Analysis of FitBit Fitness Tracker Data for Bellabeat

Wadi Wahyudin

2024-06-20

This is a dataset about FitBit Fitness Tracker Data for Bellabeat, if you want to know me you can click this link > Personal Site I have experience in:

- Data Analyst
- Mobile Developer
- Game Developer
- Web Developer

Okay, to the next step!!!

```r
library(tidyverse)
```

Installing and loading common packages and libraries

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

Load CSV files

Here we'll create a dataframe named 'daily_activity' and read in one of the CSV files from the dataset.

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

Create another dataframe for the sleep data.

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

Create another dataframe for weight_log.

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

Create another dataframe for daily__intensities.

```
 hourly_intensities <- read.csv("hourlyIntensities_merged.csv")
```

Create another dataframe for daily__calories.

```
 hourly_calories <- read.csv("hourlyCalories_merged.csv")
```

Create another dataframe for daily__steps.

```
 hourly_steps <- read.csv("hourlySteps_merged.csv")
```

Exploring a few key tables

Take a look at the daily_activity data.

```
as_tibble(daily_activity)
```

```
## # A tibble: 940 x 15
##            Id ActivityDate TotalSteps TotalDistance TrackerDistance
##         <dbl> <chr>             <int>         <dbl>           <dbl>
##  1 1503960366 4/12/2016         13162          8.5             8.5
##  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
##  7 1503960366 4/18/2016         13019          8.59            8.59
##  8 1503960366 4/19/2016         15506          9.88            9.88
##  9 1503960366 4/20/2016         10544          6.68            6.68
## 10 1503960366 4/21/2016          9819          6.34            6.34
## # i 930 more rows
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <int>, FairlyActiveMinutes <int>,
## #   LightlyActiveMinutes <int>, SedentaryMinutes <int>, Calories <int>
```

Take a look at the weight__log data.

```
as_tibble(weight_log)
```

```
## # A tibble: 67 x 7
##            Id Date                  WeightKg WeightPounds   BMI IsManualReport LogId
##         <dbl> <chr>                    <dbl>        <dbl> <dbl> <lgl>          <chr>
##  1 1503960366 05/02/2016 23:59     52.6          116.   22.6 TRUE           1,46E~
##  2 1503960366 05/03/2016 23:59     52.6          116.   22.6 TRUE           1,46E~
##  3 1927972279 04/13/2016 1:08     134.           294.   47.5 FALSE          1,46E~
##  4 2873212765 04/21/2016 23:59     56.7          125    21.4 TRUE           1,46E~
##  5 2873212765 05/12/2016 23:59     57.3         126.  21.7 TRUE           1,46E~
##  6 4319703577 04/17/2016 23:59     72.4         160.  27.4 TRUE           1,46E~
##  7 4319703577 05/04/2016 23:59     72.3         159.  27.4 TRUE           1,46E~
##  8 4558609924 04/18/2016 23:59     69.7         154.  27.2 TRUE           1,46E~
##  9 4558609924 04/25/2016 23:59     70.3         155.  27.5 TRUE           1,46E~
## 10 4558609924 05/01/2016 23:59     69.9         154.  27.3 TRUE           1,46E~
## # i 57 more rows
```

Take a look at the sleep_day data.

```
as_tibble(sleep_day)
```

```
## # A tibble: 413 x 5
##            Id SleepDay       TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##         <dbl> <chr>                      <int>              <int>          <int>
##  1 1503960366 4/12/2016 12:~                 1                327            346
##  2 1503960366 4/13/2016 12:~                 2                384            407
##  3 1503960366 4/15/2016 12:~                 1                412            442
##  4 1503960366 4/16/2016 12:~                 2                340            367
##  5 1503960366 4/17/2016 12:~                 1                700            712
##  6 1503960366 4/19/2016 12:~                 1                304            320
##  7 1503960366 4/20/2016 12:~                 1                360            377
##  8 1503960366 4/21/2016 12:~                 1                325            364
##  9 1503960366 4/23/2016 12:~                 1                361            384
## 10 1503960366 4/24/2016 12:~                 1                430            449
## # i 403 more rows
```

Take a look at the daily_intensities data.

```
as_tibble(hourly_intensities)
```

```
## # A tibble: 12,455 x 4
##            Id ActivityHour       TotalIntensity AverageIntensity
##         <dbl> <chr>                       <int>            <dbl>
##  1 1503960366 05/01/2016 0:00:00              9             0.15
##  2 1624580081 05/01/2016 0:00:00              1            16.7
##  3 1844505072 05/01/2016 0:00:00             39             0.65
##  4 2320127002 05/01/2016 0:00:00             15             0.25
##  5 3372868164 05/01/2016 0:00:00              2            33.3
##  6 4388161847 05/01/2016 0:00:00              1            16.7
##  7 4445114986 05/01/2016 0:00:00              8           133.
##  8 6775888955 05/01/2016 0:00:00             29           483.
##  9 6962181067 05/01/2016 0:00:00              1            16.7
## 10 7086361926 05/01/2016 0:00:00              5            83.3
## # i 12,445 more rows
```

Take a look at the daily_calories data.

```
as_tibble(hourly_calories)
```

```
## # A tibble: 22,099 x 3
##            Id ActivityHour          Calories
##         <dbl> <chr>                    <int>
##  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
##  7 1503960366 4/12/2016 6:00:00 AM        48
##  8 1503960366 4/12/2016 7:00:00 AM        47
##  9 1503960366 4/12/2016 8:00:00 AM        68
## 10 1503960366 4/12/2016 9:00:00 AM       141
## # i 22,089 more rows
```

