Bellabeat Case Study

Madison McClure

2023-02-12

Preparing the Data

Setting up my environment

Notes: setting up my R environment by loading the `tidyverse`, `ggplot`, `dplyr` , `janitor`, `readr`, and `lubridate` packages:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
library(janitor)
```

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

```
library(readr)
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Loading the CSV data files

Here I loaded the files I planned to use.

```
daily_activity <- read_csv("/cloud/project/Bellabeat Case Study/Fitabase Data 4.12.16-5.12.16/dailyActi
```

```
## 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.
```

```
sleep <- read_csv("/cloud/project/Bellabeat Case Study/Fitabase Data 4.12.16-5.12.16/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.
```

```
weight <- read_csv("/cloud/project/Bellabeat Case Study/Fitabase Data 4.12.16-5.12.16/weightLogInfo_merg
```

```
## 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.
```

Viewing the data

Here I looked at the data.

```
View(daily_activity)
View(sleep)
View(weight)
```

Processing the data

Checking for consistent column names

```
clean_names(daily_activity)
```

```
## # A tibble: 940 x 15
##            id activity~1 total~2 total~3 track~4 logge~5 very_~6 moder~7 light~8
##         <dbl> <chr>        <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 1503960366 4/12/2016    13162    8.5     8.5        0    1.88   0.550    6.06
##  2 1503960366 4/13/2016    10735    6.97    6.97       0    1.57   0.690    4.71
##  3 1503960366 4/14/2016    10460    6.74    6.74       0    2.44   0.400    3.91
##  4 1503960366 4/15/2016     9762    6.28    6.28       0    2.14   1.26     2.83
##  5 1503960366 4/16/2016    12669    8.16    8.16       0    2.71   0.410    5.04
##  6 1503960366 4/17/2016     9705    6.48    6.48       0    3.19   0.780    2.51
##  7 1503960366 4/18/2016    13019    8.59    8.59       0    3.25   0.640    4.71
##  8 1503960366 4/19/2016    15506    9.88    9.88       0    3.53   1.32     5.03
##  9 1503960366 4/20/2016    10544    6.68    6.68       0    1.96   0.480    4.24
## 10 1503960366 4/21/2016     9819    6.34    6.34       0    1.34   0.350    4.65
## # ... with 930 more rows, 6 more variables: sedentary_active_distance <dbl>,
## #   very_active_minutes <dbl>, fairly_active_minutes <dbl>,
## #   lightly_active_minutes <dbl>, sedentary_minutes <dbl>, calories <dbl>, and
## #   abbreviated variable names 1: activity_date, 2: total_steps,
## #   3: total_distance, 4: tracker_distance, 5: logged_activities_distance,
## #   6: very_active_distance, 7: moderately_active_distance,
## #   8: light_active_distance
```

```
clean_names(sleep)
```

```
## # A tibble: 413 x 5
##            id sleep_day             total_sleep_records total_minutes_~1 total~2
##         <dbl> <chr>                               <dbl>            <dbl>   <dbl>
##  1 1503960366 4/12/2016 12:00:00 AM                   1              327     346
##  2 1503960366 4/13/2016 12:00:00 AM                   2              384     407
##  3 1503960366 4/15/2016 12:00:00 AM                   1              412     442
##  4 1503960366 4/16/2016 12:00:00 AM                   2              340     367
##  5 1503960366 4/17/2016 12:00:00 AM                   1              700     712
##  6 1503960366 4/19/2016 12:00:00 AM                   1              304     320
##  7 1503960366 4/20/2016 12:00:00 AM                   1              360     377
##  8 1503960366 4/21/2016 12:00:00 AM                   1              325     364
##  9 1503960366 4/23/2016 12:00:00 AM                   1              361     384
## 10 1503960366 4/24/2016 12:00:00 AM                   1              430     449
## # ... with 403 more rows, and abbreviated variable names
## #   1: total_minutes_asleep, 2: total_time_in_bed
```

```
clean_names(weight)
```

```
## # A tibble: 67 x 8
##            id date                  weight~1 weigh~2   fat   bmi is_ma~3  log_id
##         <dbl> <chr>                    <dbl>   <dbl> <dbl> <dbl> <lgl>     <dbl>
##  1 1503960366 5/2/2016 11:59:59 PM      52.6    116.    22  22.6 TRUE    1.46e12
##  2 1503960366 5/3/2016 11:59:59 PM      52.6    116.    NA  22.6 TRUE    1.46e12
##  3 1927972279 4/13/2016 1:08:52 AM     134.     294.    NA  47.5 FALSE   1.46e12
##  4 2873212765 4/21/2016 11:59:59 PM     56.7    125.    NA  21.5 TRUE    1.46e12
##  5 2873212765 5/12/2016 11:59:59 PM     57.3    126.    NA  21.7 TRUE    1.46e12
##  6 4319703577 4/17/2016 11:59:59 PM     72.4    160.    25  27.5 TRUE    1.46e12
##  7 4319703577 5/4/2016 11:59:59 PM      72.3    159.    NA  27.4 TRUE    1.46e12
##  8 4558609924 4/18/2016 11:59:59 PM     69.7    154.    NA  27.2 TRUE    1.46e12
##  9 4558609924 4/25/2016 11:59:59 PM     70.3    155.    NA  27.5 TRUE    1.46e12
## 10 4558609924 5/1/2016 11:59:59 PM      69.9    154.    NA  27.3 TRUE    1.46e12
## # ... with 57 more rows, and abbreviated variable names 1: weight_kg,
## #   2: weight_pounds, 3: is_manual_report
```

