Case Study 2: How Can a Wellness Technology Company Play It Smart?

Thuan Ha

2024-01-23

Introduction:

This is a case study from the Google Data Analytics Course on Coursera, where I will be using what I have learn throughout the course to perform a real-world tasks of a junior data analyst.

Scenario:

You are a junior data analyst working on the marketing analyst team at Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, co-founder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. You have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights you discover will then help guide marketing strategy for the company.

Dataset:

The dataset provide is FitBit Fitness Tracker Data from Kaggle. 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.

Bussiness Tasks:

1. What are some trends in smart device usage?/
2. How could these trends apply to Bellabeat customers?/
3. How could these trends help influence Bellabeat marketing strategy?/

Packages use for this study:

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     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 error
```

```
library(janitor)
```

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

```
library(lubridate)
library(skimr)
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'
##
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
library(stringr)
```

Importing the data:

```
daily_activity <- read.csv("Bellabeat Data/dailyActivity_merged.csv")
daily_sleep <- read.csv("Bellabeat Data/sleepDay_merged.csv")
weight_log_info <- read.csv("Bellabeat Data/weightLogInfo_merged.csv")
hourly_intensities <- read.csv("Bellabeat Data/hourlyIntensities_merged.csv")
```

Checking the data:

Using the head function let me have a quick overview of the data to make sure each columns have the correct format. It look like the date columns for the four dataset is in character format instead of date format, and IsManualReport for the weight__log__info dataset could be change to a Boolean format.

```
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_sleep)
```

```
##           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_info)
```

```
##           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_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
```

Cleaning columns:

I'll be using the clean_names() function to have a consistent columns name for the dataset I'll be using and to remove space, letter cases, etc. to a consistent format. Also, I'll be changing the date format and adding new columns that could help with my analysis.

```
daily_activity <- clean_names(daily_activity)
daily_sleep <- clean_names(daily_sleep)
weight_log_info <- clean_names(weight_log_info)
hourly_intensities <- clean_names(hourly_intensities)
```

```
n_distinct(weight_log_info$id)
```

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

After looking over the data I decide to check the weight_log_info dataset since it contain many null value in one of the columns. It seem there are only 8 participant willing to give their weight info which is hard to have a analysis for the weight dataset.

```
#as.Date function let me convert chr column into date
daily_activity$activity_date <- as.Date(daily_activity$activity_date, "%m/%d/%y")
```

```
daily_sleep$sleep_day <- as.Date(daily_sleep$sleep_day, "%m/%d/%y")
```

```
hourly_intensities$activity_hour <- parse_date_time(hourly_intensities$activity_hour,
                                               "%m/%d/%y %H:%M:%S %p")
```

```
#for this dataset I use parse_date_time instead since it can handle the AM/PM in the date
weight_log_info$date <- parse_date_time(weight_log_info$date, "%m/%d/%y %H:%M:%S %p")
```

```
#the is_manual_report contain True or False so I change it into a Boolean
weight_log_info$is_manual_report <- as.logical(weight_log_info$is_manual_report)
```

Adding new columns:

Adding a total active hours and days of week columns to help further analyze trend such as which day are participants most active.

```
daily_activity <- daily_activity %>%
  mutate(total_active_hours = round((very_active_minutes + fairly_active_minutes + lightly_active_minute
         days_of_week = wday(activity_date, label = T))
```

```
hourly_intensities$Time <- format(as.POSIXct(hourly_intensities$activity_hour,format="%Y:%m:%d %H:%M:%S
```

```
hourly_intensities$Date <- format(as.POSIXct(hourly_intensities$activity_hour,format="%Y:%m:%d %H:%M:%S
```

```
daily_activity_cleaned <- daily_activity[!(daily_activity$total_active_hours <= 0.00),]
```

Dataset summary:

```
# daily activiy summary
daily_activity %>%
  select(total_steps,
         total_distance,
         total_active_hours) %>%
  summary()
```

```
##   total_steps    total_distance   total_active_hours
##  Min.   :    0   Min.   : 0.000   Min.   :0.000
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:2.000
##  Median : 7406   Median : 5.245   Median :4.000
##  Mean   : 7638   Mean   : 5.490   Mean   :3.776
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:5.000
##  Max.   :36019   Max.   :28.030   Max.   :9.000
```

```
# summary of active by minutes
daily_activity %>%
  select(very_active_minutes,
         fairly_active_minutes,
         lightly_active_minutes) %>%
  summary()
```

```
##  very_active_minutes fairly_active_minutes lightly_active_minutes
##  Min.   :  0.00      Min.   :  0.00        Min.   :  0.0
##  1st Qu.:  0.00      1st Qu.:  0.00        1st Qu.:127.0
##  Median :  4.00      Median :  6.00        Median :199.0
##  Mean   : 21.16      Mean   : 13.56        Mean   :192.8
##  3rd Qu.: 32.00      3rd Qu.: 19.00        3rd Qu.:264.0
##  Max.   :210.00      Max.   :143.00        Max.   :518.0
```

```
# calories and minutes spend sitting
daily_activity %>%
  select(sedentary_minutes,
         calories) %>%
  summary()
```

```
##  sedentary_minutes    calories
##  Min.   :   0.0    Min.   :   0
##  1st Qu.: 729.8    1st Qu.:1828
##  Median :1057.5    Median :2134
##  Mean   : 991.2    Mean   :2304
##  3rd Qu.:1229.5    3rd Qu.:2793
##  Max.   :1440.0    Max.   :4900
```