Take a look at the daily_steps data.

```
as_tibble(hourly_steps)
```

```
## # A tibble: 12,802 x 3
##            Id ActivityHour       StepTotal
##         <dbl> <chr>                  <int>
##  1 1503960366 4/12/2016 0:00:00        373
##  2 1503960366 4/12/2016 1:00:00        160
##  3 1503960366 4/12/2016 2:00:00        151
##  4 1503960366 4/12/2016 8:00:00        250
##  5 1503960366 4/12/2016 9:00:00       1864
##  6 1503960366 4/12/2016 10:00:00       676
##  7 1503960366 4/12/2016 11:00:00       360
##  8 1503960366 4/12/2016 12:00:00       253
##  9 1503960366 4/12/2016 13:00:00       221
## 10 1503960366 4/12/2016 14:00:00      1166
## # i 12,792 more rows
```

Identify all the columsn in the daily_activity data.

```
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"
```

Take a look at the sleep_day data.

```
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
```

Identify all the columsn in the daily_activity data.

```
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"
```

Identify all the columsn in the sleep_day data.

```
colnames(sleep_day)
```

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

Identify all the columsn in the weight_log data.

```
colnames(weight_log)
```

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

Identify all the columsn in the daily_intensities data.

```
colnames(hourly_intensities)
```

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

Identify all the columsn in the daily_calories data.

```
colnames(hourly_calories)
```

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

Identify all the columsn in the daily_steps data.

```
colnames(hourly_steps)
```

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

Understanding some summary statistics

See how many unique participants are there in each dataframe

```
n_distinct(daily_activity$Id)
```

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

```
n_distinct(sleep_day$Id)
```

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

```
n_distinct(weight_log$Id)
```

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

```
n_distinct(hourly_intensities$Id)
```

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

```
n_distinct(hourly_calories$Id)
```

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

```
n_distinct(hourly_steps$Id)
```

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

See how many observations are there in each dataframe

```
nrow(daily_activity)
```

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

```
nrow(sleep_day)
```

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

```
nrow(weight_log)
```

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

```
nrow(hourly_intensities)
```

```
## [1] 12455
```

```
nrow(hourly_calories)
```

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

```
nrow(hourly_steps)
```

```
## [1] 12802
```

Summary

See what are some quick summary statistics about each data frame

For the daily activity dataframe:

```
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
```

For the sleep dataframe:

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

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

For the weight log dataframe:

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

```
##     WeightKg       WeightPounds        BMI
##  Min.   : 52.60   Min.   :116.0   Min.   :21.45
##  1st Qu.: 61.40   1st Qu.:135.4   1st Qu.:23.96
##  Median : 62.50   Median :137.8   Median :24.39
##  Mean   : 72.04   Mean   :158.8   Mean   :25.19
##  3rd Qu.: 85.05   3rd Qu.:187.5   3rd Qu.:25.56
##  Max.   :133.50   Max.   :294.3   Max.   :47.54
```

Cleaning data for hourly_intensities:

```
# Identify non-numeric values in AverageIntensity and replace them with NA
hourly_intensities <- hourly_intensities %>%
  mutate(AverageIntensity = as.numeric(AverageIntensity))

# Calculate the mean of the valid numeric values in AverageIntensity
mean_intensity <- mean(hourly_intensities$AverageIntensity, na.rm = TRUE)

# Replace NA values in AverageIntensity with the mean value
hourly_intensities <- hourly_intensities %>%
  mutate(AverageIntensity = ifelse(is.na(AverageIntensity), mean_intensity, AverageIntensity))

# Confirm the replacement
sum(is.na(hourly_intensities$AverageIntensity))
```

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

```
# Select the relevant columns and display summary statistics
hourly_intensities %>%
  select(TotalIntensity, AverageIntensity) %>%
  summary()
```

```
##  TotalIntensity  AverageIntensity
##  Min.   :  1.0   Min.   :  0.05
##  1st Qu.:  5.0   1st Qu.:  0.50
##  Median : 13.0   Median : 83.33
##  Mean   : 17.2   Mean   :169.84
##  3rd Qu.: 23.0   3rd Qu.:283.33
##  Max.   :180.0   Max.   :983.33
```

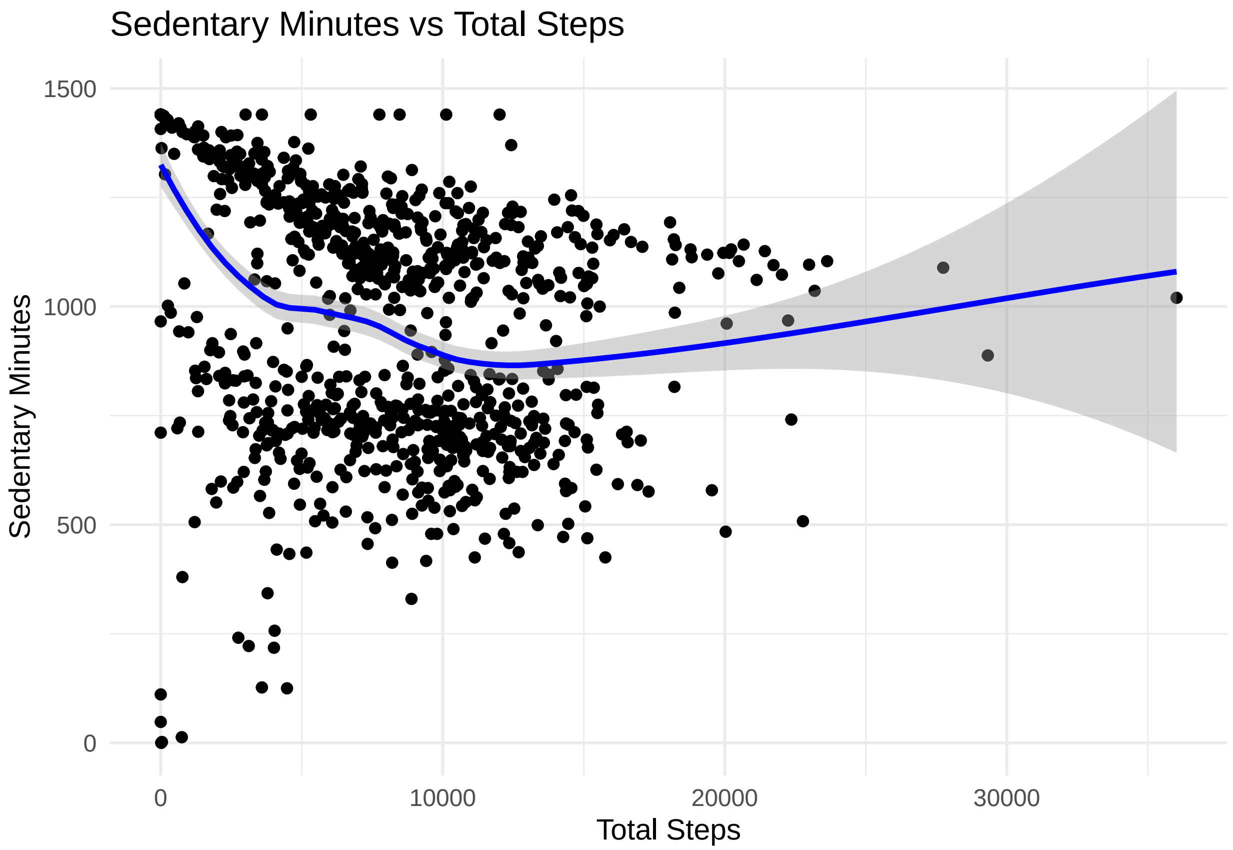
What does this tell us about how this sample of people's activities?

