Google Data Analytics Bellabeat project

Olanrewaju

2023-10-19

This is the Prepare stage of the analysis where I am going to clean the data and make sure its ready for analysis

We Start by installing all the packages necessary for our analysis

```
install.packages('tidyverse')
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages('janitor')
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages('skimr')
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages('lubridate')
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

We load the packages installed

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     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 1.9.3     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
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

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

Loading the Csv files for the analysis

I created a Dataframe named daily_activity

```
daily_activity <- read.csv("dailyActivity_merged.csv")
```

I also want to create a dataframe named sleep_day

```
sleep_day <- read.csv("sleepDay_merged.csv")
```

I am also interested in creating a data frame named weight_log

```
weight_log <- read.csv("weightLogInfo_merged.csv")
```

Inspecting the dataset to see if there are any errors with the formatting

```
str(daily_activity)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate             : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps               : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance            : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes         : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                 : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
str(sleep_day)
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM" "
```

```
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
str(weight_log)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "4/21/20
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

After evaluting the formattng I discovored that the date column for all the data frame are formated as CHR and not in a date format

I want to clean the column names as well to make it consistent

```
daily_activity <- clean_names(daily_activity)
sleep_day <- clean_names(sleep_day)
weight_log <- clean_names(weight_log)
```

Convert the string to date data as appropriate

```
daily_activity$activity_date <- as.Date(daily_activity$activity_date, '%m/%d/%y')
sleep_day$sleep_day <- as.Date(sleep_day$sleep_day, '%m/%d/%y')
```

The weight_log dataframe has a AM/PM indicator. hence, I am going to use Parse_date_time command

```
weight_log$date <- parse_date_time(weight_log$date, '%m/%d/%y %H:%M:%S %p')
```

I also saw that the is_manual_report is the weight_log data frame is chr so i will convert it to logical format

```
weight_log $ is_manual_report <- as.logical(weight_log $ is_manual_report)
```

I want to add days of the week to the dataframe and also remove fat from the data set since it wont be used in the analysis

```
daily_activity$day_of_week <- wday(daily_activity$activity_date, label = T, abbr = T)
daily_activity$total_active_hours = round((daily_activity$very_active_minutes + daily_activity$fairly_a
daily_activity$sedentary_hours = round((daily_activity$sedentary_minutes)/60, digits = 2)
```

```r
sleep_day$hours_in_bed = round((sleep_day$total_time_in_bed)/60, digits = 2)
sleep_day$hours_asleep = round((sleep_day$total_minutes_asleep)/60, digits = 2)
sleep_day$time_taken_to_sleep = (sleep_day$total_time_in_bed - sleep_day$total_minutes_asleep)
```

Removing the fat column

```r
weight_log <- weight_log %>%
  select(-c(fat))
```

I want to also add a column that tells me if the user or over or underweight

```r
weight_log <- weight_log %>%
  mutate(bmi2 = case_when(
    bmi > 24.9 ~ 'Overweight',
    bmi < 18.5 ~ 'Underweight',
    TRUE ~ 'Healthy'
  ))
```

I want to remove rows in which total_active_hours and calories burned are zero. if the watch is not been worned then its not going to collect any data

```r
daily_activity_cleaned <- daily_activity[!(daily_activity$calories<=0),]
daily_activity_cleaned <- daily_activity_cleaned[!(daily_activity_cleaned$total_active_hours<=0.00),]
```

I have prepared and clean my data appropriately and I want to get into the Analysis Stage

I will be using the ggplot to understand some observation and i want to get the average of the steps taken, sedentary hours, very active minutes etc and also know the days where users are ost active.

summary(daily_activity_cleaned$total_steps) Min. 1st Qu. Median Mean 3rd Qu. Max. 0 4920 8053 8319 11100 36019

