Google Capstone Project: BellaBeat

JGM

2024-03-21

Load library

The following packages are loaded to be used for analysis.

```
library(tidyverse)
library(lubridate)
library(ggstatsplot)
library(ggplot2)
library(dplyr)
library(skimr)
```

Import dataset files to R

From the Kaggle website the files are then downloaded and stored in a folder where it can be accessed and analyzed. The files in csv format are then imported into RStudio.

```
daily_activity <-read_csv("dailyActivity_merged.csv")
daily_sleep <-read_csv("sleepDay_merged.csv")
```

Inspect loaded dataset

To check whether the correct files are load we then use the head function to get a quick look at our data. In this analysis only data activity and daily sleep are going to be used.

```
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>
```

```
head(daily_sleep)
```

```
## # A tibble: 6 x 5
##           Id SleepDay        TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <chr>                       <dbl>              <dbl>          <dbl>
## 1 1503960366 4/12/2016 12:0~                1                327            346
## 2 1503960366 4/13/2016 12:0~                2                384            407
## 3 1503960366 4/15/2016 12:0~                1                412            442
## 4 1503960366 4/16/2016 12:0~                2                340            367
## 5 1503960366 4/17/2016 12:0~                1                700            712
## 6 1503960366 4/19/2016 12:0~                1                304            320
```

Inspect columns of loaded dataset

```
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"
```

```
colnames(daily_sleep)
```

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

Convert dates to date type

In order to manipulate our data effectively we have to change our date which is in character type to a date type. This way we can further analyzed our data through labelling them in their proper days.

```
daily_activity1 <-daily_activity %>%
  mutate_at(vars(Id), as.character) %>%
  mutate_at(vars(ActivityDate), as.Date, format = "%m/%d/%y") %>%
  rename("Day"="ActivityDate")
daily_sleep1 <-daily_sleep %>%
  mutate_at(vars(Id), as.character) %>%
  mutate_at(vars(SleepDay), as.Date, format = "%m/%d/%y") %>%
  rename("Day"="SleepDay")
head(daily_activity1)
```

```
## # A tibble: 6 x 15
##   Id         Day        TotalSteps TotalDistance TrackerDistance
##   <chr>      <date>          <dbl>         <dbl>           <dbl>
```

```
## 1 1503960366 2020-04-12      13162          8.5             8.5
## 2 1503960366 2020-04-13      10735          6.97            6.97
## 3 1503960366 2020-04-14      10460          6.74            6.74
## 4 1503960366 2020-04-15       9762          6.28            6.28
## 5 1503960366 2020-04-16      12669          8.16            8.16
## 6 1503960366 2020-04-17       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>
```

```
head(daily_sleep1)
```

```
## # A tibble: 6 x 5
##   Id         Day        TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##   <chr>      <date>                 <dbl>              <dbl>          <dbl>
## 1 1503960366 2020-04-12                 1                327            346
## 2 1503960366 2020-04-13                 2                384            407
## 3 1503960366 2020-04-15                 1                412            442
## 4 1503960366 2020-04-16                 2                340            367
## 5 1503960366 2020-04-17                 1                700            712
## 6 1503960366 2020-04-19                 1                304            320
```

Join datasets to one dataset

Merging our dataset makes our study easier since we can perform analysis by just working with one dataset.

```
daily_data <-daily_sleep1 %>%
  right_join(daily_activity1, by=c("Id","Day")) %>%
  mutate(Weekday = weekdays(as.Date(Day, "m/%d/%Y")))
```

Check for duplicated and null data

We check for duplicated and null data in order to see whether we want to remove them and avoid contamination within our results.

```
daily_data <-daily_data[!duplicated(daily_data), ]
sum(is.na(daily_data))
```

```
## [1] 1590
```

```
n_distinct(daily_data$Id)
```

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

Check for unique IDs

We check for unique IDs to see the amount of data we are working with. Within daily activity we have 33 unique IDs and in daily sleep we have 24 unique IDs. In general these data sets are lacking and adding more would benefit the results of our analysis.

```
n_unique(daily_activity$Id)
```

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

```
n_unique(daily_sleep$Id)
```

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

Gather the summarized data

By viewing the summary of the data we have preliminary look at our analysis. We can see the results of the gathered data using the FitBit Tracker.