Renaming

Here I renamed the distance column to reflect the unit of measurement in the daily_activity table.

```
daily_activity <- rename(daily_activity, TotalDistanceKm = TotalDistance)
```

Checking the datatypes

Here I checked the datatypes of the variables in each table to make sure they were consistent and made sense.

```
glimpse(daily_activity)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019~
## $ TotalDistanceKm          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```
glimpse(sleep)
```

```
## Rows: 413
## Columns: 5
## $ Id                 <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150~
## $ SleepDay           <chr> "4/12/2016 12:00:00 AM", "4/13/2016 12:00:00 AM", "~
## $ TotalSleepRecords  <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, 430, 2~
## $ TotalTimeInBed     <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, 449, 3~
```

```
glimpse(weight)
```

```
## Rows: 67
## Columns: 8
## $ Id             <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 2873212~
## $ Date           <chr> "5/2/2016 11:59:59 PM", "5/3/2016 11:59:59 PM", "4/13/2~
## $ WeightKg       <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70.3, ~
## $ WeightPounds   <dbl> 115.9631, 115.9631, 294.3171, 125.0021, 126.3249, 159.6~
## $ Fat            <dbl> 22, NA, NA, NA, NA, 25, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BMI            <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27.25,~
## $ IsManualReport <lgl> TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, ~
## $ LogId          <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.461283e+12,~
```

Checking for duplicates

Here I checked each of the tables for any duplicate rows of data.

```
get_dupes(daily_activity)
```

```
## No variable names specified - using all columns.

## No duplicate combinations found of: Id, ActivityDate, TotalSteps, TotalDistanceKm, TrackerDistance,

## # A tibble: 0 x 16
## # ... with 16 variables: Id <dbl>, ActivityDate <chr>, TotalSteps <dbl>,
## #   TotalDistanceKm <dbl>, TrackerDistance <dbl>,
## #   LoggedActivitiesDistance <dbl>, VeryActiveDistance <dbl>,
## #   ModeratelyActiveDistance <dbl>, LightActiveDistance <dbl>,
## #   SedentaryActiveDistance <dbl>, VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>, dupe_count <int>
```

```
get_dupes(sleep)
```

```
## No variable names specified - using all columns.
```

```
## # A tibble: 6 x 6
##           Id SleepDay              TotalSleepRecords TotalMinu~1 Total~2 dupe_~3
##        <dbl> <chr>                             <dbl>       <dbl>   <dbl>   <int>
## 1 4388161847 5/5/2016 12:00:00 AM                  1         471     495       2
## 2 4388161847 5/5/2016 12:00:00 AM                  1         471     495       2
## 3 4702921684 5/7/2016 12:00:00 AM                  1         520     543       2
## 4 4702921684 5/7/2016 12:00:00 AM                  1         520     543       2
## 5 8378563200 4/25/2016 12:00:00 AM                 1         388     402       2
## 6 8378563200 4/25/2016 12:00:00 AM                 1         388     402       2
## # ... with abbreviated variable names 1: TotalMinutesAsleep, 2: TotalTimeInBed,
## #   3: dupe_count
```

```
get_dupes(weight)
```

```
## No variable names specified - using all columns.
```

```
## No duplicate combinations found of: Id, Date, WeightKg, WeightPounds, Fat, BMI, IsManualReport, LogI
```

```
## # A tibble: 0 x 9
## # ... with 9 variables: Id <dbl>, Date <chr>, WeightKg <dbl>,
## #   WeightPounds <dbl>, Fat <dbl>, BMI <dbl>, IsManualReport <lgl>,
## #   LogId <dbl>, dupe_count <int>
```

Here I removed duplicates found in the sleep table.

```
sleep <- sleep %>% distinct()
```

Data cleanup

Here I removed rows where users took zero steps that day since this is likely due to users not wearing their trackers.

```
daily_activity <- filter(daily_activity, TotalSteps != 0)
```

Adding columns

Here I added a column to each table to specify which day of the week the data is from.

```
daily_activity$ActivityDate <- mdy(daily_activity$ActivityDate)
daily_activity$Weekday <- weekdays(daily_activity$ActivityDate)

sleep$Date <- as.Date(sleep$SleepDay, format = "%m/%d/%Y")
sleep$Weekday <- weekdays(sleep$Date)

weight$Date_Only <- as.Date(weight$Date, format = "%m/%d/%Y")
weight$Weekday <- weekdays(weight$Date_Only)
```

