Fitbit Data Analysis/Viz

Josh Huang

2023-09-08

Installing and loading packages for analysis

These packages are mostly installed for the purpose of accessing functions that help clean, sort, filter, and visualize the data used within an analysis process.

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dz/j8x5jwvj1wsbzwlzl_pcyz4w0000gn/T//RtmpNqCovX/downloaded_packages
```

```
install.packages("ggplot2", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dz/j8x5jwvj1wsbzwlzl_pcyz4w0000gn/T//RtmpNqCovX/downloaded_packages
```

```
install.packages("readr", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/dz/j8x5jwvj1wsbzwlzl_pcyz4w0000gn/T//RtmpNqCovX/downloaded_packages
```

```
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 1.9.2     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 errors
```

```r
library(ggplot2)
library(readr)
library(dplyr)
```

Setting up the directory

Used to help establish the location for which the files will be retrieved.

```r
getwd()
```

```
## [1] "/Users/josh7/Documents/CAPSTONE/Fitbit_Data_4.12.16-5.12.16"
```

```r
setwd("/Users/josh7/Documents/CAPSTONE/Fitbit_Data_4.12.16-5.12.16")
```

Importing files for analysis

Using the readr function to assign variable names to each individual csv file.

```r
Daily_Activity <- read_csv("dailyActivity_merged.csv")
```

```
## 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.
```

```r
head(Daily_Activity)
```

```
## # A tibble: 6 x 15
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
##        <dbl> <chr>             <dbl>         <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
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

```r
Daily_Calories <- read_csv("dailyCalories_merged.csv")
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, Calories
##
## 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.
```

```
Daily_Intensities <- read_csv("dailyIntensities_merged.csv")
```

```
## Rows: 940 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (9): Id, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, Ve...
##
## 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.
```

```
Daily_Steps <- read_csv("dailySteps_merged.csv")
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, StepTotal
##
## 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.
```

```
Heartrate_Seconds <- read_csv("heartrate_seconds_merged.csv")
```

```
## Rows: 2483658 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Time
## dbl (2): Id, Value
##
## 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.
```

```
Hourly_Calories <- read_csv("hourlyCalories_merged.csv")
```

```
## Rows: 22099 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, Calories
##
## 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.
```

```
Hourly_Intensities <- read_csv("hourlyIntensities_merged.csv")
```

```
## Rows: 22099 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
##
## 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.
```

```
Hourly_Steps <- read_csv("hourlySteps_merged.csv")
```

```
## Rows: 22099 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (2): Id, StepTotal
##
## 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.
```

```
Minute_Calories_Narrow <- read_csv("minuteCaloriesNarrow_merged.csv")
```

```
## Rows: 1325580 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Calories
##
## 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.
```

```
Minute_Calories_Wide <- read_csv("minuteCaloriesWide_merged.csv")
```

```
## Rows: 21645 Columns: 62
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityHour
## dbl (61): Id, Calories00, Calories01, Calories02, Calories03, Calories04, Ca...
##
## 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.
```

```
Minute_METs_Narrow <- read_csv("minuteMETsNarrow_merged.csv")
```

```
## Rows: 1325580 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityMinute
```

```
## dbl (2): Id, METs
##
## 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.
```

```
Minute_Sleep <- read_csv("minuteSleep_merged.csv")
```

```
## Rows: 188521 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): date
## dbl (3): Id, value, logId
##
## 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.
```

```
Minute_Steps_Narrow <- read_csv("minuteStepsNarrow_merged.csv")
```

```
## Rows: 1325580 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityMinute
## dbl (2): Id, Steps
##
## 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.
```

```
Minute_Steps_Wide <- read_csv("minuteStepsWide_merged.csv")
```

```
## Rows: 21645 Columns: 62
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityHour
## dbl (61): Id, Steps00, Steps01, Steps02, Steps03, Steps04, Steps05, Steps06,...
##
## 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_Day <- read_csv("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_Log_Info <- read_csv("weightLogInfo_merged.csv")
```

```
## 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.
```

Quick overview of data

head() function returns the first six rows of the dataset.

```
head(Daily_Activity)
```

```
## # A tibble: 6 x 15
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
##        <dbl> <chr>             <dbl>         <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
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

Check for missing values

```
sum(is.na(Daily_Activity))
```

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

```
sum(is.na(Daily_Calories))
```

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