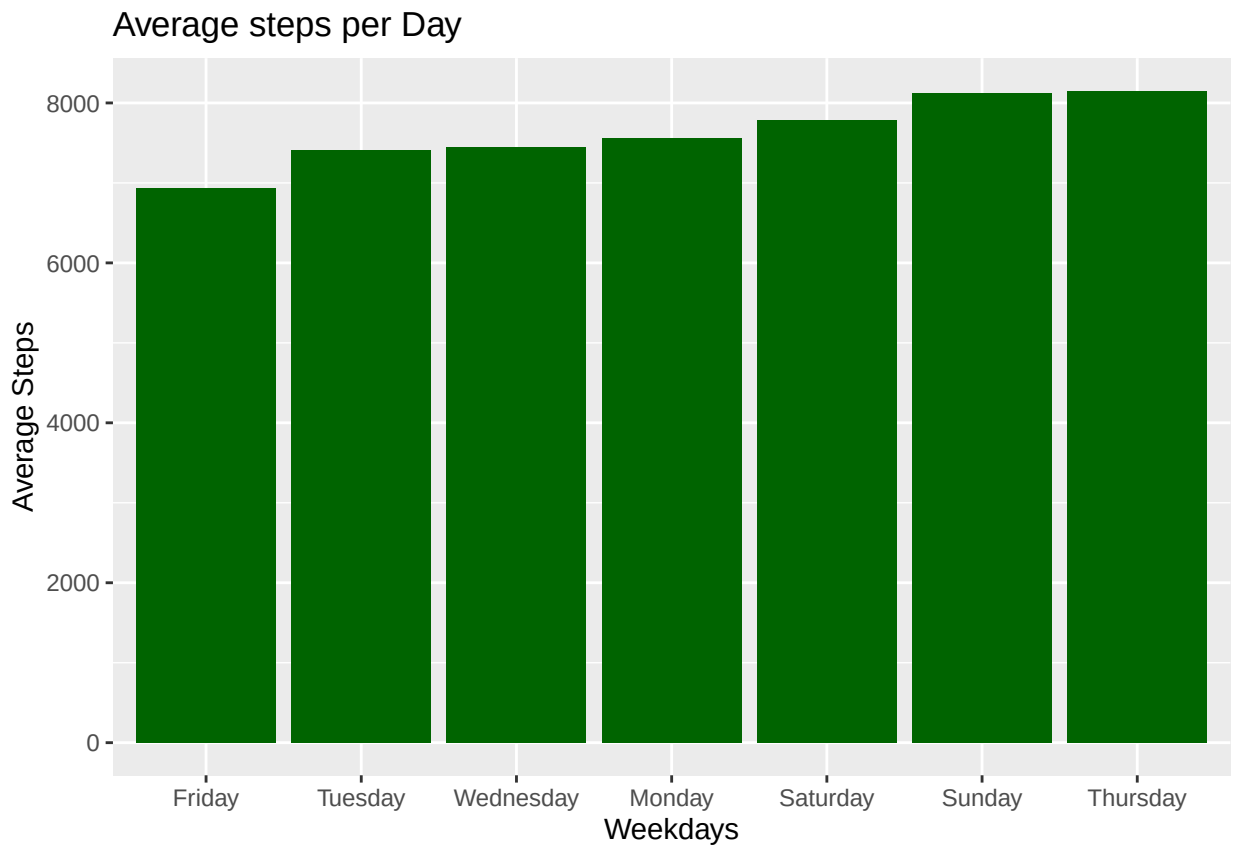
```
daily_data %>%
  select(Calories,TotalMinutesAsleep, TotalDistance, TotalSteps, VeryActiveMinutes, FairlyActiveMinutes
  summary()
```

```
##     Calories    TotalMinutesAsleep TotalDistance      TotalSteps
##  Min.   :   0   Min.   : 58.0      Min.   : 0.000   Min.   :    0
##  1st Qu.:1828   1st Qu.:361.0      1st Qu.: 2.620   1st Qu.: 3790
##  Median :2134   Median :432.5      Median : 5.245   Median : 7406
##  Mean   :2304   Mean   :419.2      Mean   : 5.490   Mean   : 7638
##  3rd Qu.:2793   3rd Qu.:490.0      3rd Qu.: 7.713   3rd Qu.:10727
##  Max.   :4900   Max.   :796.0      Max.   :28.030   Max.   :36019
##                 NA's   :530
##  VeryActiveMinutes FairlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :1440.0
##
```