```
# time spend sleeping
daily_sleep %>%
  select(total_minutes_asleep,
         total_time_in_bed) %>%
  summary()
```

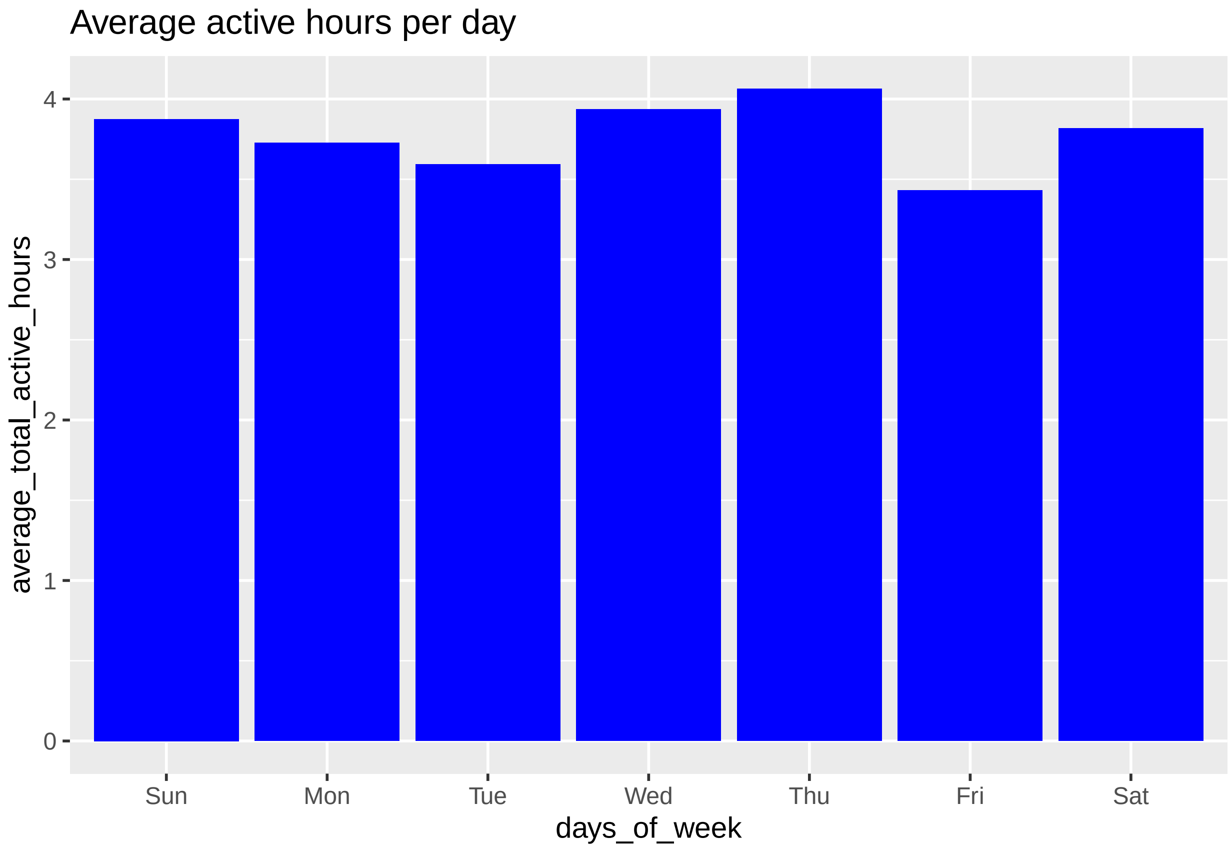
```
##  total_minutes_asleep total_time_in_bed
##  Min.   : 58.0        Min.   : 61.0
##  1st Qu.:361.0        1st Qu.:403.0
##  Median :433.0        Median :463.0
##  Mean   :419.5        Mean   :458.6
##  3rd Qu.:490.0        3rd Qu.:526.0
##  Max.   :796.0        Max.   :961.0
```

Info. from the summary:

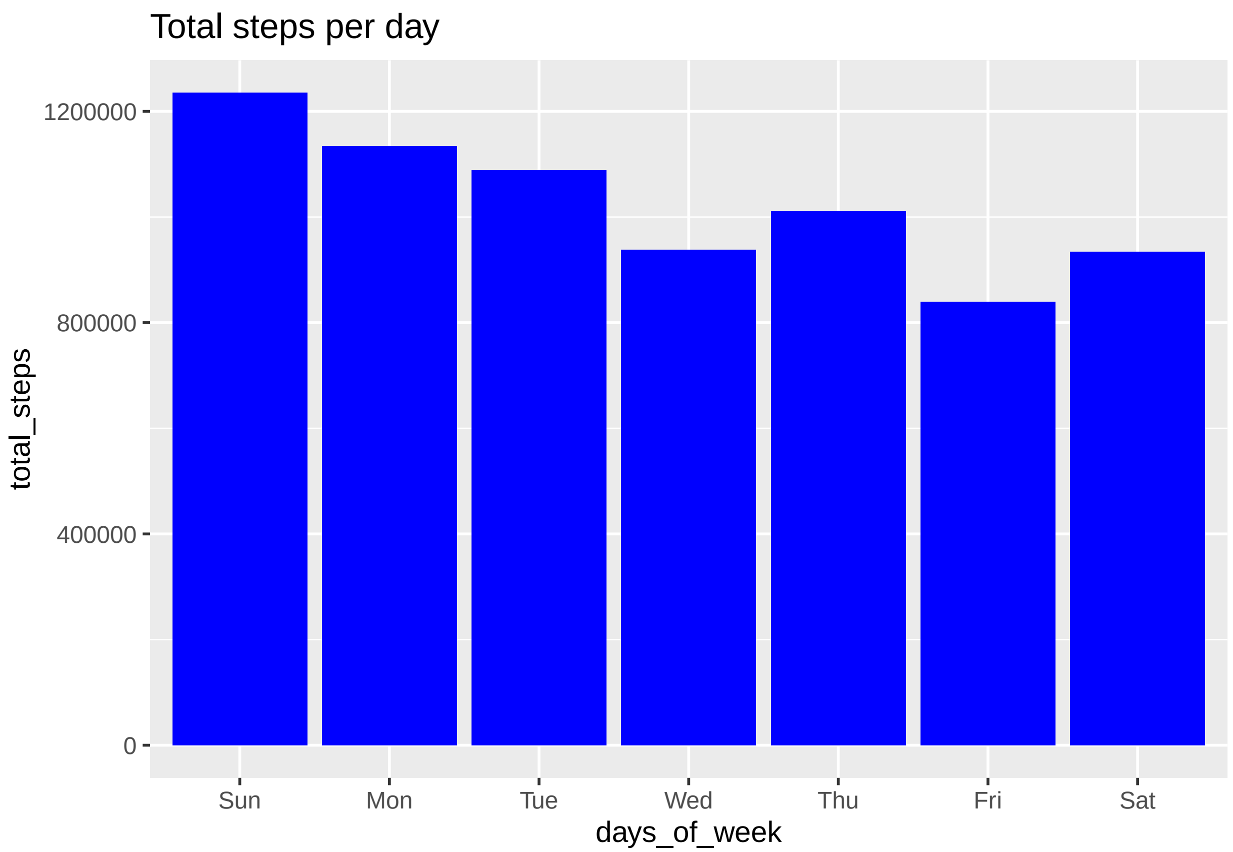
- The most stand out observation would be the average sedentary time or time spend sitting is 991 minutes or around 16-17 hours, which is consider a high risk according to this article.linked phrase/
- The average total steps is 7,638 which is under the recommended average of 10,000 steps by the CDC.linked phrase/
