

# Bellabeat Case Study

Grace Heaberlin

2023-11-25

## Introduction

Bellabeat's cofounder and Chief Creative Officer Urška Sršen 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. After researching the various products, I will be focusing on the Bellabeat app.

## Company information

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. 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. 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.

## Stakeholders 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: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

### Products:

- 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.
- 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.
- 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.
- 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.
- 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.

## Business task

- What are some trends in smart device usage?
- How could these trends apply to Bellabeat customers?
- How could these trends help influence Bellabeat marketing strategy?

## Data information

FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius): This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users 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. All data was cleaned, analyzed, and visualized in R Studio.

## High-Level Recommendations

1. Reminders for extended sedentary time and bedtime, along with “wind down period” an hour before.
2. Encourage users with in app suggestions to create a personalized schedule depending on their most commonly active times.
3. Notifications to encourage users to work out in any intensity level, such as light housework or a short walk.
4. Explore the idea of connecting screentime and producing a report at the end of the week for users to aid in sleep quality.

## Summary of Analysis

### Installing packages

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'  
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v dplyr      1.1.4      v readr      2.1.4  
## v forcats    1.0.0      v stringr    1.5.1  
## v ggplot2    3.4.4      v tibble     3.2.1  
## v lubridate  1.9.3      v tidyr      1.3.0  
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
install.packages("tidyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'  
## (as 'lib' is unspecified)
```

```
library(tidyr)
```

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'  
## (as 'lib' is unspecified)
```

```

library(dplyr)
install.packages("ggplot2")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
library(ggplot2)
install.packages("rmarkdown")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
library(rmarkdown)
install.packages("here")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
library(here)

## here() starts at /cloud/project
install.packages("janitor")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
library(janitor)

##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##   chisq.test, fisher.test
install.packages("skimr")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
library(skimr)
install.packages("lubridate")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
library(lubridate)
install.packages("wesanderson")

## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
## (as 'lib' is unspecified)
library(wesanderson)

```

## Importing

```

daily_activity <- read.csv("dailyActivity_merged.csv")
daily_steps <- read.csv("dailySteps_merged.csv")

```

```

daily_intensities <- read.csv("dailyIntensities_merged.csv")
daily_calories <- read.csv("dailyCalories_merged.csv")
sleep_day <- read.csv("sleepDay_merged.csv")
weight_log <- read.csv("weightLogInfo_merged.csv")
hourly_steps <- read.csv("hourlySteps_merged.csv")
hourly_intensities <- read.csv("hourlyIntensities_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")

```

## Summary stats

```
head(daily_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(daily_steps)
```

```

##           Id ActivityDay StepTotal
## 1 1503960366 4/12/2016      13162
## 2 1503960366 4/13/2016      10735
## 3 1503960366 4/14/2016      10460
## 4 1503960366 4/15/2016       9762
## 5 1503960366 4/16/2016      12669
## 6 1503960366 4/17/2016       9705

```

```
head(daily_intensities)
```

```

##           Id ActivityDay SedentaryMinutes LightlyActiveMinutes
## 1 1503960366 4/12/2016              728              328

```

```
## 2 1503960366 4/13/2016 776 217
## 3 1503960366 4/14/2016 1218 181
## 4 1503960366 4/15/2016 726 209
## 5 1503960366 4/16/2016 773 221
## 6 1503960366 4/17/2016 539 164
## FairlyActiveMinutes VeryActiveMinutes SedentaryActiveDistance
## 1 13 25 0
## 2 19 21 0
## 3 11 30 0
## 4 34 29 0
## 5 10 36 0
## 6 20 38 0
## LightActiveDistance ModeratelyActiveDistance VeryActiveDistance
## 1 6.06 0.55 1.88
## 2 4.71 0.69 1.57
## 3 3.91 0.40 2.44
## 4 2.83 1.26 2.14
## 5 5.04 0.41 2.71
## 6 2.51 0.78 3.19
```

```
head(daily_calories)
```

```
##      Id ActivityDay Calories
## 1 1503960366 4/12/2016 1985
## 2 1503960366 4/13/2016 1797
## 3 1503960366 4/14/2016 1776
## 4 1503960366 4/15/2016 1745
## 5 1503960366 4/16/2016 1863
## 6 1503960366 4/17/2016 1728
```

```
head(sleep_day)
```

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

```
head(weight_log)
```

```
##      Id      Date WeightKg WeightPounds Fat BMI
## 1 1503960366 5/2/2016 11:59:59 PM 52.6 115.9631 22 22.65
## 2 1503960366 5/3/2016 11:59:59 PM 52.6 115.9631 NA 22.65
## 3 1927972279 4/13/2016 1:08:52 AM 133.5 294.3171 NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM 56.7 125.0021 NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM 57.3 126.3249 NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM 72.4 159.6147 25 27.45
```

```
##      IsManualReport      LogId
## 1              True 1.462234e+12
## 2              True 1.462320e+12
## 3             False 1.460510e+12
## 4              True 1.461283e+12
## 5              True 1.463098e+12
## 6              True 1.460938e+12
```

```
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(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_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
```

```
colnames(daily_activity)
```

```
## [1] "Id" "ActivityDate"
## [3] "TotalSteps" "TotalDistance"
## [5] "TrackerDistance" "LoggedActivitiesDistance"
## [7] "VeryActiveDistance" "ModeratelyActiveDistance"
## [9] "LightActiveDistance" "SedentaryActiveDistance"
## [11] "VeryActiveMinutes" "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes" "SedentaryMinutes"
## [15] "Calories"
```

```
colnames(daily_steps)
```

```
## [1] "Id" "ActivityDay" "StepTotal"
```

```
colnames(daily_intensities)
```

```
## [1] "Id" "ActivityDay"
## [3] "SedentaryMinutes" "LightlyActiveMinutes"
## [5] "FairlyActiveMinutes" "VeryActiveMinutes"
```

```

## [7] "SedentaryActiveDistance" "LightActiveDistance"
## [9] "ModeratelyActiveDistance" "VeryActiveDistance"

colnames(daily_calories)

## [1] "Id" "ActivityDay" "Calories"

colnames(sleep_day)

## [1] "Id" "SleepDay" "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"

colnames(weight_log)

## [1] "Id" "Date" "WeightKg" "WeightPounds"
## [5] "Fat" "BMI" "IsManualReport" "LogId"

colnames(hourly_steps)

## [1] "Id" "ActivityHour" "StepTotal"

colnames(hourly_intensities)

## [1] "Id" "ActivityHour" "TotalIntensity" "AverageIntensity"

colnames(hourly_calories)

## [1] "Id" "ActivityHour" "Calories"

n_distinct(daily_activity$Id)

## [1] 33

n_distinct(daily_steps$Id)

## [1] 33

n_distinct(daily_intensities$Id)

## [1] 33

n_distinct(daily_calories$Id)

## [1] 33

n_distinct(sleep_day$Id)

## [1] 24

n_distinct(weight_log$Id)

## [1] 8

n_distinct(hourly_steps$Id)

## [1] 33

n_distinct(hourly_intensities$Id)

## [1] 33

n_distinct(hourly_calories$Id)

## [1] 33

