Bellabeat Case Study

I have already used SQL and pivot tables in spreadsheets to do some initial analysis. I want to rely on R programming language to help create visalizations of the data. I will begin by loading the appropriate packages.

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
```

I will now upload my datasets. Because I was able to combine the hourly data in spreadsheets, I can upload that on it's own, but I am going to have to rely on joins to aggregate the sleep data into the daily dataset.

```
dailyactivity <- read_csv("Bellabeat_Case_Study/Bellabeat_daily.csv")
```

```
## Rows: 940 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDate
## dbl (6): Id, TotalSteps, TotalDistance, TotalActiveMinutes, SedentaryMinutes...
##
## 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.
```

```
dailysleep <- read_csv("Bellabeat_Case_Study/Bellabeat_sleep.csv")
```

```
## Rows: 413 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDate
## dbl (3): Id, 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.
```

```
hourly <- read_csv("Bellabeat_Case_Study/Bellabeat_hourly.csv")
```

```
## Rows: 22099 Columns: 6
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Activity Day
## dbl  (4): Id, Calories, Total Intensities, Steps
## time (1): Activity Hour
##
## 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.
```

```
weekdaysleep <- read_csv("Bellabeat_Case_Study/Bellabean Case Study - Avg_Day_Sleep.csv")
```

```
## Rows: 504 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Weekday, ActivityDate, Avg_Distance
## dbl (4): Avg_Sedementary, Avg_Sleep, Avg_Steps, Avg_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.
```

```
head(dailyactivity)
```

```
## # A tibble: 6 x 7
##       Id ActivityDate TotalSteps TotalDistance TotalActiveMinu~ SedentaryMinutes
##    <dbl> <chr>             <dbl>         <dbl>            <dbl>            <dbl>
## 1 1.50e9 4/12/2016         13162          8.5               366              728
## 2 1.50e9 4/13/2016         10735          6.97              257              776
## 3 1.50e9 4/14/2016         10460          6.74              222             1218
## 4 1.50e9 4/15/2016          9762          6.28              272              726
## 5 1.50e9 4/16/2016         12669          8.16              267              773
## 6 1.50e9 4/17/2016          9705          6.48              222              539
## # ... with 1 more variable: Calories <dbl>
```

```
head(dailysleep)
```

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

```
head(hourly)
```

```
## # A tibble: 6 x 6
##           Id `Activity Day` `Activity Hour` Calories `Total Intensities` Steps
##        <dbl> <chr>          <time>             <dbl>                <dbl> <dbl>
## 1 1503960366 4/12/2016      00:00                 81                   20   373
## 2 1503960366 4/12/2016      01:00                 61                    8   160
## 3 1503960366 4/12/2016      02:00                 59                    7   151
## 4 1503960366 4/12/2016      03:00                 47                    0     0
## 5 1503960366 4/12/2016      04:00                 48                    0     0
## 6 1503960366 4/12/2016      05:00                 48                    0     0
```

```
head(weekdaysleep)
```

```
## # A tibble: 6 x 7
##   Weekday   ActivityDate Avg_Sedementary Avg_Sleep Avg_Steps Avg_Calories
##   <chr>     <chr>                  <dbl>     <dbl>     <dbl>        <dbl>
## 1 Tuesday   4/12/2016               750.      442.     7506.        2439.
## 2 Wednesday 4/13/2016               766.      430.     6103         2208.
## 3 Thursday  4/14/2016               743.      445.     7626.        2426.
## 4 Friday    4/15/2016               777.      427.     7472.        2317.
## 5 Saturday  4/16/2016               710.      392.     8615.        2494.
## 6 Sunday    4/17/2016               707.      464.     6530.        2252.
## # ... with 1 more variable: Avg_Distance <chr>
```

And now to join the daily data.

```
daily_join <- right_join(dailyactivity, dailysleep, by = c('Id'='Id','ActivityDate'='SleepDate'))
head(daily_join)
```

