Bellabeat_Final

Muazzam

2024-07-22

Contents

This is a case study completed on Bellabeat - the wellness company. This study focuses on the effectiveness of their smart watches and analyses data from 31 users over a 1 month period.

Functions and Setup

The following code defines the setup and libraries that are used in this analysis

```r
#set root directory for csv files
params <- list(
  root = getwd()
)
params$root <- paste0(params$root, "/") ##CHANGE the backslash to "\\" if on UNIX based OS

# library(ggplot2)
library(tidyr)
library(stringr)
library(janitor)
```

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

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

```r
library(here)
```

```
## here() starts at C:/Users/muazzam/Desktop/Github_Projects/bellabeat_case_study
```

```
library(ggplot2)
library(skimr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(forcats)
library(scales)
library(ggrepel)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

The following code defines the functions that will be used in the case study

```
read = function(s1){
  paste(params$root,s1,sep="") %>%
    read.csv()
}

camel_to_snake <- function(camel_case_names) {
  # Use str_replace_all from the stringr package to add underscores before uppercase letters
  snake_case_names <- str_replace_all(camel_case_names, "(?<!^)([A-Z])", "_\\1")
  # Convert all characters to lowercase
  snake_case_names <- tolower(snake_case_names)
  return(snake_case_names)
}
```

Below, the appropriate data tables from the Bellabeat dataset are loaded in as local variables

```
daily_activity <- read("dailyActivity_merged.csv")
sleep_day <- read("sleepDay_merged.csv")
sleep_min <- read("minuteSleep_merged.csv")
intensity_min <- read("minuteStepsNarrow_merged.csv")
heart_rate <- read("heartrate_seconds_merged.csv")
weight <- read("weightLogInfo_merged.csv")
```

```
calories_min <- read("minuteCaloriesNarrow_merged.csv")

#hourly data frames

#minutely data frames
minute_intensities_narrow_merged <- read("minuteIntensitiesNarrow_merged.csv")
```

Cleaning & Manipulation

The data tables are then cleaned, first checked for any duplicates

```
sum(duplicated(daily_activity))
```

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

```
sum(duplicated(sleep_day))
```

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

```
sum(duplicated(sleep_min))
```

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

```
sum(duplicated(intensity_min))
```

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

```
sum(duplicated(heart_rate))
```

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

```
sum(duplicated(weight))
```

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

```
sum(duplicated(calories_min))
```

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

removing the duplicates yields

```
sleep_day <- sleep_day %>%
  distinct()
sleep_min <- sleep_min %>%
  distinct()
```

```
#edit variable names in daily_activity
daily_activity <- setNames(daily_activity,camel_to_snake(names(daily_activity)))

#edit variable names in sleep_day
sleep_day <- setNames(sleep_day,camel_to_snake(names(sleep_day)))


#cleaning the names
clean_names(daily_activity)
```

In order to better analyse the data, the `daily_activity` and `sleep_day` tables must be merged as that will display full activity of the user over 24 hours.

```
sleep_day_sep <- separate(sleep_day, sleep_day, into = c("activity_date","time","am/pm"), sep = " ", rem

full_day <- daily_activity %>%
  full_join(sleep_day_sep, by = c("id" = "id", "activity_date" = "activity_date"))
```

Checking to ensure that all recorded metrics with minutes add up to 1440 which indicates that all 24 hours of the day are accounted for

```
full_day <- mutate(full_day, total_time=(very_active_minutes+fairly_active_minutes+lightly_active_minut

full_day_check1 <- filter(full_day, total_time > 1440)
glimpse(full_day_check1)
```

```
## Rows: 155
## Columns: 21
## $ id                          <dbl> 1503960366, 1644430081, 1844505072, 1844505~
## $ activity_date               <chr> "4/17/2016", "5/2/2016", "4/15/2016", "5/1/~
## $ total_steps                 <int> 9705, 3758, 3844, 2573, 678, 980, 3761, 441~
## $ total_distance              <dbl> 6.48, 2.73, 2.54, 1.70, 0.47, 0.68, 2.60, 2~
## $ tracker_distance            <dbl> 6.48, 2.73, 2.54, 1.70, 0.47, 0.68, 2.60, 2~
## $ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_distance        <dbl> 3.19, 0.07, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
## $ moderately_active_distance <dbl> 0.78, 0.31, 0.00, 0.26, 0.00, 0.00, 0.00, 0~
## $ light_active_distance       <dbl> 2.51, 2.35, 2.54, 1.45, 0.47, 0.68, 2.60, 2~
## $ sedentary_active_distance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_minutes         <int> 38, 1, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, ~
## $ fairly_active_minutes       <int> 20, 7, 0, 7, 0, 0, 0, 8, 0, 0, 0, 0, 0, 0, ~
## $ lightly_active_minutes      <int> 164, 148, 176, 75, 55, 51, 192, 181, 238, 1~
## $ sedentary_minutes           <int> 539, 682, 527, 585, 734, 941, 1058, 706, 66~
## $ calories                    <int> 1728, 2580, 1725, 1541, 2220, 2221, 2638, 1~
## $ time                        <chr> "12:00:00", "12:00:00", "12:00:00", "12:00:~
## $ ‘am/pm‘                     <chr> "AM", "AM", "AM", "AM", "AM", "AM", "AM", "~
## $ total_sleep_records         <int> 1, 1, 1, 1, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ total_minutes_asleep        <int> 700, 796, 644, 590, 750, 475, 296, 503, 531~
## $ total_time_in_bed           <int> 712, 961, 961, 961, 775, 499, 315, 546, 565~
## $ total_time                  <int> 1473, 1799, 1664, 1628, 1564, 1491, 1565, 1~
```

```
full_day_check2 <- filter(full_day, total_time < 1440)
glimpse(full_day_check2)
```

```
## Rows: 129
## Columns: 21
## $ id                         <dbl> 1503960366, 1503960366, 1844505072, 2026352~
## $ activity_date              <chr> "4/16/2016", "5/11/2016", "4/30/2016", "4/1~
## $ total_steps                <int> 12669, 12770, 4014, 3335, 2467, 2915, 6088,~
## $ total_distance             <dbl> 8.16, 8.13, 2.67, 2.07, 1.53, 1.81, 3.77, 4~
## $ tracker_distance           <dbl> 8.16, 8.13, 2.67, 2.07, 1.53, 1.81, 3.77, 4~
## $ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_distance       <dbl> 2.71, 2.56, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
## $ moderately_active_distance <dbl> 0.41, 1.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
## $ light_active_distance      <dbl> 5.04, 4.55, 2.65, 2.05, 1.53, 1.81, 3.77, 4~
## $ sedentary_active_distance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_minutes        <int> 36, 36, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ fairly_active_minutes      <int> 10, 23, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ lightly_active_minutes     <int> 221, 251, 184, 197, 153, 162, 286, 352, 233~
## $ sedentary_minutes          <int> 773, 669, 218, 653, 749, 712, 586, 492, 594~
## $ calories                   <int> 1863, 1783, 1763, 1431, 1370, 1399, 1593, 1~
## $ time                       <chr> "12:00:00", "12:00:00", "12:00:00", "12:00:~
## $ `am/pm`                    <chr> "AM", "AM", "AM", "AM", "AM", "AM", "AM", "~
## $ total_sleep_records        <int> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ total_minutes_asleep       <int> 340, 285, 722, 545, 477, 520, 508, 490, 573~
## $ total_time_in_bed          <int> 367, 306, 961, 568, 514, 545, 545, 510, 607~
## $ total_time                 <int> 1407, 1285, 1363, 1418, 1416, 1419, 1417, 1~
```

The result above shows that there are 155 rows where total minutes is more than 24 hours and 129 rows where it is less than 24 hours. This results from an inaccuracy in the data. To resolve this, the data from `sleep_min` provides a more accurate representation of how much each person slept, meaning we can count the minutes sleeping from this table instead of using the `full_day_total_time_in_bed` metric. This is done with the following code