```

```
nrow(daily_activity)
```

```
## [1] 940
```

```
nrow(daily_steps)
```

```
## [1] 940
```

```
nrow(daily_intensities)
```

```
## [1] 940
```

```
nrow(daily_calories)
```

```
## [1] 940
```

```
nrow(sleep_day)
```

```
## [1] 413
```

```
nrow(weight_log)
```

```
## [1] 67
```

```
nrow(hourly_steps)
```

```
## [1] 22099
```

```
nrow(hourly_intensities)
```

```
## [1] 22099
```

```
nrow(hourly_calories)
```

```
## [1] 22099
```

## Identifying which data needs cleaned

```
print("Duplicates")
```

```
## [1] "Duplicates"
```

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

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

```
sum(duplicated(daily_steps))
```

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

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

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

```
sum(duplicated(daily_calories))
```

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

```
sum(duplicated(sleep_day))
```

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

```
sum(duplicated(weight_log))
```



```
## [1] 0
sum(duplicated(hourly_steps))

## [1] 0
sum(duplicated(hourly_intensities))

## [1] 0
sum(duplicated(hourly_calories))

## [1] 0
print("Nulls")

## [1] "Nulls"
sum(is.na(daily_activity))

## [1] 0
sum(is.na(daily_steps))

## [1] 0
sum(is.na(daily_intensities))

## [1] 0
sum(is.na(daily_calories))

## [1] 0
sum(is.na(sleep_day))

## [1] 0
sum(is.na(weight_log))

## [1] 65
sum(is.na(hourly_steps))

## [1] 0
sum(is.na(hourly_intensities))

## [1] 0
sum(is.na(hourly_calories))

## [1] 0
```

## Removing duplicate data

```
sleep <- sleep_day %>%
  distinct()
sum(duplicated(sleep))

## [1] 0
```

## Viewing which column majority of nulls are in

```
summary(weight_log)
```

```
##           Id           Date           WeightKg           WeightPounds
## Min.      :1.504e+09   Length:67       Min.       : 52.60   Min.       :116.0
## 1st Qu.:6.962e+09   Class :character   1st Qu.: 61.40   1st Qu.:135.4
## Median :6.962e+09   Mode  :character   Median : 62.50   Median :137.8
## Mean      :7.009e+09                Mean      : 72.04   Mean      :158.8
## 3rd Qu.:8.878e+09                3rd Qu.: 85.05   3rd Qu.:187.5
## Max.      :8.878e+09                Max.      :133.50   Max.      :294.3
##
##           Fat           BMI           IsManualReport           LogId
## Min.      :22.00   Min.      :21.45   Length:67       Min.      :1.460e+12
## 1st Qu.:22.75   1st Qu.:23.96   Class :character   1st Qu.:1.461e+12
## Median :23.50   Median :24.39   Mode  :character   Median :1.462e+12
## Mean      :23.50   Mean      :25.19                Mean      :1.462e+12
## 3rd Qu.:24.25   3rd Qu.:25.56                3rd Qu.:1.462e+12
## Max.      :25.00   Max.      :47.54                Max.      :1.463e+12
## NA's      :65
```

```
## All of the nulls are in the Fat column.
```

## Standardizing date columns

```
cleaned_daily_activity <- daily_activity %>%
  rename(Date = ActivityDate) %>%
  mutate(Date = as.Date(Date, format = '%m/%d/%Y'),
         Day = weekdays(Date),
         Month = months(Date))

cleaned_daily_steps <- daily_steps %>%
  rename(Date = ActivityDay) %>%
  mutate(Date = as.POSIXct(Date, format = '%m/%d/%Y %I:%M:%S %p'),
         Time = format(Date, format = '%H:%M:%S'),
         Day = weekdays(Date),
         Month = months(Date))

cleaned_daily_intensities <- daily_intensities %>%
  rename(Date = ActivityDay) %>%
  mutate(Date = as.POSIXct(Date, format = '%m/%d/%Y %I:%M:%S %p'),
         Time = format(Date, format = '%H:%M:%S'),
         Day = weekdays(Date),
         Month = months(Date))

cleaned_daily_calories <- daily_calories %>%
  rename(Date = ActivityDay) %>%
  mutate(Date = as.POSIXct(Date, format = '%m/%d/%Y %I:%M:%S %p'),
         Time = format(Date, format = '%H:%M:%S'),
         Day = weekdays(Date),
         Month = months(Date))

cleaned_sleep_day <- sleep_day %>%
  rename(Date = SleepDay) %>%
```

```

mutate(Date = as.POSIXct(Date, format = '%m/%d/%Y %I:%M:%S %p'),
       Time = format(Date, format = '%H:%M:%S'),
       Day = weekdays(Date),
       Month = months(Date))

cleaned_weight_log <- weight_log %>%
  mutate(Date = as.POSIXct(Date, format = '%m/%d/%Y %I:%M:%S %p'),
         Time = format(Date, format = '%H:%M:%S'),
         Day = weekdays(Date),
         Month = months(Date))

cleaned_hourly_intensities <- hourly_intensities %>%
  mutate(Date = as.POSIXct(ActivityHour, format = '%m/%d/%Y %I:%M:%S %p'),
         Time = format(Date, format = '%H:%M:%S'),
         Day = weekdays(Date),
         Month = months(Date))

cleaned_hourly_steps <- hourly_steps %>%
  rename(Date = ActivityHour) %>%
  mutate(Date = as.POSIXct(Date, format = '%m/%d/%Y %I:%M:%S %p'),
         Time = format(Date, format = '%H:%M:%S'),
         Day = weekdays(Date),
         Month = months(Date))

cleaned_hourly_calories <- hourly_calories %>%
  rename(Date = ActivityHour) %>%
  mutate(Date = as.POSIXct(Date, format = '%m/%d/%Y %I:%M:%S %p'),
         Time = format(Date, format = '%H:%M:%S'),
         Day = weekdays(Date),
         Month = months(Date))

