Bellabeat Fitbit Case Study

Chika Ogbuokiri

2023-07-19

About the Project description:"Sršen encourages you to use public data that explores smart device users' daily habits. She points you to a specific data set: It includes info about daily activity, steps, and heart rate that can be used to explore users' habits."

Lets install all our packages and load them

```
install.packages("tidyverse")
```

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

```
install.packages("readr")
```

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

```
install.packages("ggplot2")
```

```
## 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)
```

```
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   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```
## -- 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(dplyr)
library(ggplot2)
```

```
library(readr)
library(janitor)
```

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

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

#Processing our data Import and read our data

```
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.
```

```
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.
```

```
heart_rate_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.
```

```
steps_hour<-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.
```

```
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<-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.
```

#Cleaning our data

```
daily_activity<-clean_names(daily_activity)
daily_steps<-clean_names(daily_steps)
steps_hour<-clean_names(steps_hour)
heart_rate_seconds<-clean_names(heart_rate_seconds)
sleep_day<-clean_names(sleep_day)
weight_log<-clean_names(weight_log)
glimpse(daily_activity)
```

```
## Rows: 940
## Columns: 15
## $ id                         <dbl> 1503960366, 1503960366, 1503960366, 1503960~
## $ activity_date              <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/1~
## $ total_steps                <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 130~
## $ total_distance             <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9~
## $ tracker_distance           <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9~
## $ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_distance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3~
## $ moderately_active_distance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1~
## $ light_active_distance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5~
## $ sedentary_active_distance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
## $ very_active_minutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66,~
## $ fairly_active_minutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, ~
## $ lightly_active_minutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205~
## $ sedentary_minutes          <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 8~
## $ calories                   <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2~
```

#Analyze the data

```
summary(daily_activity$total_steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    3790    7406    7638   10727   36019
```

The result of the summary shows that we even have participated who recorded 0 in their daily activity.

#Lets see the number of unique participants in these tables

```
n_distinct(daily_activity$id)
```

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

```
n_distinct(daily_steps$id)
```

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

```
n_distinct(steps_hour$id)
```

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

```
n_distinct(heart_rate_seconds$id)
```

```
## [1] 14
```

```
n_distinct(sleep_day$id)
```

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

```
n_distinct(weight_log$id)
```

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

We can see that the users who used the Fitbit Tracker for weight logs and heart rate tracking is very small.