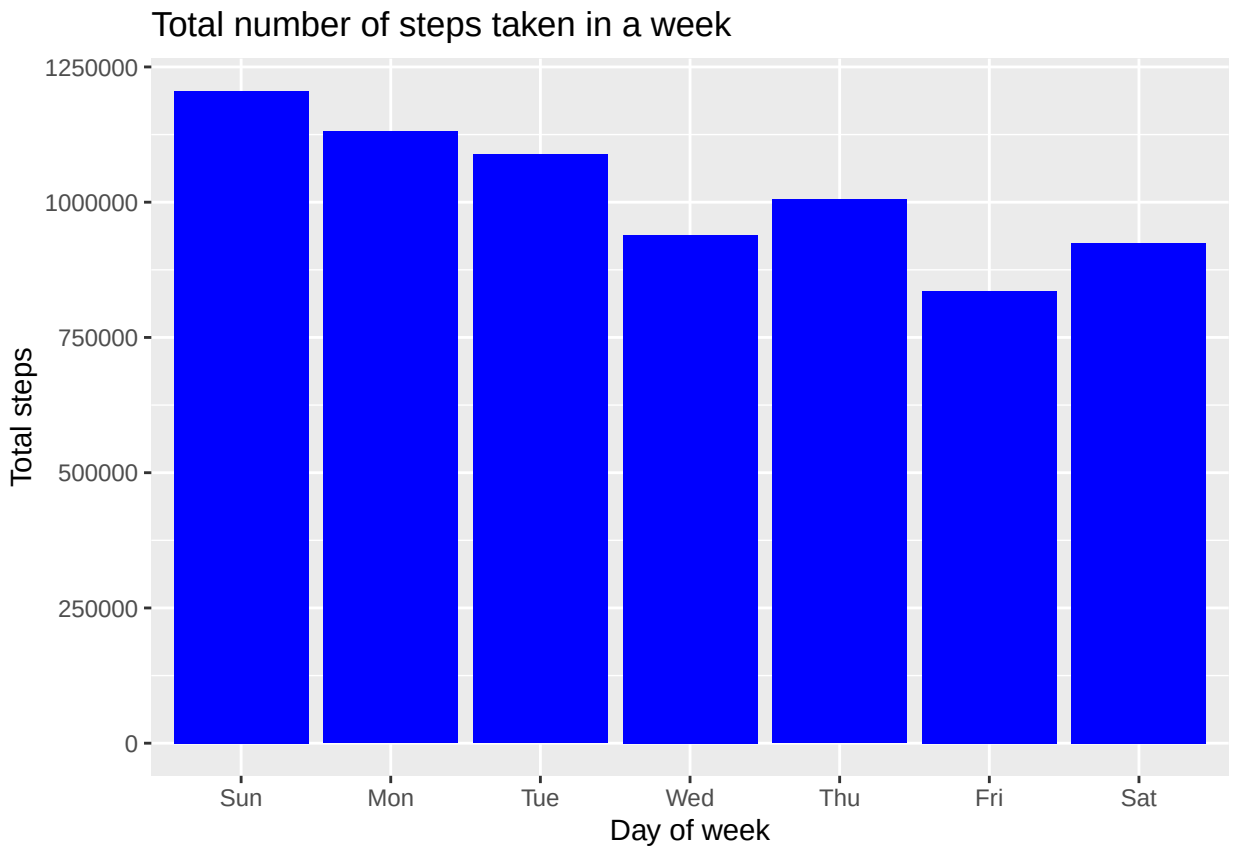
summary(daily_activity_cleaned$sedentary_hours) Min. 1st Qu. Median Mean 3rd Qu. Max. 0.00 12.02 17.00 15.87 19.80 23.98

summary(daily_activity_cleaned$very_active_minutes) Min. 1st Qu. Median Mean 3rd Qu. Max. 0.00 0.00 7.00 23.21 36.00 210.00

summary(sleep_day$hours_asleep) Min. 1st Qu. Median Mean 3rd Qu. Max. 0.970 6.020 7.220 6.992 8.170 13.270