Plotting a few explorations

What's the relationship between steps taken in a day and sedentary minutes? How could this help inform the customer segments that we can market to? E.g. position this more as a way to get started in walking more? Or to measure steps that you're already taking?

```
# Create the scatter plot with axis labels and a smoothing line
ggplot(data=daily_activity, aes(x=TotalSteps, y=SedentaryMinutes)) +
  geom_point() +
  geom_smooth(method="loess", se=TRUE, color="blue") +
  labs(
    title = "Sedentary Minutes vs Total Steps",
    x = "Total Steps",
    y = "Sedentary Minutes"
  ) +
  theme_minimal()
```

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



Check the corelation between TotalSteps and SedentaryMinutes

```
cor.test(daily_activity$TotalSteps, daily_activity$SedentaryMinutes)
```

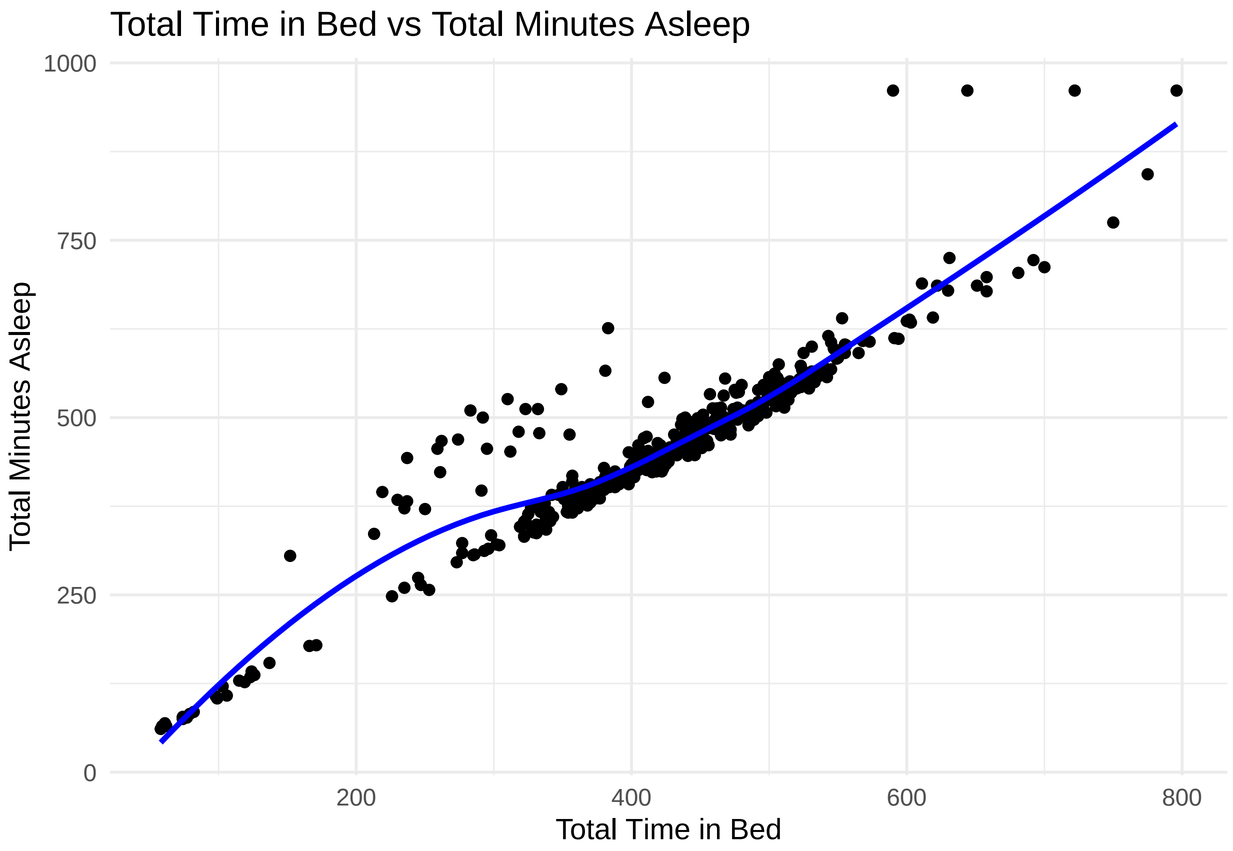
```
##
##  Pearson's product-moment correlation
##
## data:  daily_activity$TotalSteps and daily_activity$SedentaryMinutes
## t = -10.615, df = 938, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3833971 -0.2691782
## sample estimates:
##        cor
## -0.3274835
```

Description: The Pearson's product-moment correlation analysis between TotalSteps and SedentaryMinutes in daily_activity indicates a statistically significant negative correlation (cor = -0.3274835, p < 2.2e-16). This negative correlation suggests that as the number of total steps increases, the time spent in sedentary activities decreases. In other words, higher physical activity (measured in steps) is associated with less sedentary behavior in the dataset.