```
## # A tibble: 6 x 9
##       Id ActivityDate TotalSteps TotalDistance TotalActiveMinu~ SedentaryMinutes
##    <dbl> <chr>             <dbl>         <dbl>            <dbl>            <dbl>
## 1 1.50e9 4/12/2016         13162          8.5               366              728
## 2 1.50e9 4/13/2016         10735          6.97              257              776
## 3 1.50e9 4/15/2016          9762          6.28              272              726
## 4 1.50e9 4/16/2016         12669          8.16              267              773
## 5 1.50e9 4/17/2016          9705          6.48              222              539
## 6 1.50e9 4/19/2016         15506          9.88              345              775
## # ... with 3 more variables: Calories <dbl>, TotalMinutesAsleep <dbl>,
## #   TotalTimeInBed <dbl>
```

Now we have a complete clean dataset for the daily data. So now we are going to try out some different vizualizations.

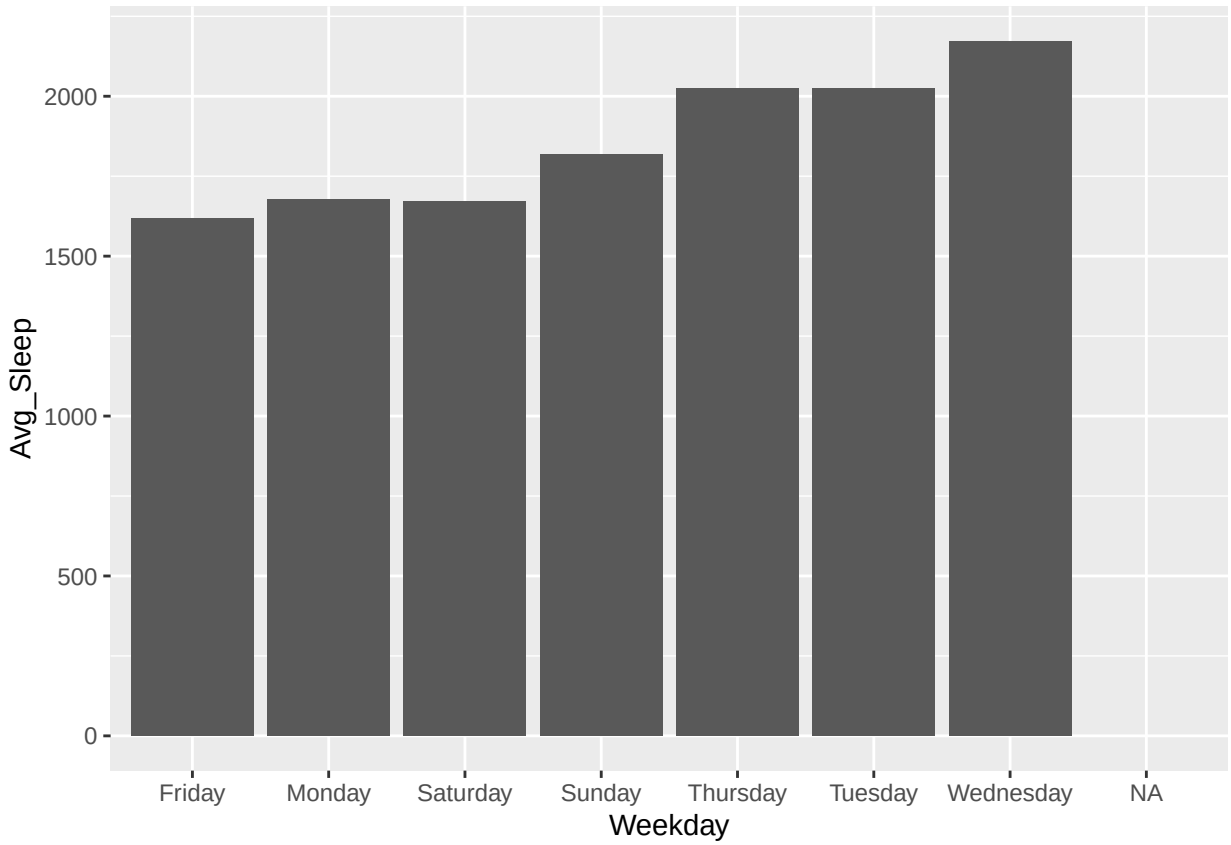
The first question became what do we qualify as a healthy user? After some research, these are the following statistics we are going to use *Even though gender is not specified, I'm going to use suggested stats related to women* - Recommended for adults to get between 7-9 hours of sleep a night - Recommended for adults to get around 10,000 steps a day - Recommended for adult women to burn around 2,000 calories a day

I will use these stats to qualify a "healthy" user.

Using spreadsheets and SQL I have already discovered: - 33 users provided data - 24 users provided sleep data - 413 inidividual sleeps recorded - Only 119 sleep days that fall between 7-9 hours - 19 out of the 24 users averaged suggested sleep over the 30 days