days user are most active

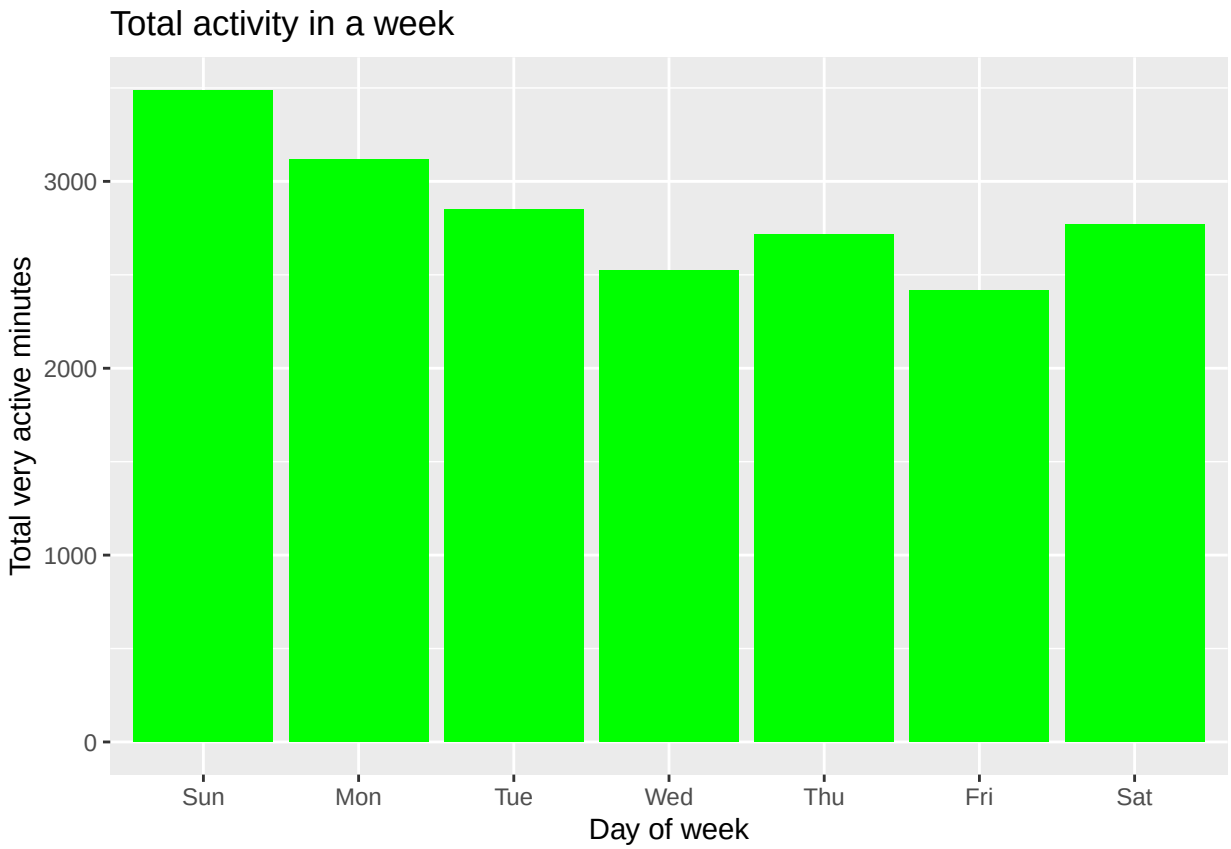
```
options(scipen = 999)
ggplot(data = daily_activity_cleaned) +
  aes(x = day_of_week, y = total_steps) +
  geom_col(fill =  'blue') +
  labs(x = 'Day of week', y = 'Total steps', title = 'Total number of steps taken in a week')
```




```
ggsave('total_steps.pdf')
```

```
## Saving 6.5 x 4.5 in image
```

```
ggplot(data = daily_activity_cleaned) +
  aes(x = day_of_week, y = very_active_minutes) +
  geom_col(fill =  'green') +
  labs(x = 'Day of week', y = 'Total very active minutes', title = 'Total activity in a week')
```




```
ggsave('total_activity.pdf')
```

```
## Saving 6.5 x 4.5 in image
```

```
ggplot(data = daily_activity_cleaned) +
  aes(x = day_of_week, y = calories) +
  geom_col(fill =  'red') +
  labs(x = 'Day of week', y = 'Calories burned', title = 'Total calories burned in a week')
```