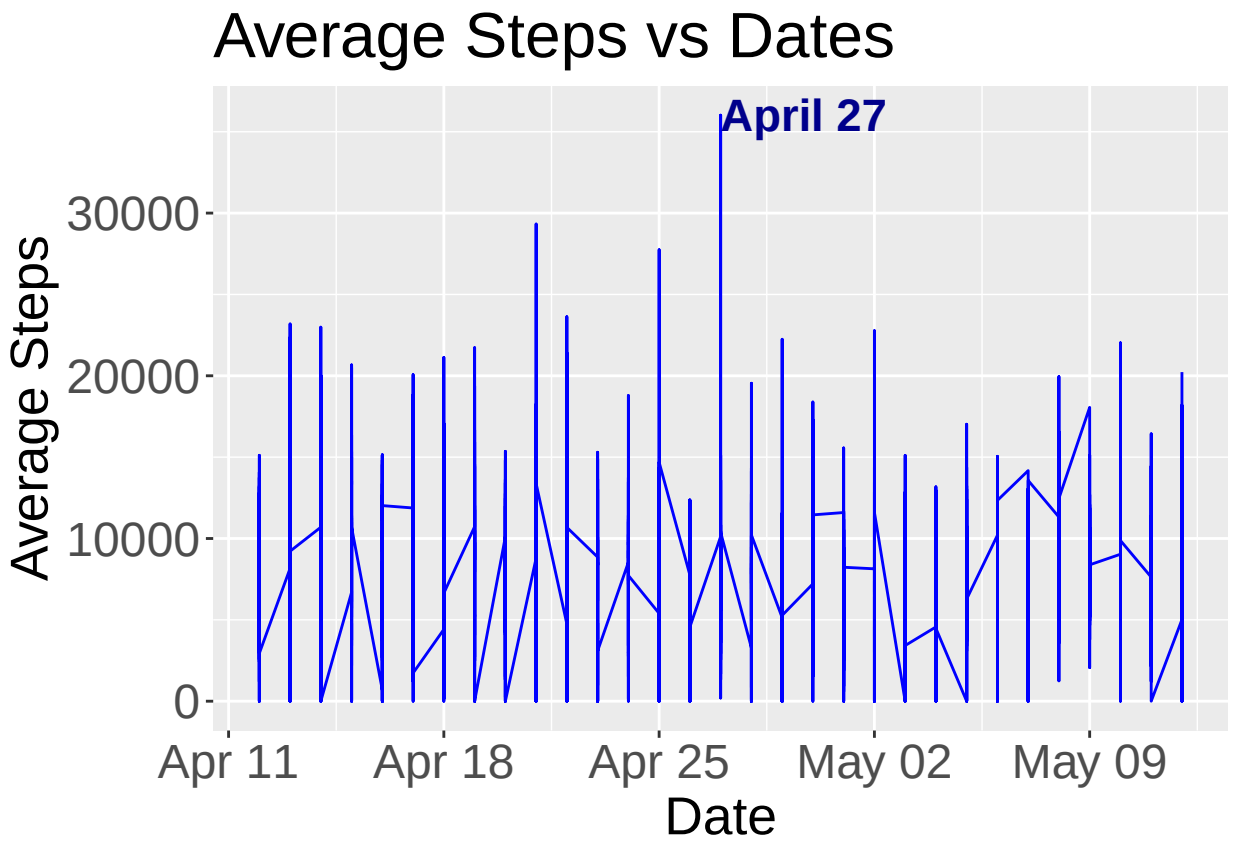
```
### We need to convert the activity day to Year , month and day for easy analysis
daily_steps$Ymd<-mdy(daily_steps$activity_day)

###Lets see the average daily steps by the diff users
average_steps_per_day <- daily_steps %>%
  group_by(activity_day, id) %>%
  summarise(average_step = mean(step_total, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'activity_day'. You can override using the
## `.groups` argument.
```

```
m_daily_steps<-mutate(daily_steps,average_steps_per_day)
### lets plot the Dates Vs Average steps; we noticed that there is spike on apr 27th
####was this day world health day? what could have caused such spike? We need more data to answer this
options(scipen=999)
ggplot(m_daily_steps) +
  geom_line(mapping = aes(x = Ymd, y = average_step), color = "blue") +
  labs(title = "Average Steps vs Dates", x = "Date", y = "Average Steps") +
  theme(axis.text = element_text(size = 18),  # Increase font size of axis labels
```

```
      axis.title = element_text(size = 20),
      plot.title = element_text(size = 24))+ annotate("text", x = m_daily_steps$Ymd[which.max(m_daily_s
          y = max(m_daily_steps$average_step), label = "April 27", size = 6,fontface = "bold", hjust =
```