```
sleep_min <- sleep_min %>%
  mutate(mins = value)
sleep_min$mins[sleep_min$mins>0] <- 1
summary(sleep_min)
```

```
##        Id                  date               value           logId
##  Min.   :1.504e+09   Length:187978      Min.   :1.000   Min.   :1.137e+10
##  1st Qu.:3.977e+09   Class :character   1st Qu.:1.000   1st Qu.:1.144e+10
##  Median :4.703e+09   Mode  :character   Median :1.000   Median :1.150e+10
##  Mean   :4.997e+09                      Mean   :1.096   Mean   :1.150e+10
##  3rd Qu.:6.962e+09                      3rd Qu.:1.000   3rd Qu.:1.155e+10
##  Max.   :8.792e+09                      Max.   :3.000   Max.   :1.162e+10
##       mins
##  Min.   :1
##  1st Qu.:1
##  Median :1
##  Mean   :1
##  3rd Qu.:1
##  Max.   :1
```

Preparing sleep day to merge into `full_day`and finally completing the merge

```
sleep_min <- separate(sleep_min, date, into=c('sleep_date','time', 'am/pm'), sep=" ", remove=TRUE)
sleep_min_prep <- sleep_min %>%
  group_by(Id,sleep_date) %>%
  summarise(total_sleep = sum(mins))
```

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

```
full_day_new <- full_day %>% full_join(sleep_min_prep, by=c('id'='Id', 'activity_date'='sleep_date'))
full_day_new[is.na(full_day_new)] <-0
head(full_day_new)
```

```
##           id activity_date total_steps total_distance tracker_distance
## 1 1503960366     4/12/2016       13162           8.50             8.50
## 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
##   logged_activities_distance very_active_distance moderately_active_distance
## 1                          0                 1.88                       0.55
## 2                          0                 1.57                       0.69
## 3                          0                 2.44                       0.40
## 4                          0                 2.14                       1.26
## 5                          0                 2.71                       0.41
## 6                          0                 3.19                       0.78
##   light_active_distance sedentary_active_distance very_active_minutes
## 1                  6.06                         0                  25
## 2                  4.71                         0                  21
## 3                  3.91                         0                  30
## 4                  2.83                         0                  29
## 5                  5.04                         0                  36
## 6                  2.51                         0                  38
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    13                    328               728     1985
## 2                    19                    217               776     1797
## 3                    11                    181              1218     1776
## 4                    34                    209               726     1745
## 5                    10                    221               773     1863
## 6                    20                    164               539     1728
##       time am/pm total_sleep_records total_minutes_asleep total_time_in_bed
## 1 12:00:00    AM                   1                  327               346
## 2 12:00:00    AM                   2                  384               407
## 3        0     0                   0                    0                 0
## 4 12:00:00    AM                   1                  412               442
## 5 12:00:00    AM                   2                  340               367
## 6 12:00:00    AM                   1                  700               712
##   total_time total_sleep
## 1       1440         346
## 2       1440         407
## 3          0           0
## 4       1440         442
## 5       1407         400
## 6       1473         679
```

with the following code we can confirm whether or not the total sleep time is more accurate than before

```
full_day_new <- mutate(full_day_new, total_time_new=(very_active_minutes+fairly_active_minutes+lightly_a

full_day_check1 <- filter(full_day_new, total_time_new > 1440)
glimpse(full_day_check1)
```

```
## Rows: 0
## Columns: 23
## $ id                         <dbl>
## $ activity_date              <chr>
## $ total_steps                <dbl>
## $ total_distance             <dbl>
## $ tracker_distance           <dbl>
## $ logged_activities_distance <dbl>
## $ very_active_distance       <dbl>
## $ moderately_active_distance <dbl>
## $ light_active_distance      <dbl>
## $ sedentary_active_distance  <dbl>
## $ very_active_minutes        <dbl>
## $ fairly_active_minutes      <dbl>
## $ lightly_active_minutes     <dbl>
## $ sedentary_minutes          <dbl>
## $ calories                   <dbl>
## $ time                       <chr>
## $ ‘am/pm‘                    <chr>
## $ total_sleep_records        <dbl>
## $ total_minutes_asleep       <dbl>
## $ total_time_in_bed          <dbl>
## $ total_time                 <dbl>
## $ total_sleep                <dbl>
## $ total_time_new             <dbl>
```

```
full_day_check2 <- filter(full_day_new, total_time_new < 1440)
glimpse(full_day_check2)
```

