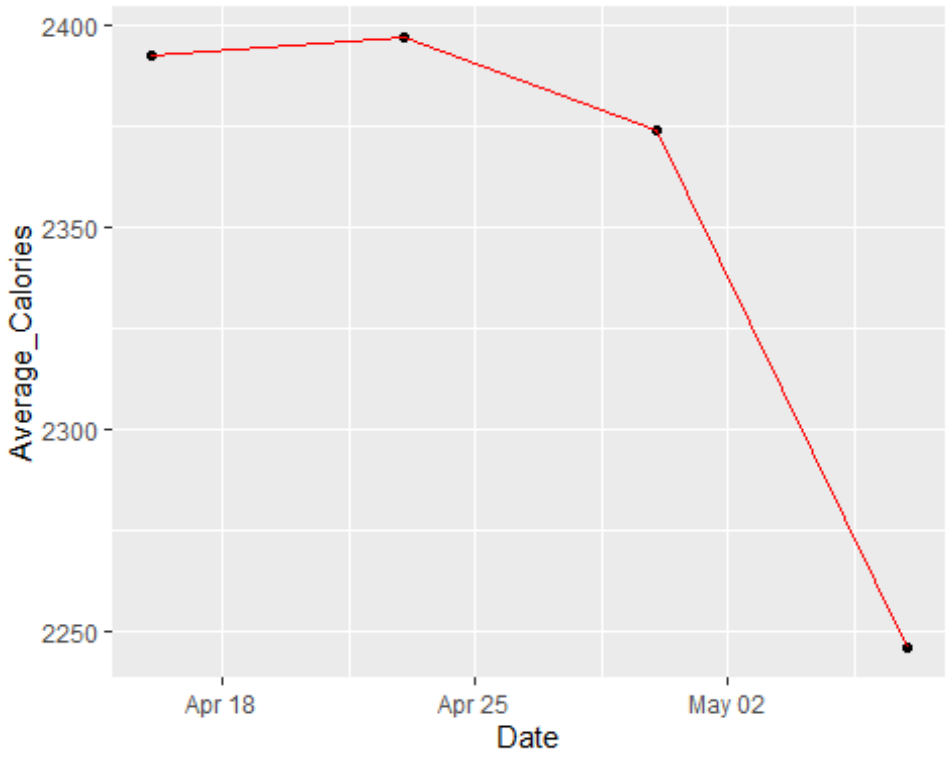


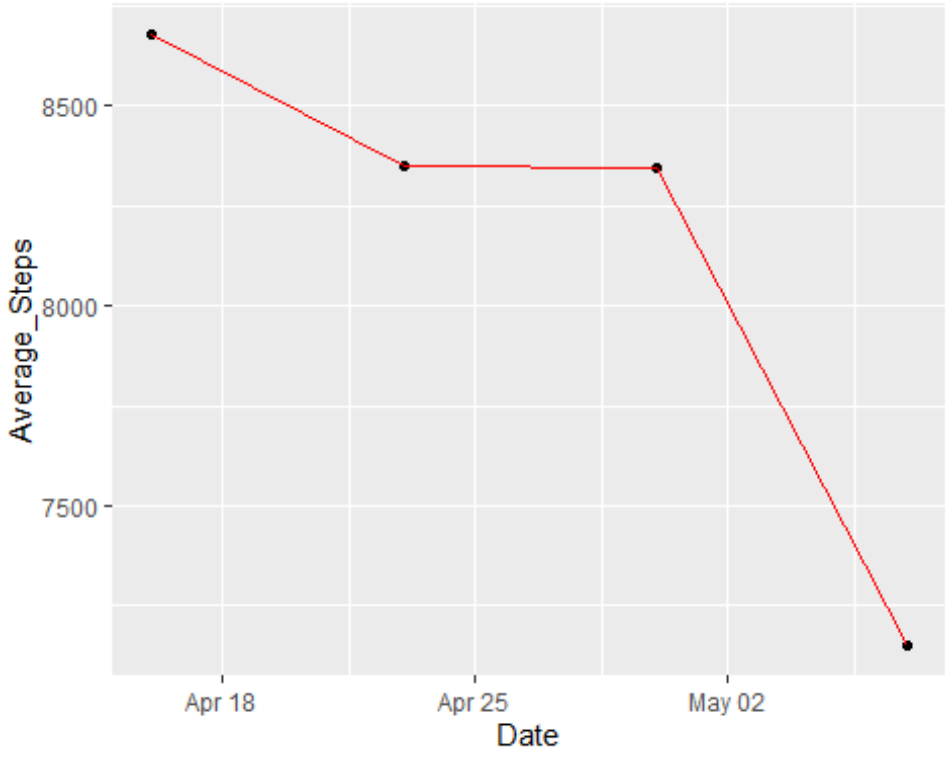
there's a Slight decrease in average steps taken by day