```
sum(is.na(Daily_Intensities))
```

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

```r
sum(is.na(Daily_Steps))
```

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

```r
sum(is.na(Heartrate_Seconds))
```

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

```r
sum(is.na(Hourly_Calories))
```

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

```r
sum(is.na(Hourly_Intensities))
```

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

```r
sum(is.na(Hourly_Steps))
```

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

```r
sum(is.na(Minute_Calories_Narrow))
```

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

```r
sum(is.na(Minute_Calories_Wide))
```

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

```r
sum(is.na(Minute_METs_Narrow))
```

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

```r
sum(is.na(Minute_Sleep))
```

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

```r
sum(is.na(Minute_Steps_Narrow))
```

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

```r
sum(is.na(Minute_Steps_Wide))
```

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

```r
sum(is.na(Sleep_Day))
```

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

```r
sum(is.na(Weight_Log_Info))
```

```
## [1] 65
```

Check for num of unique user ids

Using pipes indicated by "%>%" operator to link functions

```r
distinct_count <- Daily_Activity %>%
  select("Id") %>%
  n_distinct("Id")
View(distinct_count)
```

Check for duplicated values in this dataset

```r
sum(duplicated(Daily_Activity))
```

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

```r
sum(duplicated(Daily_Calories))
```

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

```r
sum(duplicated(Daily_Intensities))
```

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

```r
sum(duplicated(Daily_Steps))
```

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

```r
sum(duplicated(Heartrate_Seconds))
```

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

```r
sum(duplicated(Hourly_Calories))
```

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

```r
sum(duplicated(Hourly_Intensities)) # duplicates are normal in context
```

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

```r
sum(duplicated(Hourly_Steps))
```

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

```r
sum(duplicated(Minute_Calories_Narrow))
```

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

```r
sum(duplicated(Minute_Calories_Wide))
```

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

```r
sum(duplicated(Minute_METs_Narrow))
```

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

```r
sum(duplicated(Minute_Sleep)) # duplicates are normal in context
```

```
## [1] 543
```

```r
sum(duplicated(Minute_Steps_Narrow))
```

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

```r
sum(duplicated(Minute_Steps_Wide))
```

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

```r
sum(duplicated(Sleep_Day))
```

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

```r
sum(duplicated(Weight_Log_Info))
```

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

Removing the "Fat" column with missing values

Using pipes and select() function to remove the "Fat" column due to its significant lack of usable data

```
Weight_Log_Info <- Weight_Log_Info %>%
  select(-"Fat")
View(Weight_Log_Info)
```

Creating a new dataset without the duplicate values for "Sleep_Day"

unique() function used to ensure that duplicates are removed.

```
Sleep_Day <- unique(Sleep_Day)
View(Sleep_Day)
sum(duplicated(Sleep_Day))
```

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

Converting columns to date/time format

as.Date() function is used to convert incorrect data types such as char into the date data type. The as.POSIXct() is similar, but instead deals with formats that include the hours, minutes, and seconds in addition to the calendar date.

```
Daily_Activity$ActivityDate <- as.Date(Daily_Activity$ActivityDate, format = "%m/%d/%Y", tz=Sys.timezon
View(Daily_Activity)

Daily_Intensities$ActivityDay <- as.Date(Daily_Intensities$ActivityDay, format = "%m/%d/%Y", tz=Sys.tim
View(Daily_Intensities)

Hourly_Intensities$ActivityHour <- as.POSIXct(Hourly_Intensities$ActivityHour, format = "%m/%d/%Y %H:%M
View(Hourly_Intensities)

Daily_Steps$ActivityDay <- as.Date(Daily_Steps$ActivityDay, format = "%m/%d/%Y", tz=Sys.timezone())
View(Daily_Steps)

Sleep_Day$SleepDay <- as.POSIXct(Sleep_Day$SleepDay, format = "%m/%d/%Y %H:%M:%S", tz=Sys.timezone())
View(Sleep_Day)
```

KEY METRICS

These are used to gauge some interesting data points that can be used to inform potential areas of improvement for Bellabeat products.

Filtering data

```
Steps_Under_10K <- subset(Daily_Steps, subset = StepTotal < 10000)
View(Steps_Under_10K)
```

Finding proportion of daily activity with less than 10K steps

Finding the rows of the total and the desired metric helps illustrate the proportion.