```
ggplot(weekdaysleep, mapping = aes(x=Weekday, y=Avg_Sleep)) + geom_col()
```

```
## Warning: Removed 473 rows containing missing values (position_stack).
```

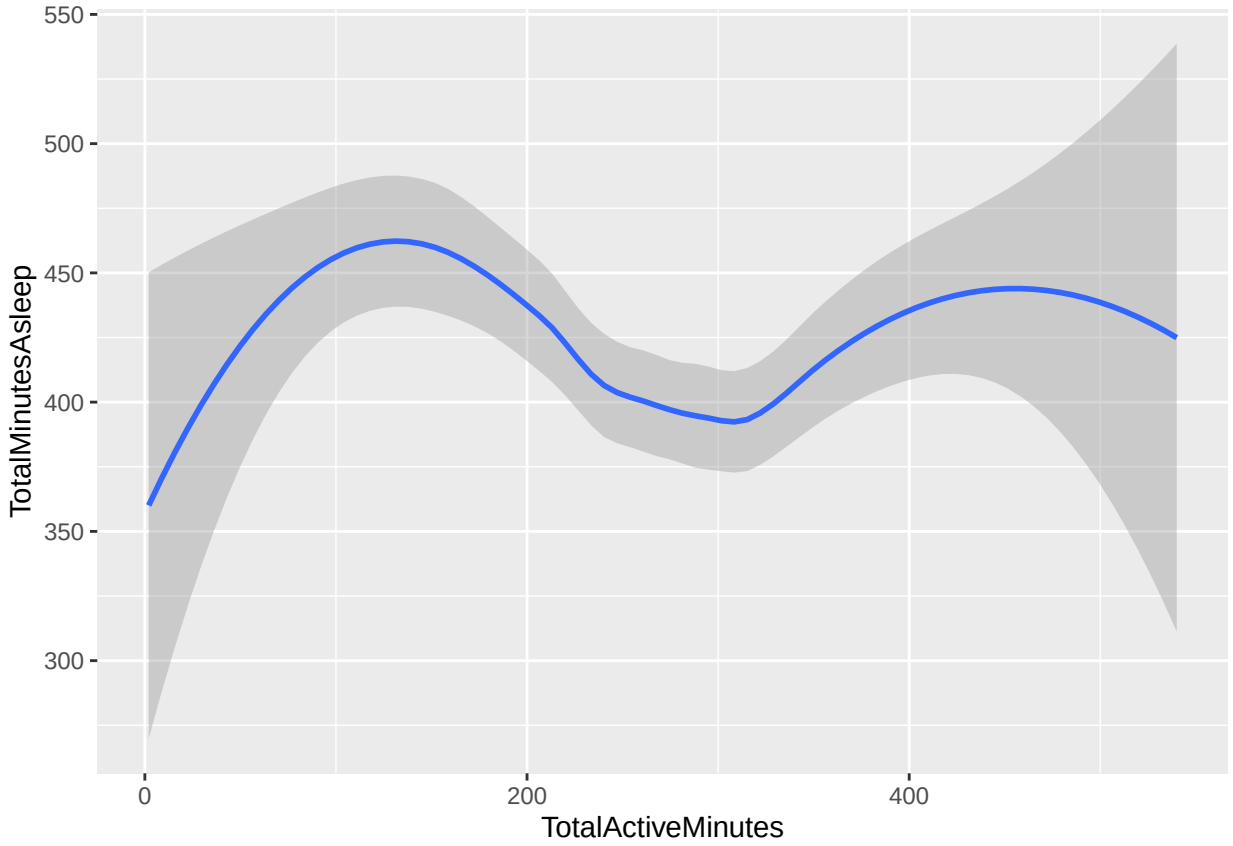


This first visualization shows the average sleep grouped by weekday. So on average Friday's, Saturday's, and Monday's get the least amount of sleep.

So the next question is how do we help users get more sleep? My hypothesis is that increased activity leads to increased sleep.