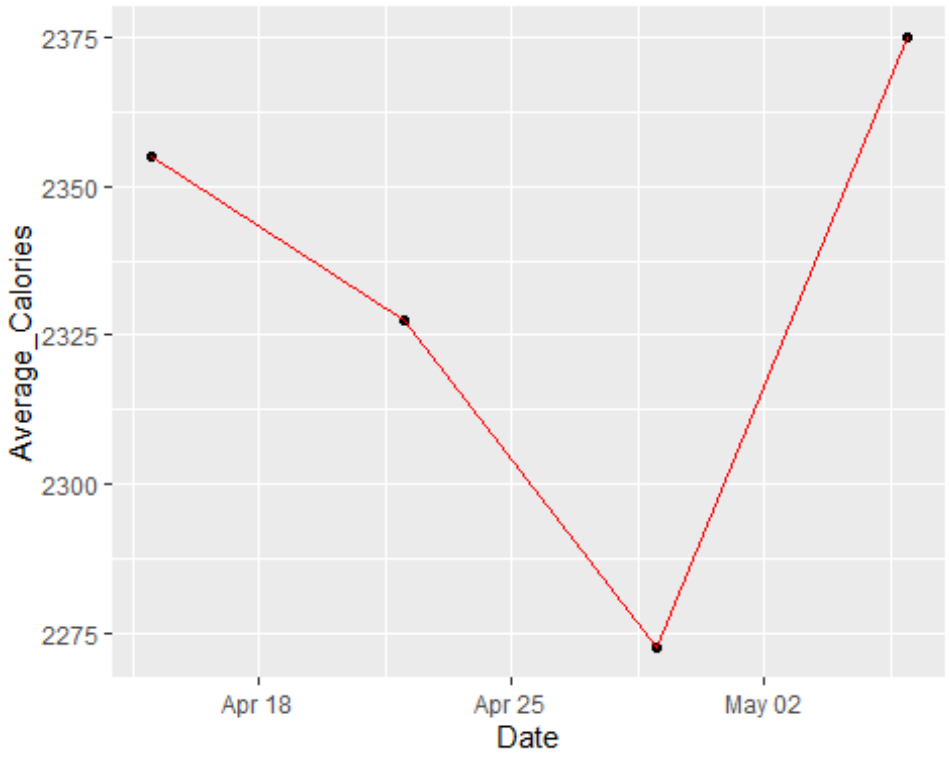
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