See what's the relationship between minutes asleep and time in bed

```
# Create the scatter plot with axis labels and a smoothing line
ggplot(data=sleep_day, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) +
  geom_point() +
  geom_smooth(method="loess", se=FALSE, color="blue") +
  labs(
    title = "Total Time in Bed vs Total Minutes Asleep",
    x = "Total Time in Bed",
    y = "Total Minutes Asleep"
  ) +
  theme_minimal()
```

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



Check the corelation between TotalMinutesAsleep and TotalTimeInBed

```
cor.test(sleep_day$TotalMinutesAsleep, sleep_day$TotalTimeInBed)
```

```
##
##  Pearson’s product-moment correlation
##
## data:  sleep_day$TotalMinutesAsleep and sleep_day$TotalTimeInBed
## t = 51.483, df = 411, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9162253 0.9423445
## sample estimates:
##       cor
## 0.9304575
```

Description: The Pearson’s product-moment correlation analysis between `TotalMinutesAsleep` and `TotalTimeInBed` in `sleep_day` shows a very strong positive correlation (cor = 0.9304575, p < 2.2e-16). This indicates that there is a high correlation between the total time spent asleep and the total time spent in bed. In other words, individuals who spend more time in bed tend to have longer periods of actual sleep, suggesting good alignment between time spent in bed and actual sleep duration in the dataset.

See what could these trends tell about how to help market this product Or areas where might want to explore further

```
cor.test(sleep_day$TotalMinutesAsleep, sleep_day$TotalTimeInBed,
 alternative = "greater")
```

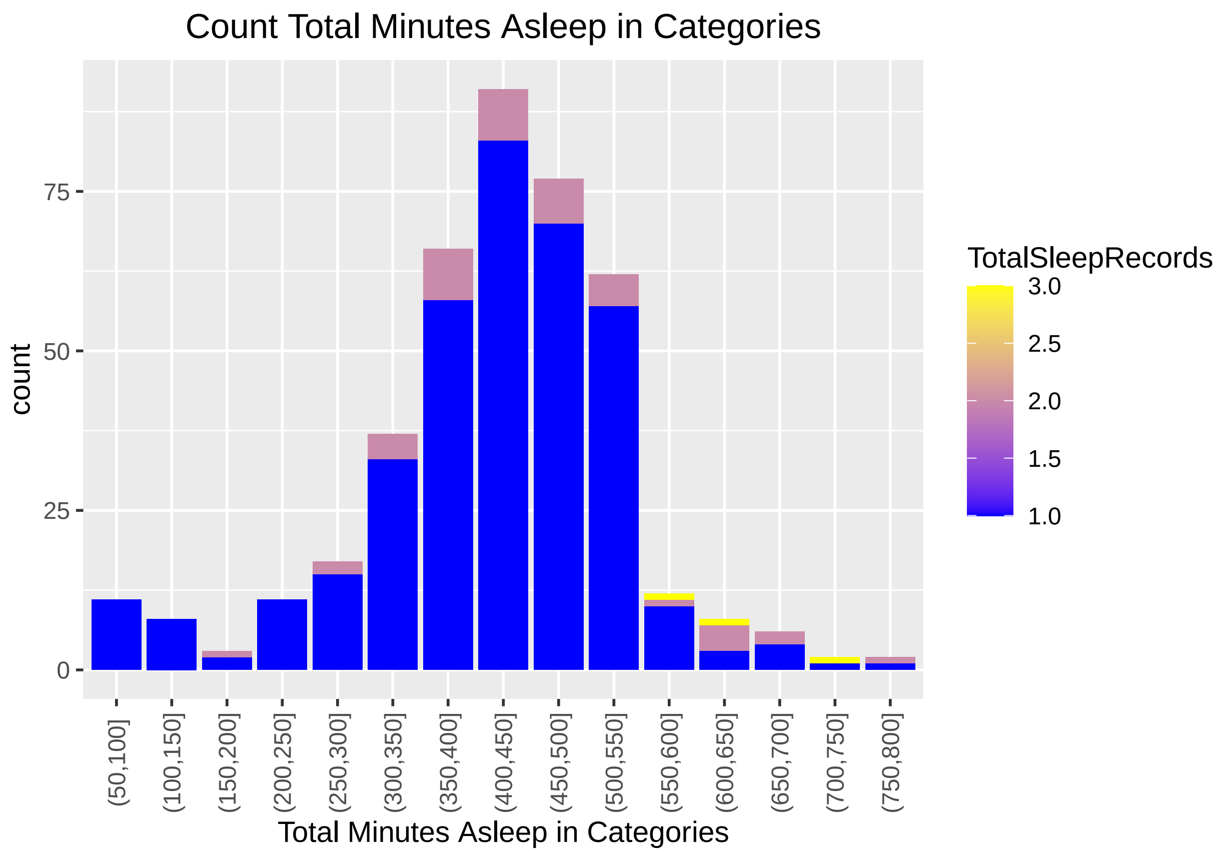
```
##
##  Pearson’s product-moment correlation
##
## data:  sleep_day$TotalMinutesAsleep and sleep_day$TotalTimeInBed
## t = 51.483, df = 411, p-value < 2.2e-16
## alternative hypothesis: true correlation is greater than 0
## 95 percent confidence interval:
##  0.9186882 1.0000000
## sample estimates:
##       cor
## 0.9304575
```

Description: Positive and strong correlation (close to 1) as expected.