```

## Detailed summary stats

```

daily_activity %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes) %>%
  summary()

```

```
##      TotalSteps      TotalDistance      SedentaryMinutes
##  Min.       :    0      Min.       : 0.000      Min.       :  0.0
##  1st Qu.: 3790      1st Qu.: 2.620      1st Qu.: 729.8
##  Median : 7406      Median : 5.245      Median :1057.5
##  Mean   : 7638      Mean   : 5.490      Mean    : 991.2
##  3rd Qu.:10727      3rd Qu.: 7.713      3rd Qu.:1229.5
##  Max.    :36019      Max.    :28.030      Max.     :1440.0

```

*## Most users spend their time sedentary, with the average sedentary minutes logged being 991.2 (or 16.*

```

daily_steps %>%
  select(ActivityDay,
         StepTotal) %>%
  summary()

```

```
## ActivityDay      StepTotal
## Length:940      Min.   :    0
## Class :character 1st Qu.: 3790
## Mode  :character Median : 7406
##                  Mean   : 7638
##                  3rd Qu.:10727
##                  Max.   :36019
```

```
daily_intensities%>%
  select(ActivityDay,
          SedentaryMinutes) %>%
  summary()
```

```
## ActivityDay      SedentaryMinutes
## Length:940      Min.   :    0.0
## Class :character 1st Qu.: 729.8
## Mode  :character Median :1057.5
##                  Mean   : 991.2
##                  3rd Qu.:1229.5
##                  Max.   :1440.0
```

```
daily_calories %>%
  select(ActivityDay,
          Calories) %>%
  summary()
```

```
## ActivityDay      Calories
## Length:940      Min.   :    0
## Class :character 1st Qu.:1828
## Mode  :character Median :2134
##                  Mean   :2304
##                  3rd Qu.:2793
##                  Max.   :4900
```

```
sleep_day %>%
  select(SleepDay,
          TotalTimeInBed,
          TotalMinutesAsleep) %>%
  summary()
```

```
## SleepDay      TotalTimeInBed TotalMinutesAsleep
## Length:413    Min.   : 61.0   Min.   : 58.0
## Class :character 1st Qu.:403.0   1st Qu.:361.0
## Mode  :character Median :463.0   Median :433.0
##                  Mean   :458.6   Mean   :419.5
##                  3rd Qu.:526.0   3rd Qu.:490.0
##                  Max.   :961.0   Max.   :796.0
```

*## There is an almost perfectly linear trend in Total Time in Bed vs. Total Minutes Asleep, with some e*

```
weight_log %>%
  select(BMI,
          WeightPounds) %>%
  summary()
```

```
## BMI      WeightPounds
## Min.   :21.45   Min.   :116.0
```

```
## 1st Qu.:23.96 1st Qu.:135.4
## Median :24.39 Median :137.8
## Mean :25.19 Mean :158.8
## 3rd Qu.:25.56 3rd Qu.:187.5
## Max. :47.54 Max. :294.3
```

*## Most participants are considered overweight with an average BMI of 25.19 and a "healthy" BMI range b*

```
hourly_steps %>%
  select(ActivityHour,
         StepTotal) %>%
  summary()
```

```
## ActivityHour      StepTotal
## Length:22099      Min.   :    0.0
## Class :character  1st Qu.:    0.0
## Mode  :character  Median :   40.0
##                      Mean    :  320.2
##                      3rd Qu.:  357.0
##                      Max.    :10554.0
```

*## There is a large deviation between the median and max, possibly due to a large outlier.*

```
hourly_intensities %>%
  select(ActivityHour,
         TotalIntensity,
         AverageIntensity) %>%
  summary()
```

```
## ActivityHour      TotalIntensity  AverageIntensity
## Length:22099      Min.   :    0.00  Min.   :0.0000
## Class :character  1st Qu.:    0.00  1st Qu.:0.0000
## Mode  :character  Median :    3.00  Median :0.0500
##                      Mean    :   12.04  Mean    :0.2006
##                      3rd Qu.:   16.00  3rd Qu.:0.2667
##                      Max.    :   180.00  Max.    :3.0000
```

```
hourly_calories %>%
  select(ActivityHour,
         Calories) %>%
  summary()
```

```
## ActivityHour      Calories
## Length:22099      Min.   :  42.00
## Class :character  1st Qu.:  63.00
## Mode  :character  Median :  83.00
##                      Mean    :  97.39
##                      3rd Qu.: 108.00
##                      Max.    : 948.00
```

## Calculations

```
sleep_day$time_to_fall_asleep <- (sleep_day$TotalTimeInBed-sleep_day$TotalMinutesAsleep)
summary(sleep_day$time_to_fall_asleep)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
##      0.00   17.00   25.00   39.17   40.00  371.00
```

```
##The average person takes 39 minutes to fall asleep. The average person should only take 15 to 20 minu
```