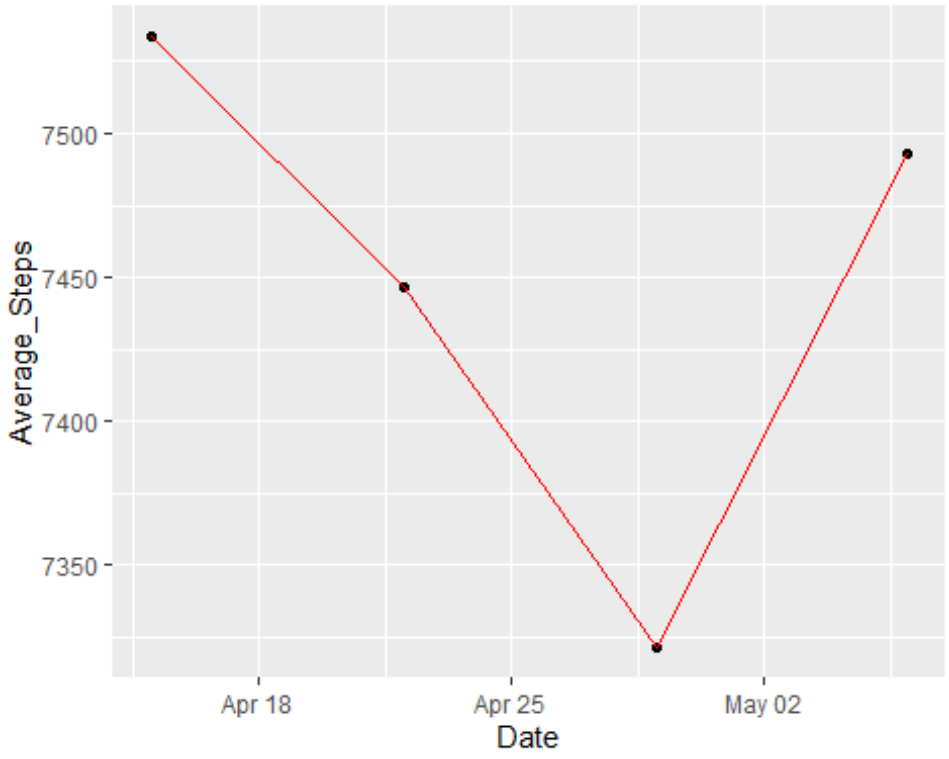


Verifying the decline by day filtering saturdays and tuesdays


2400
2350
2300
2250
Average_Calories
Apr 18
Apr 25
May 02
Date


8500
8000
7500
Average_Steps
Apr 18
Apr 25
May 02
Date





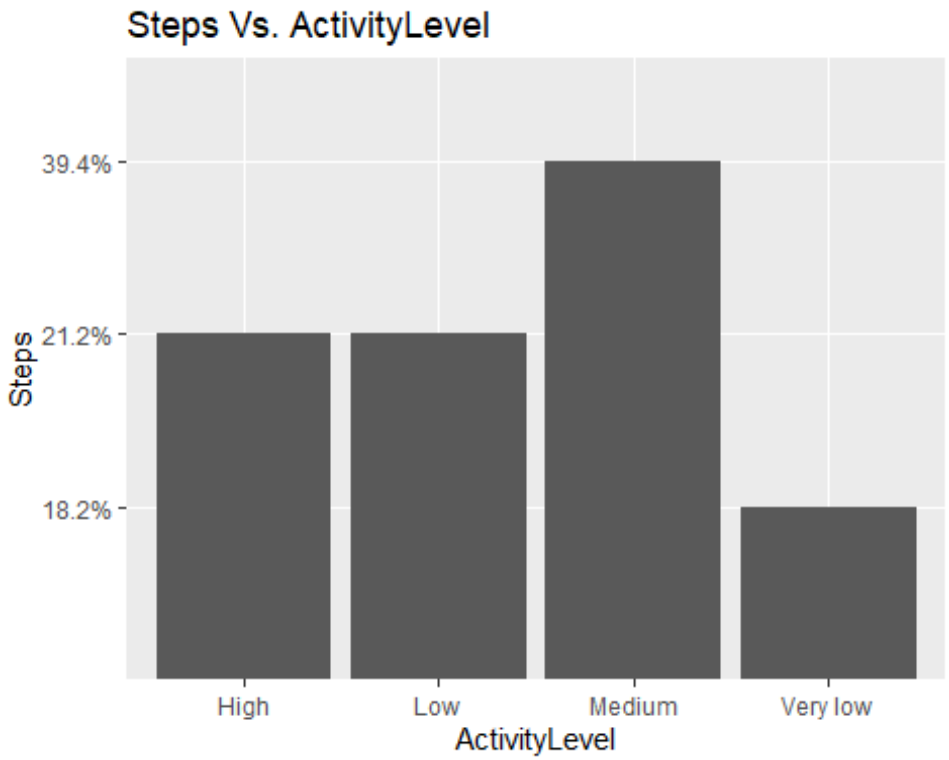
From the Plots above, there's a decline on saturdays but inconsistent variation on tuesdays

Checking for the general percentage of Users' activities based on levels:

The average calories and steps by day representation

```
##   UserActivity_info1 UserActivity_infoCalories UserActivity_infoSteps
## 1      ActivityLevel                 Calories                Calories
## 2               High               Above 2700             Above 10000
## 3                Low     between 1600 and 2000  between 4000 and 7000
## 4             Medium     between 2000 and 2700 between 7000 and 10000
## 5           Very Low            less than 1600          less than 4000

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

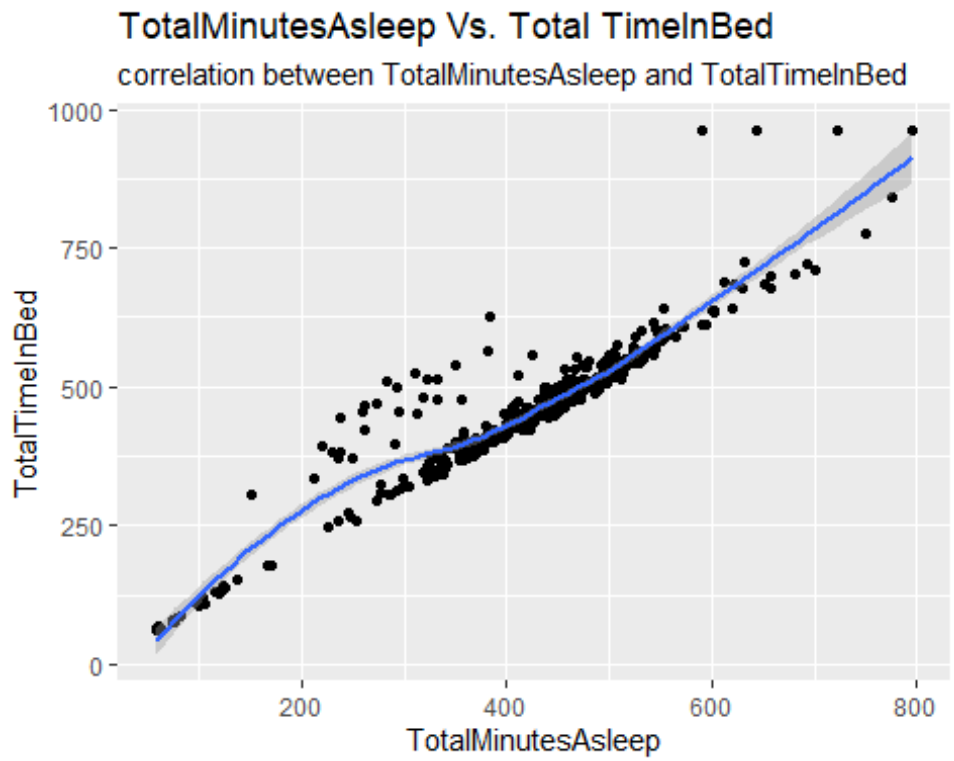
Summary of user's average activity by day



Checking correlation between users and Total minutes as sleep

```
ggplot(data=sleepDay, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) +
geom_point()+geom_smooth()+
  labs(title = "TotalMinutesAsleep Vs. Total TimeInBed", subtitle =
"correlation between TotalMinutesAsleep and TotalTimeInBed")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
cor(sleepDay$TotalMinutesAsleep, sleepDay$TotalTimeInBed)
```

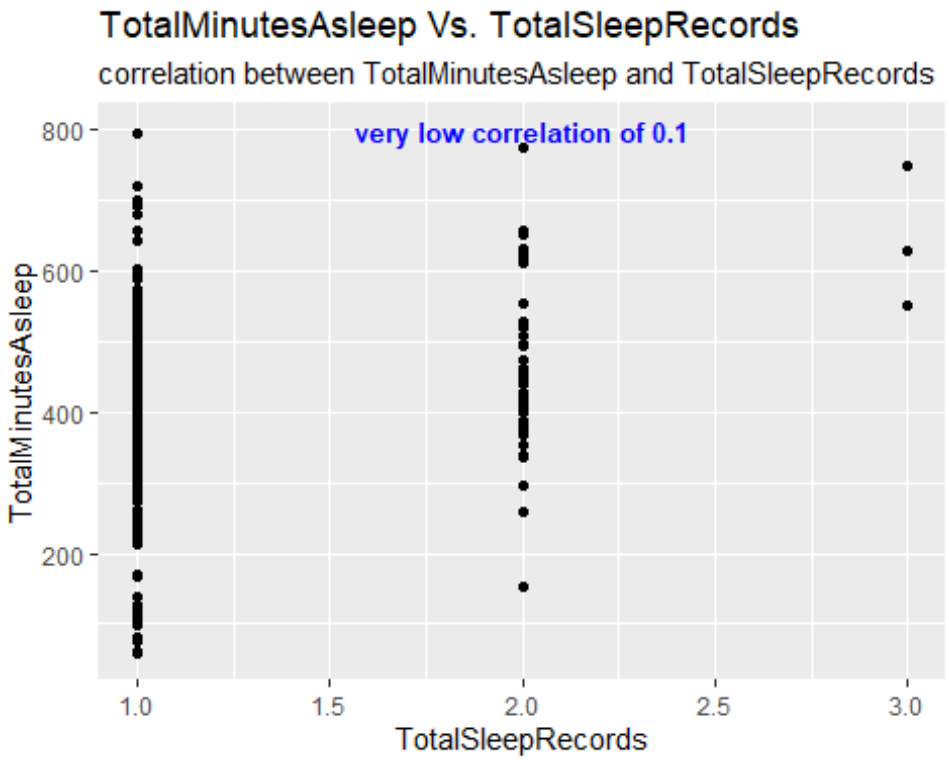
```
## [1] 0.9304575
```

The above shows a strong relationship and high correlation between Total time in bed and total minutes asleep

Relation ship between total minutes asleep and total sleep record

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at 0.99

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 0.0001

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.99

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.01

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 4.0401

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning: Computation failed in `stat_smooth()`:
## NA/NaN/Inf in foreign function call (arg 5)
```