Data Visualizations

Settings

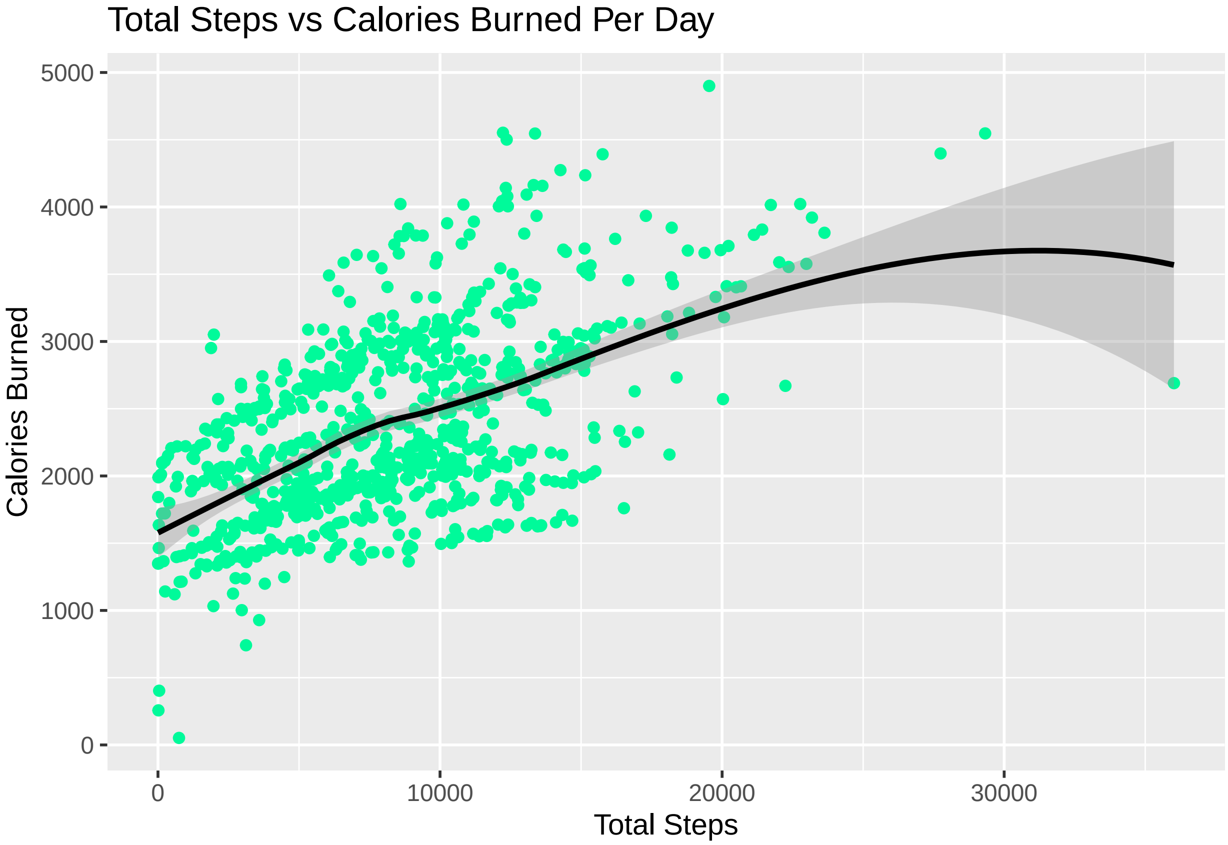
Here I turned off scientific notation for the graphs.

```
options(scipen=999)
```

Steps vs calories burned

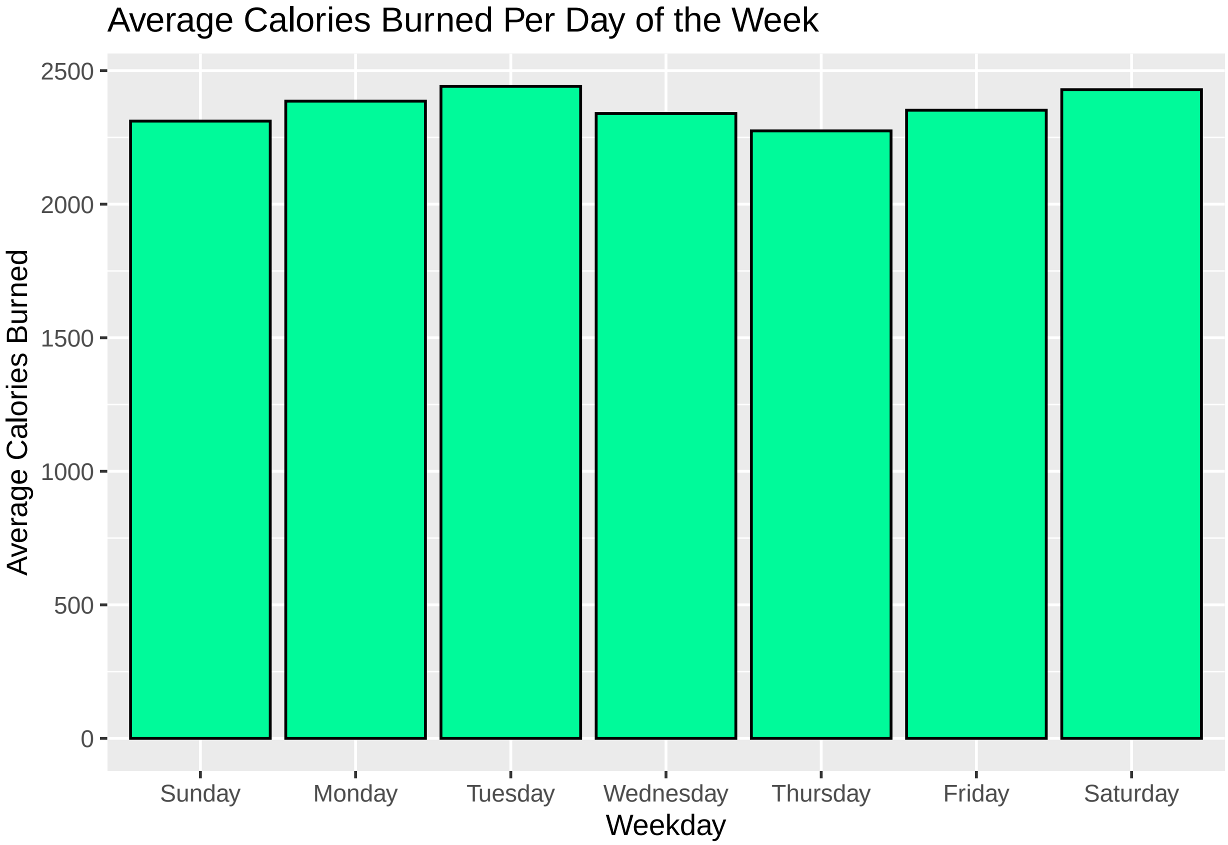
```
ggplot(data=daily_activity) + geom_point(mapping=aes(x=TotalSteps, y=Calories),
                                         color="mediumspringgreen") +
  labs(title="Total Steps vs Calories Burned Per Day",x="Total Steps",
       y="Calories Burned") +
  geom_smooth(mapping=aes(x=TotalSteps,y=Calories),color="black")
```