## Joins

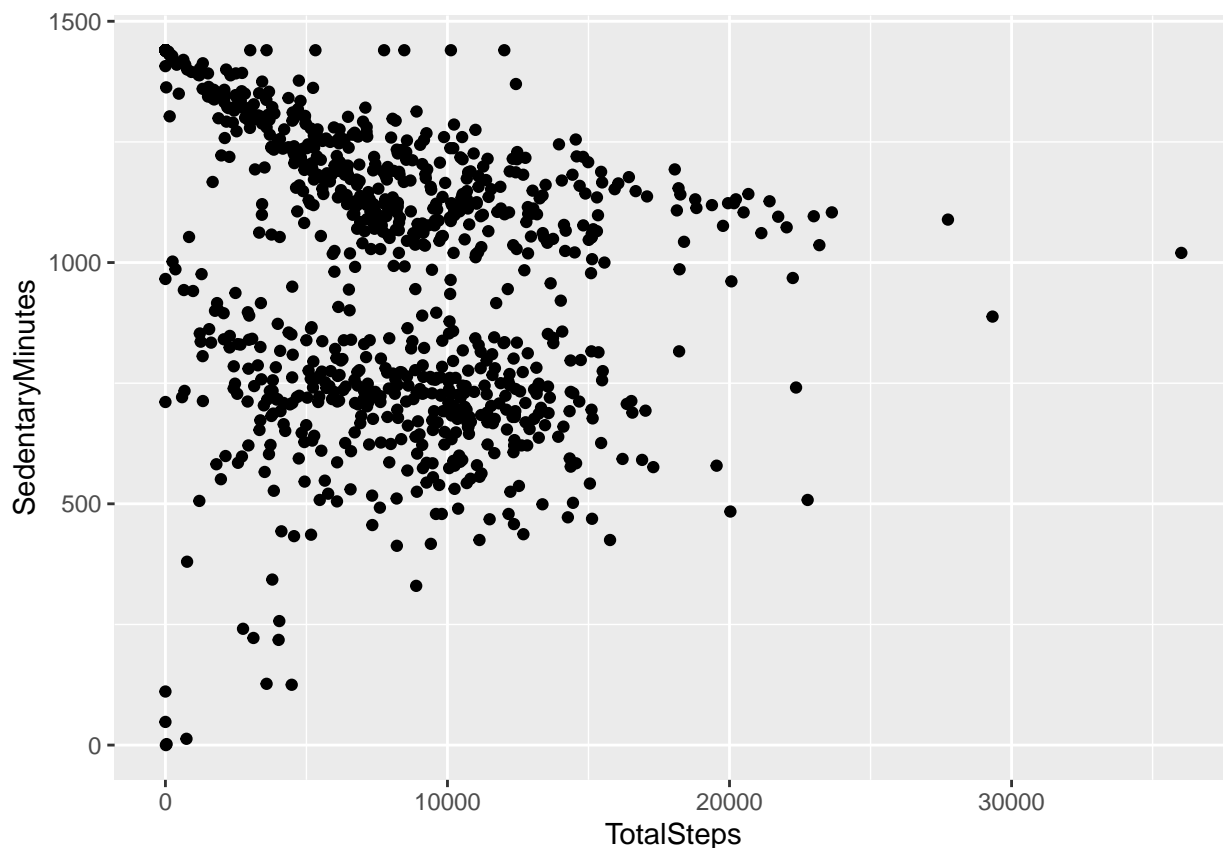
```
joined_sleep_day <- sleep_day %>%
  select(Id, SleepDay, TotalMinutesAsleep, TotalTimeInBed)
joined_weight_log <- weight_log %>%
  select(Id, Date, WeightPounds, BMI)

daily_activity_sleep <- merge(cleaned_daily_activity, joined_sleep_day, by = 'Id')
daily_activity_weight <- merge(cleaned_daily_activity, joined_weight_log, by = 'Id')

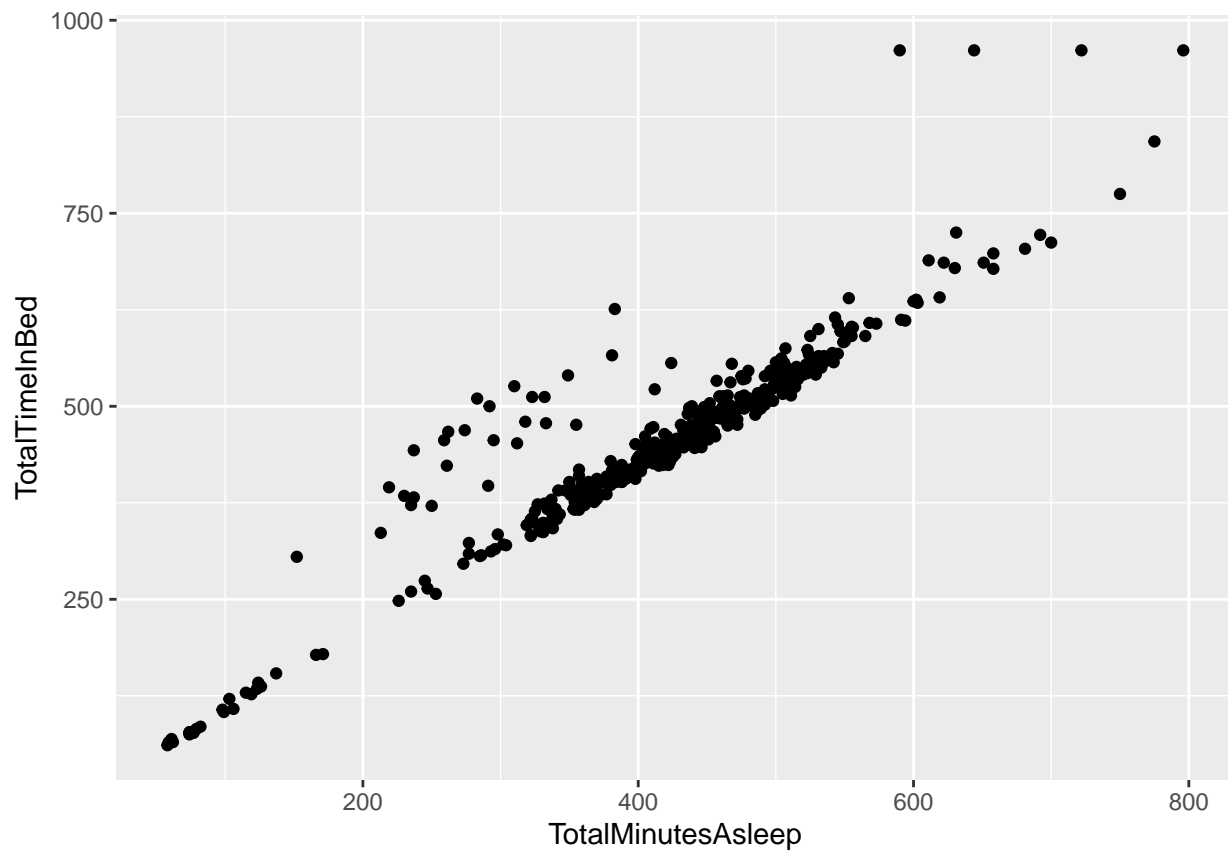
daily_activity_intensities <- merge(cleaned_daily_activity, cleaned_hourly_intensities, by = c('Id', 'Date'))
hourly_intensities_steps <- merge(cleaned_hourly_intensities, cleaned_hourly_steps, by = c('Id', 'Date'))
```