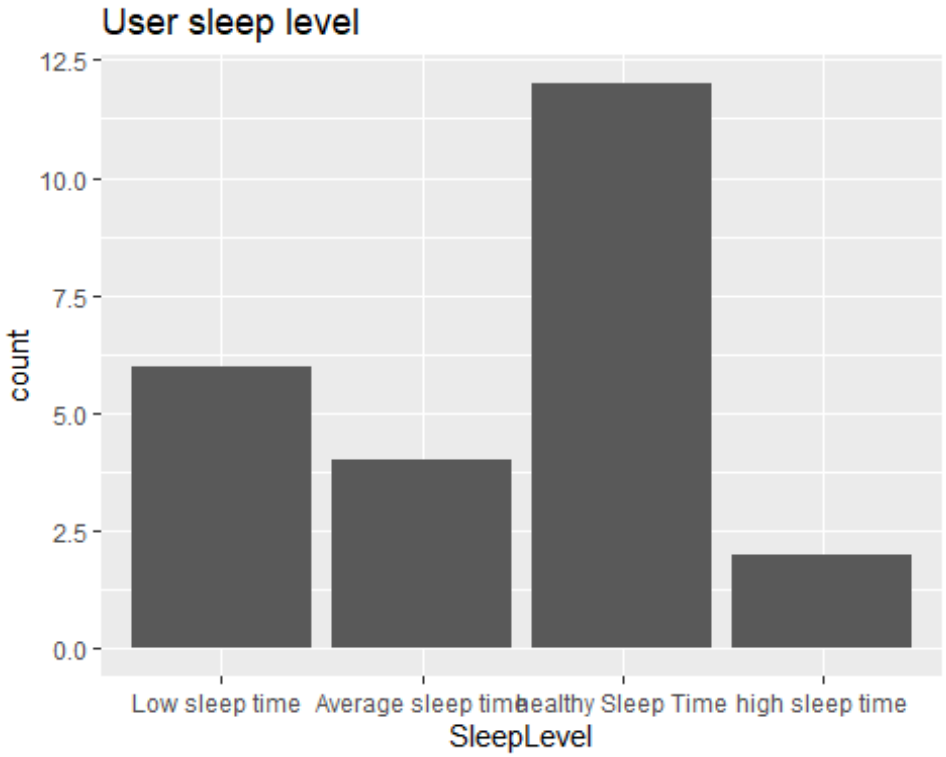

```
## [1] 0.9304575
```

The plot shows that that users who slept more than once per day got more total sleep time

###Users' sleep level

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

```
## # A tibble: 24 × 3
## # Groups:   Id [24]
##            Id Average_sleep SleepLevel
##         <dbl>         <dbl> <fct>
##  1 1503960366          360. "Average sleep time"
##  2 1644430081          294  "Low sleep time "
##  3 1844505072          652  "high sleep time"
##  4 1927972279          417  "healthy Sleep Time"
##  5 2026352035          506. "high sleep time"
##  6 2320127002           61  "Low sleep time "
##  7 2347167796          447. "healthy Sleep Time"
##  8 3977333714          294. "Low sleep time "
##  9 4020332650          349. "Average sleep time"
## 10 4319703577          477. "healthy Sleep Time"
## # … with 14 more rows
```




```
##           SleepLevel Percentage
## 1    Low sleep time       18.2%
## 2 Average sleep time      12.1%
## 3 healthy Sleep Time      36.4%
## 4    high sleep time       6.1%
```