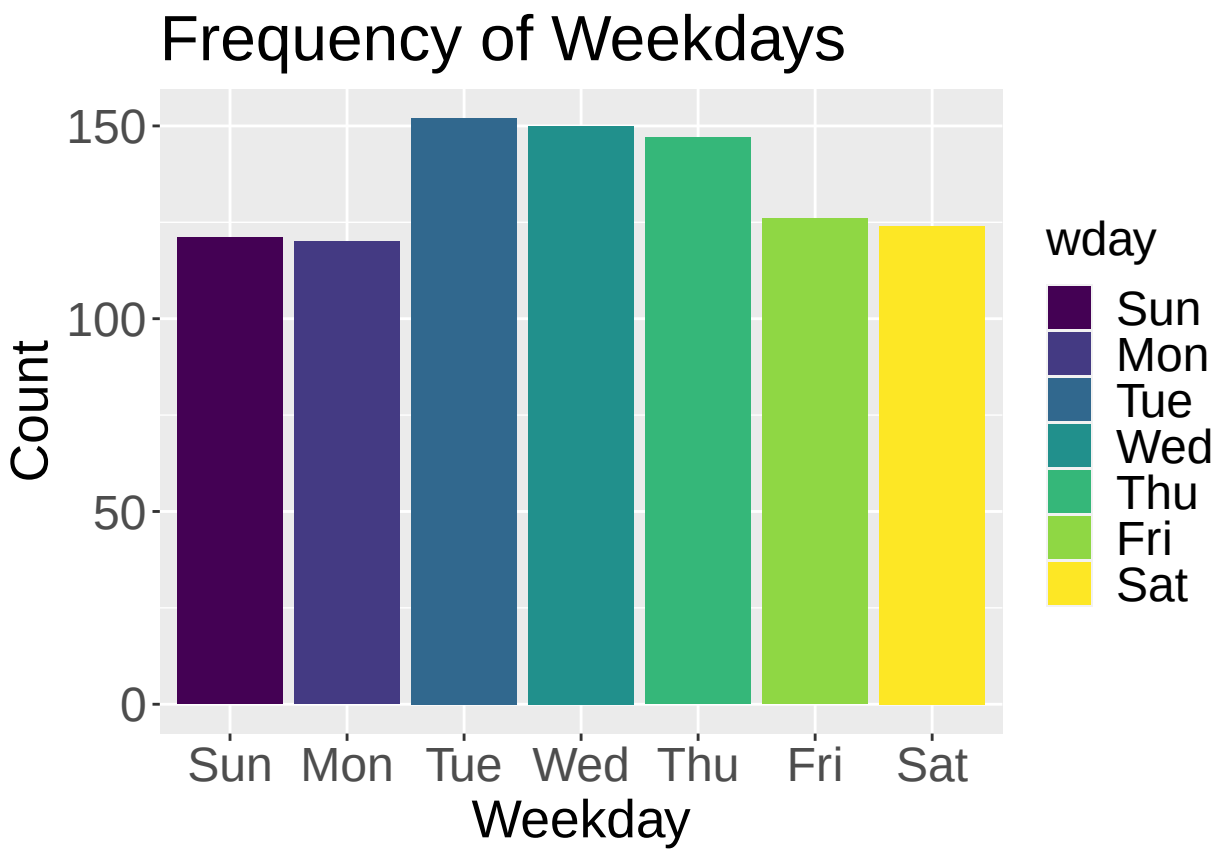

```
width<-9
height<-6
ggsave("Average Steps Vs Dates.png",width=width,height = height)
```

```
###Lets convert the date to a character and we will see a new column called Ymd
daily_activity$Ymd<-mdy(daily_activity$activity_date)

daily_activity <- daily_activity %>%
  mutate(wday = wday(ymd(Ymd), label = TRUE)) %>%
  mutate(wday = factor(wday))

### View the updated daily_activity table and plot a bar chart to ###see the day with the highest activ

library(ggplot2)
### Bar chart with title
ggplot(daily_activity) +
  geom_bar(mapping = aes(x = wday, fill = wday)) +
  labs(title = "Frequency of Weekdays", x = "Weekday", y = "Count") +
  theme(axis.text = element_text(size = 18),  # Increase font size of axis labels
        axis.title = element_text(size = 20),
        legend.title = element_text(size=18),
        legend.text = element_text(size=18),
        plot.title = element_text(size = 24))
```