- While the average active hours is around the estimated of 4 hours, majority came from lightly active. It is recommended that adults aged between 18 - 64 years should do at least 150–300 minutes of moderate-intensity physical activity.linked phrase/

Visualization:

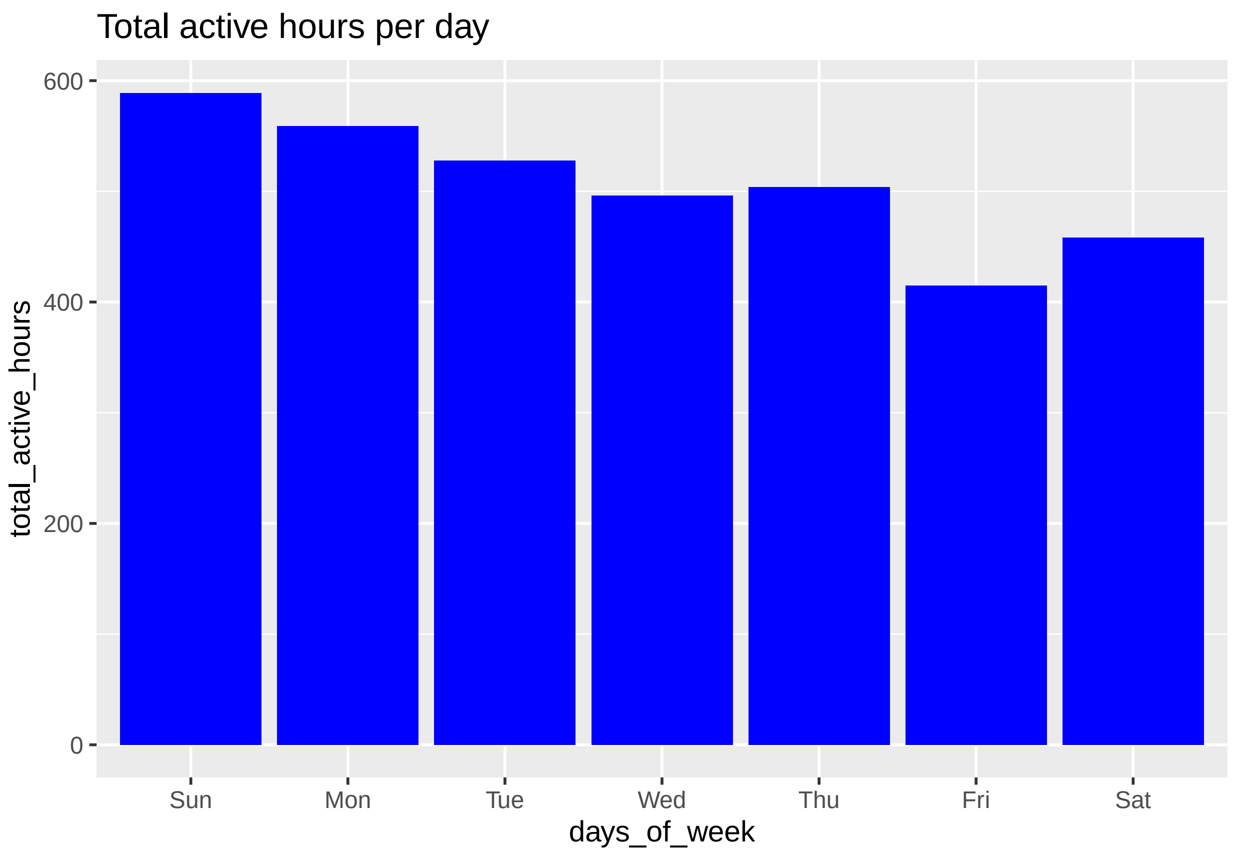
```
daily_activity %>%
  group_by(days_of_week) %>%
  summarise(average_total_active_hours = mean(total_active_hours)) %>%
  ggplot(aes(x = days_of_week, y = average_total_active_hours)) +
  geom_col(fill = "blue") +
  labs(title = "Average active hours per day")
```



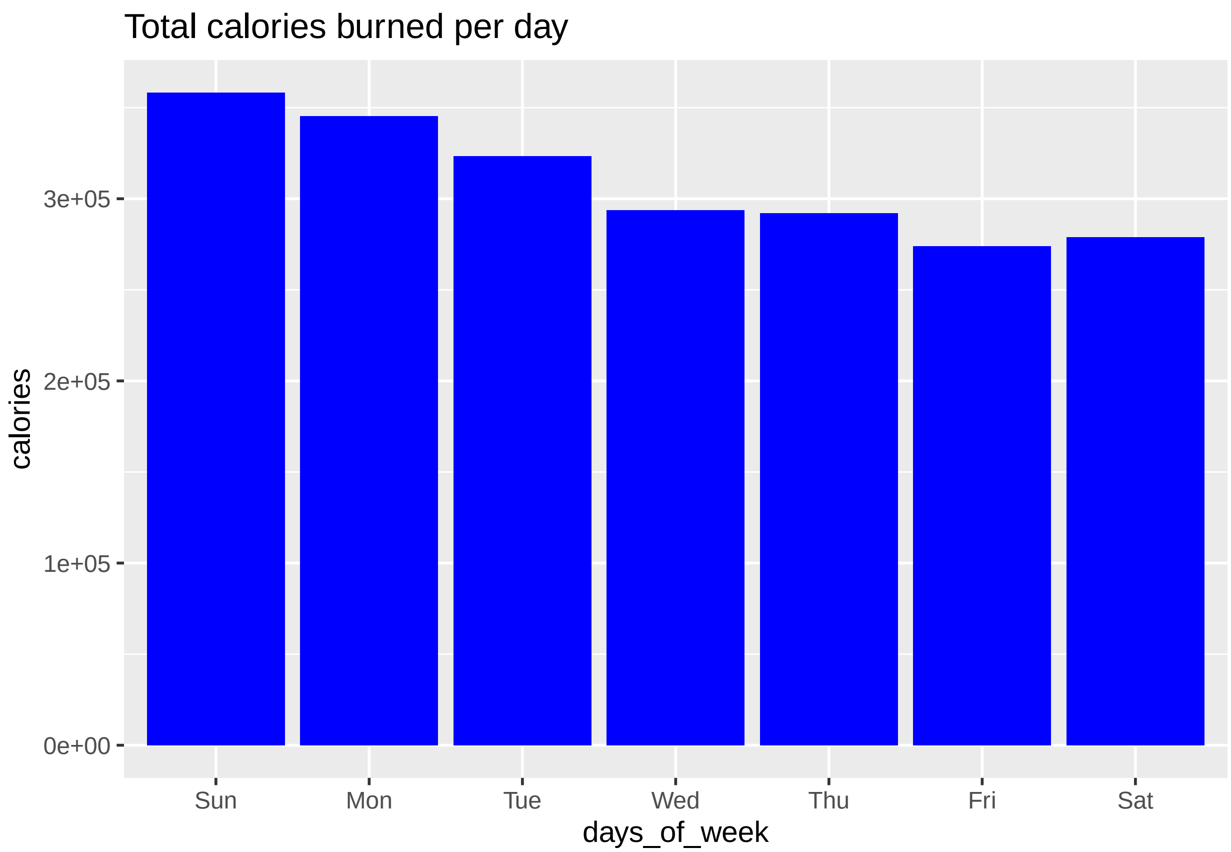
```
daily_activity %>%
  group_by(days_of_week) %>%
  ggplot(aes(x = days_of_week, y = total_steps)) +
  geom_col(fill = "blue") +
  labs(title = "Total steps per day")
```