```
ggplot(daily_join, mapping = aes(x=TotalActiveMinutes,y=TotalMinutesAsleep)) + geom_smooth()
```

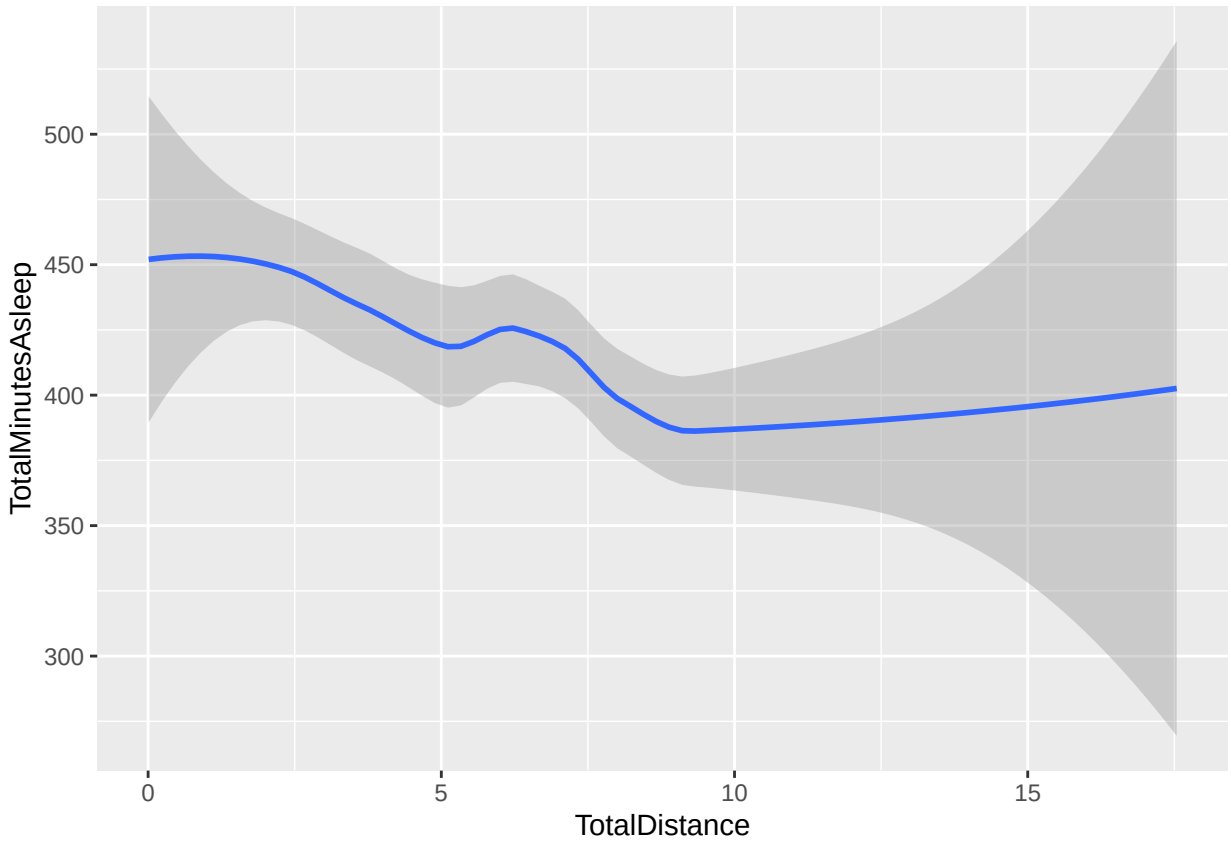
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



There actually isn't a real correlation between the two. So using my remaing datapoints, I'm going to check for correlation between them

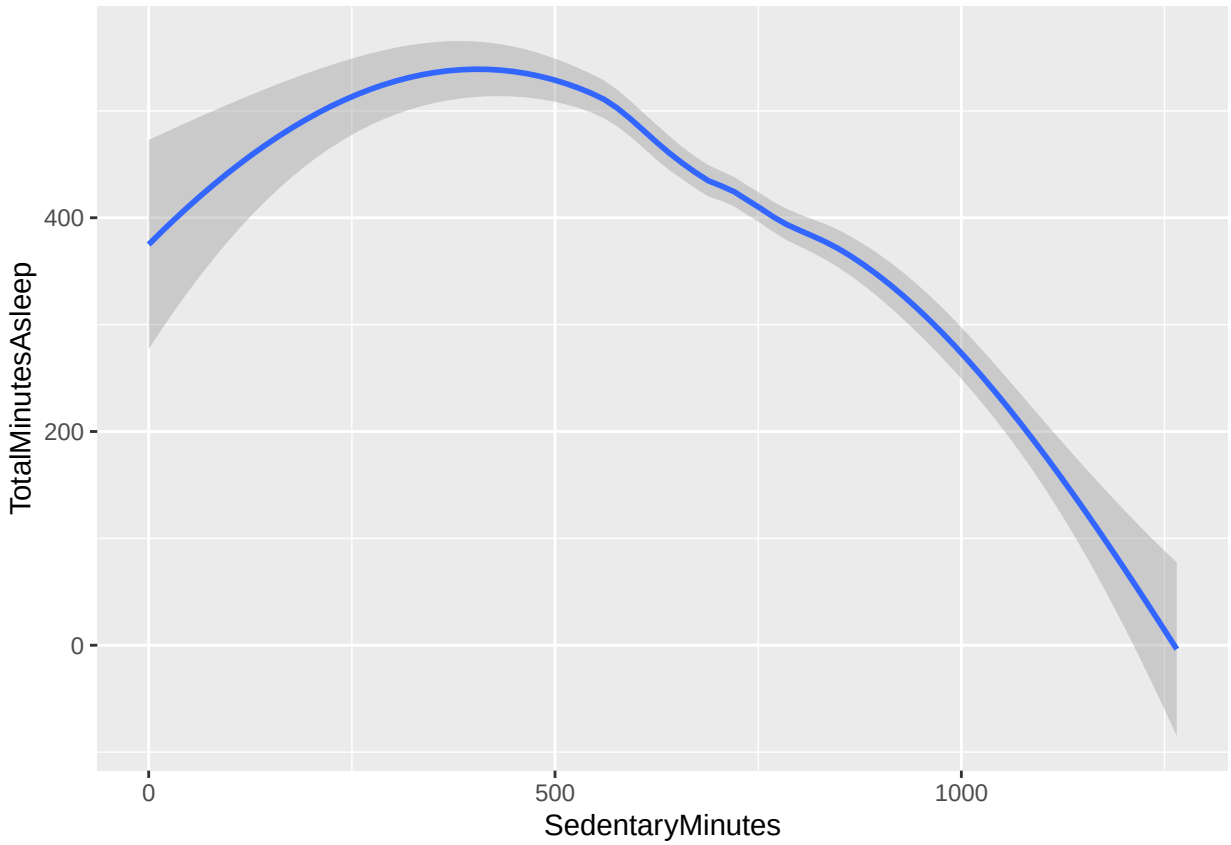
```
ggplot(daily_join, mapping = aes(x=TotalDistance, y=TotalMinutesAsleep)) + geom_smooth()
```

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




```
ggplot(daily_join, mapping = aes(x=SedentaryMinutes,y=TotalMinutesAsleep)) + geom_smooth()
```

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



Of the three there is a slight negative correlation between sedentary minutes and total minutes asleep. So rather than increased exercise, it just comes down to reducing sedementary time to increase sleep.