## Plotting

```
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes)) + geom_point()
```



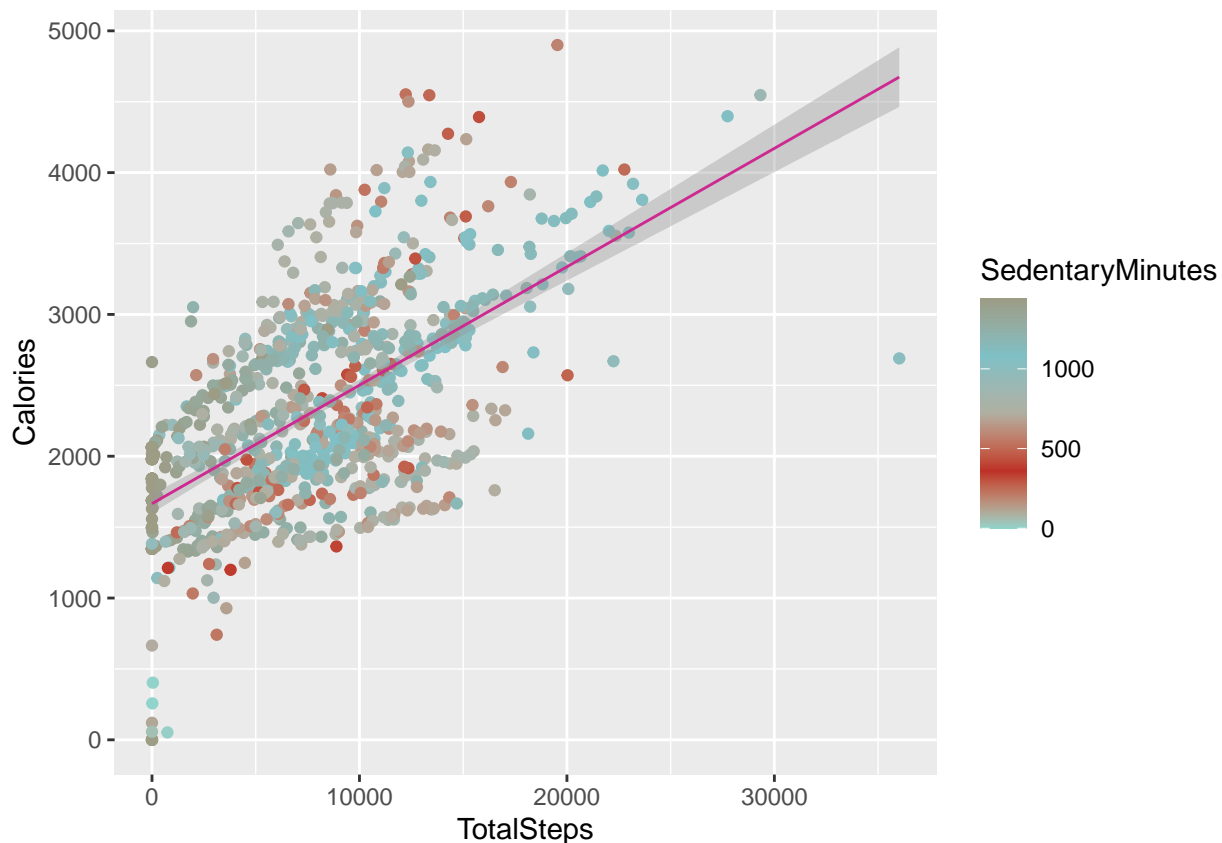
```
ggplot(data=sleep_day, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) + geom_point()
```



## Steps vs. Calories

```
max_date = max(cleaned_daily_activity$Date)
min_date = min(cleaned_daily_activity$Date)
sample_size = n_distinct(cleaned_daily_activity$Id)
ggplot(data=cleaned_daily_activity,
       aes(x = TotalSteps, y = Calories, color = SedentaryMinutes)) + geom_point() + scale_color_gradientn

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



*##The more steps taken, more calories burned.*

## Time Sedentary vs. Weight

```
max_date = max(daily_activity_weight$Date)

## Warning in max(daily_activity_weight$Date): no non-missing arguments to max;
## returning -Inf

min_date = min(daily_activity_weight$Date)

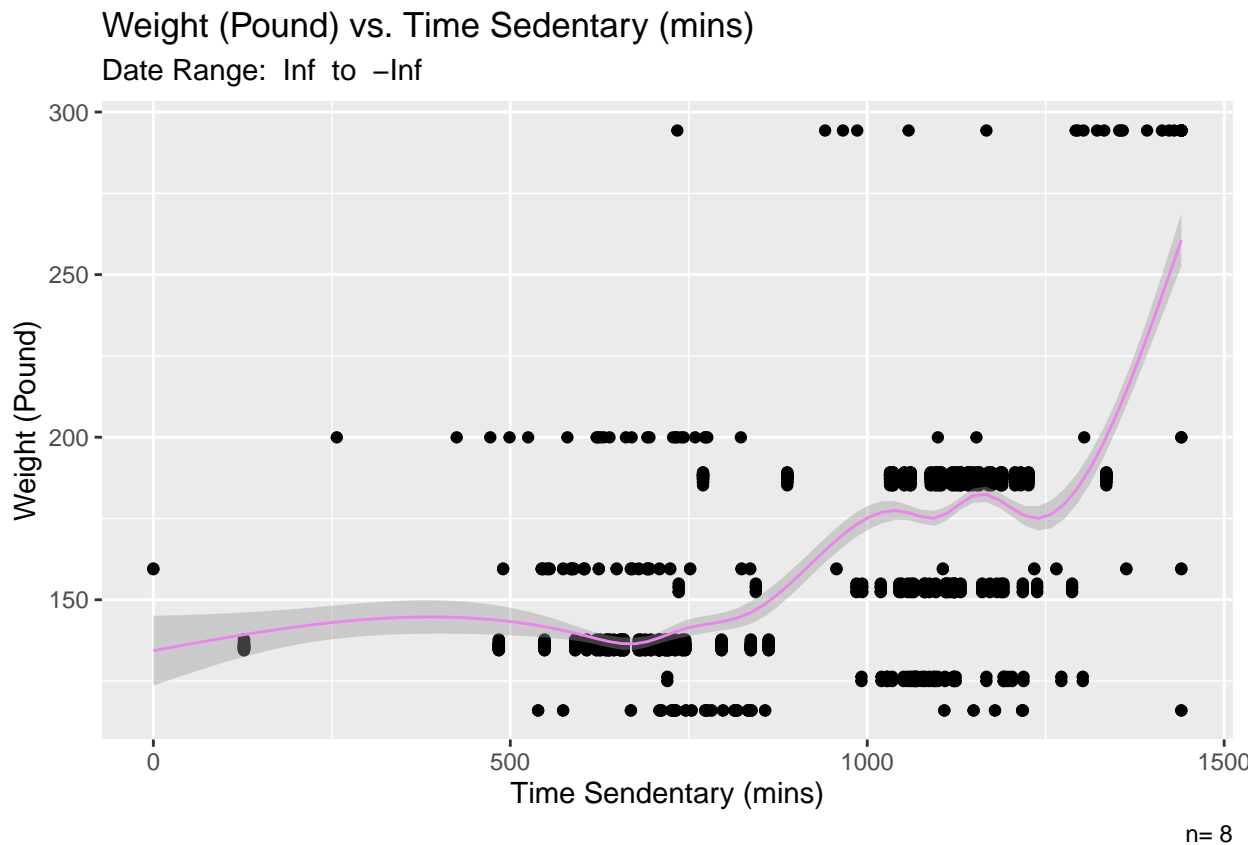
## Warning in min(daily_activity_weight$Date): no non-missing arguments to min;
## returning Inf