```r
daily_activity %>%
  group_by(days_of_week) %>%
  ggplot(aes(x = days_of_week, y = total_active_hours)) +
  geom_col(fill = "blue") +
  labs(title = "Total active hours per day")
```



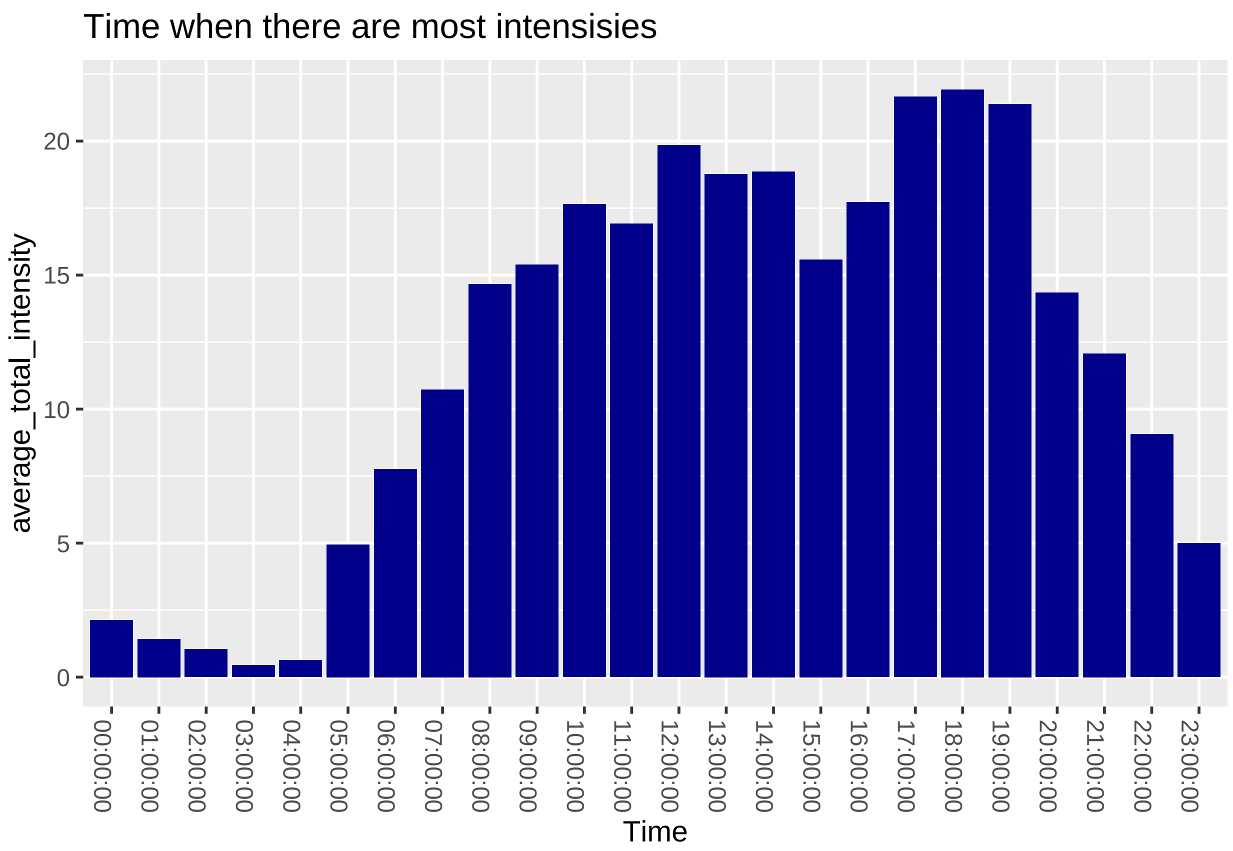
```r
daily_activity %>%
  group_by(days_of_week) %>%
  ggplot(aes(x = days_of_week, y = calories)) +
  geom_col(fill = "blue") +
  labs(title = "Total calories burned per day")
```