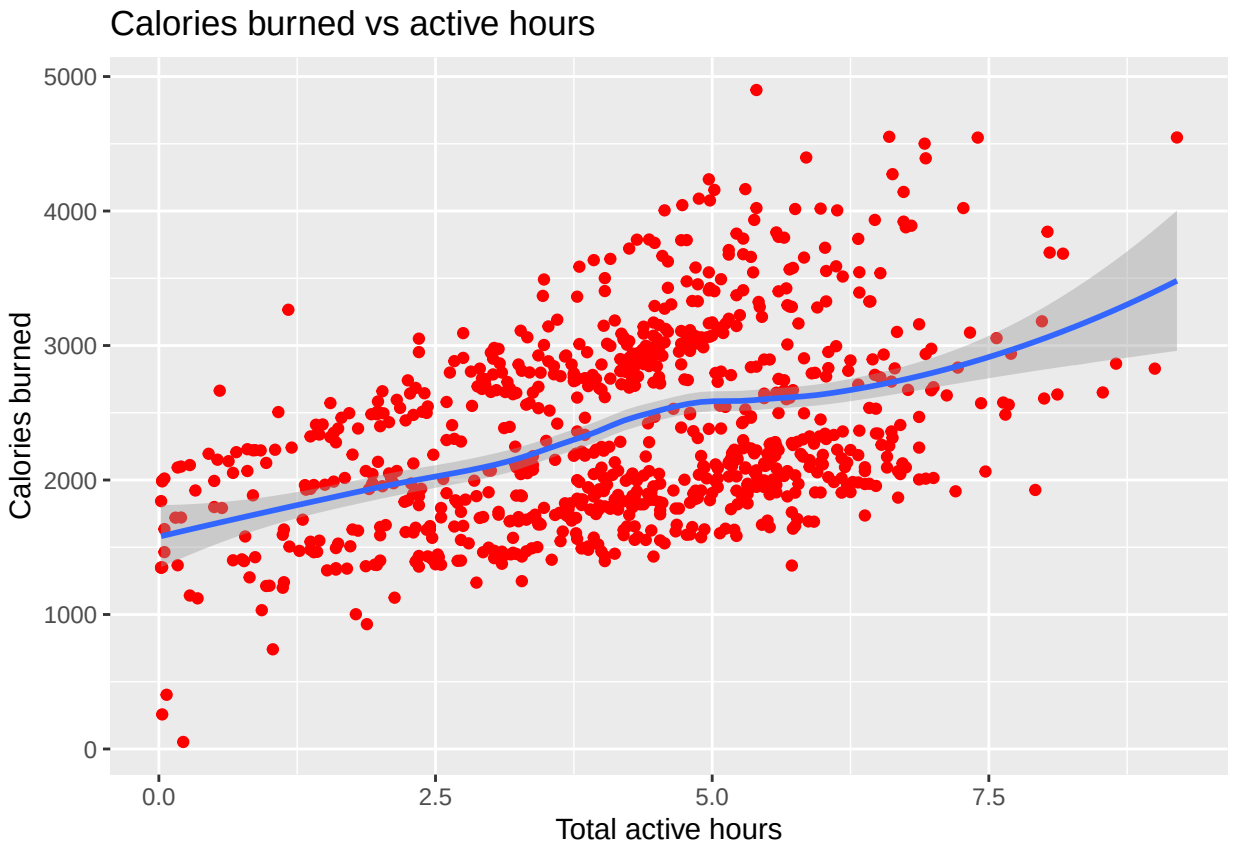

```
ggsave('total_calories.pdf')
```

```
## Saving 6.5 x 4.5 in image
```

lets see the relationship between total active hours, total steps taken and sedentary hours against calories burned

```
ggplot(data = daily_activity_cleaned) +
  aes(x= total_active_hours, y = calories) +
  geom_point(color = 'red') +
  geom_smooth() +
  labs(x = 'Total active hours', y = 'Calories burned', title = 'Calories burned vs active hours')
```