```
## Rows: 76
## Columns: 23
## $ id                         <dbl> 1503960366, 1624580081, 1644430081, 1844505~
## $ activity_date              <chr> "5/11/2016", "5/12/2016", "5/11/2016", "5/1~
## $ total_steps                <dbl> 12770, 2971, 1329, 0, 0, 9117, 6564, 4193, ~
## $ total_distance             <dbl> 8.13, 1.93, 0.97, 0.00, 0.00, 6.41, 4.07, 2~
## $ tracker_distance           <dbl> 8.13, 1.93, 0.97, 0.00, 0.00, 6.41, 4.07, 2~
## $ logged_activities_distance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ very_active_distance       <dbl> 2.56, 0.00, 0.00, 0.00, 0.00, 1.28, 0.00, 0~
## $ moderately_active_distance <dbl> 1.01, 0.00, 0.00, 0.00, 0.00, 0.67, 0.00, 0~
## $ light_active_distance      <dbl> 4.55, 1.92, 0.95, 0.00, 0.00, 4.44, 4.07, 2~
## $ sedentary_active_distance  <dbl> 0.00, 0.01, 0.01, 0.00, 0.00, 0.00, 0.00, 0~
## $ very_active_minutes        <dbl> 36, 0, 0, 0, 0, 16, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ fairly_active_minutes      <dbl> 23, 0, 0, 0, 0, 16, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ lightly_active_minutes     <dbl> 251, 107, 49, 0, 0, 236, 345, 229, 343, 128~
## $ sedentary_minutes          <dbl> 669, 890, 713, 711, 966, 728, 530, 665, 330~
## $ calories                   <dbl> 1783, 1002, 1276, 665, 1383, 1853, 1658, 14~
## $ time                       <chr> "12:00:00", "0", "0", "0", "0", "0", "12:00~
## $ ‘am/pm‘                    <chr> "AM", "0", "0", "0", "0", "0", "AM", "AM", ~
## $ total_sleep_records        <dbl> 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0~
## $ total_minutes_asleep       <dbl> 285, 0, 0, 0, 0, 0, 538, 511, 456, 0, 436, ~
## $ total_time_in_bed          <dbl> 306, 0, 0, 0, 0, 0, 560, 521, 485, 0, 490, ~
## $ total_time                 <dbl> 1285, 0, 0, 0, 0, 0, 1435, 1415, 1158, 0, 1~
## $ total_sleep                <dbl> 306, 0, 0, 0, 0, 0, 564, 545, 330, 0, 480, ~
## $ total_time_new             <dbl> 1285, 997, 762, 711, 966, 996, 1439, 1439, ~
```

Upon further analysis, it seems that not all the observation for which a `total_distance` value is recorded has a corresponding and equal `tracker_distance` value. When running the code: `sum(daily_activity$total_distance == daily_activity$tracker_distance)`, the results show that only 925/940 observations include a `tracker_distance` record which matches the `total_distance` observation.

The subset of `daily_activity` which includes the faulty data can be seen with the following code.

```
subset(full_day_new, total_distance != tracker_distance)%>%
  print()
```

```
##             id activity_date total_steps total_distance tracker_distance
## 690 6962181067     4/21/2016       11835           9.71             7.88
## 694 6962181067     4/25/2016       13239           9.27             9.08
## 708 6962181067      5/9/2016       12342           8.72             8.68
## 712 7007744171     4/12/2016       14172          10.29             9.48
## 713 7007744171     4/13/2016       12862           9.65             8.60
## 714 7007744171     4/14/2016       11179           8.24             7.48
## 718 7007744171     4/18/2016       14816          10.98             9.91
## 719 7007744171     4/19/2016       14194          10.48             9.50
## 720 7007744171     4/20/2016       15566          11.31            10.41
## 725 7007744171     4/25/2016       18229          13.34            12.20
## 727 7007744171     4/27/2016       13541          10.22             9.06
## 729 7007744171     4/29/2016       20067          14.30            13.42
## 732 7007744171      5/2/2016       13041           9.18             8.72
## 733 7007744171      5/3/2016       14510          10.87             9.71
## 735 7007744171      5/5/2016       15010          11.10            10.04
##     logged_activities_distance very_active_distance moderately_active_distance
## 690                   4.081692                 3.99                       2.10
## 694                   2.785175                 3.02                       1.68
## 708                   3.167822                 3.90                       1.18
## 712                   4.869783                 4.50                       0.38
## 713                   4.851307                 4.61                       0.56
## 714                   3.285415                 2.95                       0.34
## 718                   4.930550                 3.79                       2.12
## 719                   4.942142                 4.41                       0.76
## 720                   4.924841                 4.79                       0.67
## 725                   4.861792                 4.31                       1.37
## 727                   4.885605                 4.27                       0.66
## 729                   4.911146                 4.31                       2.05
## 732                   2.832326                 4.64                       0.70
## 733                   4.912368                 4.48                       1.02
## 735                   4.878232                 4.33                       1.29
##     light_active_distance sedentary_active_distance very_active_minutes
## 690                  3.51                      0.11                  53
## 694                  4.46                      0.10                  35
## 708                  3.65                      0.00                  43
## 712                  5.41                      0.00                  53
## 713                  4.48                      0.00                  56
## 714                  4.96                      0.00                  34
## 718                  5.05                      0.02                  48
## 719                  5.31                      0.00                  53
## 720                  5.86                      0.00                  60
## 725                  7.67                      0.00                  51
## 727                  5.29                      0.00                  50
## 729                  7.95                      0.00                  55
## 732                  3.83                      0.00                  64
## 733                  5.36                      0.00                  58
## 735                  5.48                      0.00                  53
##     fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 690                    27                    214               708     2179
## 694                    31                    282               637     2194
## 708                    21                    231               607     2105
## 712                     8                    355              1024     2937
## 713                    22                    261              1101     2742
## 714                     6                    304              1096     2668
## 718                    31                    284              1077     2832
## 719                    17                    304              1066     2812
## 720                    33                    347              1000     3096
## 725                    24                    379               986     3055
## 727                    12                    337              1041     2830
## 729                    42                    382               961     3180
## 732                    14                    250              1112     2642
## 733                    31                    330              1021     2976
## 735                    23                    317              1047     2933
##         time am/pm total_sleep_records total_minutes_asleep total_time_in_bed
## 690 12:00:00    AM                   1                  451               457
## 694 12:00:00    AM                   1                  400               415
## 708 12:00:00    AM                   1                  489               497
## 712        0     0                   0                    0                 0
## 713        0     0                   0                    0                 0
## 714        0     0                   0                    0                 0
## 718        0     0                   0                    0                 0
## 719        0     0                   0                    0                 0
## 720        0     0                   0                    0                 0
## 725        0     0                   0                    0                 0
## 727        0     0                   0                    0                 0
## 729        0     0                   0                    0                 0
## 732        0     0                   0                    0                 0
## 733        0     0                   0                    0                 0
## 735        0     0                   0                    0                 0
##     total_time total_sleep total_time_new
## 690       1459         438           1440
## 694       1400         455           1440
## 708       1399         538           1440
## 712          0           0           1440
## 713          0           0           1440
## 714          0           0           1440
## 718          0           0           1440
## 719          0           0           1440
## 720          0           0           1440
## 725          0           0           1440
## 727          0           0           1440
## 729          0           0           1440
## 732          0           0           1440
## 733          0           0           1440
## 735          0           0           1440
```

This discrepancy can be caused by a faulty observation from the Bellabeat wellness tracker and thus should be removed from the data set since the tracker distance and total distance will be used for analysis.

```
full_day_cleaned <- subset(full_day_new, total_distance == tracker_distance)
```

Running the following code shows that there are 33 distinct user IDs, or people which were studied across the month.There are only 33 distinct users as this is the max value for any of the sheets from the dataset.