```
avg_int_by_hour <- hourly_intensities %>%
  group_by(Time) %>%
  drop_na() %>%
  summarise(average_total_intensity = mean(total_intensity))

ggplot(avg_int_by_hour, aes(x = Time, y = average_total_intensity)) +
  geom_histogram(stat = "identity", fill = "dark blue") +
  theme(axis.text.x = element_text(angle = 270)) +
  labs(title = "Time when there are most intensisies")
```

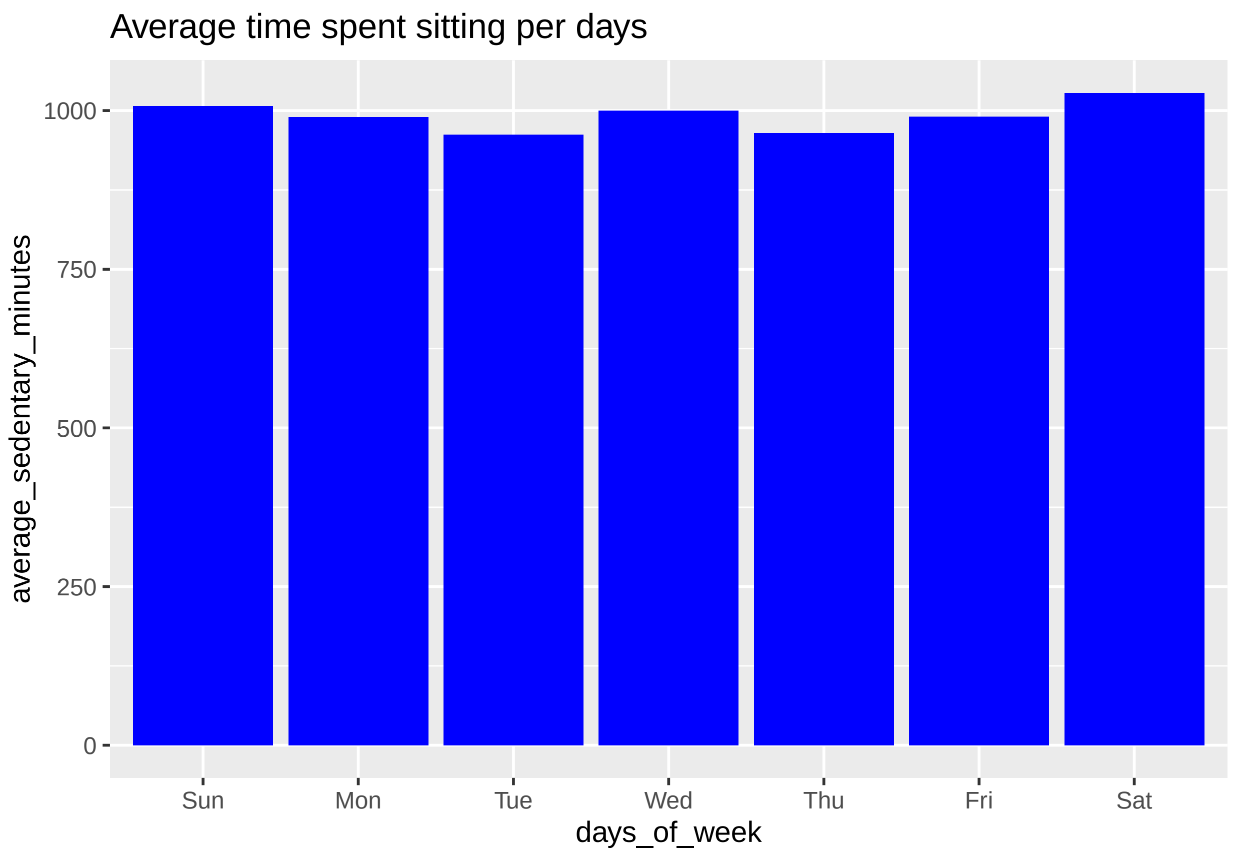
```
## Warning in geom_histogram(stat = "identity", fill = "dark blue"): Ignoring
## unknown parameters: ‘binwidth‘, ‘bins‘, and ‘pad‘
```



```
aggregate(daily_activity$total_active_hours, list(daily_activity$days_of_week), FUN = sum)
```

```
##   Group.1   x
## 1     Sun 589
## 2     Mon 559
## 3     Tue 528
## 4     Wed 496
## 5     Thu 504
## 6     Fri 415
## 7     Sat 458
```

```
daily_activity %>%
  group_by(days_of_week) %>%
  summarise(average_sedentary_minutes = mean(sedentary_minutes)) %>%
  ggplot(aes(x = days_of_week, y = average_sedentary_minutes)) +
  geom_col(fill = "blue") +
  labs(title = "Average time spent sitting per days")
```



### Correlations:

```
cor(daily_activity$total_steps, daily_activity$calories)
```

```
## [1] 0.5915681
```

```
cor(daily_activity$total_active_hours, daily_activity$calories)
```

```
## [1] 0.4669456
```

```
cor(daily_activity$very_active_minutes, daily_activity$calories)
```

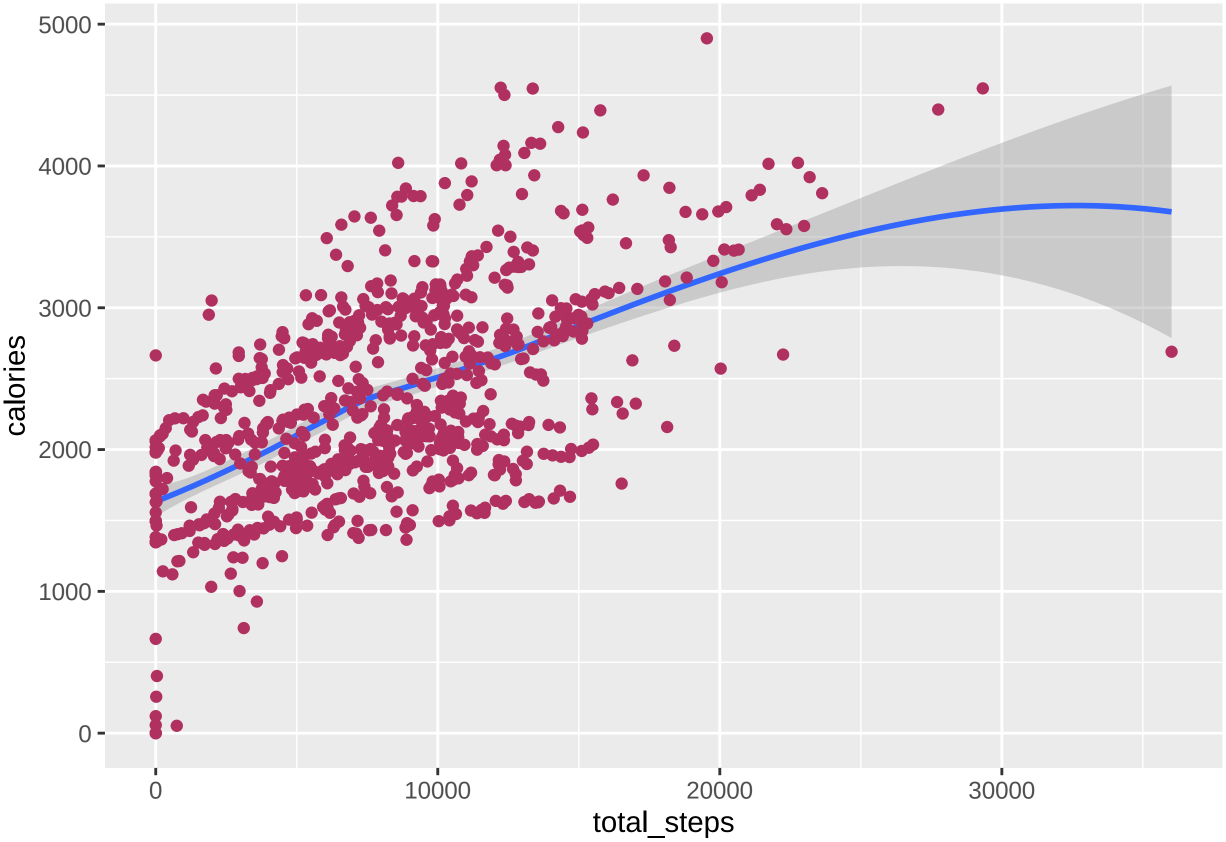
```
## [1] 0.6158383
```