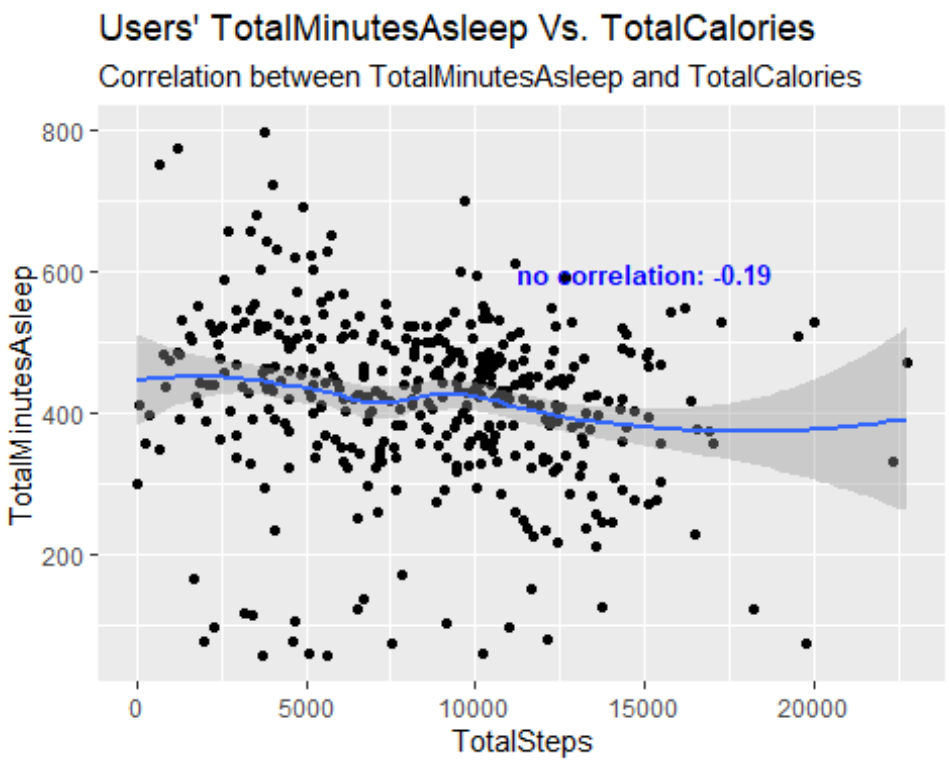
Relationship between daily activity and minutes asleep

```
Sleepday_sep<-sleepDay %>%
  separate(col=SleepDay,into = c('the_date', 'the_time'),sep=" ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 413 rows [1, 2,
3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].

Sleepday_sep2 <- Sleepday_sep %>%
                            group_by(Id,
the_date=as.Date(the_date,"%m/%d/%Y"))%>%
                            summarise(TotalMinutesAsleep,TotalTimeInBed,
TotalSleepRecords)

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

DailyActivity_df_sep2 <-DailyActivity_df_sep%>%
  summarise(the_date=as.Date(the_date,"%m/%d/%Y"),Id, the_date,TotalSteps,
Calories, TotalActivity = VeryActiveDistance + LightlyActiveMinutes+
SedentaryMinutes + FairlyActiveMinutes + SedentaryActiveDistance)

SleepVsActivity<-merge(Sleepday_sep2,
DailyActivity_df_sep2,by=c("Id","the_date"))

write.csv(SleepVsActivity, "C:/Users/Koyenikan Arinola/Desktop/New
folder/SleepVsActivity2.csv", row.names = TRUE)
```

The relationship between Steps/Calories and Total minutes asleep

```
ggplot(data=SleepVsActivity, aes(x=TotalSteps, y=TotalMinutesAsleep)) +
geom_point()+geom_smooth()+
  labs(title = "Users' TotalMinutesAsleep Vs. TotalCalories", subtitle =
"Correlation between TotalMinutesAsleep and TotalCalories") +
  annotate("text", x=15000,y=600,label="no correlation: -0.19",color="Blue",
           fontface="bold", size=3.5)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

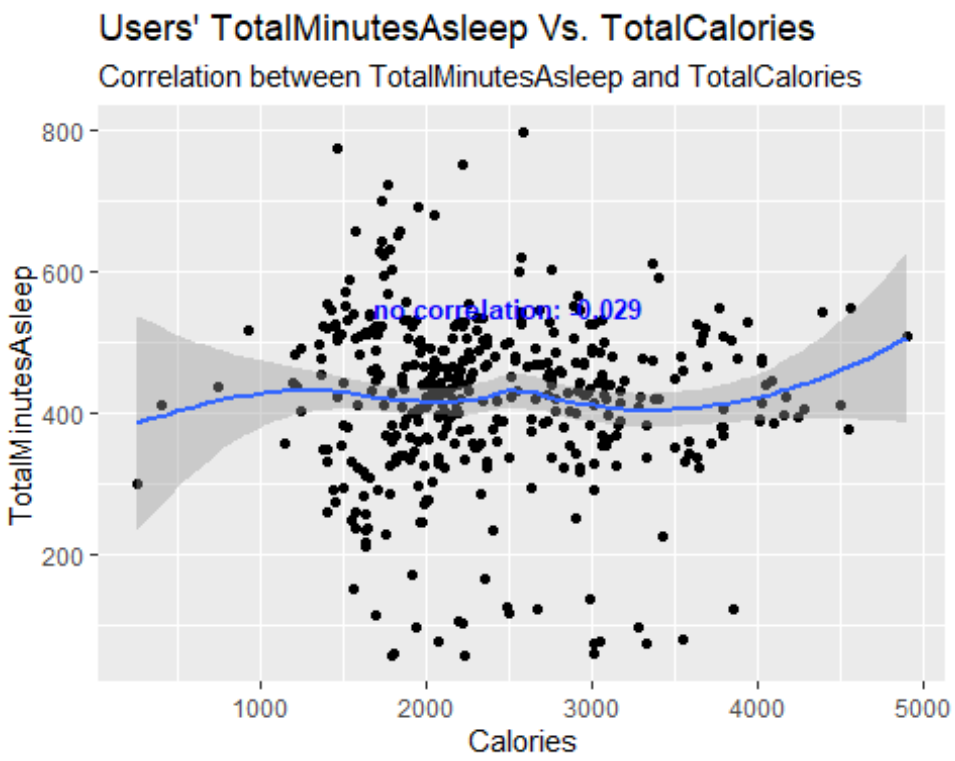