```
n_distinct(full_day_cleaned$id)
```

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

Therefore, to ease the analysis and plotting, a `user_id` table will be made assigining each of the user ids to a user number as done below

```
Users <- paste("User", c(1:33), sep = " ")

Id <- distinct(full_day_new, id)

users_id <- tibble(Users, Id)
```

An activity level table will be used to categorize the individuals who spend more than 30 minutes in ver active exercise as `high`, between 10 and 30 as `medium` and below 10 as `low`

```
AL <- full_day_cleaned %>%
  group_by(id) %>%
  summarize(
    median_very_active = median(very_active_minutes)
  ) %>%
  mutate(
    activity_level = case_when(
      median_very_active < 10 ~ "Low",
      median_very_active <= 30 ~ "Med",
      median_very_active < 210 ~ "High"
    ),
    activity_level = factor(
      activity_level,
      levels=c('Low', 'Med', 'High')
    )
  ) %>%
  subset(select = -c(median_very_active))
```

Plots

Initially, we set up a data table to demonstrate the proportion of time spent exercising in each of the categories indicated by the `full_day_new` table. All the minutes spent except for sleeping were used as the whole and each section `fairly_active`, `lightly_active`, etc were used as the proportions. This is achieved in the following code

```
AL <- full_day_cleaned %>% #AL indicates activity level
  group_by(id) %>%
  summarize(
    median_very_active = median(very_active_minutes)
  ) %>%
  mutate(
    activity_level = case_when(
      median_very_active < 10 ~ "Low",
      median_very_active <= 30 ~ "Med",
      median_very_active < 210 ~ "High"
    ),
    activity_level = factor(
      activity_level,
      levels=c('Low', 'Med', 'High')
    )
  ) %>%
  subset(select = -c(median_very_active))
```

Table `df_formatted` will be created to properly format the data in order to generate the pie charts

```
df_2 <- full_day_cleaned %>%
  group_by(id) %>%
  summarise(very_active_mins = median(very_active_minutes),
            lightly_active_mins = median(lightly_active_minutes),
            fairly_active_mins = median(fairly_active_minutes),
            sedentary_mins = median(sedentary_minutes))

df_2 <- df_2 %>%
  mutate(total = rowSums(df_2[, -1]))

df_2 <- df_2 %>%
  mutate(
    fairly_active_prop = fairly_active_mins / total,
    lightly_active_prop = lightly_active_mins / total,
    very_active_prop = very_active_mins / total,
    sedentary_prop = sedentary_mins / total
  )
```

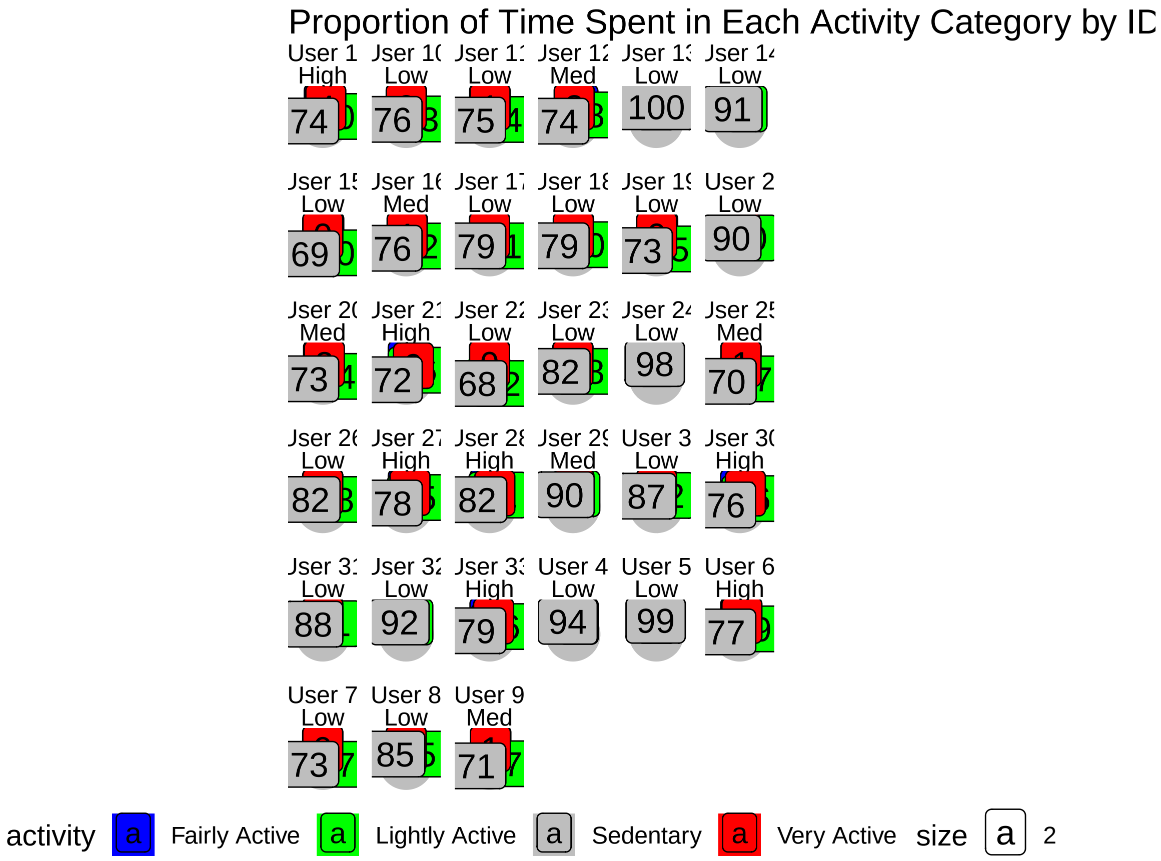
Since the pie chart can only be generated with long data, the `df_2` table is formatted in a long format as follows

```
df_long <- df_2 %>%
  select(id, fairly_active_prop, lightly_active_prop, very_active_prop, sedentary_prop) %>%
  pivot_longer(cols = -id, names_to = "activity", values_to = "proportion")

df_long <- df_long %>%
  full_join(users_id, by = c("id" = "id")) %>%
  full_join(AL, by = c("id" = "id"))
```

The table is then finally graphed as follows

```
labels <- c(fairly_active_prop = "Fairly Active", lightly_active_prop = "Lightly Active",
            very_active_prop = "Very Active",
            sedentary_prop = "Sedentary")
colors <- c(fairly_active_prop = "blue", lightly_active_prop = "green", very_active_prop = "red", seden

ggplot(df_long, aes(x = "", y = proportion, fill = activity)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  facet_wrap(~Users~activity_level) +
  labs(title = "Proportion of Time Spent in Each Activity Category by ID and Activity Level") +
  theme_void() +
  theme(legend.position = "bottom") +
  scale_fill_manual(values = colors, labels = labels)+
  geom_label(data = df_long, aes(label = 100*round(proportion,2), size = 2.0, show.legend = FALSE))
```