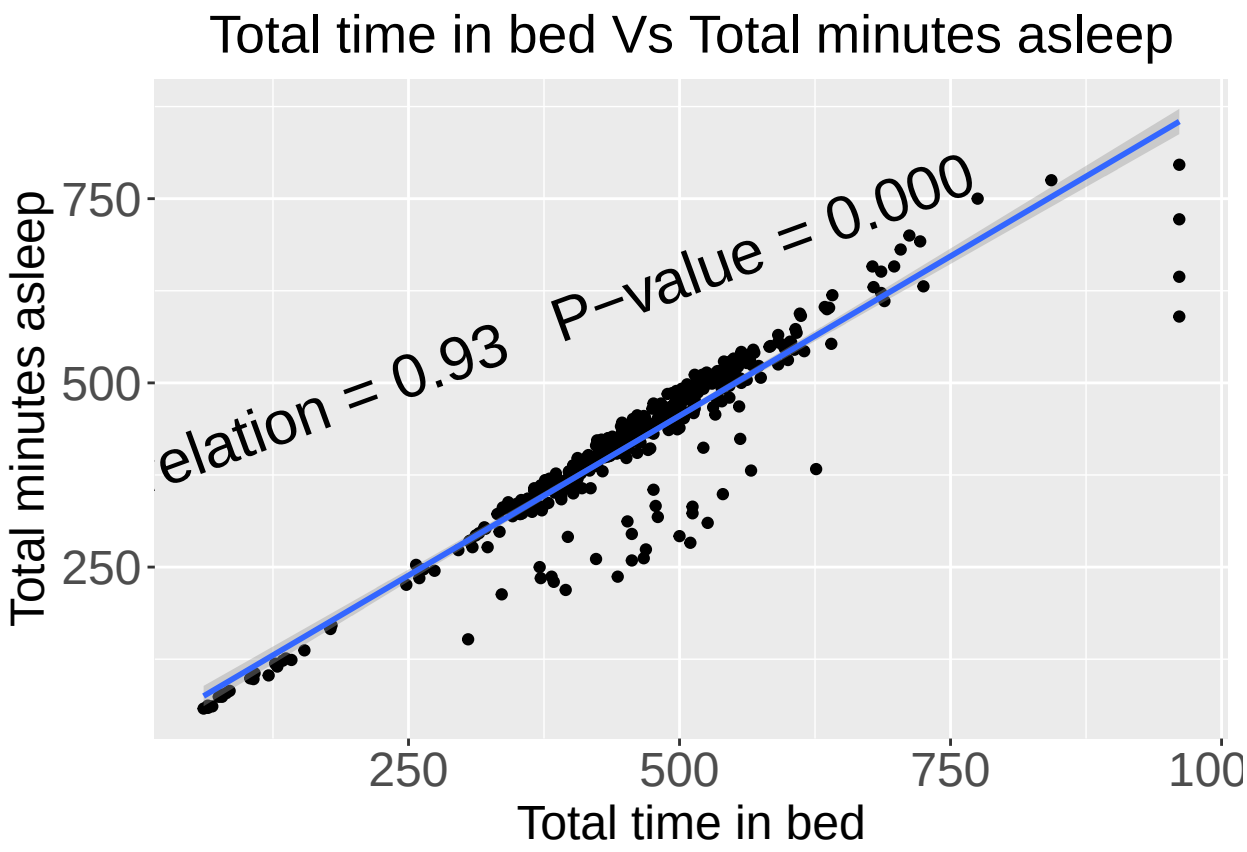



```
ggsave("Frequency of Weekdays.png",width=10,height=6)
```

```
cor_result <- cor.test(sleep_day$total_time_in_bed, sleep_day$total_minutes_asleep)
p_value <- cor_result$p.value
rounded_p_value <- sprintf("%.3f", p_value)

### Plot with correlation coefficient and p-value annotation
###This shows a strong relationship between the two varaiables and ###they are statistically signifcant
ggplot(data = sleep_day, aes(x = total_time_in_bed, y = total_minutes_asleep)) +
  geom_point() +
  geom_smooth(method = lm) +
  labs(title = "Total time in bed Vs Total minutes asleep",
       x = "Total time in bed", y = "Total minutes asleep") +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.title.x = element_text(size = 18),
        axis.title.y = element_text(size = 18),
        axis.text = element_text(size = 18)) +
  annotate("text", x = max(sleep_day$total_time_in_bed), y = max(sleep_day$total_minutes_asleep),
           label = paste("Correlation =", round(cor_result$estimate, 2), "  P-value =", rounded_p_value
           size = 8, color = "black",angle=20,hjust = 1.2, vjust = -1)
```

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




```
ggsave("Time In Bed Vs Time Asleep.png",width=9,height=6)
```