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



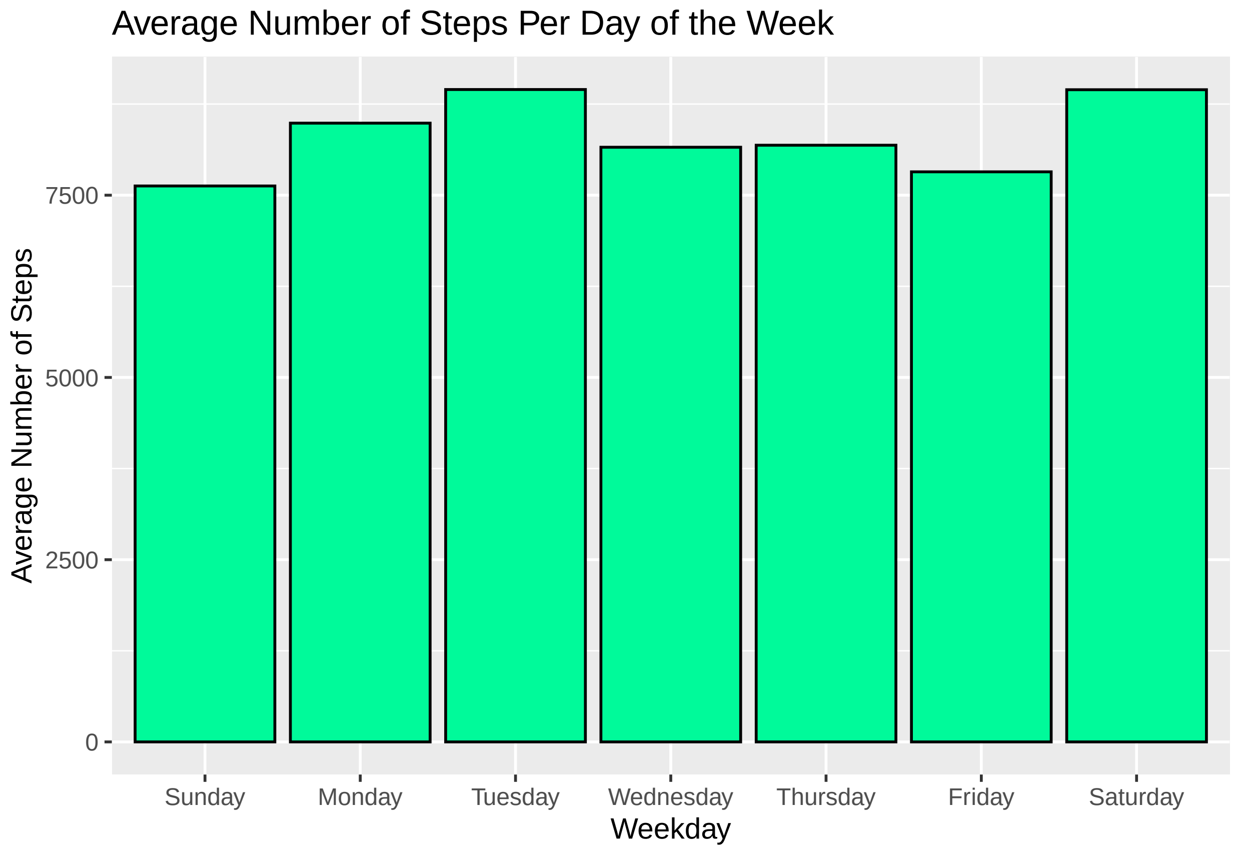
Calories burned vs day of week

```
avg_calories <- daily_activity %>% group_by(Weekday) %>%
  summarize(calories_avg = mean(Calories))

ggplot(data=avg_calories) +
  geom_bar(mapping=aes(x=Weekday, y=calories_avg), stat='identity',
           fill="mediumspringgreen",color="black") +
  labs(title="Average Calories Burned Per Day of the Week",
      y="Average Calories Burned") +
  xlim("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")
```