```
## Warning in geom_label(data = df_long, aes(label = 100 * round(proportion, :
## Ignoring unknown aesthetics: show.legend
```

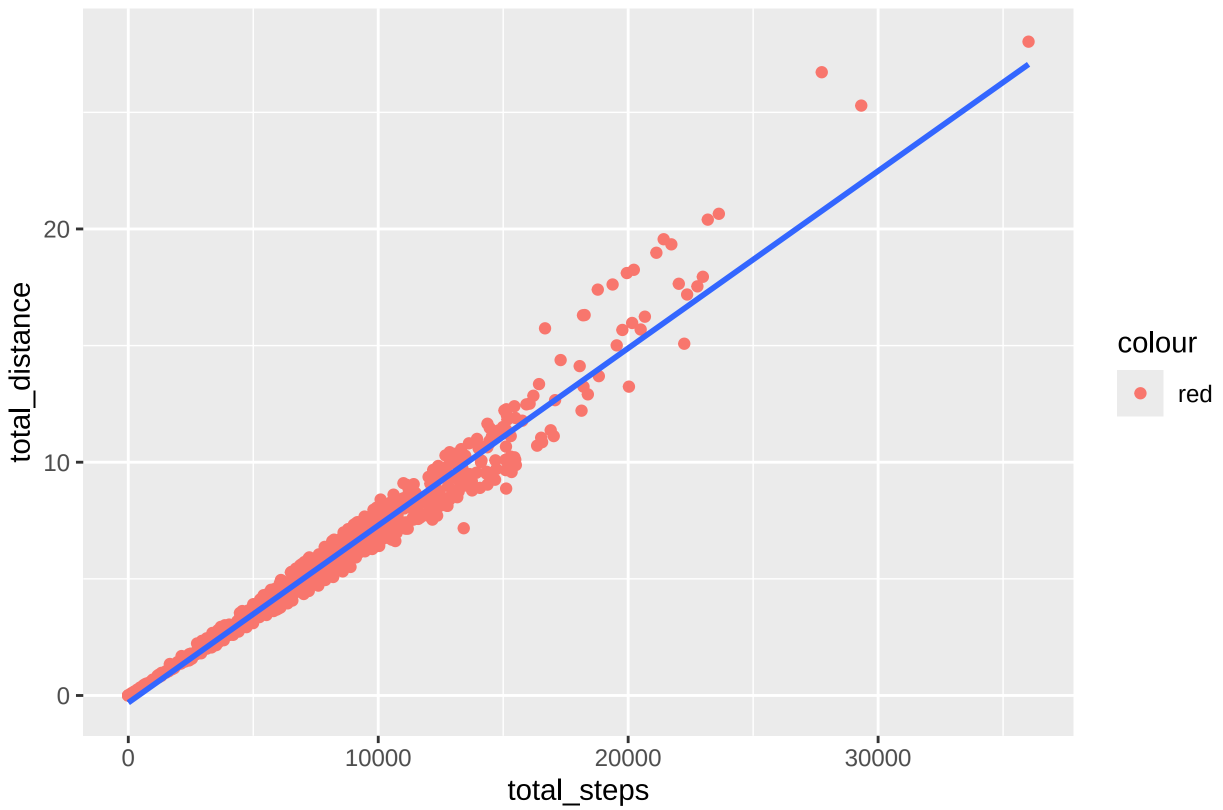


From the graphs, we can see that those who spend at least 30 minutes in very active exercise each day (high), always spend at least 2% of their total exercise time in very active exercise. All the users classified as medium exercise have a proportion of very active exercise at 1%. This can be used for marketing and demographic research purposes. It seems that people spend at most 2-3% of their time exercising at a high intensity which can influence marketing strategies and change the demographic of people.

Further, we compare total steps to total distance to observe if a trend emerges.

```r
ggplot(full_day_cleaned, aes(x=total_steps, y=total_distance)) +
  geom_point(aes(color="red")) +
  geom_smooth(method="lm",se=FALSE) +
  labs(caption = lm(daily_activity$total_distance ~ daily_activity$total_steps))
```

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



c(`(Intercept)` = −0.316536036252617, `daily_activity$total_steps` = 0.000760186709838487)

The correlation function indicates that the correlation coefficient between the `total_steps` and `total_distance` is equal to 0.9852245 which indicates that the more steps people take, the greater distance they travel.

However, `n_distinct(full_day_cleaned$id)` outputs that there are only 33 distinct user IDs (people) that represent the data, therefore, most of the points on the scatter plot are multiple observations from a single person. Thus, by creating a pivot table that calculates the average of all the fields per ID, a more comprehensive scatter plot can be made.

```
daily_activity_pivot_table <- full_day_cleaned %>%
  group_by(id) %>%
  summarise(
   across(where(is.numeric),mean,na.rm = TRUE)
  ) %>%
  rename_with(~ paste0("avg_", .), -id)
```

```
## Warning: There was 1 warning in ‘summarise()‘.
## i In argument: ‘across(where(is.numeric), mean, na.rm = TRUE)‘.
## i In group 1: ‘id = 1503960366‘.
## Caused by warning:
## ! The ‘...‘ argument of ‘across()‘ is deprecated as of dplyr 1.1.0.
## Supply arguments directly to ‘.fns‘ through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
```