Time Asleep & Total Sleep Records

```
 sleep_day$asleep_categories <- cut(sleep_day$TotalMinutesAsleep, seq(from = 0, to = 800, by = 50))
 sleep_day %>%
 group_by(asleep_categories,TotalSleepRecords) %>%
 summarise(count = n()) %>%
 ggplot(aes(x = asleep_categories, y = count, fill = TotalSleepRecords)) +
 geom_bar(position= "stack",stat="identity") +
 scale_fill_gradient(low = "blue", high = "yellow") +
 labs(x = "Total Minutes Asleep in Categories", title = "Count Total Minutes Asleep in Categories")+
 theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(vjust = 0.5, angle = 90))
```

```
## ‘summarise()‘ has grouped output by ’asleep_categories’. You can override using
## the ‘.groups‘ argument.
```



Merging these two datasets together

```
combined_data <- right_join(sleep_day, daily_activity, by = "Id")
```

```
## Warning in right_join(sleep_day, daily_activity, by = "Id"): Detected an unexpected many-to-many rela
## i Row 1 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 1 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

Take a look at how many participants are in this data set.

```
n_distinct(combined_data$Id)
```

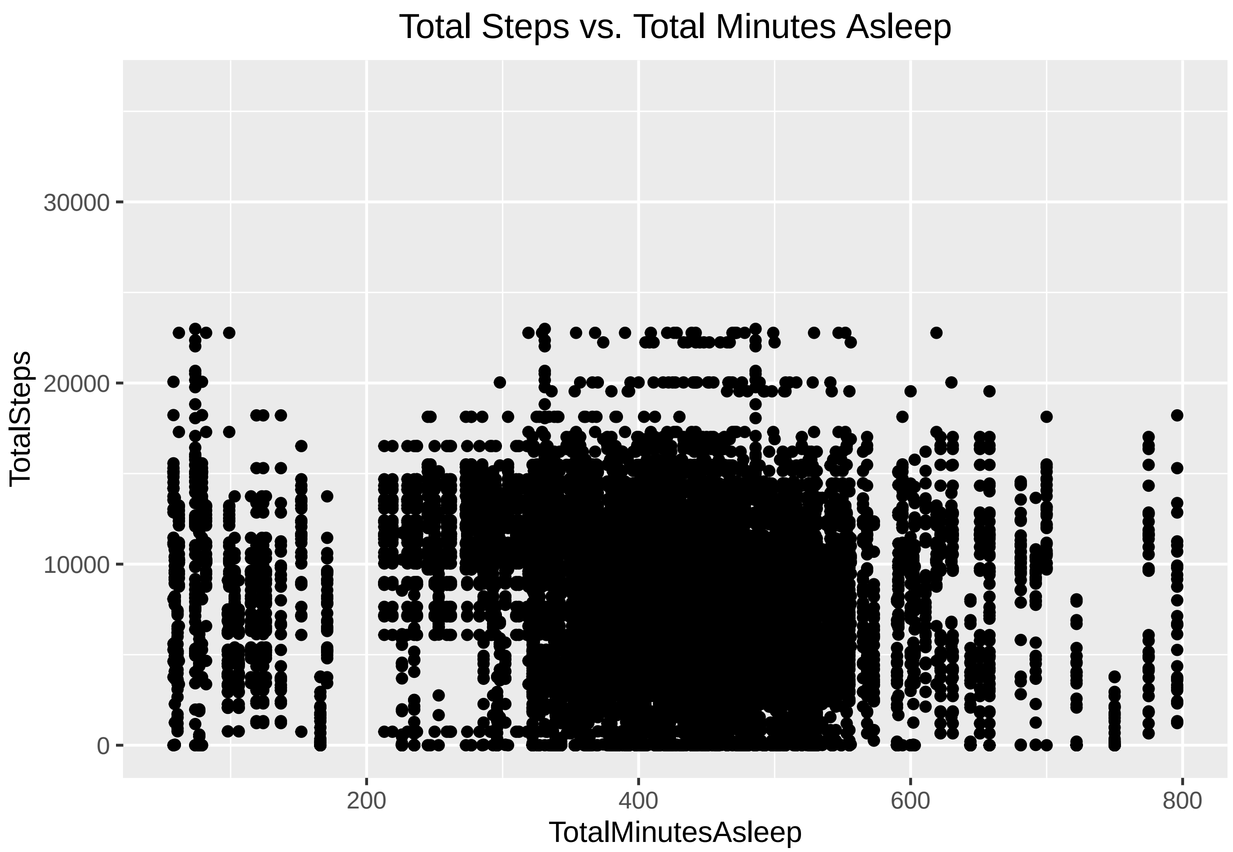
```
## [1] 33
```

Exploring Relationships Between Activity and Sleep

Let’s explore if participants who sleep more also take more steps or fewer steps per day.

```
ggplot(data = combined_data, aes(x = TotalMinutesAsleep, y = TotalSteps)) +
 geom_point() + labs(title = "Total Steps vs. Total Minutes Asleep") +
 theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Removed 227 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



Check the correlation

```
cor.test(combined_data$TotalMinutesAsleep, combined_data$TotalSteps)
```

```
##
##  Pearson's product-moment correlation
##
## data:  combined_data$TotalMinutesAsleep and combined_data$TotalSteps
## t = -11.044, df = 12439, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.11591302 -0.08110962
## sample estimates:
##         cor
## -0.09854146
```