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

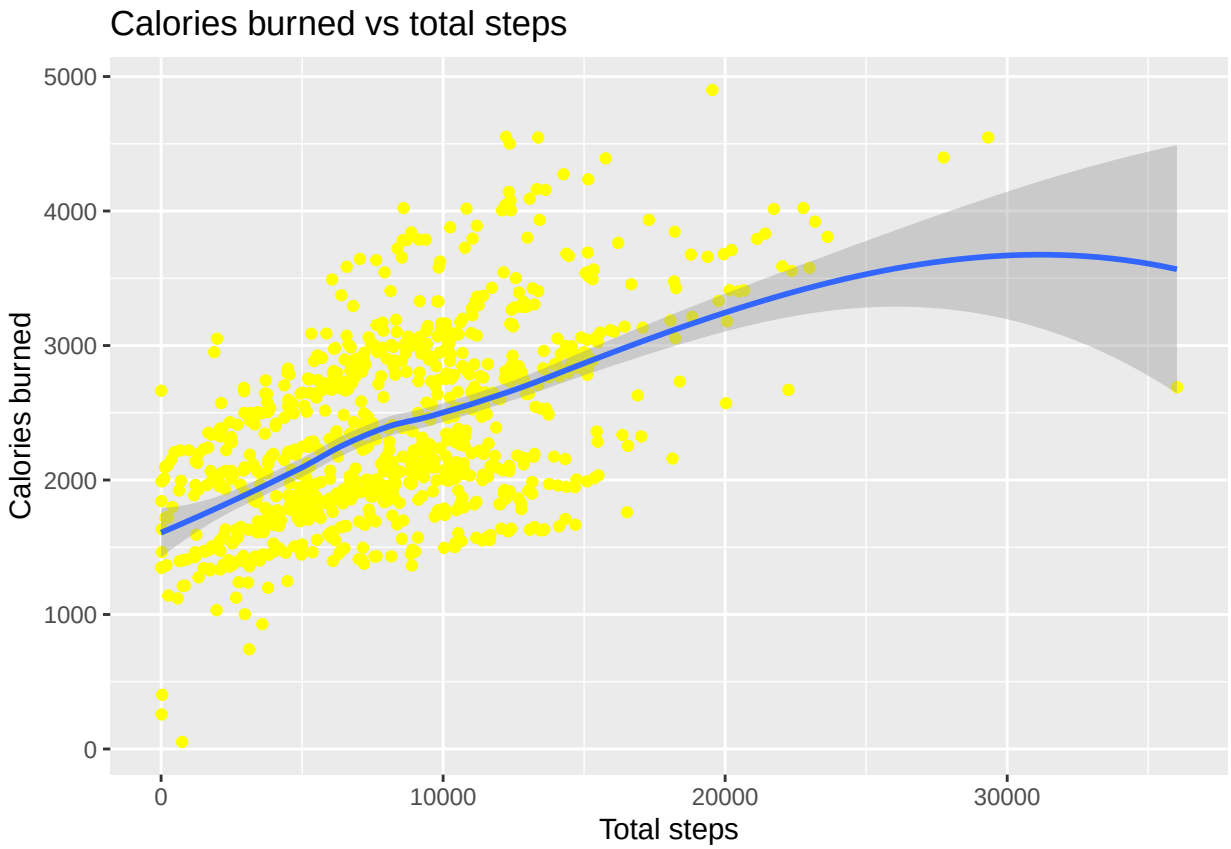



```
ggsave('calories_burned_vs_active_hours.pdf')
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
ggplot(data = daily_activity_cleaned) +
  aes(x= total_steps, y = calories) +
  geom_point(color = 'yellow') +
  geom_smooth() +
  labs(x = 'Total steps', y = 'Calories burned', title = 'Calories burned vs total steps')
```