sample_size = n_distinct(daily_activity_weight$Id)

ggplot(data=daily_activity_weight) +
  geom_point(mapping=aes(x=SedentaryMinutes,
                        y=WeightPounds)) +
  geom_smooth(mapping=aes(x=SedentaryMinutes,
                        y=WeightPounds), color='Violet', linewidth=0.5) +
  labs(title = 'Weight (Pound) vs. Time Sedentary (mins)',
       subtitle = paste("Date Range: ", min_date, " to ", max_date),
       caption = paste('n=', sample_size),
       x='Time Sedentary (mins)',
       y='Weight (Pound)')

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



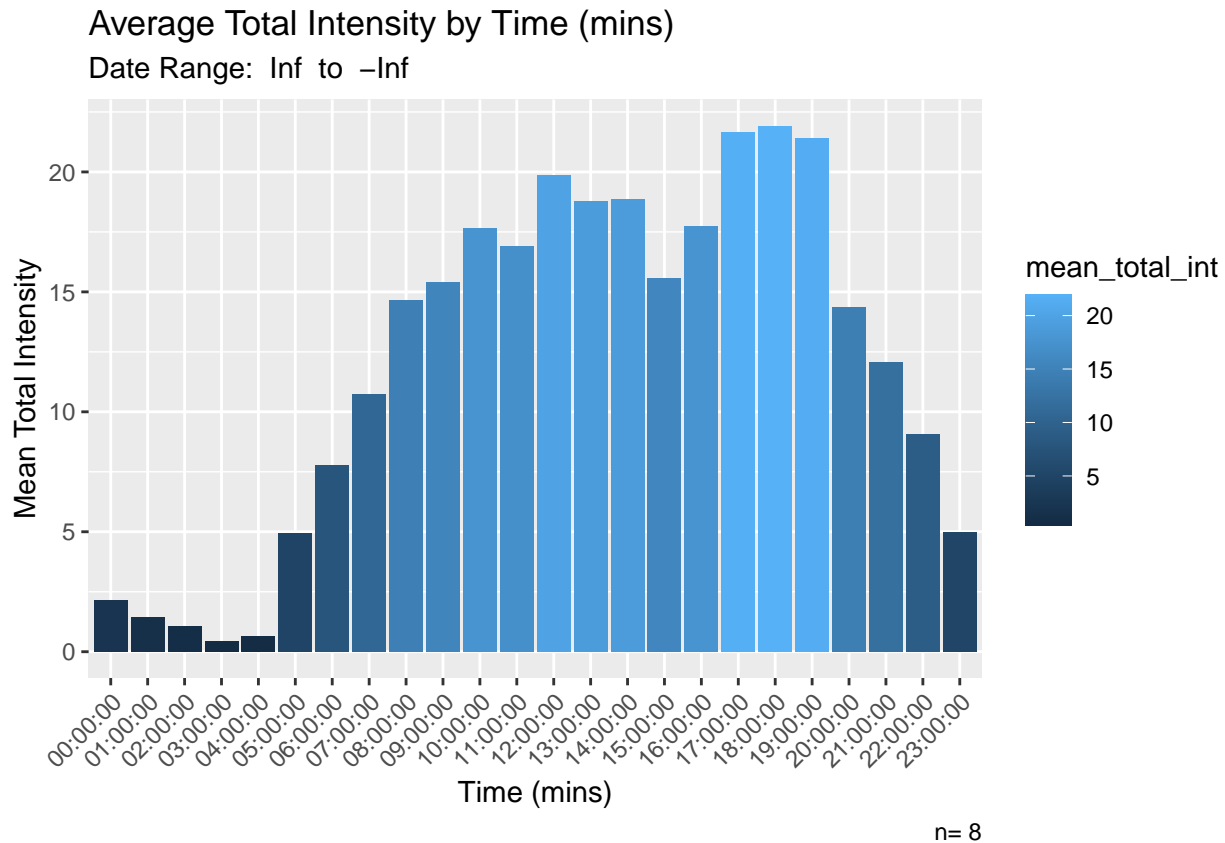


*##This indicates a higher weight given more time spent sedentary. There are 2 sharp inclines: between 5*

*##Avg Intensity by Time*

```
grouped_time_intensities <- hourly_intensities_steps %>%
  drop_na() %>%
  group_by(Time) %>%
  summarize(mean_total_int = mean(TotalIntensity),
            mean_steps = mean(StepTotal),
            sample_size = n_distinct(Id),
            min_date = min(Date),
            max_date = max(Date))

ggplot(data = grouped_time_intensities) + geom_bar(aes(x = Time, y = mean_total_int, fill = mean_total_int)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  labs(title = "Average Total Intensity by Time (mins)",
       x = "Time (mins)", y = "Mean Total Intensity",
       subtitle = paste('Date Range: ', min_date, ' to ', max_date),
       caption = paste('n=', sample_size))
```



##Majority of time spent working out is between 5 AM - 11 PM, with activity peaking between 5 PM - 7 PM