```
cor(SleepVsActivity$TotalSteps, SleepVsActivity$TotalMinutesAsleep)
```

```
## [1] -0.1868665
```

```
ggplot(data=SleepVsActivity, aes(x=Calories, y=TotalMinutesAsleep)) +
geom_point()+geom_smooth()+
  labs(title = "Users' TotalMinutesAsleep Vs. TotalCalories", subtitle =
"Correlation between TotalMinutesAsleep and TotalCalories") +
  annotate("text", x=2500,y=550,label="no correlation: -0.029",color="Blue",
           fontface="bold", size=3.5)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
cor(SleepVsActivity$Calories, SleepVsActivity$TotalMinutesAsleep)
```

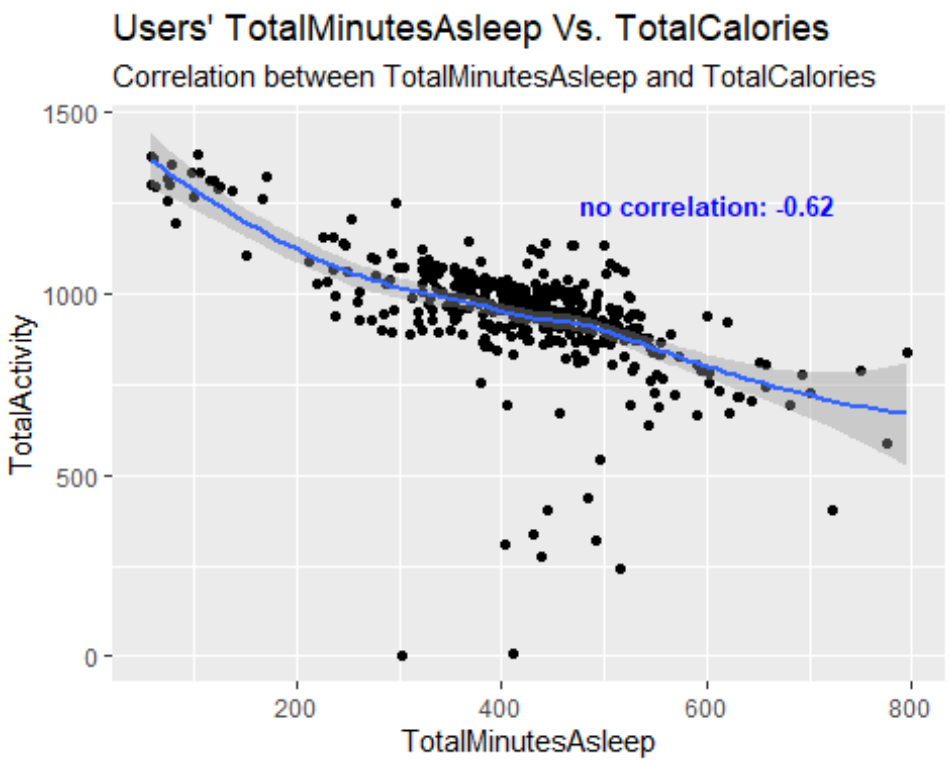
```
## [1] -0.02852571
```

The plot above shows that there's no relationship between steps/calories burnt and total minutes asleep

The relationship between total sleep minutes and total activity

```
ggplot(data=SleepVsActivity, aes(x=TotalMinutesAsleep, y=TotalActivity)) +
geom_point()+geom_smooth()+
  labs(title = "Users' TotalMinutesAsleep Vs. TotalCalories", subtitle =
"Correlation between TotalMinutesAsleep and TotalCalories") +
  annotate("text", x=600,y=1250,label="no correlation: -0.62",color="Blue",
           fontface="bold", size=3.5)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