```
nrow(Daily_Steps) # 940 total
```

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

```
nrow(Steps_Under_10K) # 637 under 10k
```

```
## [1] 637
```

```
# ~68% under 10K steps overall
```

Comparison of minutes asleep to time in bed

480 minutes is equivalent to the 8 hours of daily sleep recommended by health professionals.

```
Sleep_Under_8hrs <- subset(Sleep_Day, subset = TotalMinutesAsleep < 480)
View(Sleep_Under_8hrs)
```

Finding porportion of daily sleep under 8 hours

Finding the rows of the total and the desired metric helps illustrate the proportion.

```
nrow(Sleep_Day) # 410 total
```

```
## [1] 410
```

```
nrow(Sleep_Under_8hrs) # 294 under 8hrs
```

```
## [1] 294
```

```
# ~72% under 8hrs sleep overall
```

Finding average active minutes for daily intensity lvls (total = 1218 min daily avg)

Creating individual variables to store each subcategory of intensity levels.

```
Avg_Sed_Min <- mean(Daily_Intensities$SedentaryMinutes)
View(Avg_Sed_Min) # ~991 min daily avg, 81% of total min

Avg_Light_Act_Min <- mean(Daily_Intensities$LightlyActiveMinutes)
View(Avg_Light_Act_Min) # ~192 min daily avg, 16% of total min

Avg_Fair_Act_Min <- mean(Daily_Intensities$FairlyActiveMinutes)
```

```r
View(Avg_Fair_Act_Min) # ~14 min daily avg, ~1% of total min

Avg_Very_Act_Min <- mean(Daily_Intensities$VeryActiveMinutes)
View(Avg_Very_Act_Min) # ~21 min daily avg, ~2% of total min
```

Visualizations

Daily intensity avg pie chart

Summary of values for quick overview/comparison.

```r
Daily_Intensity_Summary<- Daily_Intensities %>%
  summarize(
    Avg_Sed_Min,
    Avg_Light_Act_Min,
    Avg_Fair_Act_Min,
    Avg_Very_Act_Min
  )

View(Daily_Intensity_Summary)
```

Setup the values to be included

This is to help establish the foundation of the visual.

```r
Daily_Intensity_Averages <- c(991.2106,192.8128,13.56489,21.16489)
Daily_Int_Avg_Labels <- c("Sedentary", "Lightly Active", "Moderately Active", "Very Active")
Daily_Int_Avg_Colors <- c("red", "green", "blue", "purple")
```

Calculate percentages

Converts the daily intensity averages into percentage format and then stores under a new variable.

```r
Daily_Int_Averages <- (Daily_Intensity_Averages / sum(Daily_Intensity_Averages)) * 100
```

Create the pie chart

Use the pie() function to customize labels, colors, positioning, and add a legend. The visual within this document unfortunately features an overlap with labels. The correct version should show "Very Active" as 1.7%. A complete and corrected version is available in the final PowerPoint project.

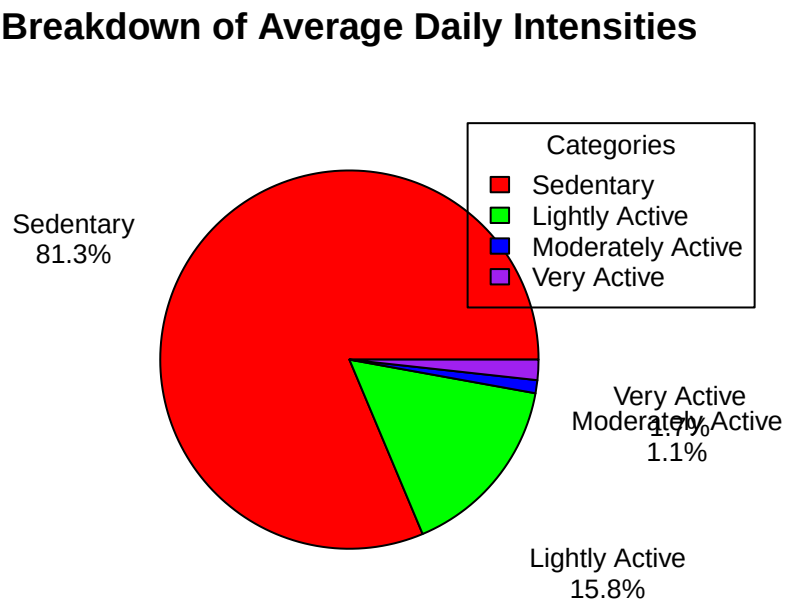
```r
pie(Daily_Intensity_Averages, labels = NA, main = "Breakdown of Average Daily Intensities", col = Daily