```
AvgId <- aggregate(cbind(Active <- daily_join$TotalActiveMinutes, Sedentary <- daily_join$SedentaryMinu
AvgId <- setNames(AvgId, c("Id", "ActivityMin", "SedentaryMin", "Steps", "SleepMin"))
arrange(AvgId, desc(SleepMin), .by_group = FALSE)
```

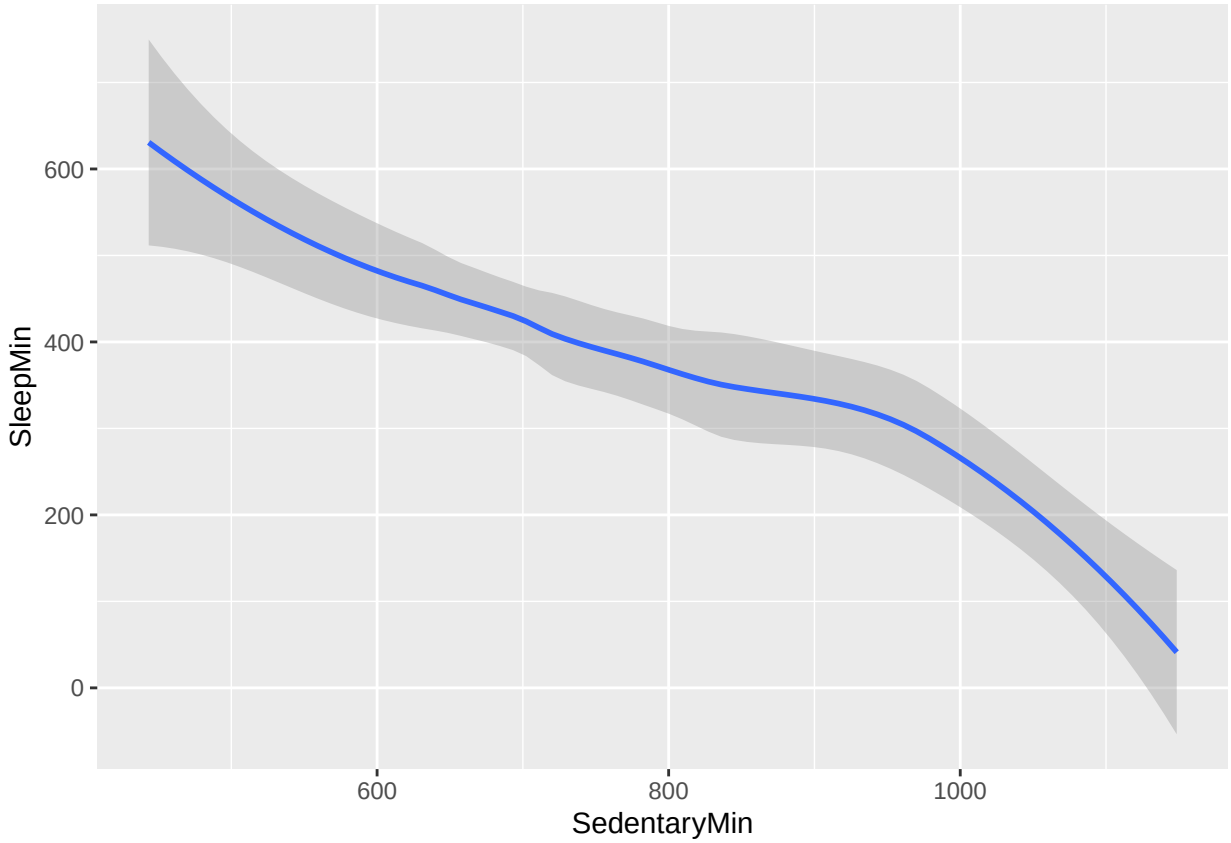
```
##            Id ActivityMin SedentaryMin     Steps SleepMin
## 1  1844505072    147.3333     443.3333  3477.000 652.0000
## 2  2026352035    256.8929     653.9643  5618.679 506.1786
## 3  6117666160    364.5556     531.9444  8823.833 478.7778
## 4  4319703577    259.2308     642.6923  7125.423 476.6538
## 5  5553957443    242.6129     668.3548  8612.581 463.4839
## 6  7086361926    234.0000     723.6667 10290.500 453.1250
## 7  6962181067    287.1290     662.3226  9794.806 448.0000
## 8  2347167796    271.2000     628.4000  8533.200 446.8000
## 9  8378563200    226.5625     715.3750  8832.938 443.3438
## 10 8792009665    178.4000     807.8000  3443.267 435.6667
## 11 5577150313    296.2308     667.3077  9260.077 432.0000
## 12 4702921684    292.1429     693.0357  9226.357 421.1429
## 13 1927972279     85.0000     977.2000  1490.000 417.0000
## 14 4388161847    286.7500     751.4583 10974.708 403.1250
## 15 4445114986    217.9643     787.3214  4756.179 385.1786
## 16 1503960366    291.3200     759.2800 12405.680 360.2800
## 17 6775888955    107.0000     964.0000  3499.000 349.6667
## 18 4020332650    249.0000     841.8750  6596.750 349.3750
## 19 8053475328    301.0000     837.3333 19078.667 297.0000
## 20 1644430081    263.2500     920.5000  7967.750 294.0000
```