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

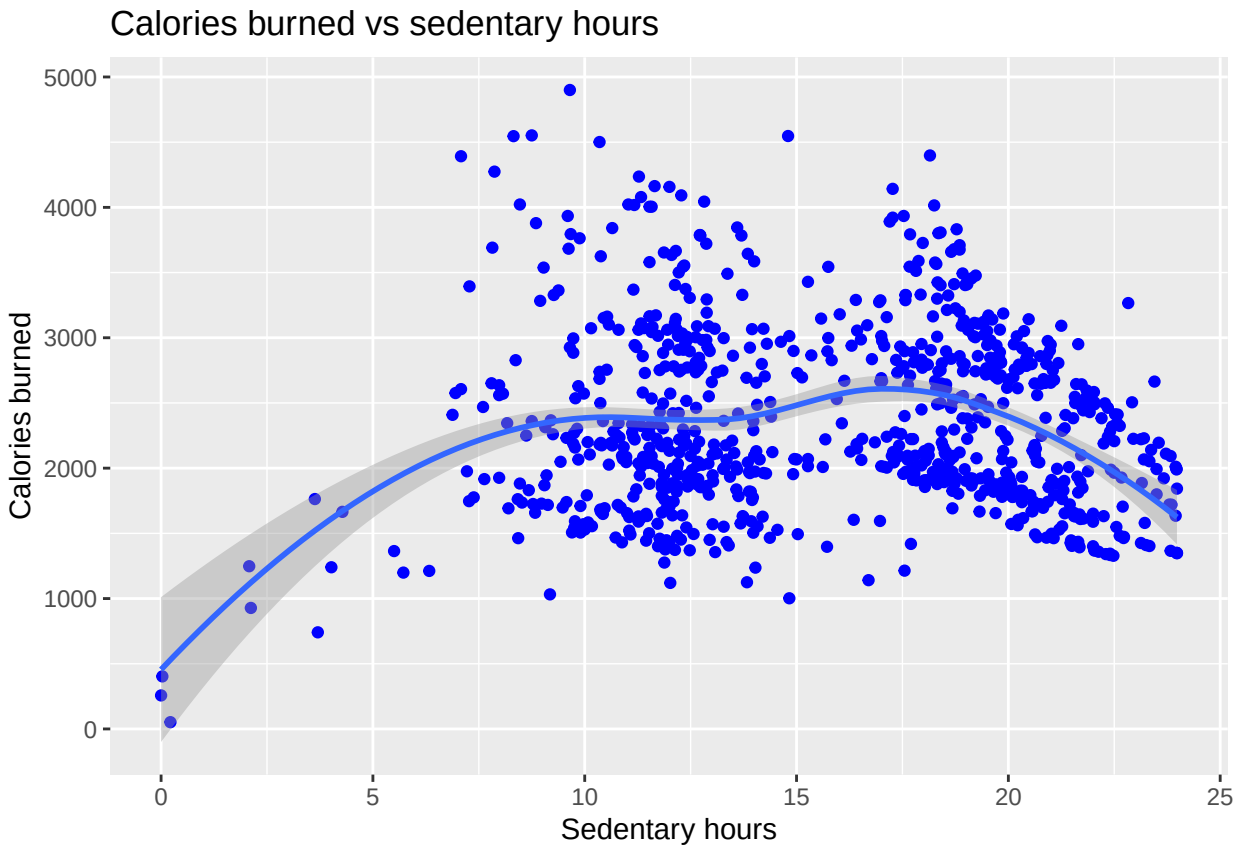



```
ggsave('calories_burned_vs_total_steps.pdf')
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
ggplot(data = daily_activity_cleaned) +
  aes(x= sedentary_hours, y = calories) +
  geom_point(color = 'blue') +
  geom_smooth() +
  labs(x = 'Sedentary hours', y = 'Calories burned', title = 'Calories burned vs sedentary hours')
```