# Calculate label positions
label_positions <- cumsum(Daily_Intensity_Averages) - 0.5 * Daily_Intensity_Averages

# Add labels to the chart
text(
```

```
  x = 1.4 * cos(2 * pi * label_positions / sum(Daily_Intensity_Averages)),
  y = 1.2 * sin(2 * pi * label_positions / sum(Daily_Intensity_Averages)),
  labels = paste0(Daily_Int_Avg_Labels, "\n", round(Daily_Int_Averages, 1), "%"),
  pos = 1,
  cex = 0.8
)

# Add a legend
legend(x = 0.5, y = 1, legend = Daily_Int_Avg_Labels, fill = Daily_Int_Avg_Colors, title = "Categories"
```



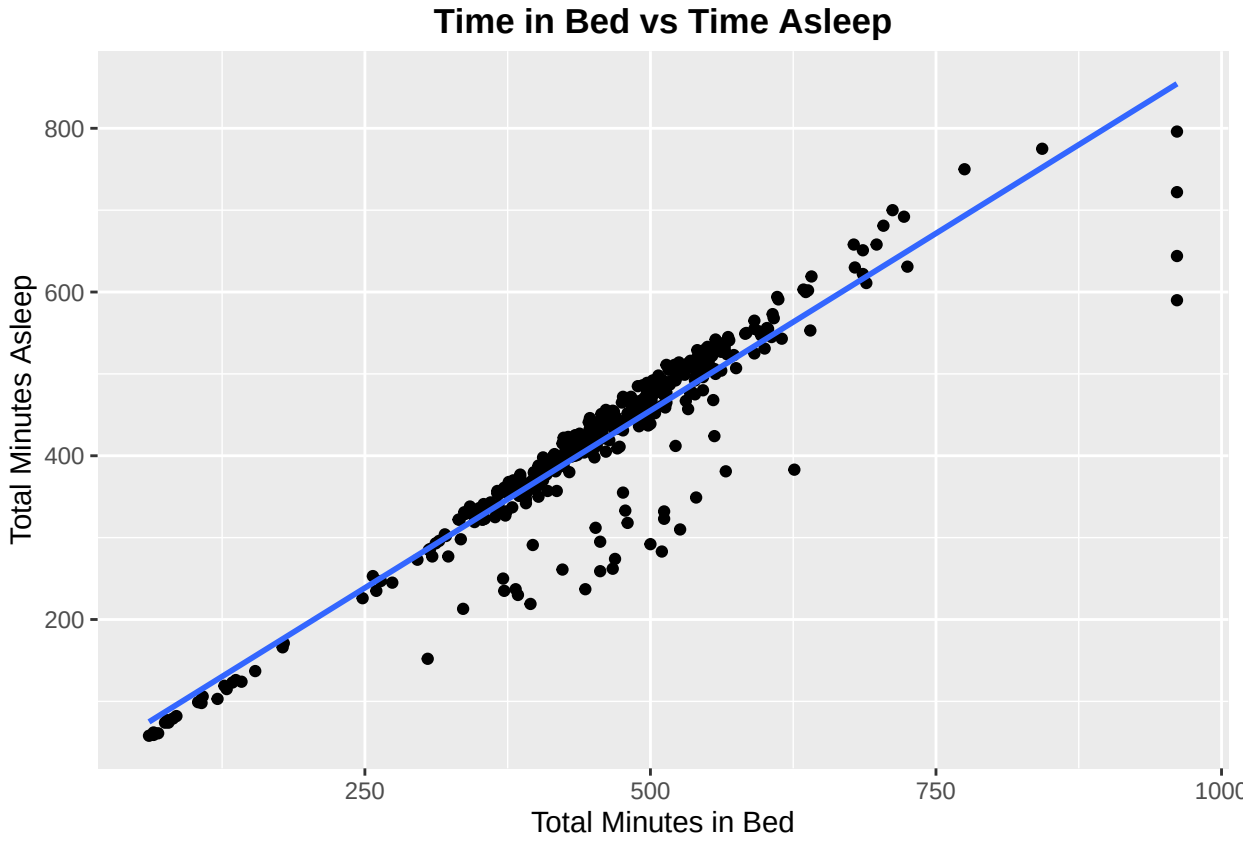
Scatterplot showing correlation between time in bed and time asleep

ggplot() is a function that allows you to customize many aspects such as the regression type, shapes, and the thematic elements such as positioning or bolding. The geom_point() function indicates that this is a scatterplot type of graph.