##Sedentary Mins vs. Time Asleep

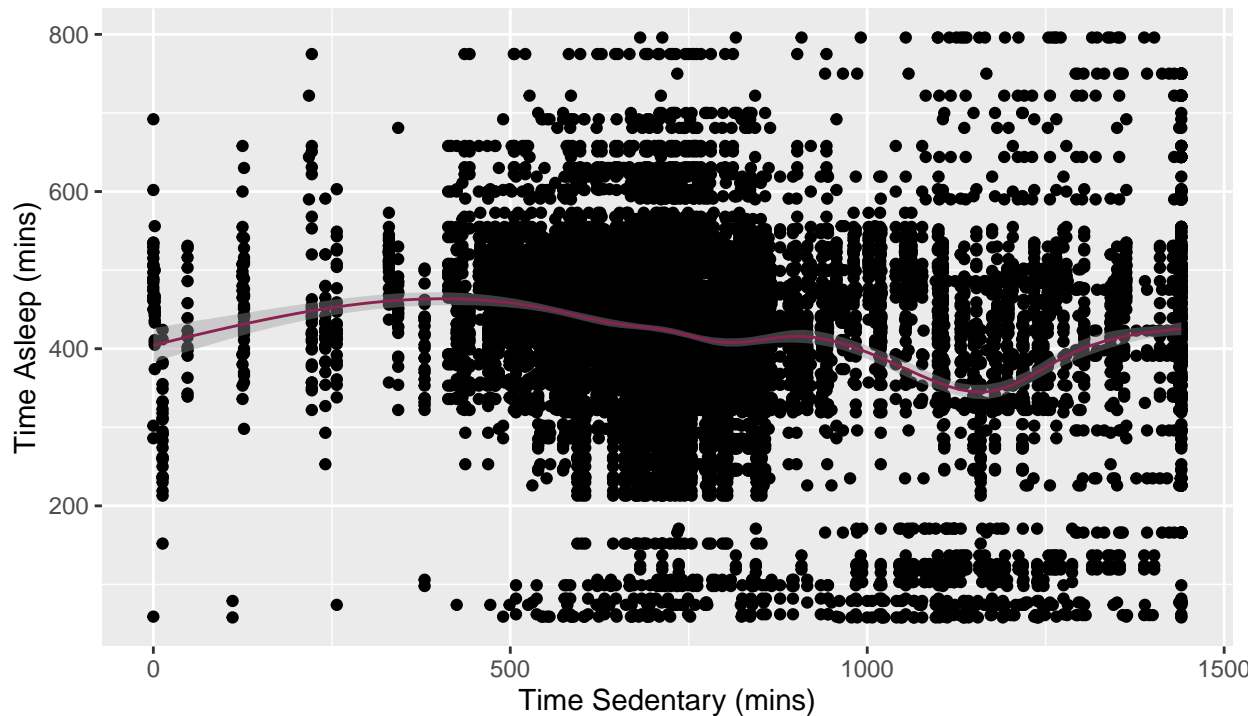
```
max_date = max(daily_activity_sleep$Date)
min_date = min(daily_activity_sleep$Date)
sample_size = n_distinct(daily_activity_sleep$Id)

ggplot(data=daily_activity_sleep) +
  geom_point(mapping=aes(x=SedentaryMinutes,
                        y=TotalMinutesAsleep)) +
  labs(title=paste('Time Sedentary vs. Time Asleep (mins)'),
       subtitle=paste("Date Range: ", min_date, " to ", max_date),
       caption=paste('n=', sample_size),
       x='Time Sedentary (mins)',
       y='Time Asleep (mins)') +
  geom_smooth(mapping=aes(x=SedentaryMinutes,
                        y=TotalMinutesAsleep),
             colour = 'violetred4',
             linewidth = 0.5)
```

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

## Time Sedentary vs. Time Asleep (mins)

Date Range: 2016-04-12 to 2016-05-12



n= 24

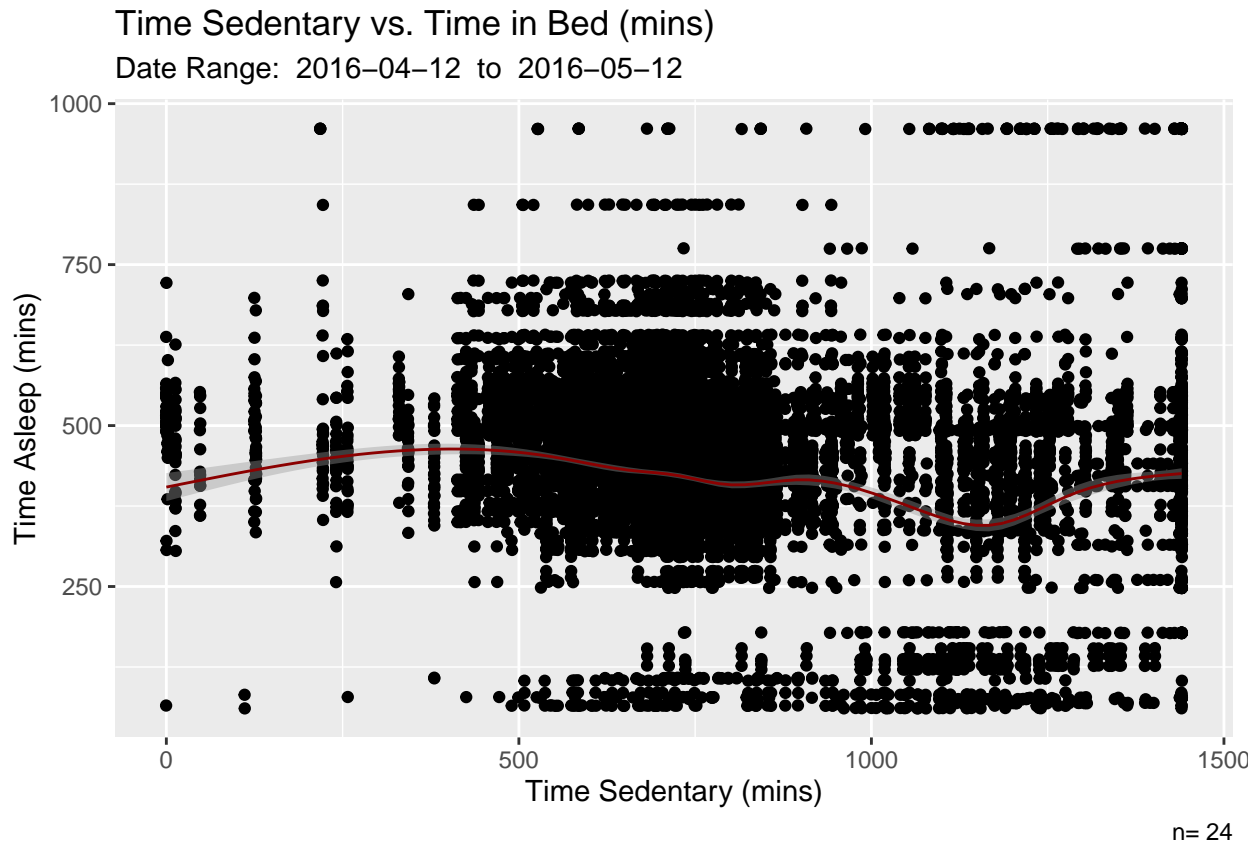
*##Looks like time spent sedentary reduces sleep time. Could remind users to move if watch detects they're*