```
## 21 3977333714    262.6429     716.2143 11218.000 293.6429
## 22 4558609924    313.0000    1028.4000  8139.000 127.6000
## 23 7007744171    220.0000    1148.5000  5115.500  68.5000
## 24 2320127002    242.0000    1129.0000  5079.000  61.0000
```

Ok so that gave me a tibble of the basic averages per each user. So I have average active minutes, average sedentary minutes, and average sleep respectively. I now want to see if the averages have any more correlation.

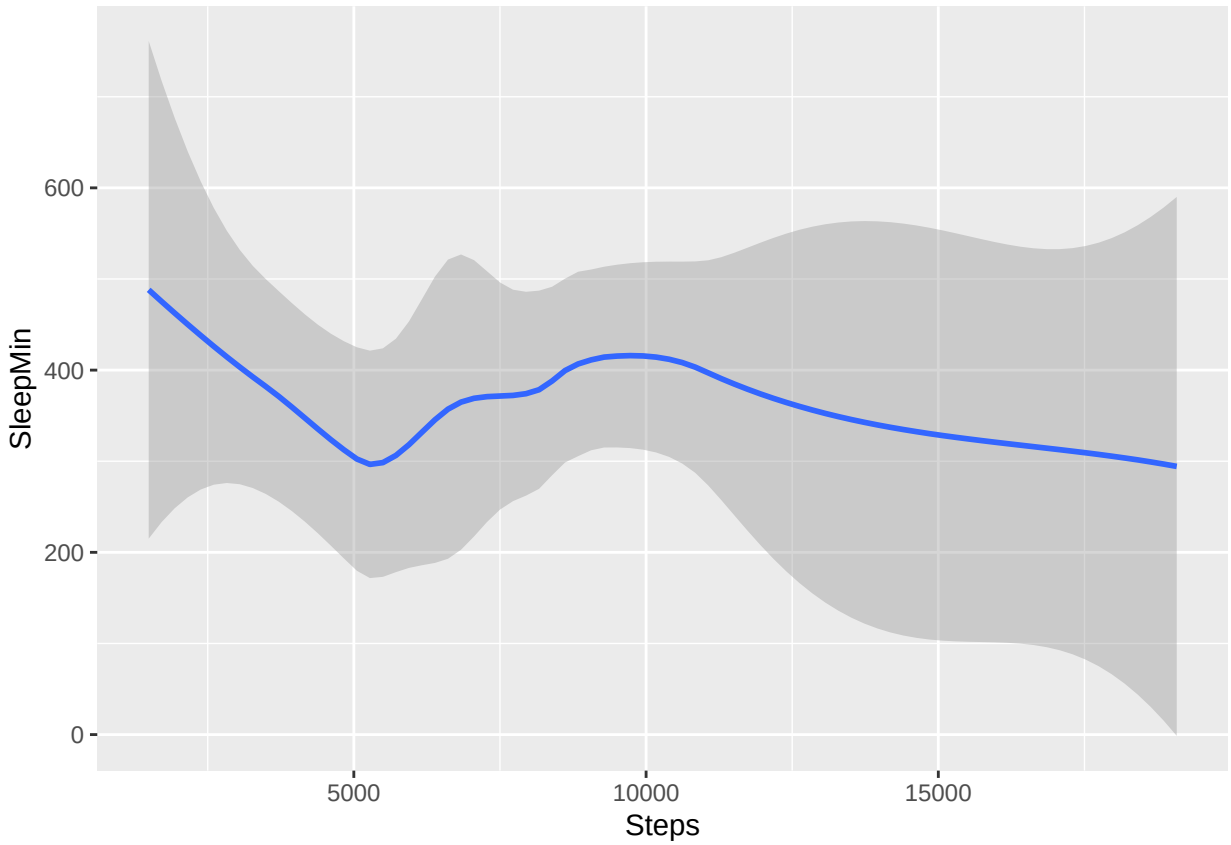
```
ggplot(AvgId, mapping = aes(x=SedentaryMin, y=SleepMin)) + geom_smooth()
```

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



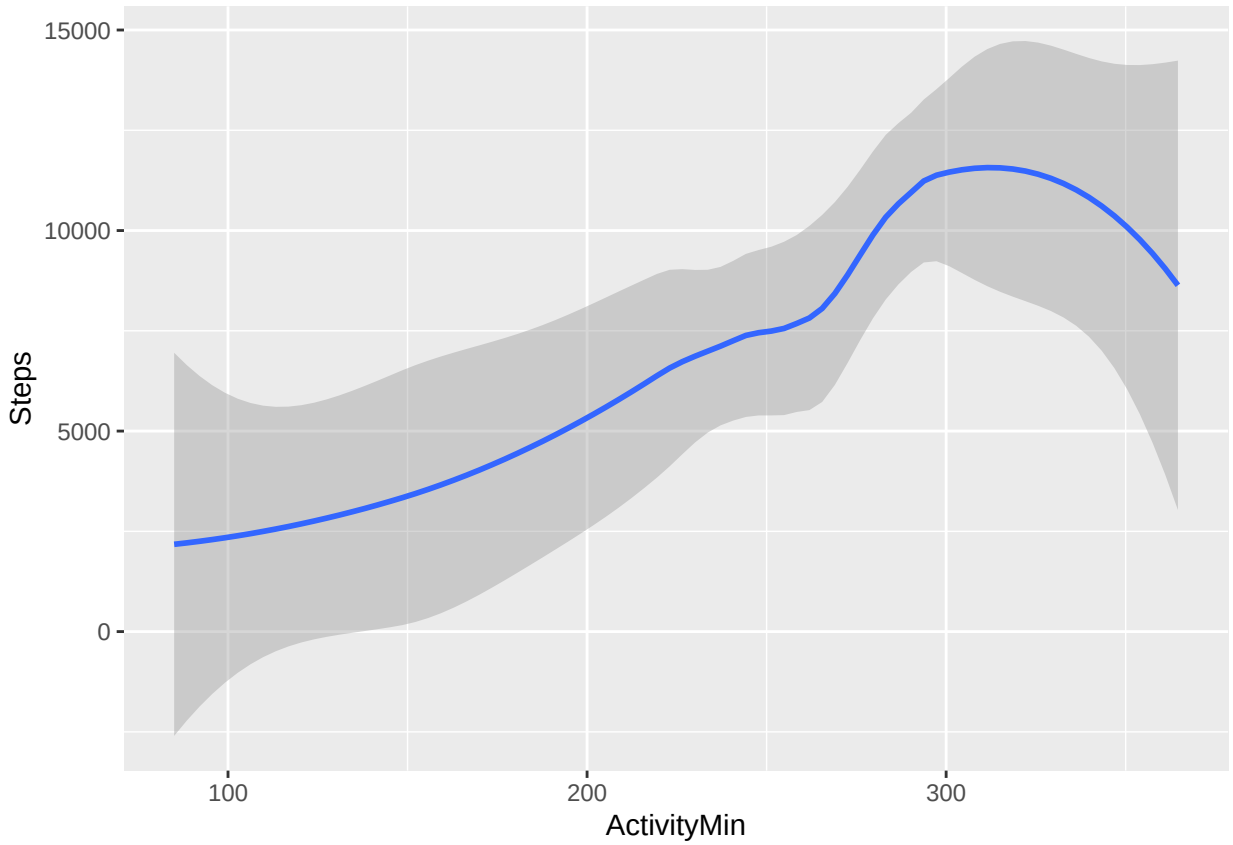