```
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

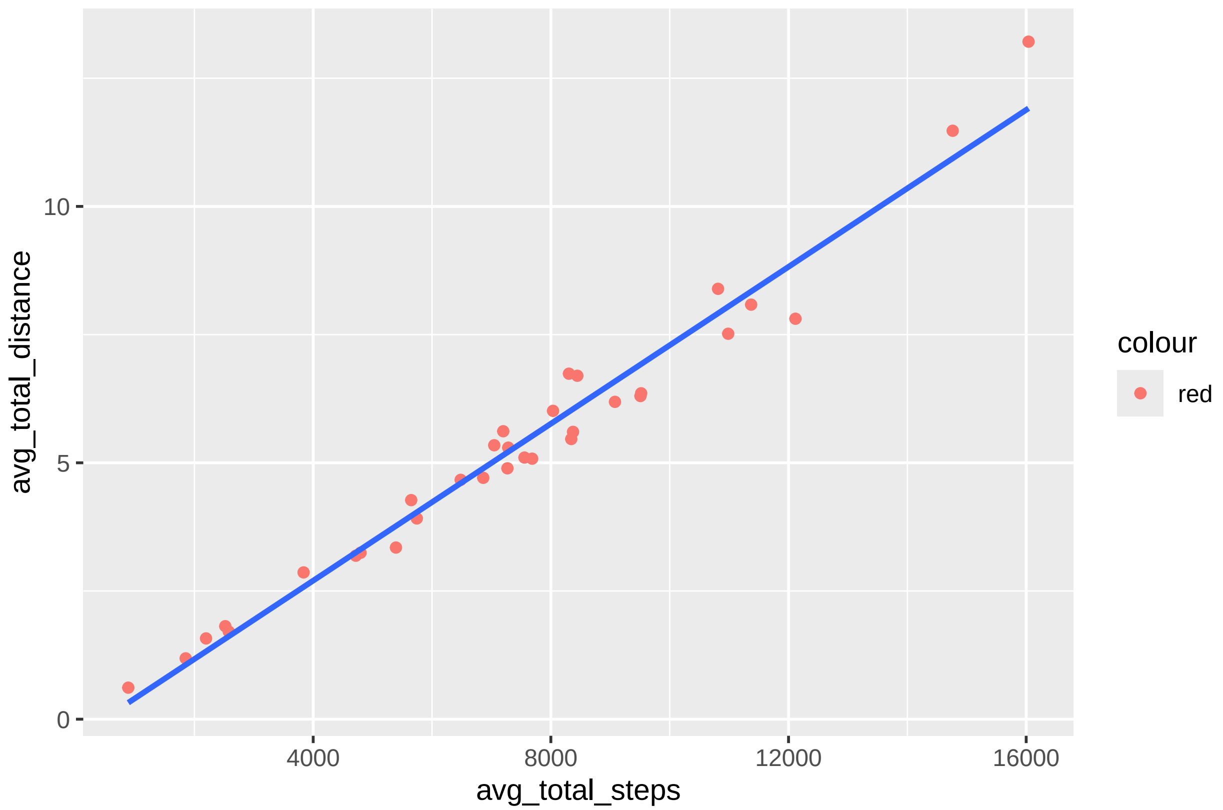
```
print(daily_activity_pivot_table)
```

```
## # A tibble: 33 x 20
##            id avg_total_steps avg_total_distance avg_tracker_distance
##         <dbl>           <dbl>              <dbl>                <dbl>
##  1 1503960366          12117.              7.81                 7.81
##  2 1624580081           5744.              3.91                 3.91
##  3 1644430081           7283.              5.30                 5.30
##  4 1844505072           2580.              1.71                 1.71
##  5 1927972279            888.              0.615                0.615
##  6 2022484408          11371.              8.08                 8.08
##  7 2026352035           5393.              3.35                 3.35
##  8 2320127002           4717.              3.19                 3.19
##  9 2347167796           9520.              6.36                 6.36
## 10 2873212765           7556.              5.10                 5.10
## # i 23 more rows
## # i 16 more variables: avg_logged_activities_distance <dbl>,
## #   avg_very_active_distance <dbl>, avg_moderately_active_distance <dbl>,
## #   avg_light_active_distance <dbl>, avg_sedentary_active_distance <dbl>,
## #   avg_very_active_minutes <dbl>, avg_fairly_active_minutes <dbl>,
## #   avg_lightly_active_minutes <dbl>, avg_sedentary_minutes <dbl>,
## #   avg_calories <dbl>, avg_total_sleep_records <dbl>, ...
```

Now, plotting `avg_total_distance` and `avg_steps_taken` reveals

```
ggplot(data=daily_activity_pivot_table, aes(x=avg_total_steps, y=avg_total_distance)) +
  geom_point(aes(color="red")) +
  geom_smooth(method="lm",se=FALSE) +
  labs(caption = lm(daily_activity_pivot_table$avg_total_distance ~ daily_activity_pivot_table$avg_total
```

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



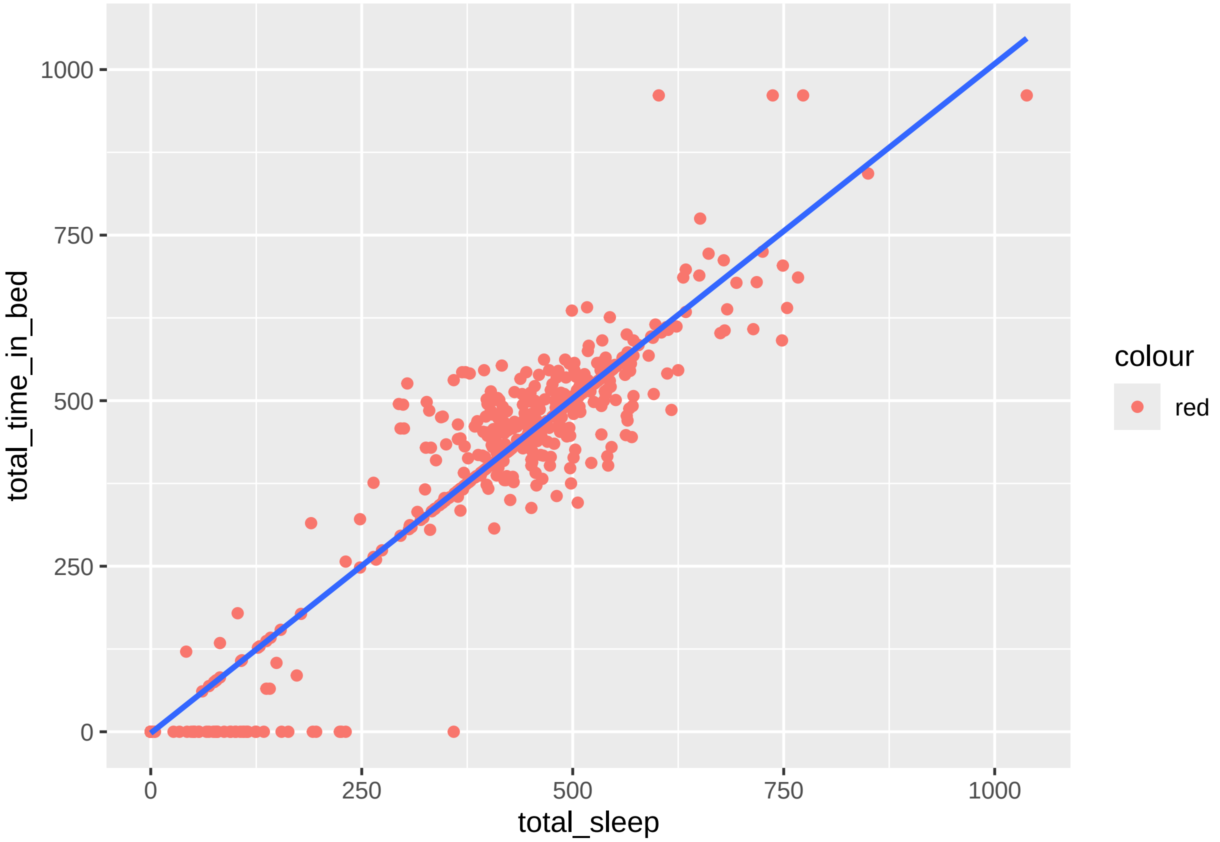
ept)` = −0.357857659404528, `daily_activity_pivot_table$avg_total_steps` = 0.000765000539793802)

This analysis returns a correlation of 0.9841508 which is a very strong positive correlation, indicating that steps actually translate into distance traveled and are a good measure of activity.