Steps vs day of week

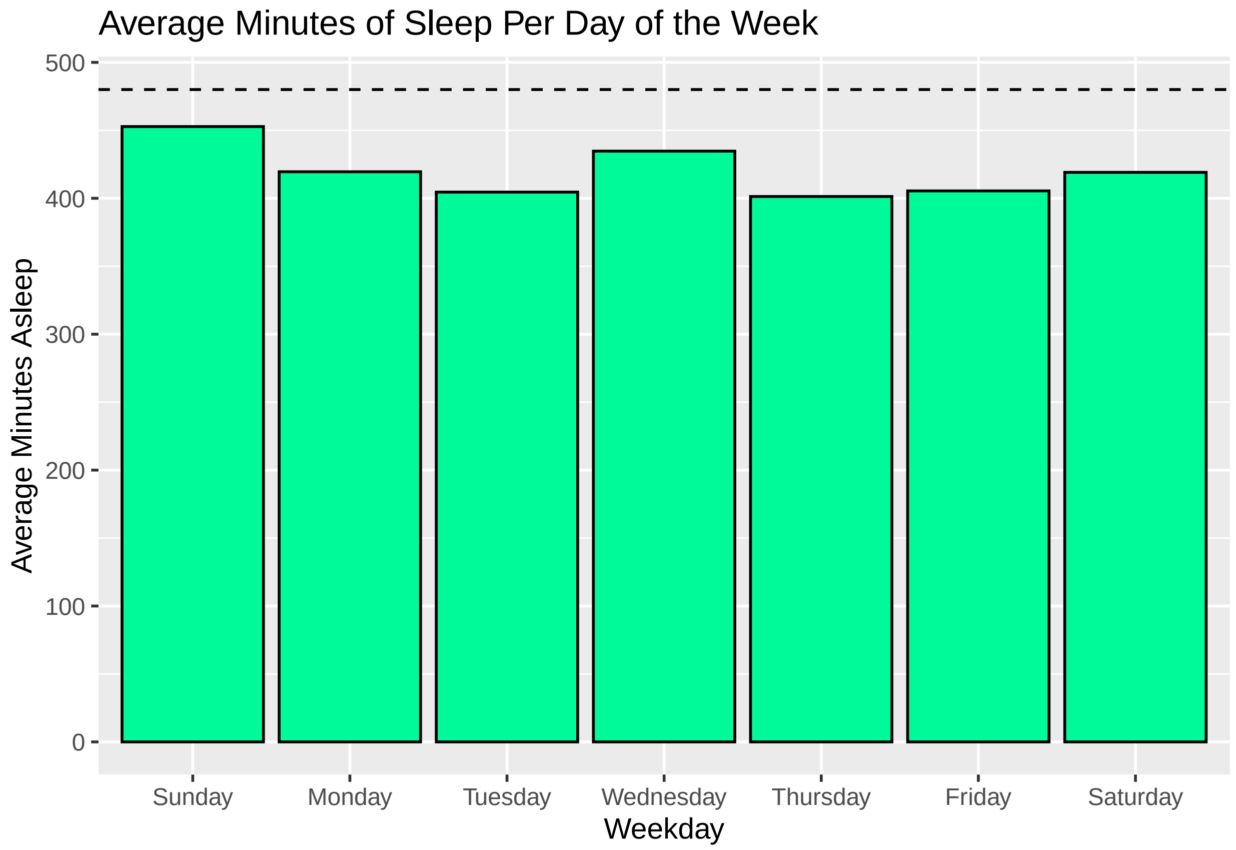
```
avg_steps <- daily_activity %>% group_by(Weekday) %>%
  summarize(avg_num_steps = mean(TotalSteps))

ggplot(data=avg_steps) +
  geom_bar(mapping=aes(x=Weekday, y=avg_num_steps), stat='identity',
           fill="mediumspringgreen",color="black") +
  labs(title="Average Number of Steps Per Day of the Week",
      y="Average Number of Steps") +
  xlim("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")
```



Sleep vs day of week

```
avg_sleep <- sleep %>% group_by(Weekday) %>%
  summarize(avg_sleep_minutes = mean(TotalMinutesAsleep))

ggplot(data=avg_sleep) +
  geom_bar(mapping=aes(x=Weekday, y=avg_sleep_minutes),stat='identity',
           fill="mediumspringgreen",color="black") +
  labs(title="Average Minutes of Sleep Per Day of the Week",
       y="Average Minutes Asleep") +
  geom_hline(yintercept=480, linetype="dashed", color = "black") +
  annotate("text", x = "", y = 480, label = "Recommended amount of sleep") +
  xlim("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")
```

```
## Warning: Removed 1 rows containing missing values (`geom_text()`).
```