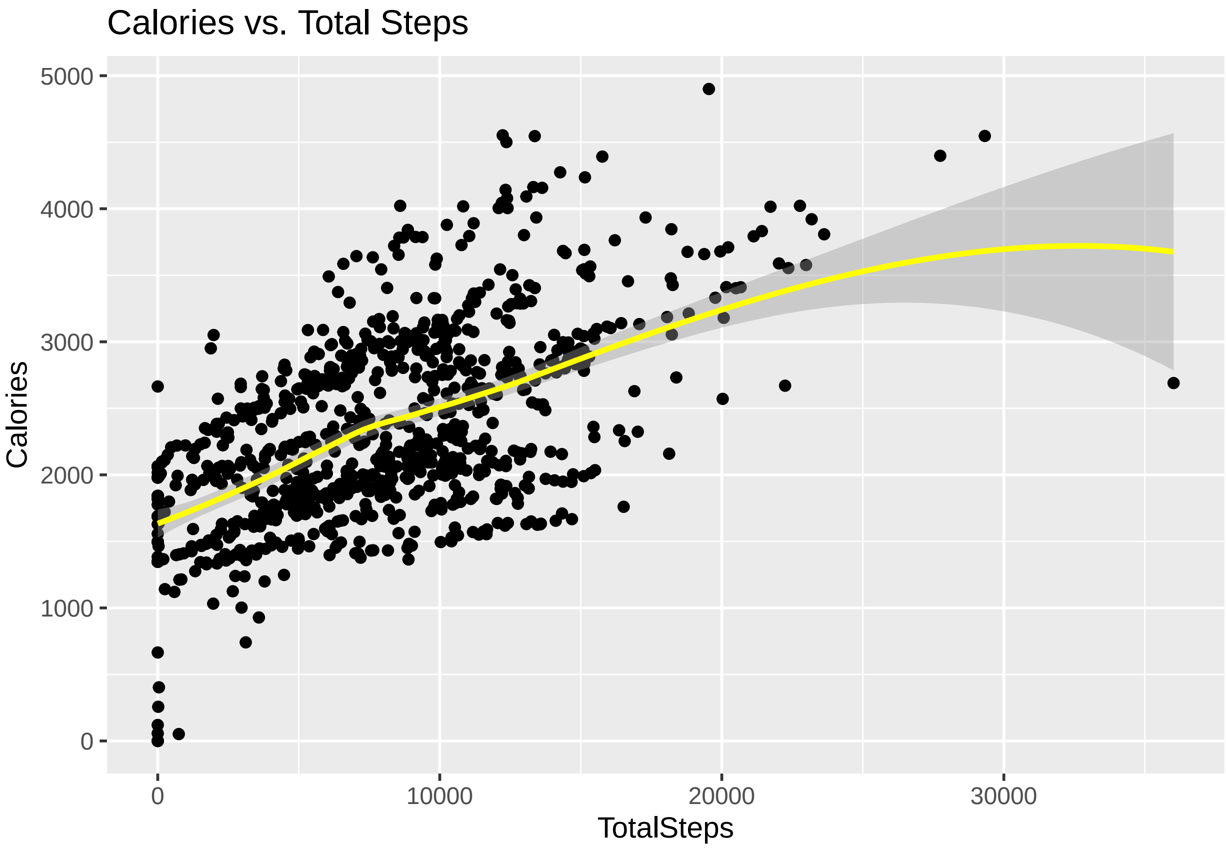
Description: There is a statistically significant negative correlation between daily step count (TotalSteps) and total minutes asleep (TotalMinutesAsleep). The correlation coefficient of -0.09854 indicates that individuals who sleep more tend to take fewer steps per day, and vice versa. Statistical analysis shows that this difference is not due to random chance, with a very low p-value (< 2.2e-16), allowing us to reject the null hypothesis that there is no correlation in the population. The 95% confidence interval for the correlation is between -0.11591302 and -0.08110962, indicating with 95% confidence that the population correlation lies within this range.

Thus, these findings suggest that shorter or longer sleep durations correlate with different levels of daily physical activity, which could have significant implications for developing marketing strategies for health and fitness-related products.

Calories vs Total Step

```
ggplot(data = daily_activity, aes(x = TotalSteps, y = Calories)) + geom_point() +
geom_smooth(method = "loess",color = "yellow") + labs(title = "Calories vs. Total Steps") + theme()
```

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



Check the corelation between Calories & Total Steps

```
cor.test(daily_activity$TotalSteps, daily_activity$Calories,
 alternative = "greater")
```

```
##
##  Pearson’s product-moment correlation
##
## data:  daily_activity$TotalSteps and daily_activity$Calories
## t = 22.472, df = 938, p-value < 2.2e-16
## alternative hypothesis: true correlation is greater than 0
## 95 percent confidence interval:
##  0.5555268 1.0000000
## sample estimates:
##       cor
## 0.5915681
```

Description: There is a statistically significant positive correlation between the total number of steps taken per day (TotalSteps) and the number of calories burned (Calories). The correlation coefficient of 0.5915681 indicates a moderate to strong positive relationship, suggesting that as the number of steps increases, the number of calories burned also increases. This correlation is statistically significant with a very low p-value (< 2.2e-16), allowing us to confidently reject the null hypothesis that there is no correlation in the population. The 95% confidence interval for the correlation ranges from 0.5555268 to 1.0000000, indicating with 95% confidence that the population correlation lies within this range.

These findings suggest that physical activity, as measured by steps taken, is closely linked to energy expenditure, as measured by calories burned. This insight could be valuable for developing targeted health and fitness programs or marketing strategies for products aimed at increasing physical activity and promoting weight loss or fitness.

Time of Intensities

```
# Add Date column formatted from ActivityHour
hourly_intensities$Date <- format(as.Date(hourly_intensities$ActivityHour, format = "%m/%d/%Y"), format

# Parse ActivityHour column to datetime format
hourly_intensities$ActivityHour <- mdy_hms(hourly_intensities$ActivityHour, tz = Sys.timezone())

# Add Time column formatted from ActivityHour
hourly_intensities$Time <- format(hourly_intensities$ActivityHour, format = "%H:%M:%S")

# Add day_of_week column derived from ActivityHour
hourly_intensities$day_of_week <- format(as.Date(hourly_intensities$ActivityHour), "%A")
```

Adds new columns to the hourly_intensities dataset to enhance temporal analysis

```
hourly_intensities$day_of_week <- ordered(hourly_intensities$day_of_week, levels=c("Sunday", "Monday",
```

Orders the day_of_week column in the hourly_intensities dataset based on predefined levels