Another analysis we can look at is the relationship between minutes asleep and time in bed.

```
ggplot(data=full_day_cleaned, aes(x=total_sleep, y=total_time_in_bed)) +
  geom_point(aes(color="red")) +
  geom_smooth(method="lm",se=FALSE)
```

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



```
  labs(caption = lm(full_day_cleaned$total_time_in_bed ~ full_day_cleaned$total_sleep))
```

```
## $caption
##
## Call:
## lm(formula = full_day_cleaned$total_time_in_bed ~ full_day_cleaned$total_sleep)
##
## Coefficients:
##                  (Intercept)  full_day_cleaned$total_sleep
##                       -2.107                         1.011
##
##
## attr(,"class")
## [1] "labels"
```

The correlation between these two variables is 0.9802023 which indicates a strong positive, linear relationship. Some notable outliers are those at the far right extreme who spend the most minutes sleeping, but spend significantly more time in bed than predicted. These deviations occur between 10 to 13 hours of sleep which can indicate an oversleeping pattern that results in more time in bed. To further analyze this trend, the `sleep_day` table can be organized to show the observations of those with the most time in bed:

```
sleep_day %>%
  arrange(desc(total_time_in_bed)) %>%
  slice(1:10)
```

```
##            id             sleep_day total_sleep_records total_minutes_asleep
## 1  1644430081  5/2/2016 12:00:00 AM                   1                  796
## 2  1844505072 4/15/2016 12:00:00 AM                   1                  644
## 3  1844505072 4/30/2016 12:00:00 AM                   1                  722
## 4  1844505072  5/1/2016 12:00:00 AM                   1                  590
## 5  5553957443 4/30/2016 12:00:00 AM                   2                  775
## 6  1927972279 4/12/2016 12:00:00 AM                   3                  750
## 7  5553957443 4/23/2016 12:00:00 AM                   2                  631
## 8  4319703577 4/23/2016 12:00:00 AM                   1                  692
## 9  1503960366 4/17/2016 12:00:00 AM                   1                  700
## 10 7086361926 4/24/2016 12:00:00 AM                   1                  681
##    total_time_in_bed
## 1                961
## 2                961
## 3                961
## 4                961
## 5                843
## 6                775
## 7                725
## 8                722
## 9                712
## 10               704
```

Note that 3 of the 4 top observations for most time in bed come from one person (same user ID). In fact, there are only 24 user IDs in the entire table, so plotting each point as a separate observation does not necessarily help. Therefore, it is useful to create a pivot table that shows the `total_sleep_records`, `average_minutes_asleep` and `average_time_in_bed` for each user ID. This is accomplished with the following code:

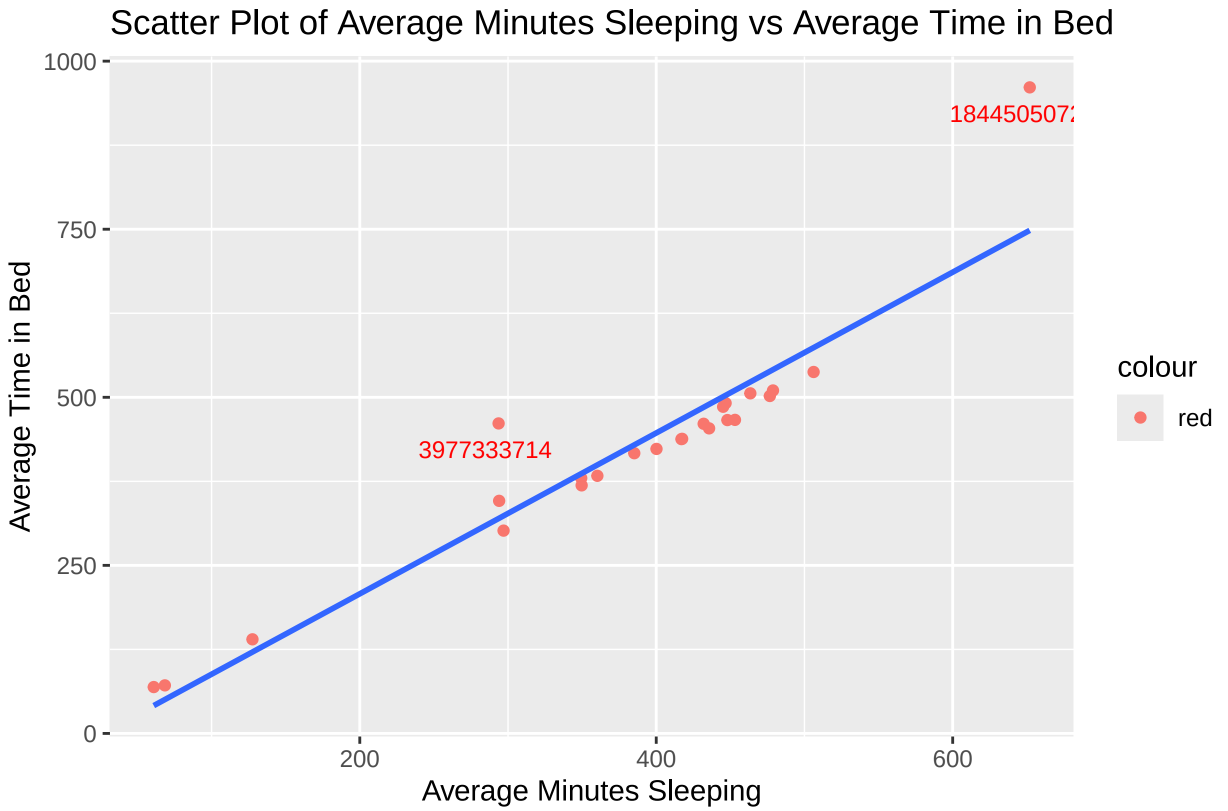
```
sleeping_pivot_table <- sleep_day %>%
  group_by(id) %>%
  summarise(
    average_minutes_sleeping = mean(total_minutes_asleep, na.rm = TRUE),
    average_time_in_bed = mean(total_time_in_bed, na.rm = TRUE)
  )
print(sleeping_pivot_table)
```

```
## # A tibble: 24 x 3
##            id average_minutes_sleeping average_time_in_bed
##         <dbl>                    <dbl>               <dbl>
##  1 1503960366                     360.                383.
##  2 1644430081                     294                 346
##  3 1844505072                     652                 961
##  4 1927972279                     417                 438.
##  5 2026352035                     506.                538.
##  6 2320127002                      61                  69
##  7 2347167796                     447.                491.
##  8 3977333714                     294.                461.
##  9 4020332650                     349.                380.
## 10 4319703577                     477.                502.
## # i 14 more rows
```

Plotting this data yields the following result.

```
outliers <- c(1844505072, 3977333714)

ggplot(data=sleeping_pivot_table, aes(x=average_minutes_sleeping, y=average_time_in_bed)) +
  geom_point(aes(color="red")) +
  geom_smooth(method="lm",se=FALSE) +
  geom_text(
    data = sleeping_pivot_table %>% filter(id %in% outliers),
    aes(label = id),
    vjust = +2,
    hjust = +0.6,
    color = "red",
    size = 3
  ) +
  labs(
    title = "Scatter Plot of Average Minutes Sleeping vs Average Time in Bed",
    x = "Average Minutes Sleeping",
    y = "Average Time in Bed"
  ) +
  labs(caption = lm(sleeping_pivot_table$average_time_in_bed ~ sleeping_pivot_table$average_minutes_sle
```

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



ept)` = −31.5273070802988, `sleeping_pivot_table$average_minutes_sleeping` = 1.19606919455067)

This scatter plot shows a correlation of 0.9403039 which indicates a strong positive linear correlation. In this scatter plot, there are namely two outliers, user 1844505072 and user 3977333714 who spend more time in bed than the average person for their given minutes asleep. This can be attributed to a possible error in the data or the fact that some of the sedentary minutes that the watch records are also considered as sleeping time even though the user is most likely awake.