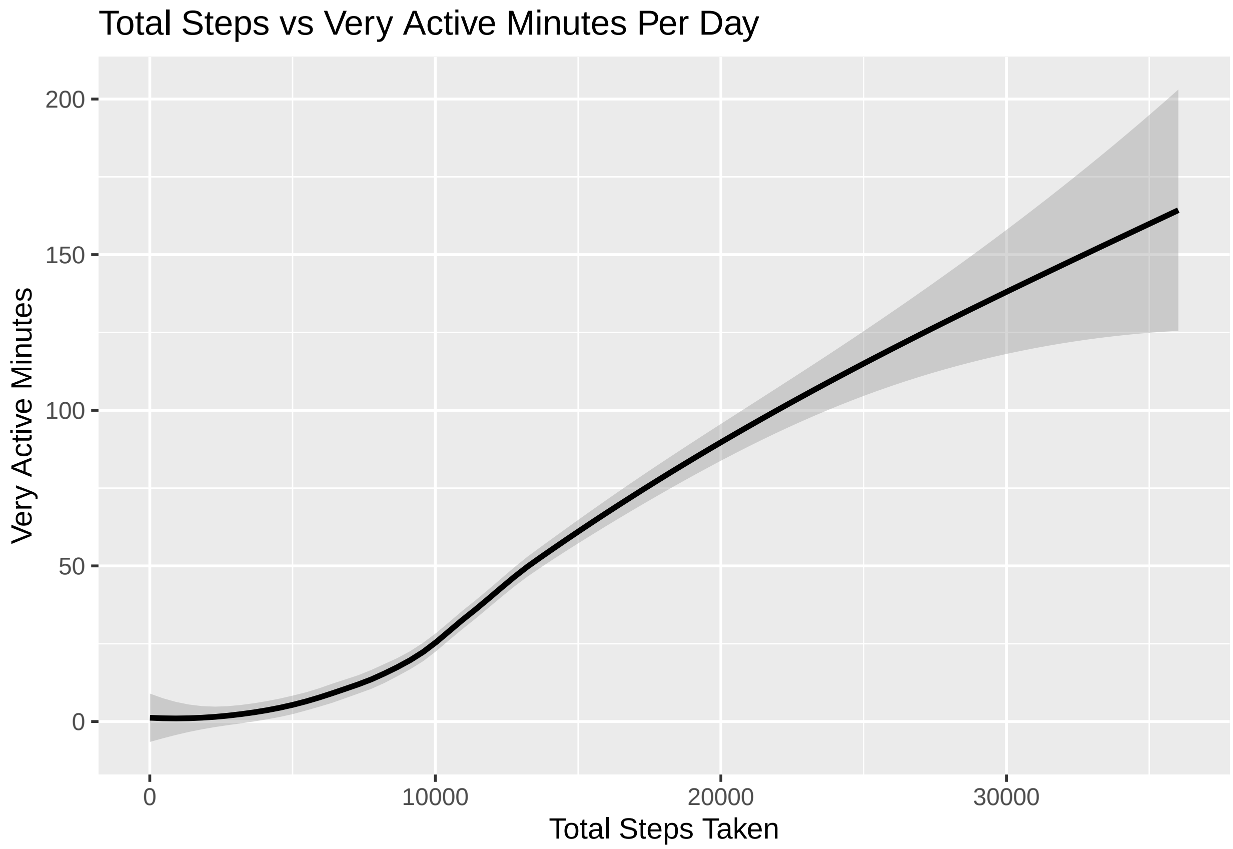


Steps vs active minutes

Very active minutes:

```
ggplot(data=daily_activity) +
  geom_smooth(mapping=aes(x=TotalSteps,y=VeryActiveMinutes),color="black") +
  labs(title="Total Steps vs Very Active Minutes Per Day",x="Total Steps Taken",
       y="Very Active Minutes")
```

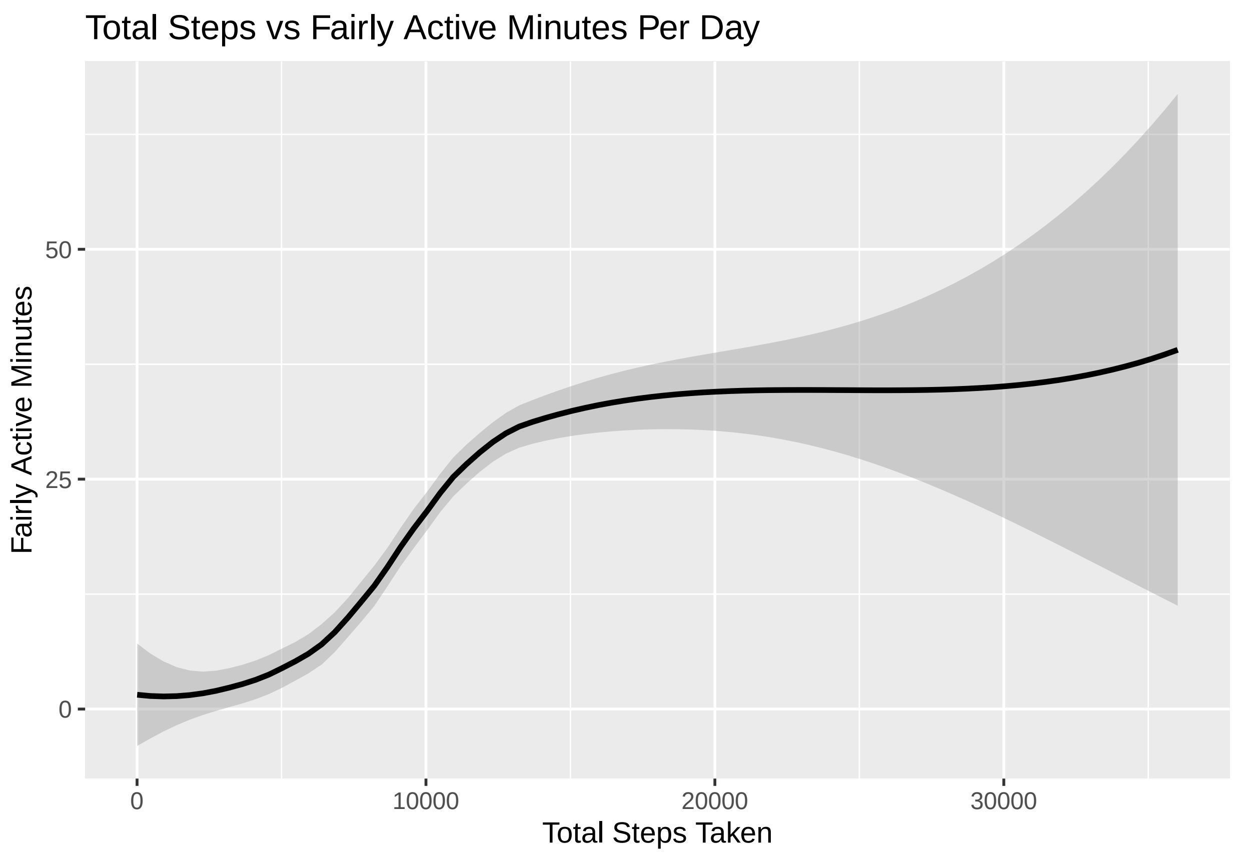
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Fairly active minutes:

```
ggplot(data=daily_activity) +
  geom_smooth(mapping=aes(x=TotalSteps,y=FairlyActiveMinutes),color="black") +
  labs(title="Total Steps vs Fairly Active Minutes Per Day",
       x="Total Steps Taken",y="Fairly Active Minutes")
```