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

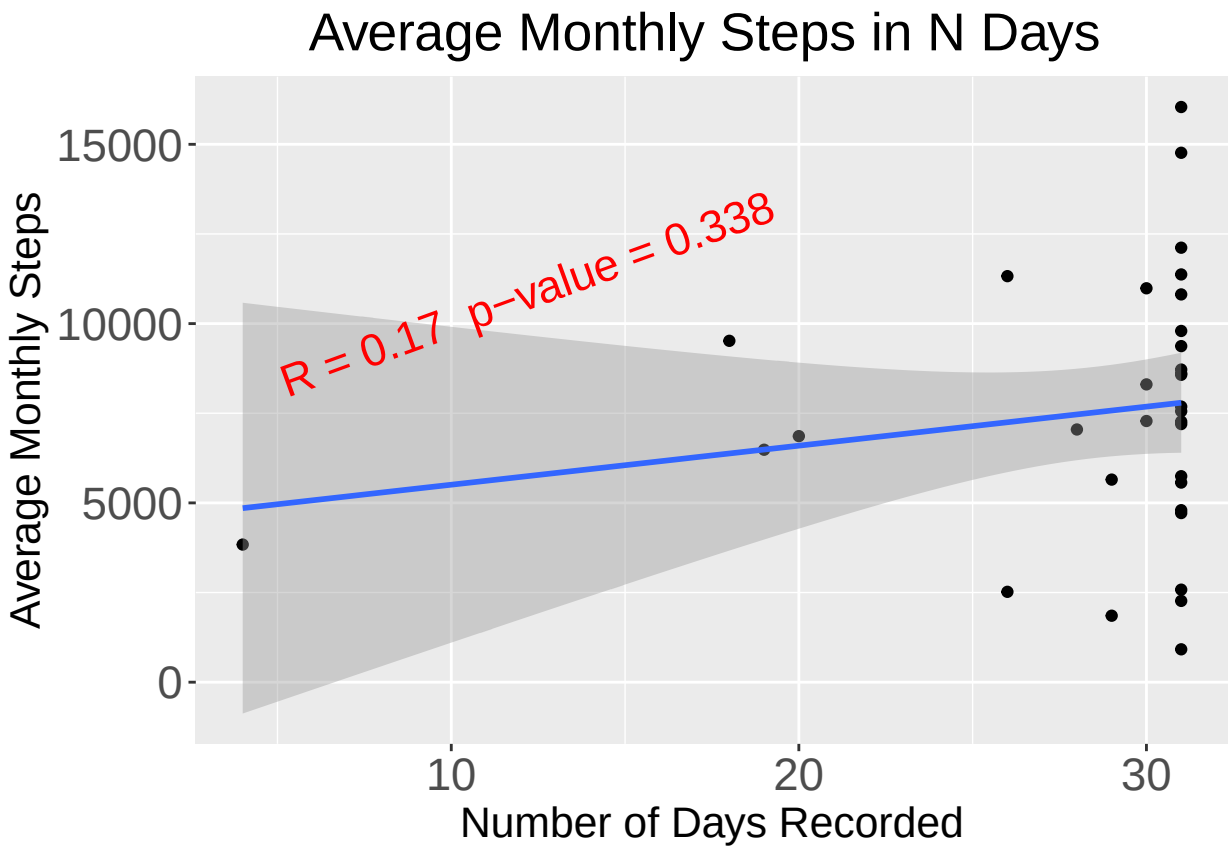
```
##Lets get a quick summary of our daily_activity table
###summarize the steps of the participants monthly
summary_user_activity_monthly <- daily_activity %>%
  group_by(id) %>%
  summarise(
    record_count = n(),
    sum_steps_monthly = sum(total_steps),
    min_steps_monthly = min(total_steps),
    max_steps_monthly = max(total_steps),
    median_steps_monthly = median(total_steps),
    avg_steps_monthly = mean(total_steps))

###We can see that the correlation is not strong and it is not statistically significant
cor_result<-cor.test(summary_user_activity_monthly$record_count,summary_user_activity_monthly$avg_steps_
p_value_result<-cor_result$p.value

###Lets plot the frequency of average steps taken monthly
summary_user_activity_monthly %>%
  ggplot(aes(x = record_count, y = avg_steps_monthly,width=40)) +
  geom_point() +
  geom_smooth(method = lm) +
  labs(title = "Average Monthly Steps in N Days ",
       x = "Number of Days Recorded", y = "Average Monthly Steps") +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
```

```
        axis.title.x = element_text(size = 16),
        axis.title.y = element_text(size = 16),
       axis.text = element_text(size=17))+ annotate("text", x = max(summary_user_activity_monthly$recor
          label = paste("R =", round(cor_result$estimate, 2), " p-value =", round(p_value_result,3)),
          size = 6, color = "red",angle=20, hjust = 1.8, vjust = -1)
```

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




```
ggsave("Number of Days recorded vs Average Monthly Steps per Participant.png",width=7,height=5)
```