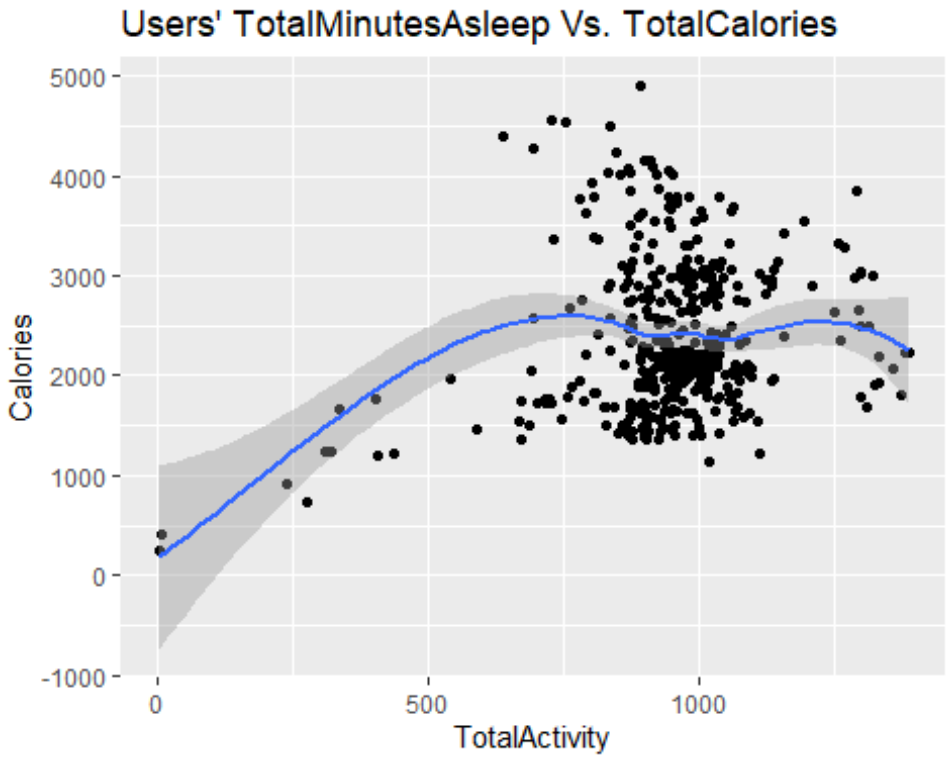



```
cor(SleepVsActivity$TotalActivity, SleepVsActivity$TotalMinutesAsleep)

## [1] -0.6154308
```

The plot above shows a negative relationship between total sleep minutes and total activities which implies that activities keep users awake

The relationship between Calories and Total activity

```
ggplot(data=SleepVsActivity, aes(x= TotalActivity, y = Calories)) +
geom_point()+geom_smooth()+
  labs(title = "Users' TotalMinutesAsleep Vs. TotalCalories")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The plot above shows a positive relationship between the total active minutes and the total calories burnt. This implies that active minute during the day contributes to the amount of calories burnt.

Recommendations for Bellabeats from the Analysis.

Observation:

63.7% of users burns less 2700 calories and 42.2% takes less than 10000 steps daily

There's a high positive relationship and high correlation between; ○ Calories burnt and steps taken; the higher steps taken the more calories burnt ○ Calories burnt and total intensity; ○ Steps taken and total intensity There's a positive relationship ○ Calories and total activity minutes

Recommendation:

Bellabeats can leverage on this finding to educate its users on impacts on intensities, steps taken, activities affects calories burnt job suits users' various needs such as gaining, burning or maintaining their calories in take for healthy living.

Observation:

○ There's higher intensities, steps taken, calories burnt at 5:00 pm, 6:00 pm, 7:00 pm and lower records between 1:00 am and 3:00 am ○ There's higher intensities, steps taken, calories burnt on Saturdays and Tuesdays ○ There's a slight decline in trends of calories and steps taken on Saturdays and inconsistent trends on other days of the week.

Recommendation:

It is observed that there's high record of intensities, steps and calories during after working hours. Bellabeats can leverage on this information to create various workout or eatot challenges and also create scheduling features so users can include these challenges to their daily or weekly routines during their free time for consistency and healthy habits.

Observation:

18.2% of users sleeps below 300 minutes per day and 6.1% of users sleeps above 500 minutes.

There's a high positive relationship and high correlation between; ○ Total minutes asleep and total minutes in bed; users who stay in bed more sleep more.

There's a negative relationship ○ Total minutes asleep and total activities minutes; this means that users who are more active sleep less.

Users who sleep more than once per day have higher minutes asleep asleep.

Calories burnt have no relationship with the total minutes asleep

Recommendation:

Bellabeats can make use of this information to educate users on the health benefits of just enough sleeps and how the amount time in bed affects the total time asleep. Creating a scheduling feature and alarm feature will remind users to take a nap wake up as the case may be.