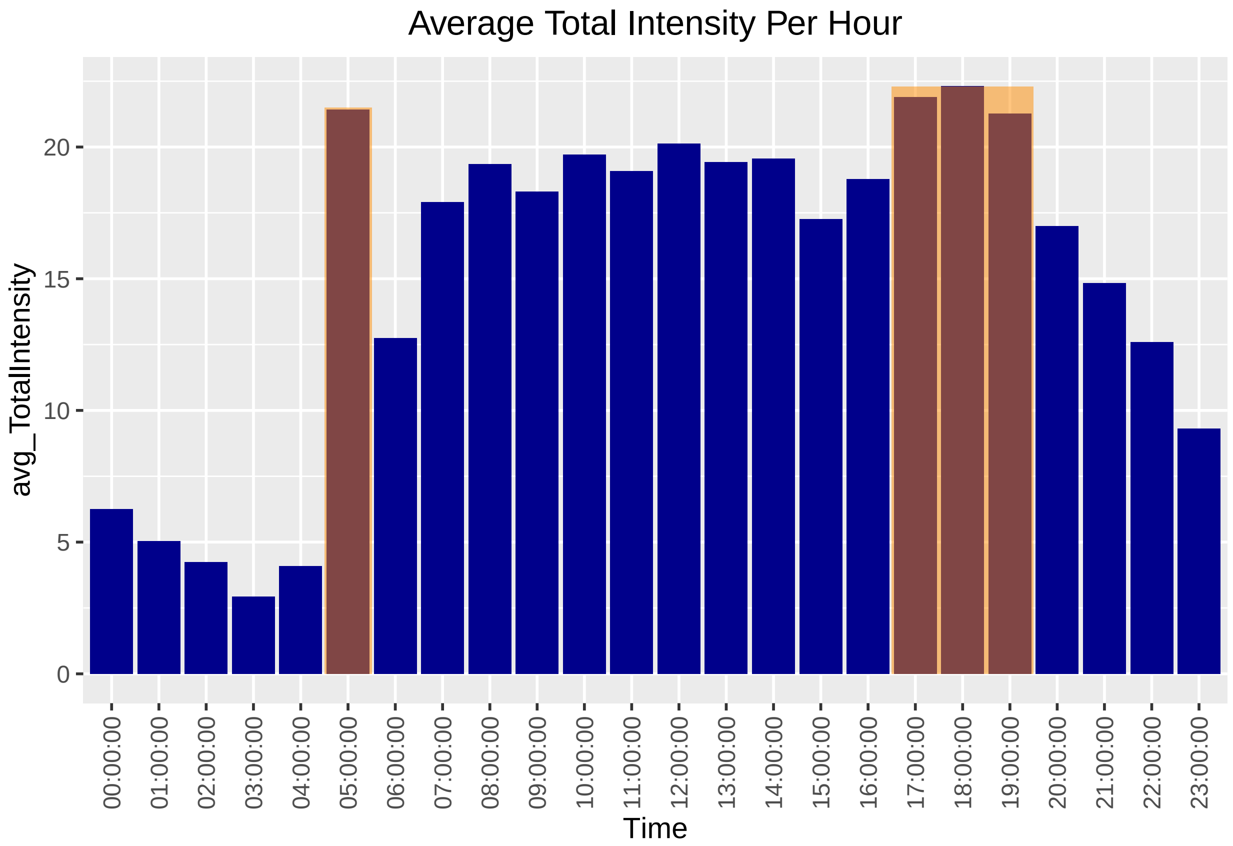
```
extract_data <- hourly_intensities[, c(3,6)]
plot_data <- extract_data %>%
  group_by(Time) %>%
  summarise(avg_TotalIntensity = mean(TotalIntensity))

extract_data2 <- hourly_intensities[, c(3,7)]
plot_data2 <- extract_data2 %>%
  group_by(day_of_week) %>%
  summarise(avg_TotalIntensity = mean(TotalIntensity))
```

Prepare data for plotting average total intensity over time and by day of the week

```
ggplot(data = plot_data, aes(x = Time, y = avg_TotalIntensity)) +
  geom_bar(stat = "identity", fill = "darkblue") +
  labs(title = "Average Total Intensity Per Hour") +
  annotate("rect", xmin = 17.5, ymin = 0, xmax = 20.5, ymax = 22.3,
           fill = "darkorange", alpha = 0.5) + annotate("rect", xmin = 5.5, ymin = 0, xmax = 6.5, ymax =
           fill = "darkorange", alpha = 0.5) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5),
        plot.title = element_text(hjust = 0.5))
```

creates a bar plot using ggplot2 to visualize the average total intensity per hour (avg_TotalIntensity) over different times (Time)



```
#ggsave("time_total_intensity.jpg")
```

Description:

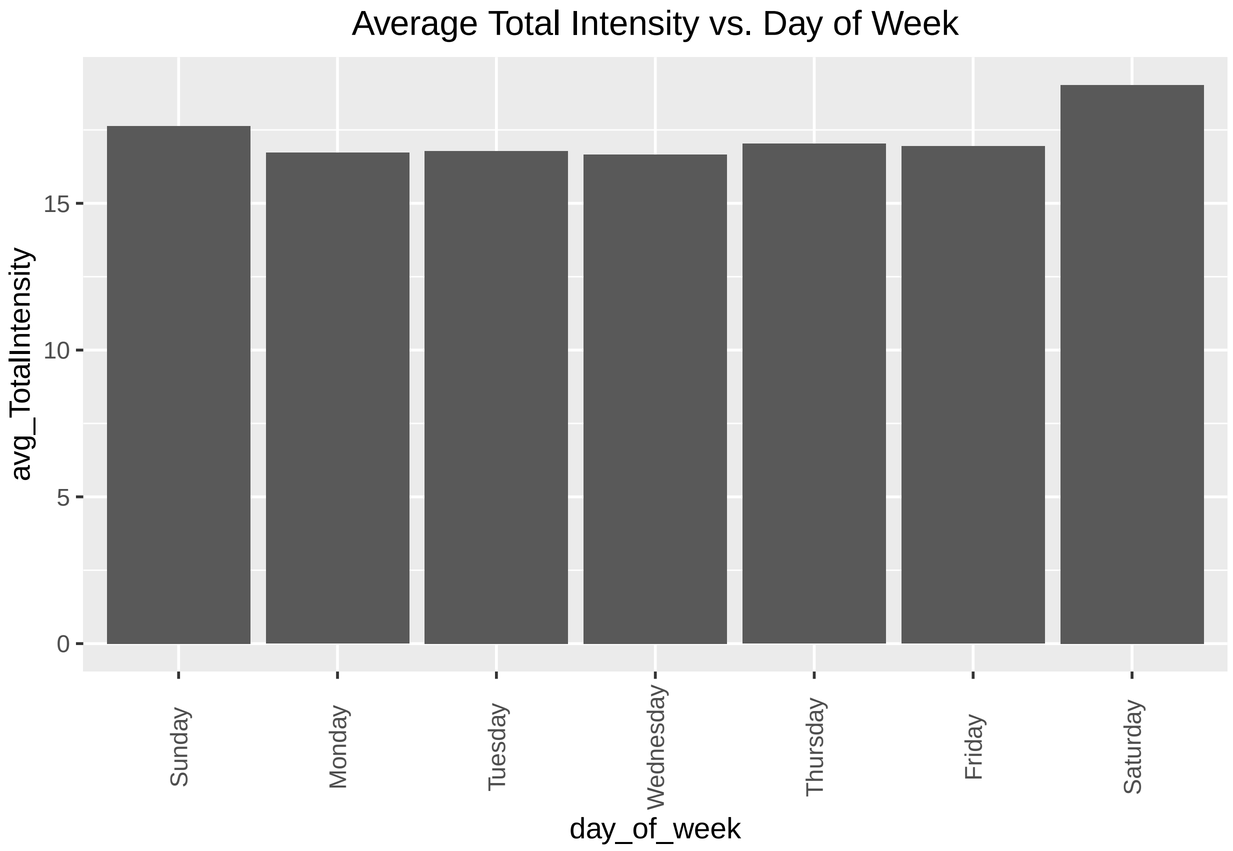
- The average hourly total intensity varied between 5 and 20 minutes per hour.
- The highest hourly total intensity occurred between 05:00, 17:00, 18:00 and 19:00.
- The lowest hourly total intensity occurred between 01:00:00 and 04:00:00.

```
str(hourly_intensities)
```

Average of Total Intensity vs. Days of Week

```
## 'data.frame':    12455 obs. of  7 variables:
##  $ Id              : num  1.50e+09 1.62e+09 1.84e+09 2.32e+09 3.37e+09 ...
##  $ ActivityHour    : POSIXct, format: "2016-05-01 00:00:00" "2016-05-01 00:00:00" ...
##  $ TotalIntensity  : int  9 1 39 15 2 1 8 29 1 5 ...
##  $ AverageIntensity: num  0.15 16.67 0.65 0.25 33.33 ...
##  $ Date            : chr  "05/01/2016" "05/01/2016" "05/01/2016" "05/01/2016" ...
##  $ Time            : chr  "00:00:00" "00:00:00" "00:00:00" "00:00:00" ...
##  $ day_of_week     : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 7 7 7 7 7 7 7 7 7 7 ...
```

```
ggplot(data = plot_data2, aes(x = day_of_week, y = avg_TotalIntensity)) +
  geom_bar(stat = "identity") + labs(title = "Average Total Intensity vs. Day of Week") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5),
        plot.title = element_text(hjust = 0.5))
```



Description: Average exercise intensity is highest on Tuesdays and Wednesdays, and lowest on Saturdays. There is a general trend of decreasing intensity from Tuesday to Saturday. The intensity on Sunday and Friday is similar to the intensity on Monday. Possible explanations:

People may be more likely to exercise on weekdays because they have more time and fewer distractions. People may exercise more intensely on Tuesdays and Wednesdays because they are preparing for the weekend. *People may exercise less intensely on Saturdays because they are tired from the week.

Conclusion

In this analysis, exploration of data from multiple datasets related to daily activities, sleep, weight, daily intensities, daily calories, and daily steps using FitBit Fitness Tracker Data for Bellabeat has been conducted. Here are the key findings and conclusions drawn:

1. Conclusion from Daily Activity Analysis:

- Participants in the dataset have an average Total Steps of approximately 7,638 with a Total Distance of about 5.49 km.
- The average time spent in sedentary activities is around 991 minutes.
- There is a significant negative correlation between Total Steps and Sedentary Minutes (cor = -0.3275, $p < 2.2e\text{-}16$), indicating that more steps correlate with less time spent sitting.

2. Conclusion from Sleep Data Analysis:

- Participants have an average Total Minutes Asleep of around 419.5 and Total Time in Bed of about 458.6 minutes.
- There is a very strong positive correlation between Total Minutes Asleep and Total Time in Bed (cor = 0.9305, $p < 2.2e-16$), indicating a strong relationship between time spent in bed and actual sleep duration.

3. Conclusion from Weight Log Analysis:

- The average weight of participants is approximately 72.04 kg with a BMI around 25.19.
- Weight and BMI vary among participants, with significant minimum and maximum values (min weight 52.6 kg, max weight 133.5 kg; min BMI 21.45, max BMI 47.54).

4. Conclusion from Hourly Intensities, Calories, and Steps Analysis:

- Daily intensities show significant variation, with an average Total Intensity of approximately 17.2 and Average Intensity around 169.84.
- Daily calories burned also vary, with an average of about 22,099 calories.
- Average daily steps are around 12,802, indicating diverse activity levels among participants.

5. Considerations for Marketing and Further Exploration:

- The correlation between Total Steps and Sedentary Minutes suggests that FitBit could be positioned as a tool to reduce sedentary time by encouraging more movement among participants.
- The strong correlation between Total Minutes Asleep and Total Time in Bed indicates that the product can promote better sleep benefits by monitoring time spent in bed.

6. Rekomendasi untuk Langkah Selanjutnya:

- Conduct further analysis to understand individual behavioral patterns based on variables such as daily intensities, calories, and steps to support more effective marketing strategies.
- Integrate data from various datasets for a more comprehensive understanding of the relationship between physical activity, sleep, and weight.

This analysis provides robust insights into participant behavior patterns using FitBit, offering a solid foundation for more effective marketing strategies and future product development.