```
ggplot(AvgId, mapping = aes(x=Steps, y=SleepMin)) + geom_smooth()
```

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




```
ggplot(AvgId, mapping = aes(x=ActivityMin, y=Steps)) + geom_smooth()
```

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



Even using a different set of averages, it still shows a negative correlation between time spent sedentary and time spend asleep.

```
hourly_mod <- aggregate(cbind(hourly$Calories, as.integer(hourly$`Total Intensities`), hourly$Steps), li
setNames(hourly_mod, c("ActivityHour", "Calories", "Intensity", "Steps"))
```

```
##    ActivityHour Calories Intensity   Steps
## 1      00:00:00    67066      6833  131124
## 2      01:00:00    65464      3272   47141
## 3      02:00:00    64551      1727   29228
## 4      03:00:00    63013       907   10893
## 5      04:00:00    63620       354    3345
## 6      05:00:00    76152       156    1972
## 7      06:00:00    80994       836   12733
## 8      07:00:00    87959      2353   35496
## 9      08:00:00    96207      8269  172714
## 10     09:00:00    98819     20870  990153
## 11     10:00:00   102618     14785  411935
## 12     11:00:00   101791     13161  428972
## 13     12:00:00   108056     17006  516302
## 14     13:00:00   106200     19611  652296
## 15     14:00:00   106590     21448  678820
## 16     15:00:00    97573     13349  412622
## 17     16:00:00   102788     17218  515561
## 18     17:00:00   111214     18882  632525
## 19     18:00:00   111884     39551 1377344
## 20     19:00:00   110065     28600  891787
## 21     20:00:00    92736     28630  983158
```

```
## 22     21:00:00    86931     33494 1177864
## 23     22:00:00    79792     18843  593381
## 24     23:00:00    70067     21705  673227
```