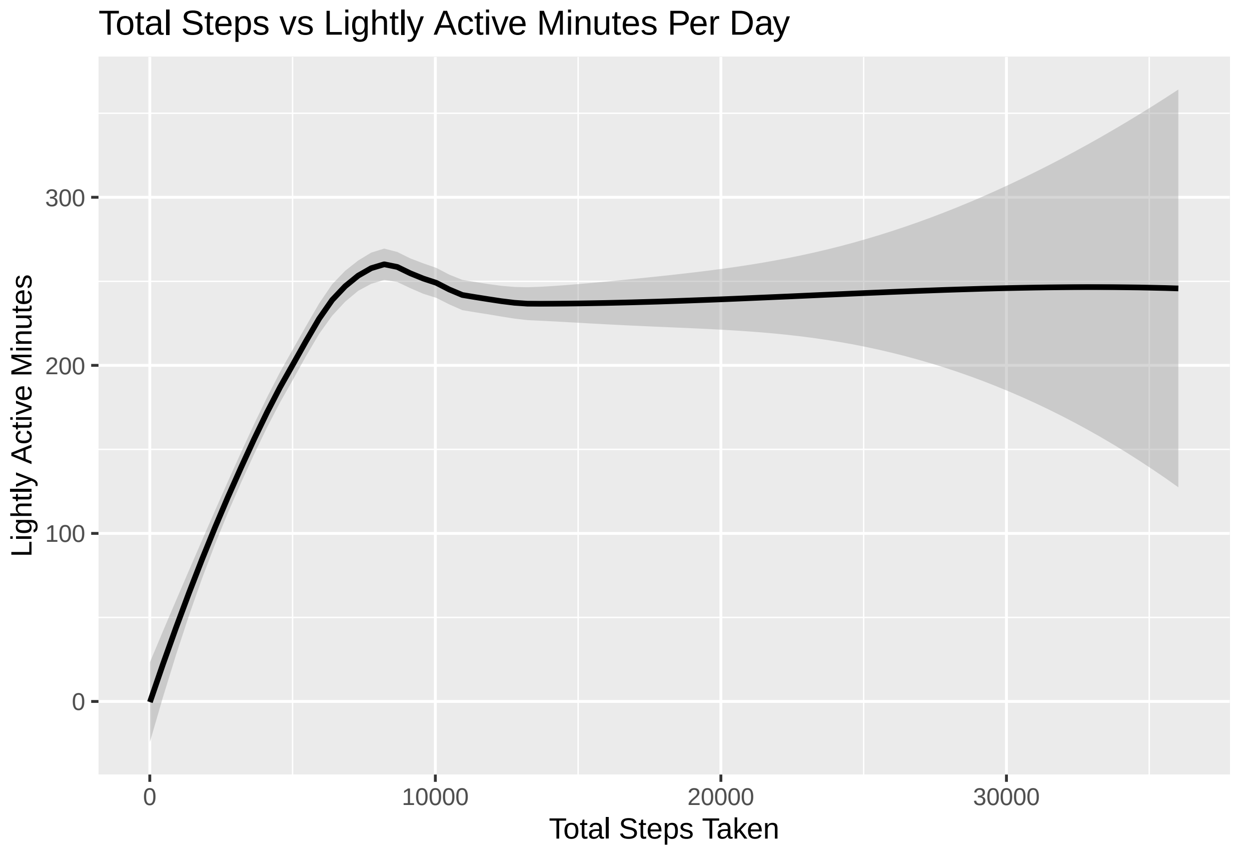
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Lightly active minutes:

```
ggplot(data=daily_activity) +
  geom_smooth(mapping=aes(x=TotalSteps,y=LightlyActiveMinutes),color="black") +
  labs(title="Total Steps vs Lightly Active Minutes Per Day",
       x="Total Steps Taken",y="Lightly Active Minutes")
```

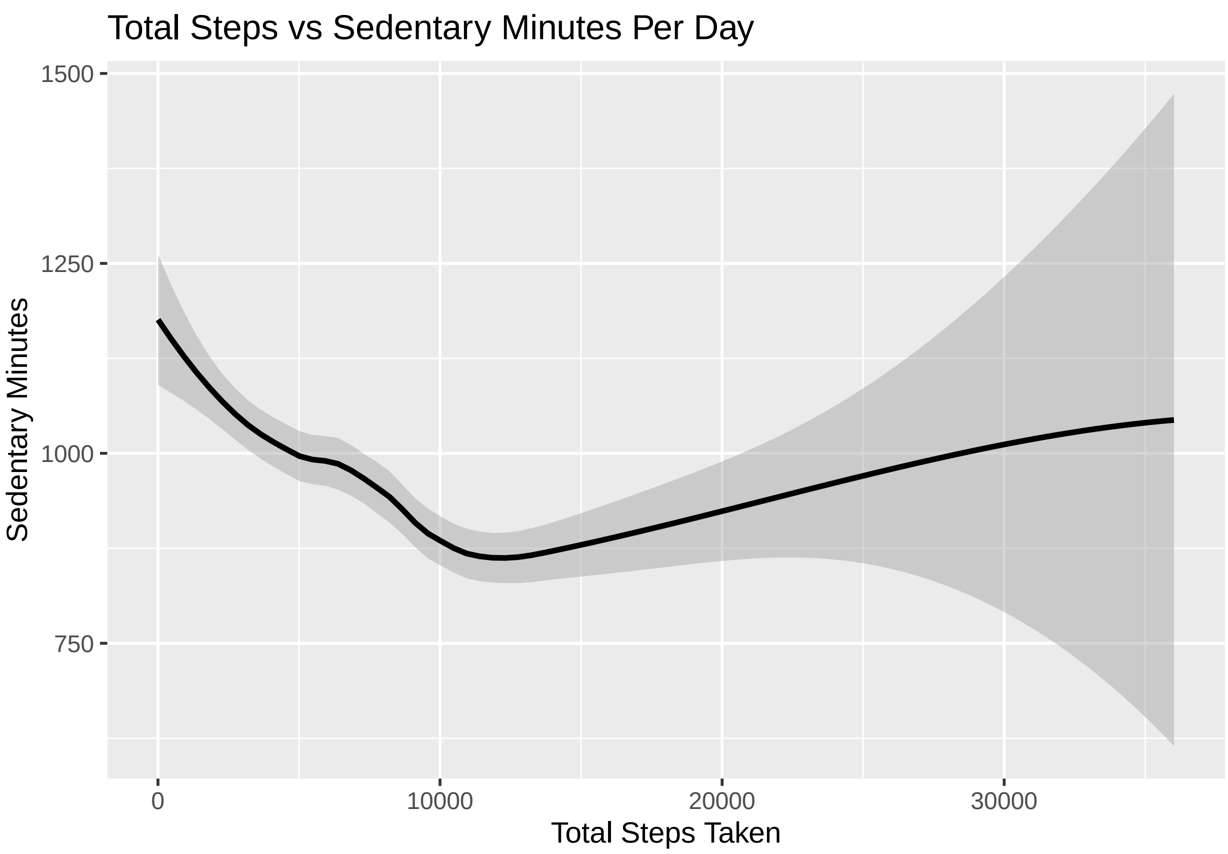
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Sedentary minutes:

```
ggplot(data=daily_activity) +
  geom_smooth(mapping=aes(x=TotalSteps,y=SedentaryMinutes),color="black") +
  labs(title="Total Steps vs Sedentary Minutes Per Day",x="Total Steps Taken",
       y="Sedentary Minutes")
```