Display Bar Chart for Average steps per Day

By performing an analysis for Average steps per day we can see that the users have actively been using their app to track their progress. Averaging within 7638 steps each day. This insight give us an understanding about the user's day to day lifestyle. A study found in 2020 that participants who take 8000 steps or more have 51% less risk from dying found in this article.

```r
average_steps <- daily_data %>%
  group_by(Weekday) %>%
  summarize(Average_steps = mean(TotalSteps, na.rm = TRUE))
ggplot(average_steps, aes(x = reorder(Weekday, Average_steps), y = Average_steps)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  labs(x = "Weekdays", y = "Average Steps", title="Average steps per Day")
```



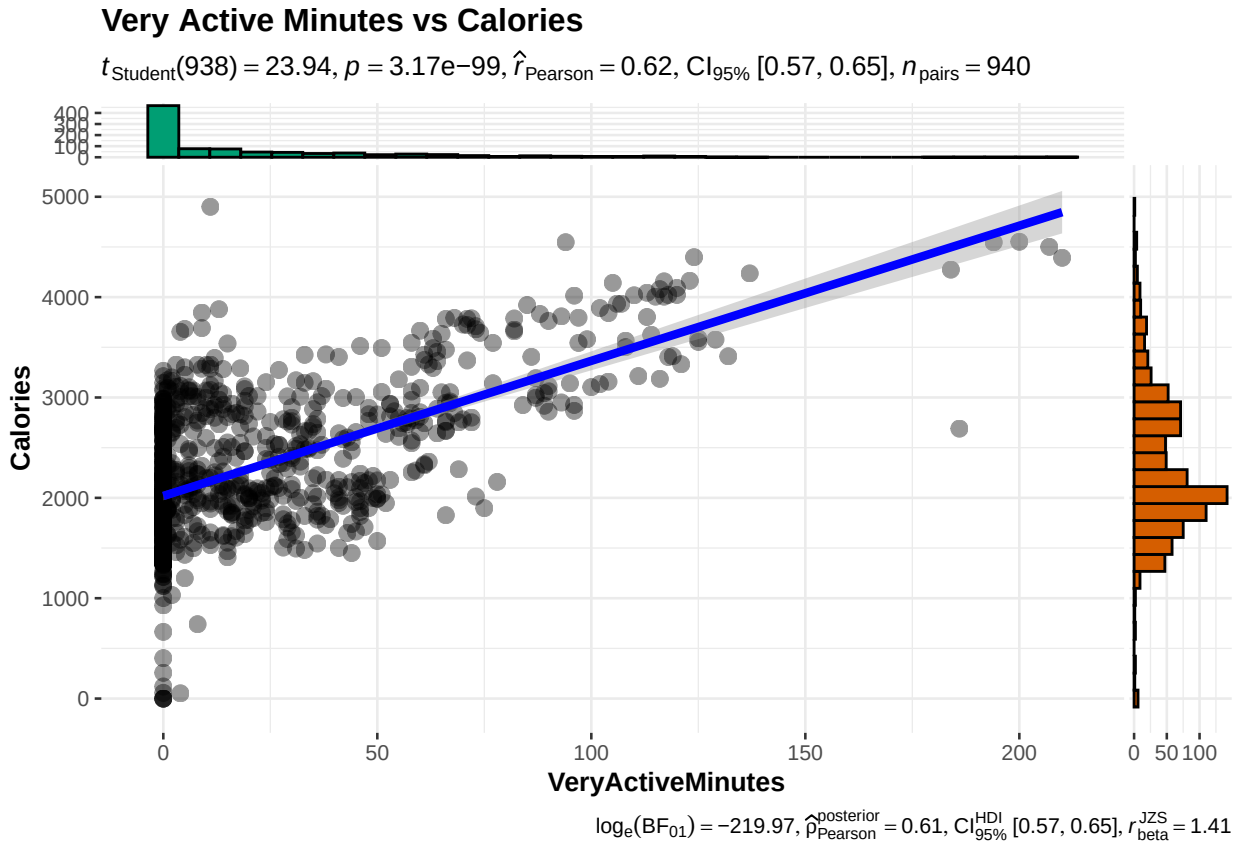
Display Plot with correlation for Calories burned per Very Active Minutes