```
cor(daily_activity$sedentary_minutes, daily_activity$calories)
```

```
## [1] -0.106973
```

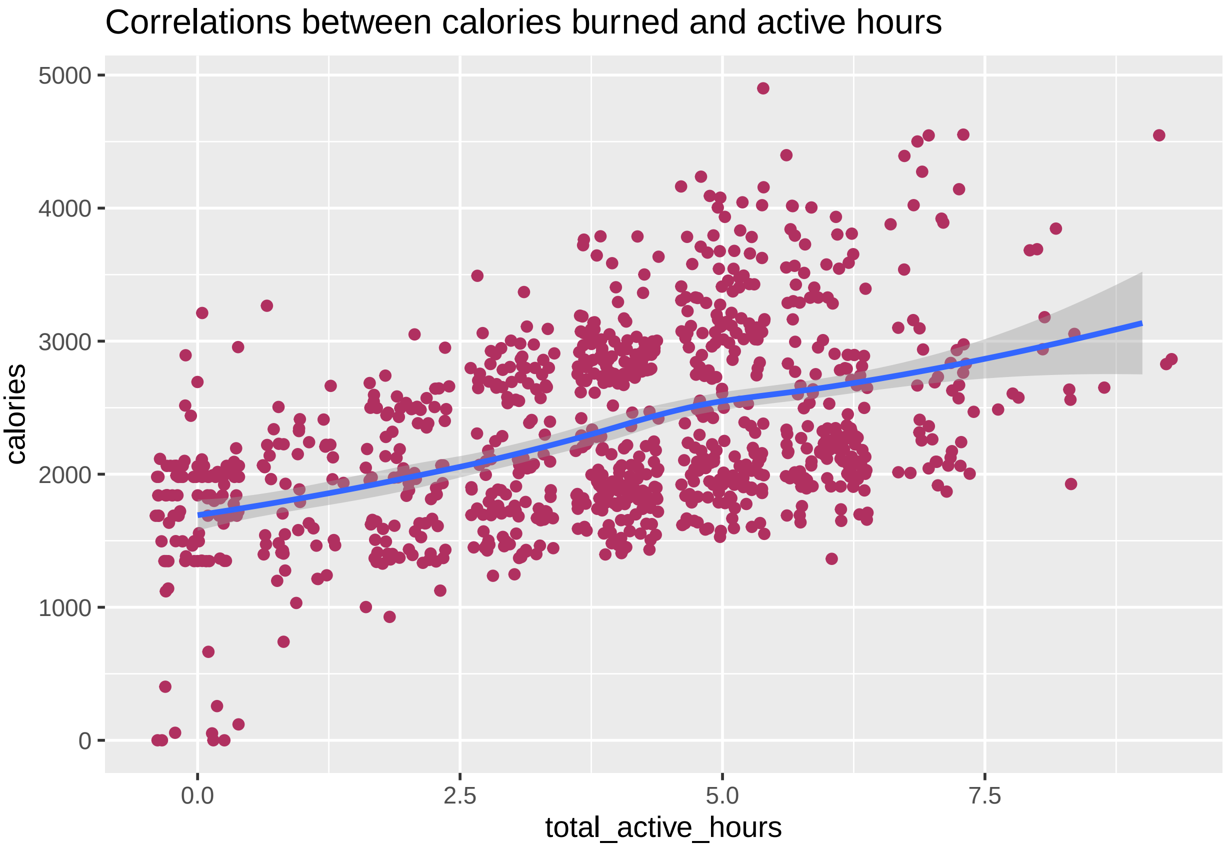
```
ggplot(daily_activity, aes(x = total_steps, y = calories)) + geom_smooth() + geom_point(color = "maroon
```

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



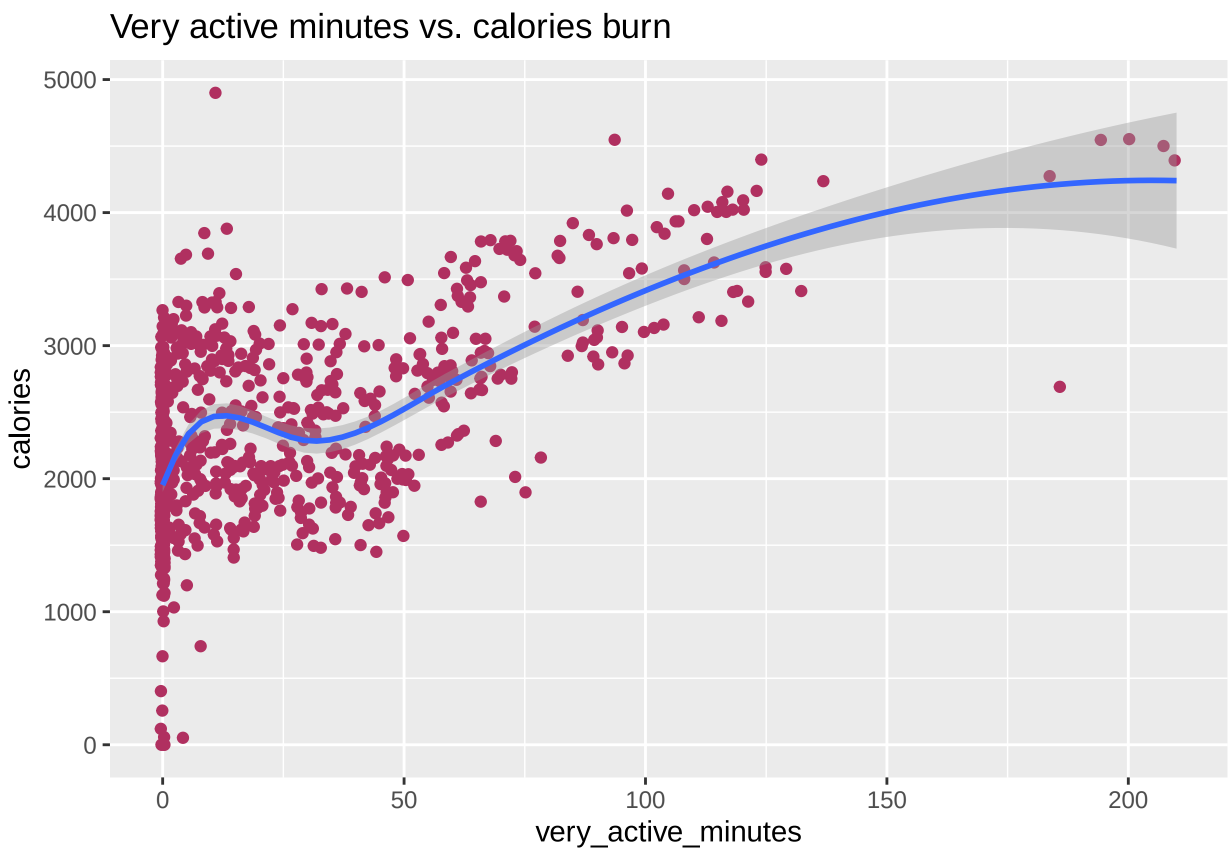
```
ggplot(daily_activity, aes(x = total_active_hours, y = calories)) + geom_jitter(color = "maroon") + geor
```

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



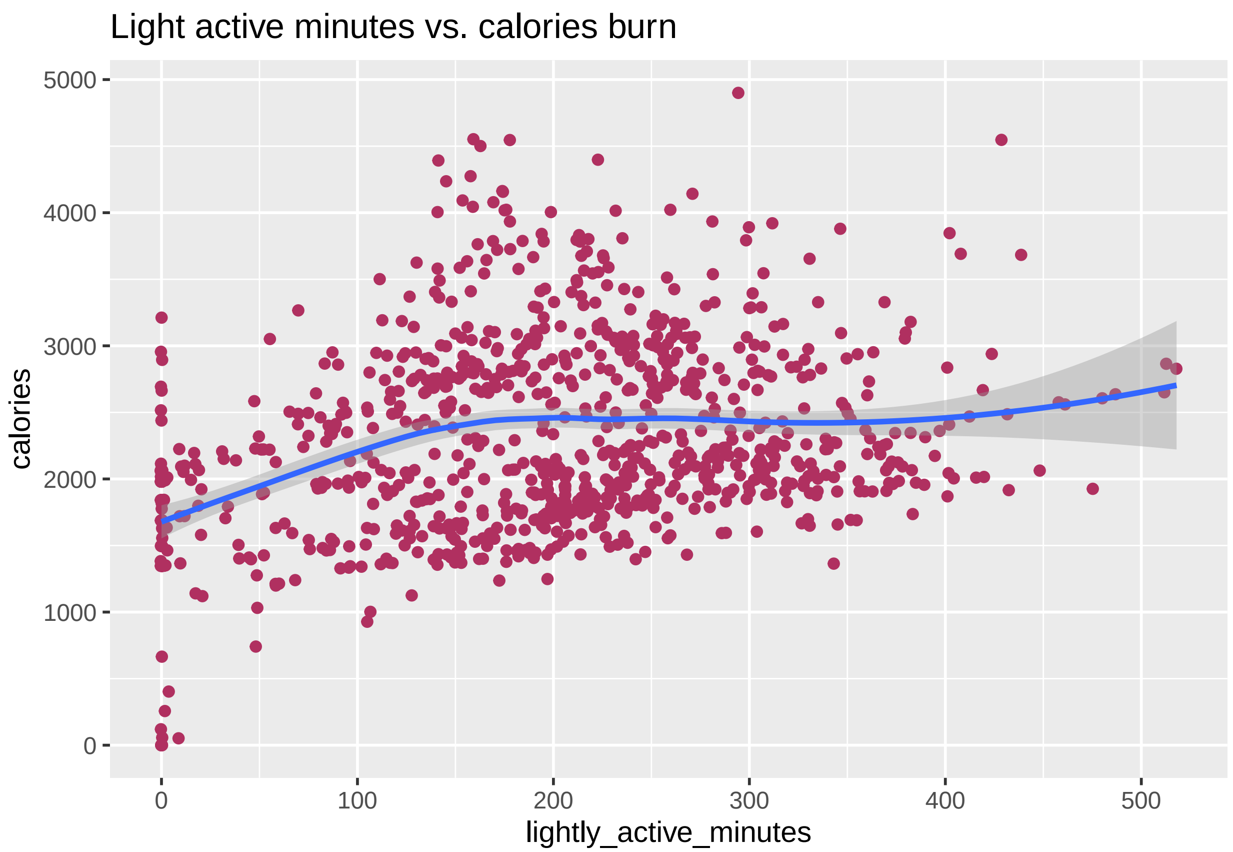
```
ggplot(daily_activity, aes(x = very_active_minutes, y = calories)) + geom_jitter(color = "maroon") + ge
```

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



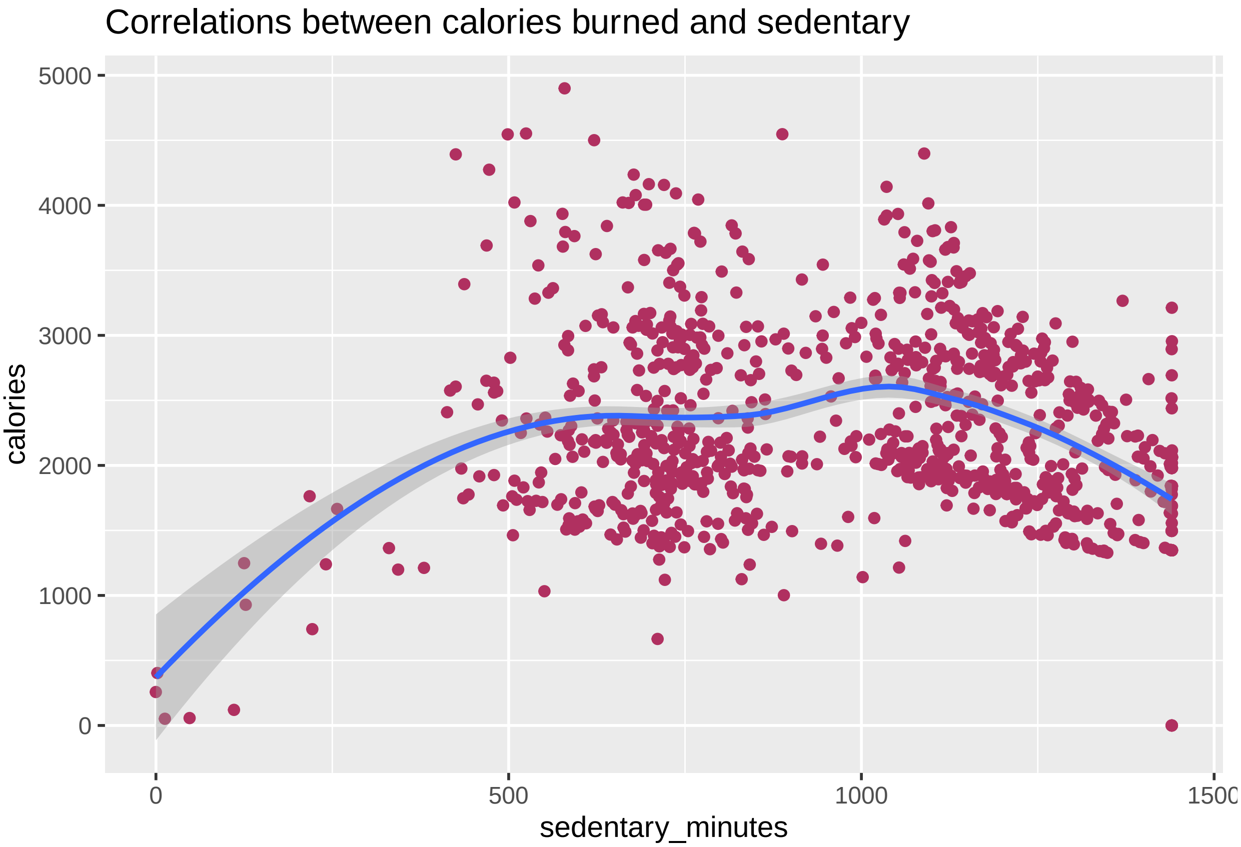
```
ggplot(daily_activity, aes(x = lightly_active_minutes, y = calories)) + geom_jitter(color = "maroon") +
```

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



```
ggplot(daily_activity, aes(x = sedentary_minutes, y = calories)) + geom_jitter(color = "maroon") + geom_
```

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



Summary

After analyzing the plot, it seem that there are a drop off in active hour after sunday, while the average active hours are around 3-4 hours a day from the first plot we know that it mostly consist of lightly active from a quick summary that we did before it./

Also, there are a strong correlation between total steps and calories burn from the correlation test and the plot. There are correlation between active hours and calories burn, however there are only moderate since the correlation coefficient is only 0.46. The relationship between active and calories burn does get better as we look at the correlation test and plot between very active and lightly active./

Recommendation and ideas

1. Bellabeat could have recommend workout routine through the app that let user know what workout they do for each day. From a personal experience, when I first start workout it very demotivating since I have no idea what to do, which in turn cause me to be lazy and not go to the gym, but after having a workout routine it give me no excuse since I have a set workout ready for me./
2. Having a notification of calories burn after a workout also motivate an individual since it show a goal that they have achieve./
3. The average sedentary is very concerning, what Bellabeat could do is have notification through there app that let user know when the sedentary time is high and recommend to do a light walk or stand up to stretch./