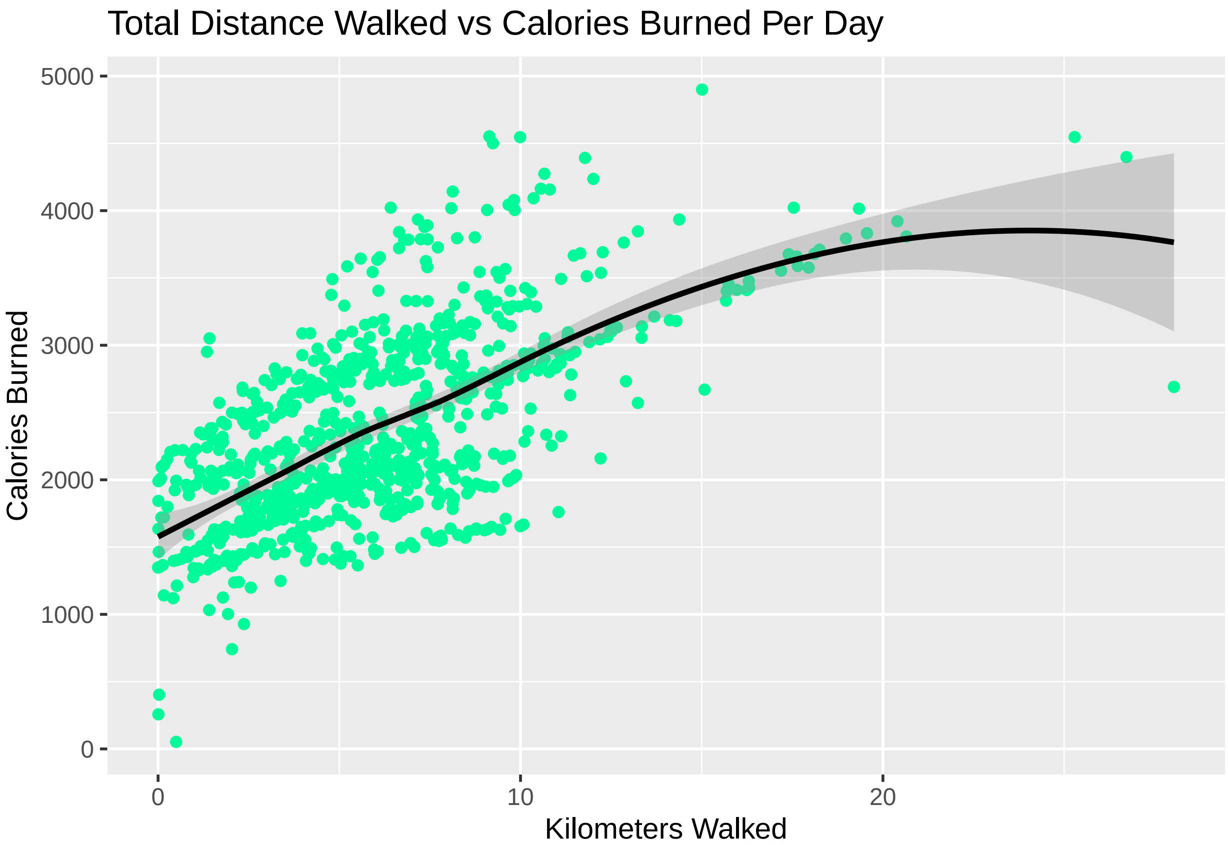
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Distance vs calories burned

```
ggplot(data=daily_activity) +
  geom_point(mapping=aes(x=TotalDistanceKm,y=Calories),
             color="mediumspringgreen") +
  geom_smooth(mapping=aes(x=TotalDistanceKm,y=Calories),color="black") +
  labs(title="Total Distance Walked vs Calories Burned Per Day",
       x="Kilometers Walked",y="Calories Burned")
```

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


Total Distance Walked vs Calories Burned Per Day
Calories Burned
5000
4000
3000
2000
1000
0
0
10
20
Kilometers Walked