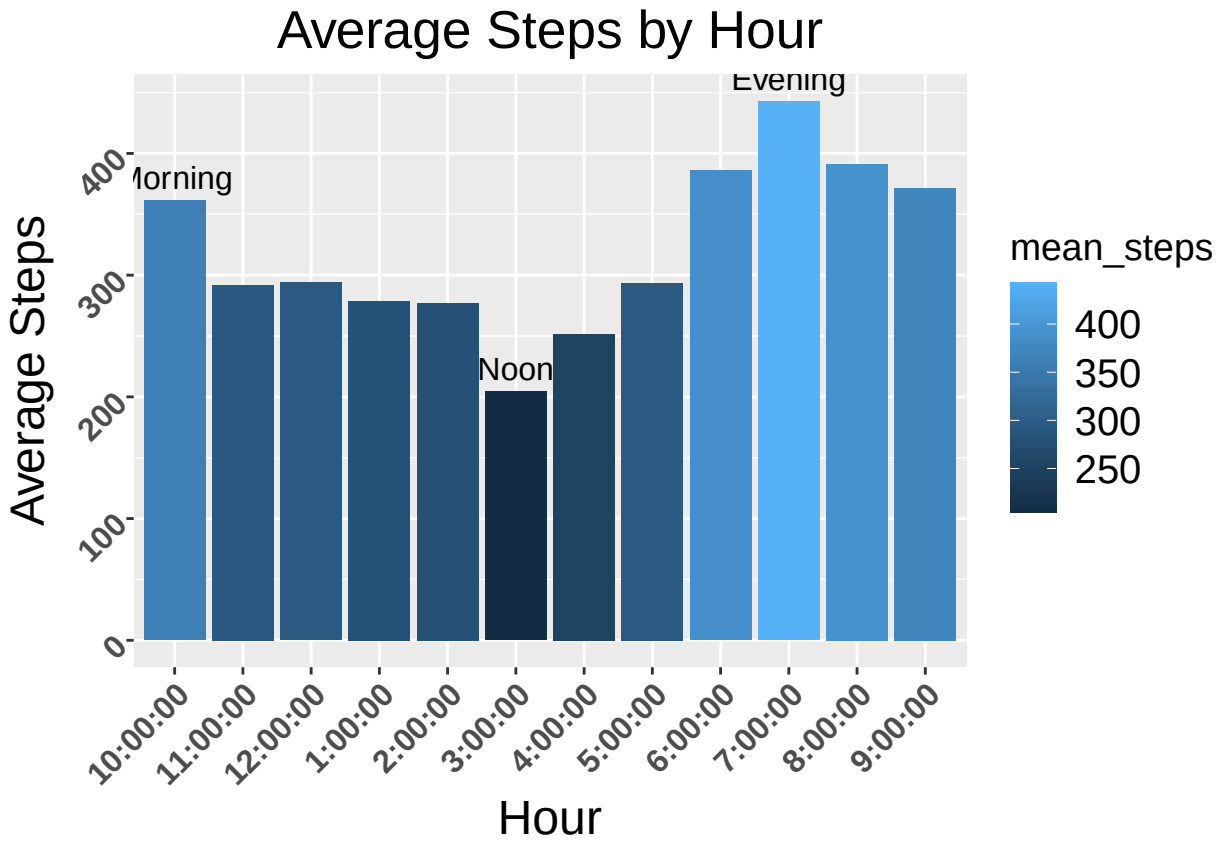
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
###Lets separate the activity_hour into date and hour
steps_hour <- separate(steps_hour, activity_hour, c("date", "hour"), sep = " ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 22099 rows [1, 2, 3, 4, 5, 6,
## 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
steps_hour$date <- as.Date(steps_hour$date)
### create a df to house the mean steps and the counts(n)
summary_steps_hour<-steps_hour%>%group_by(hour)%>%summarise(count=n(),mean_steps=mean(step_total))
###Arrange the mean steps hour to reflect the ordering of a normal time and plot a bar chart
summary_steps_hour %>%
  mutate(label = ifelse(mean_steps == max(mean_steps), "Evening",
                        ifelse(mean_steps == sort(mean_steps, decreasing = TRUE)[5], "Morning",
                               ifelse(mean_steps == min(mean_steps), "Noon", "")))) %>%
mutate(hour = factor(hour, levels = c("10:00:00", "11:00:00", "12:00:00", "1:00:00", "2:00:00",
"3:00:00", "4:00:00", "5:00:00", "6:00:00", "7:00:00", "8:00:00", "9:00:00"))) %>%
  ggplot() +
```

```
  geom_col(aes(x = hour, y = mean_steps, fill = mean_steps)) +
  scale_color_manual(values = c("Evening" = "darkblue", "Morning" = "black", "Noon" = "lightblue")) +
  geom_text(aes(x = hour, y = mean_steps, label = label), vjust = -0.5,size=4.1) +
  labs(title = "Average Steps by Hour", x = "Hour", y = "Average Steps") +
  theme(plot.title = element_text(size = 20, hjust = 0.5),
        axis.text = element_text(face="bold",size=12,angle=45,hjust=1),
       axis.title.x = element_text(size=18),
       axis.title.y = element_text(size=18),
       legend.text = element_text(size=15),
       legend.title = element_text(size=14))
```




```
ggsave("Average Steps per Hour.png",width=9,height=6)
```

```
summary(daily_activity$total_steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    3790    7406    7638   10727   36019
```

The result of the summary shows that we even have participated who recorded 0 in their daily activity.