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



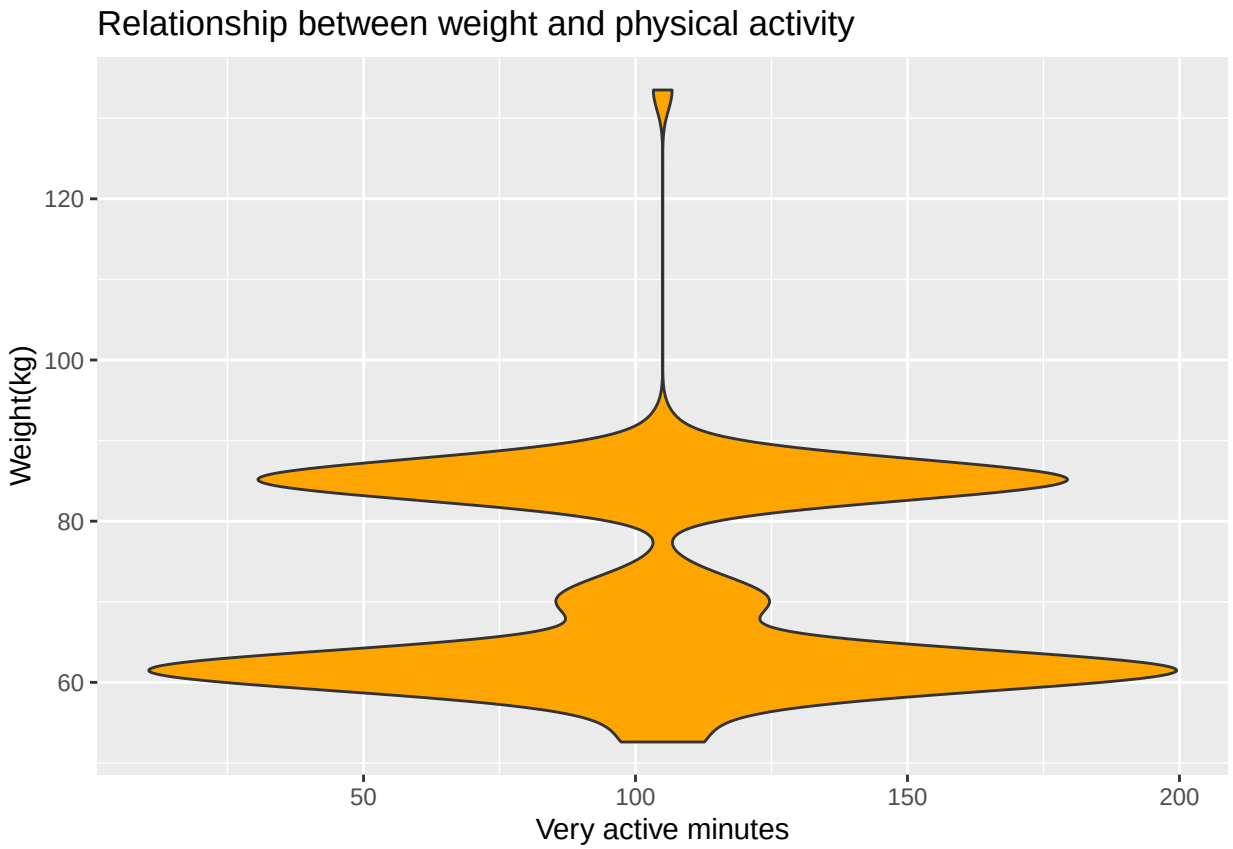

```
ggsave('sedentary_hours_vs_calories_burned.pdf')
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

Relationship between weight and physical activity

we have to merge the daily_activity and weight_log

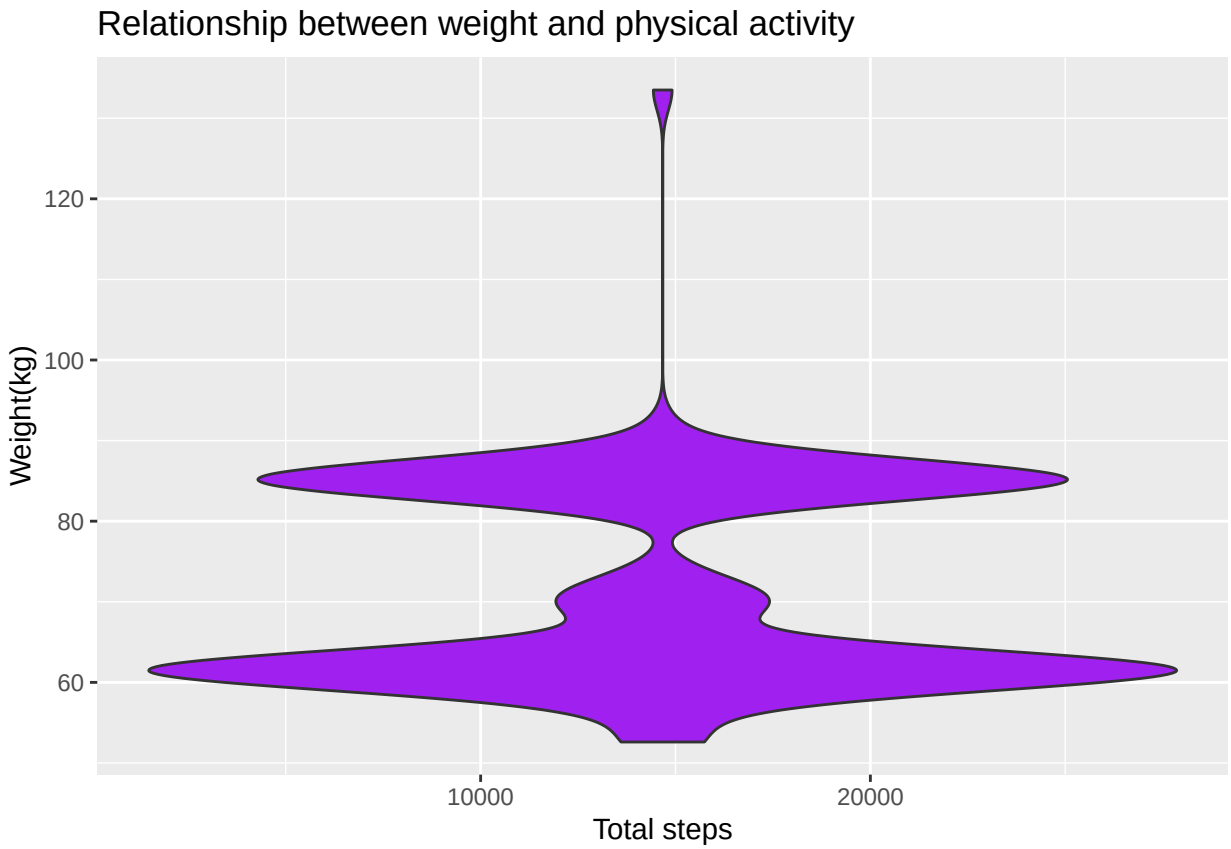
```
activity_weight <- merge(daily_activity_cleaned, weight_log, by=c('id'))

ggplot(data = activity_weight) +
  aes(x = very_active_minutes, y = weight_kg) +
  geom_violin(fill = 'orange') +
  labs(x = 'Very active minutes', y = 'Weight(kg)', title = 'Relationship between weight and physical a
```




```
ggsave('weight_physical_activity.pdf')
```

```
## Saving 6.5 x 4.5 in image
```

```
ggplot(data = activity_weight) +
  aes(x = total_steps, y = weight_kg) +
  geom_violin(fill = 'purple') +
  labs(x = 'Total steps', y = 'Weight(kg)', title = 'Relationship between weight and physical activity')
```




```
ggsave('weight_physical_activity.png')
```

```
## Saving 6.5 x 4.5 in image
```

I want to see the number of overweight and healthy users

#The amount of healthy users

```
nrow(filter(distinct(weight_log, id, .keep_all = T),bmi2 == 'Healthy'))
```

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

#The amount of underweight users

```
nrow(filter(distinct(weight_log, id, .keep_all = T),bmi2 == 'Underweight'))
```

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

#The amount of overweight users

```
nrow(filter(distinct(weight_log, id, .keep_all = T),bmi2 == 'Overweight'))
```

```
## [1] 5
```