##Sedentary vs. Time in Bed

```
max_date = max(daily_activity_sleep$Date)
min_date = min(daily_activity_sleep$Date)
sample_size = n_distinct(daily_activity_sleep$Id)
```

```
ggplot(data=daily_activity_sleep) +
  geom_jitter(mapping=aes(x=SedentaryMinutes,
                          y=TotalTimeInBed)) +
  labs(title=paste('Time Sedentary vs. Time in Bed (mins)'),
        subtitle=paste("Date Range: ", min_date, " to ", max_date),
        caption=paste('n=', sample_size),
        x='Time Sedentary (mins)',
        y='Time Asleep (mins)') +
  geom_smooth(mapping=aes(x=SedentaryMinutes,
                          y=TotalMinutesAsleep),
              colour = 'Red4',
              linewidth = 0.5)
```

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



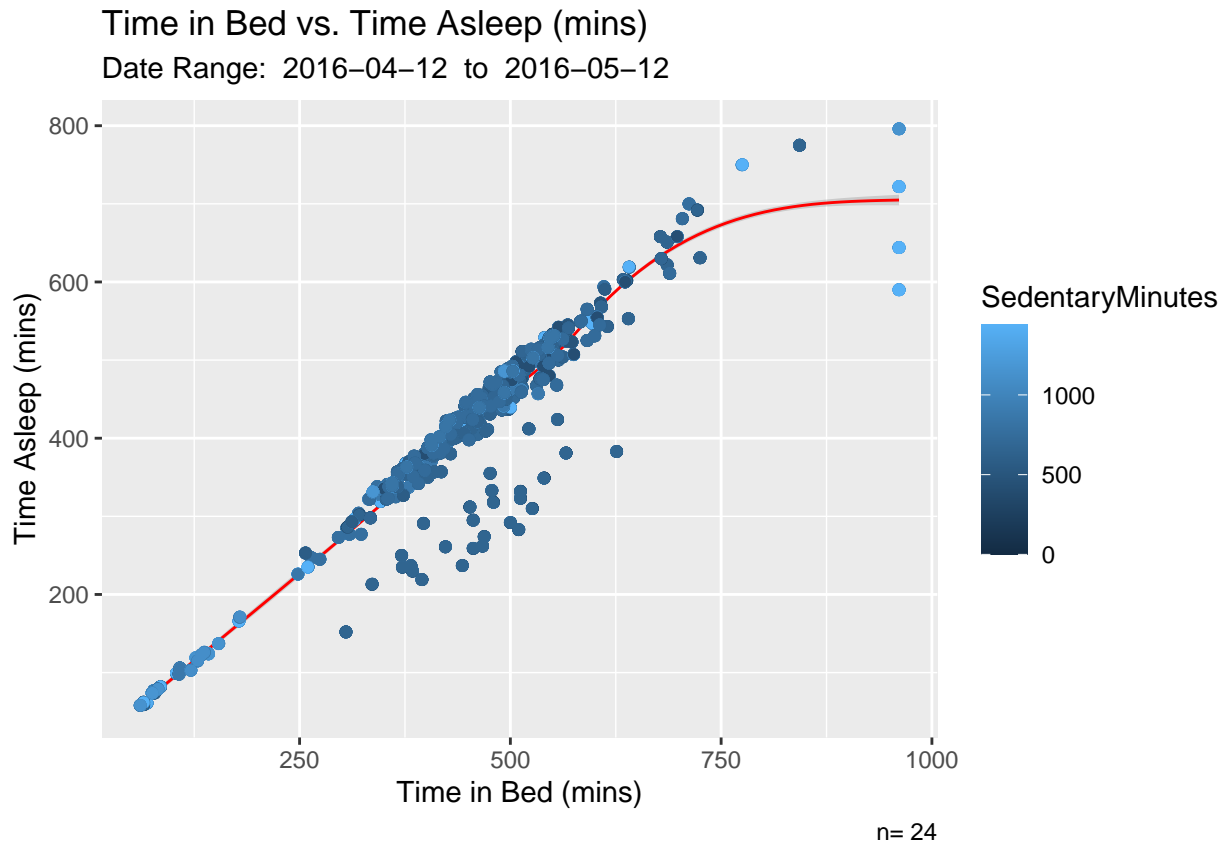
*##Highlights importance of having device send bedtime reminders, especially if users are sedentary. Use*

## Time in Bed vs. Time Asleep

```
max_date = max(daily_activity_sleep$Date)
min_date = min(daily_activity_sleep$Date)
sample_size = n_distinct(daily_activity_sleep$Id)

ggplot(data=daily_activity_sleep) +
  geom_smooth(mapping=aes(x=TotalTimeInBed,
                          y=TotalMinutesAsleep),
            color='Red',
            linewidth=0.5) +
  labs(title='Time in Bed vs. Time Asleep (mins)',
        subtitle=paste("Date Range: ", min_date, " to ", max_date),
        caption=paste('n=', sample_size),
        x='Time in Bed (mins)',
        y='Time Asleep (mins)') +
  geom_jitter(mapping=aes(x=TotalTimeInBed,
                          y=TotalMinutesAsleep, colour=SedentaryMinutes))

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



*##The average amount of sleep across all users and days is only 7hrs, which again highlights importance*

## Final Observations

Data shows high importance for reminders tailored to personalized schedules, namely for sedentary minutes and sleep. Users might also benefit from a “wind down” period an hour before bedtime to decrease time in bed and increase time asleep.