From the chart below we can see that there is correlation between calories burned per very active minutes. This is by virtue that the app tracks the heart rate of the user labelling a certain bpm per person depending if they are doing a very active work or not. In this case as the calorie increases so does the very active minutes. Whereas, if we relate the calorie burned per fairly active minutes it does not have much of a correlation which is supported with the fact that doing non work intensive tasks our body does not burn much calories compared to work intensive tasks. By viewing the correlation value we can see that for Very Active Minutes vs Calories it is 0.62 which translates to a strong association while Fairly Active Minutes vs Calories have a 0.30 correlation value which means it has weak association with each other.

```r
ggscatterstats(
  data = daily_data,
  x=VeryActiveMinutes,
  y=Calories,
  type="parametric"
) + labs(title="Very Active Minutes vs Calories")
```

```
## Registered S3 method overwritten by 'ggside':
##   method from
##   +.gg   ggplot2
```

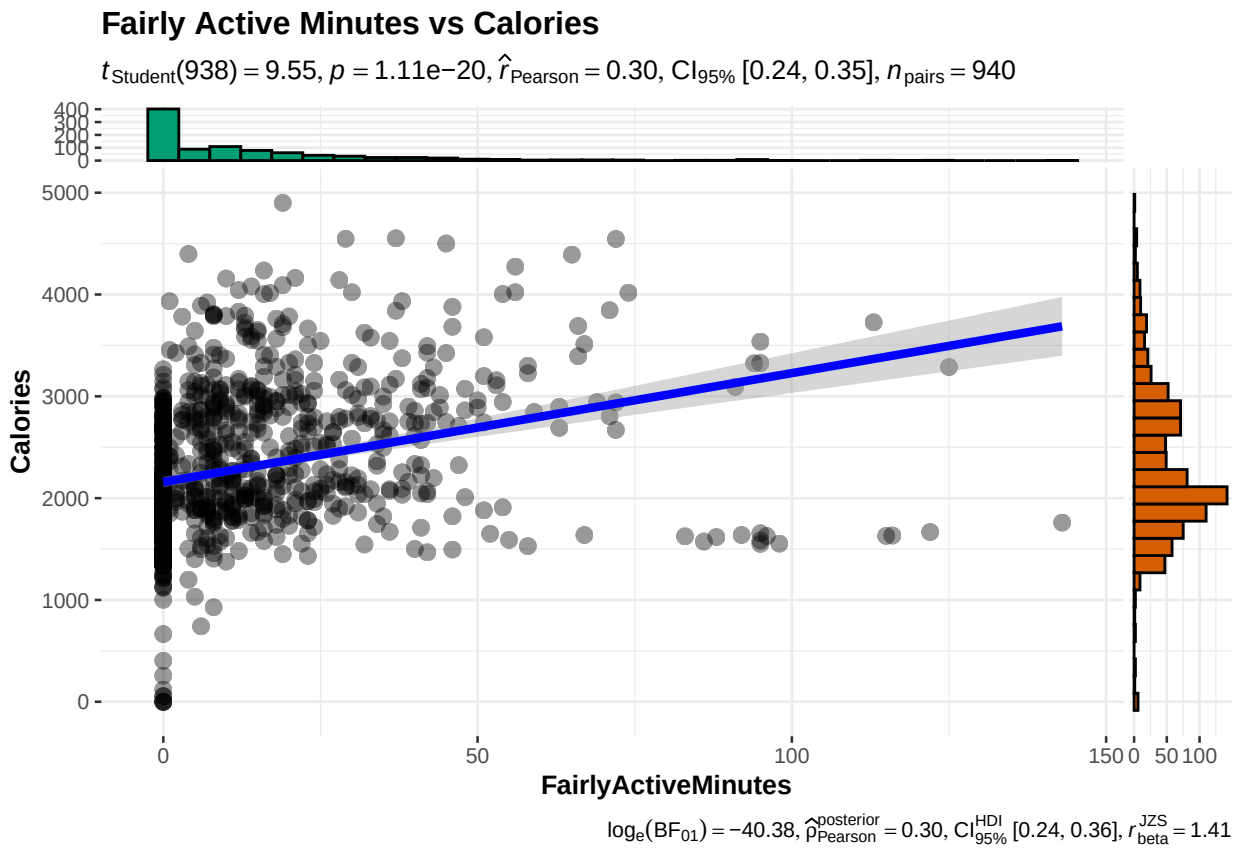
```
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```



Display Plot with correlation for Calories burned per Fairly Active Minutes