Finally, I will analyse the heart rate data to see at what times of the day users are experiencing the highest heart rates and how that corresponds to their activity level rating.

```
#format the heart_rate table to break the activity day column by data
heart_rate$Time <- parse_date_time(
  heart_rate$Time,
  "%m/%d/%y %I:%M:%S %p"
)

heart_rate <- heart_rate %>%
  mutate(ActivityDay = date(Time))

heart_rate_AL <- heart_rate %>%
  left_join(AL, by = c("Id"="id"))

heart_rate_AL <- heart_rate_AL %>%
  mutate(
    Hour = format(Time, "%k")
  )
```

Next, we pivot by hour to find the activity day with the most recorded hours

```
#pivot by hour and calculate the average bpm for each our of each day for each user
heart_rate_pivot <- heart_rate_AL %>%
  group_by(Id,ActivityDay,Hour) %>%
  summarise(mean_bpm_hour = mean(Value))
```

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

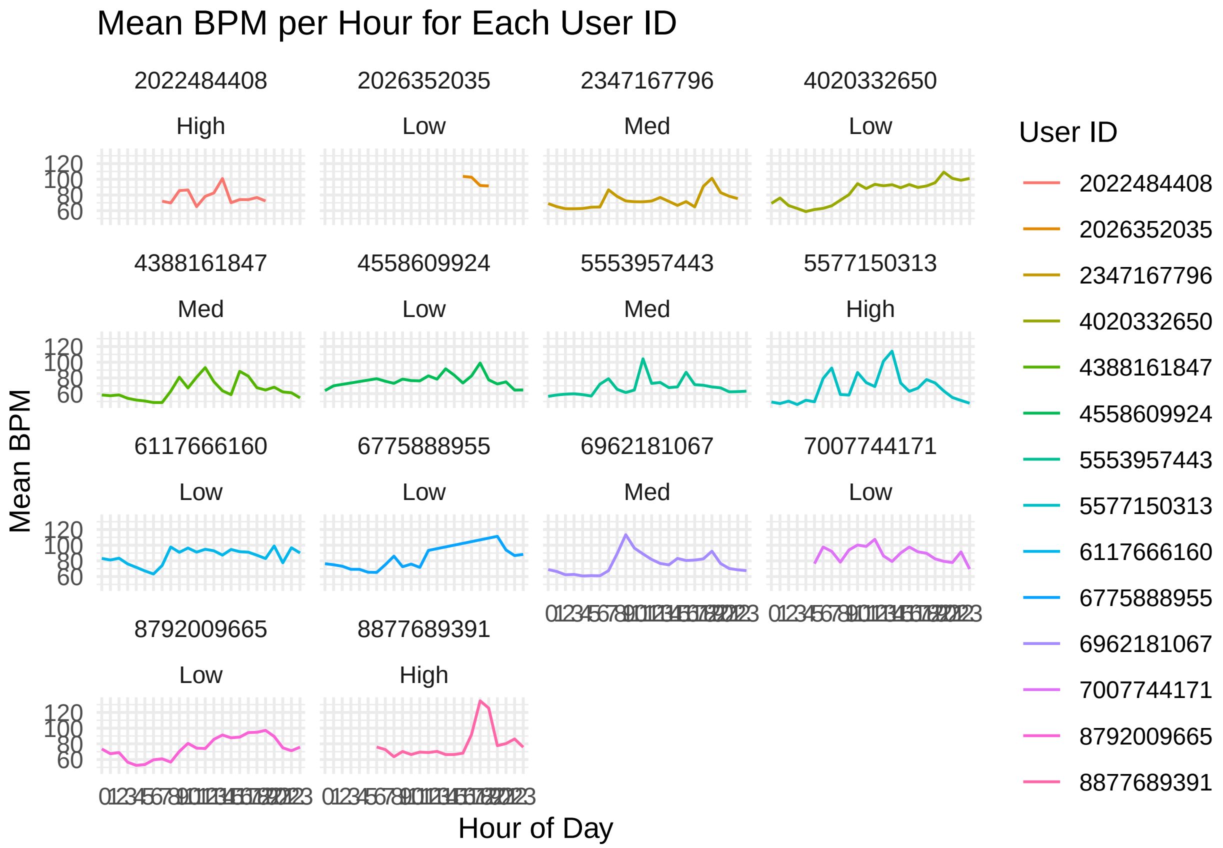
```
#select the day with the most recorded hours for each userID so that the analysis can be maximized
most_hours_day <- heart_rate_AL %>%
  group_by(Id, ActivityDay) %>%
  summarize(hours_recorded = n()) %>%
  ungroup() %>%
  group_by(Id) %>%
  filter(hours_recorded == max(hours_recorded)) %>%
  slice(1) %>%  # In case of ties, select the first occurrence
  select(Id, ActivityDay)
```

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

```
# Filter the data for the random day for each user
filtered_heart_rate <- heart_rate_pivot %>%
  inner_join(most_hours_day, by = c("Id", "ActivityDay" = "ActivityDay")) %>%
  inner_join(AL, by = c("Id" = "id"))
```

Finally plot the data

```
#finally, plot the data and facet wrap by the user ID and their acivity level
ggplot(filtered_heart_rate, aes(x=Hour, y = mean_bpm_hour, color = as.factor(Id),
                              group = interaction(Id,ActivityDay))) +
  geom_line() +
  facet_wrap(~Id~activity_level) +
  labs(title = "Mean BPM per Hour for Each User ID",
       x = "Hour of Day",
       y = "Mean BPM",
       color = "User ID") +
  theme_minimal()
```



We can see that people who Identify as high intensity users usually have one significant peak of high bpm throughout their day on average, indicating that most of their exercise is concentrated and completed at one time. For those who Identify as low exercisign people, their bpm line graphs are smoother and exhibit fewer peaks, rather more gradual increases which can suggest prolonged exercise at a lower intensity. Those who identify as medium exercisers usually have a noticeable peak, but may have more throughout the day at a lower bpm than the high intensity users.