```
ggplot(Sleep_Day, aes(x=TotalTimeInBed, y=TotalMinutesAsleep)) +
  geom_point()+
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    x = "Total Minutes in Bed",
    y = "Total Minutes Asleep",
    title = "Time in Bed vs Time Asleep"
  ) +
```

```
  theme(
    plot.title = element_text(hjust = 0.5, vjust = 0.5, face = "bold")
  )
```

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

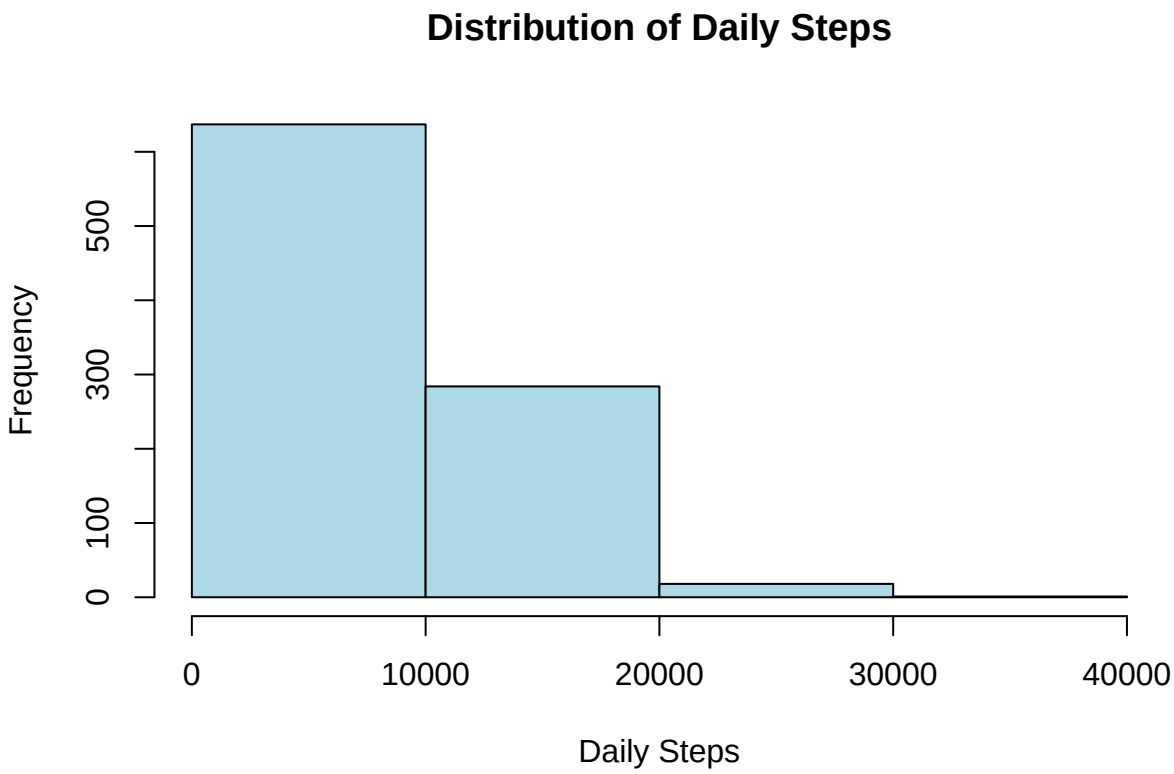


Histogram showing distribution of daily steps

A histogram is used here to illustrate how the different daily steps were distributed across different scales. Health professionals recommend around 10K steps a day for optimal benefits.

```
Only_Steps <- Daily_Steps$StepTotal

hist(Only_Steps,
     breaks = 5,
     main = "Distribution of Daily Steps",
     xlab = "Daily Steps",
     ylab = "Frequency",
     col = "lightblue",
     border = "black")
```


Distribution of Daily Steps
Frequency
0
100
300
500
0
10000
20000
30000
40000
Daily Steps