```
arrange(hourly_mod, desc("Intensity"), .by_group = TRUE)
```

```
##     Group.1     V1    V2      V3
## 1  00:00:00  67066  6833  131124
## 2  01:00:00  65464  3272   47141
## 3  02:00:00  64551  1727   29228
## 4  03:00:00  63013   907   10893
## 5  04:00:00  63620   354    3345
## 6  05:00:00  76152   156    1972
## 7  06:00:00  80994   836   12733
## 8  07:00:00  87959  2353   35496
## 9  08:00:00  96207  8269  172714
## 10 09:00:00  98819 20870  990153
## 11 10:00:00 102618 14785  411935
## 12 11:00:00 101791 13161  428972
## 13 12:00:00 108056 17006  516302
## 14 13:00:00 106200 19611  652296
## 15 14:00:00 106590 21448  678820
## 16 15:00:00  97573 13349  412622
## 17 16:00:00 102788 17218  515561
## 18 17:00:00 111214 18882  632525
## 19 18:00:00 111884 39551 1377344
## 20 19:00:00 110065 28600  891787
## 21 20:00:00  92736 28630  983158
## 22 21:00:00  86931 33494 1177864
## 23 22:00:00  79792 18843  593381
## 24 23:00:00  70067 21705  673227
```

So this now breaks down average calories, intensities, and steps based on hour of the day. Although I had issues arranging the dataset, you can still see that (excluding normal sleeping hours) 5AM-8AM have the lowest sleep times.

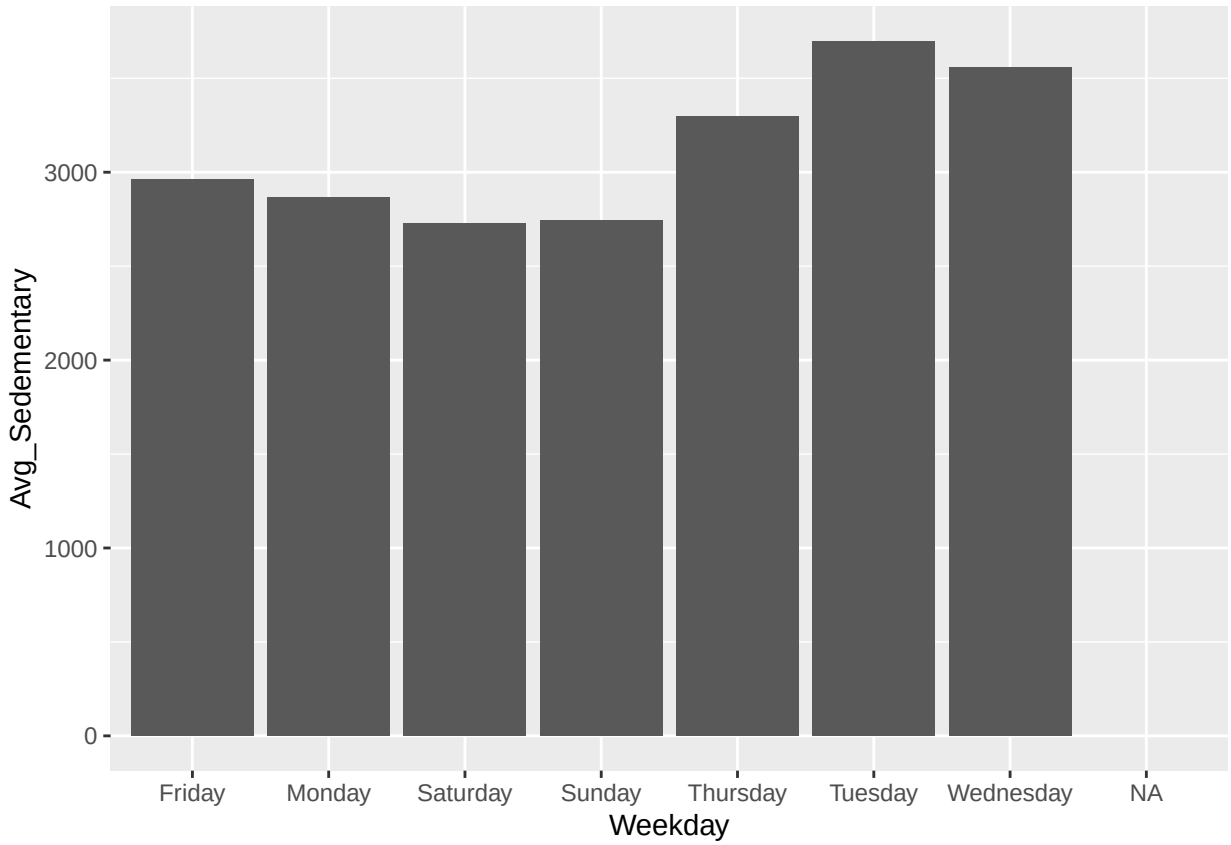
```
weekdaysleep %>%
  group_by(Weekday)%>%
  summarise(sedentary = mean(Avg_Sedementary, na.rm=FALSE), sleep = mean(Avg_Sleep, na.rm=FALSE))
```

```
## # A tibble: 8 x 3
##   Weekday   sedentary sleep
##   <chr>         <dbl> <dbl>
## 1 Friday         741.  405.
## 2 Monday         718.  419.
## 3 Saturday       682.  418.
## 4 Sunday         686.  455.
## 5 Thursday       660.  405.
## 6 Tuesday        740.  405.
## 7 Wednesday      712.  435.
## 8 <NA>            NA    NA
```

This breaks down daily averages of sedentary time and sleep time.

```
ggplot(weekdaysleep, mapping = aes(x=Weekday, y=Avg_Sedementary)) + geom_col()
```

```
## Warning: Removed 473 rows containing missing values (position_stack).
```



And as shown here, Tuesday-Thursday have the highest average sedementary time.