```
ggscatterstats(
  data = daily_data,
  x=FairlyActiveMinutes,
  y=Calories,
  type="parametric"
) + labs(title="Fairly Active Minutes vs Calories")
```

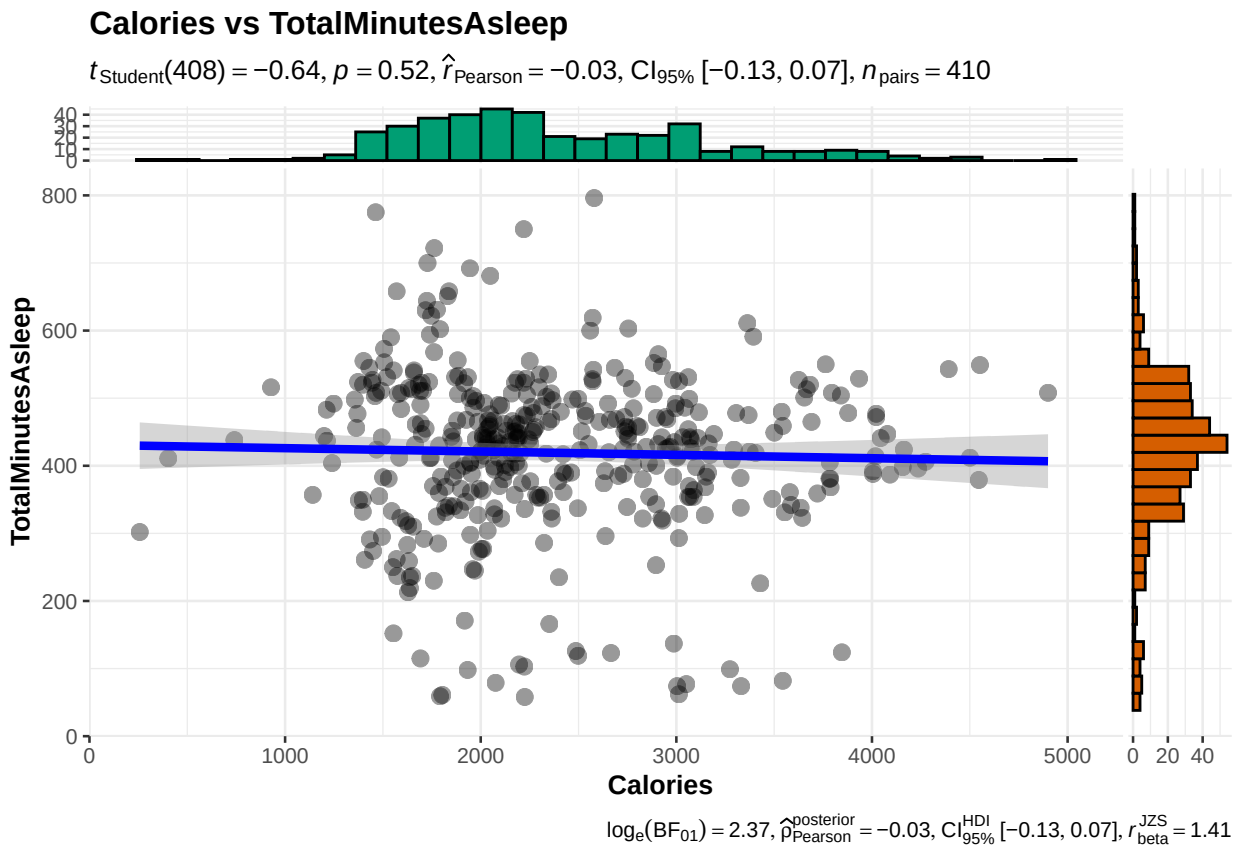
```
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```



Display Plot with correlation for Calories burned versus Total Minutes Asleep From the results shown it is clear that the calories burned does not relate to total minutes asleep. This is further supported by the correlation value of -0.03 which means that it has weak association with each other.

```
ggscatterstats(
  data = daily_data,
  x=Calories ,
  y=TotalMinutesAsleep    ,
  type="parametric"
) + labs(title="Calories vs TotalMinutesAsleep")
```

```
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```

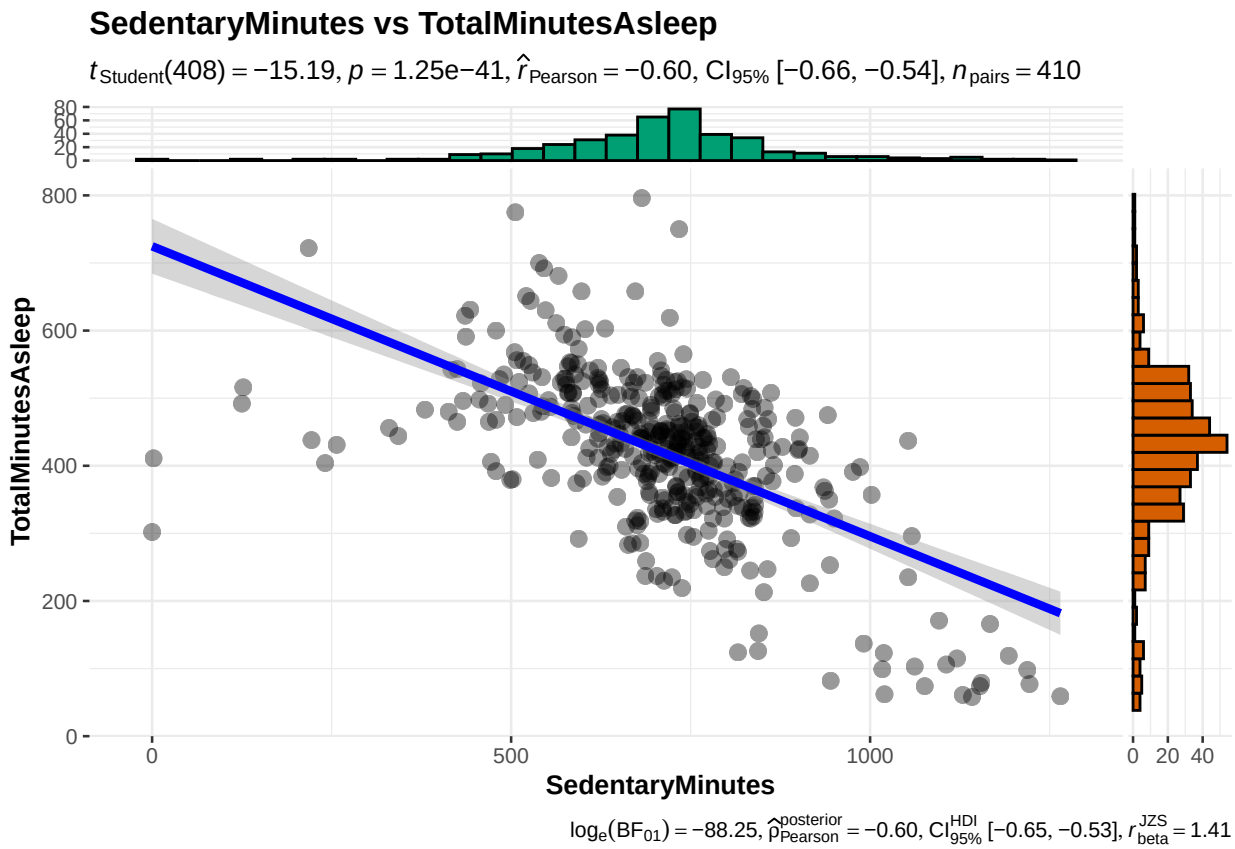


Display Plot with correlation for Sedentary Minutes versus Total Minutes Asleep

Meanwhile there is an inverse correlation between sedentary minutes versus total minutes asleep as indicated in this graph. The correlation value of the data is -0.60 which means it has a strong association. In other words, as the sedentary minutes increases the total minutes of sleep decreases.

```
ggscatterstats(
  data = daily_data,
  x=SedentaryMinutes,
  y=TotalMinutesAsleep    ,
  type="parametric"
) + labs(title="SedentaryMinutes vs TotalMinutesAsleep")
```

```
## `stat_xsidebin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_ysidebin()` using `bins = 30`. Pick better value with `binwidth`.
```


SedentaryMinutes vs TotalMinutesAsleep
$t_{Student}(408) = -15.19$, $p = 1.25e{-}41$, $\hat{r}_{Pearson} = -0.60$, $CI_{95\%}$ [−0.66, −0.54], $n_{pairs} = 410$
TotalMinutesAsleep
SedentaryMinutes
$\log_e(BF_{01}) = -88.25$, $\hat{\rho}^{posterior}_{Pearson} = -0.60$, $CI^{HDI}_{95\%}$ [−0.65, −0.53], $r^{JZS}_{beta} = 1.41$