Project 2 Report

Steven LaCroix

November 6, 2023

Introduction

In this project we were provided a lot of data. 18 different CSV files to be exact with the goal of discovering what the data was, analyzing it, and hopefully drawing some conclusions. This data was captured from various Fitbits, which track all sorts of things. Time spent in bed, workout intensity, steps taken, and much, much more. Some of the CSV files that this data was stored in tracked the users by the minute, some by the hour, and some were simply a summary of the day. Despite the data being very clean and easy to work with, it was easy to get confused on what was needed, where to find it, and then how to interpret it. Regardless, we are going to take a look at four main areas in this project. They are: distributions and outliers, data processing, correlation and plots, and t-tests. Unlike the last project, which was rather open ended, this project appeared to act more as a checklist of tasks to do. As such, this report will follow the four main sections of the project and break them down from there.

Part 1: Distributions and Outliers

a) Heart Rate

The goal of this section was to show the heart rate of at least 5 different users over time. To achieve this, the heartrate_seconds_merged dataframe was used. This dataframe provided us with three columns. A user ID, date/time, and a value (the heart rate). This dataset was massive, totaling nearly 2.5 million rows with information on 14 users, each point was captured five seconds apart. Because of that, I decided it was best to pair down the information a little. For simplicity sake and in the interest of run time, I decided to just look at the first 5 users' information. Each user's list of heart rate values were separated into 25 evenly sized bins. The mean was taken of each bin. Doing this allows us to see the progression of the user's heart rate over time in a much easier manner to process. The 25 means were then plotted as a line graph.

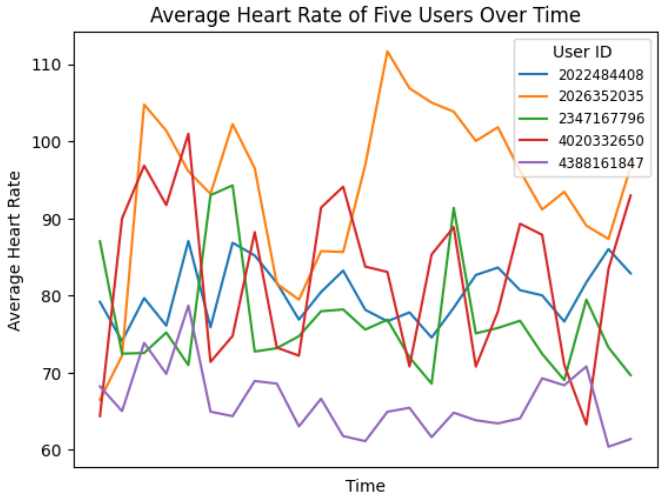

Figure 1: Average Heart Rate of Five Users Over Time

As you can see in figure 1, it's pretty easy to tell a difference between the users. For simplicity sake I will refer to the user's by the color of their line. The orange user tended to have a much higher heart rate than the others. The purple user was consistently at the bottom. Another interesting thing to note, it appears there's much more variance at the beginning, whereas things level out closer to the end. You can also see a slight downward trend of average heart rate. That could mean a lot of things. Maybe users were spooked by the fitbit at first and it being a constant reminder that they're being tracked. Maybe users were motivated to use the use it then as time went on that decreased. Either way, it's pretty clear that different users have very different average heart rates over time.

b) Daily Sleep Duration

Here we wanted to take a look at the average minutes of sleep by users. The sleepDay_merged dataframe was used for that. This gives us much more information than needed, but what we'll be looking at is the TotalMinutesAsleep column. Once the dataset was read in, we went ahead and found a list of all the unique user ID's present in the dataframe. After that was done we simply iterated through each user ID and took an average of all values for that user. These values were then added to a new list and were used in creating a bar chart with the users along the x-axis and average minutes of sleep on the y-axis.

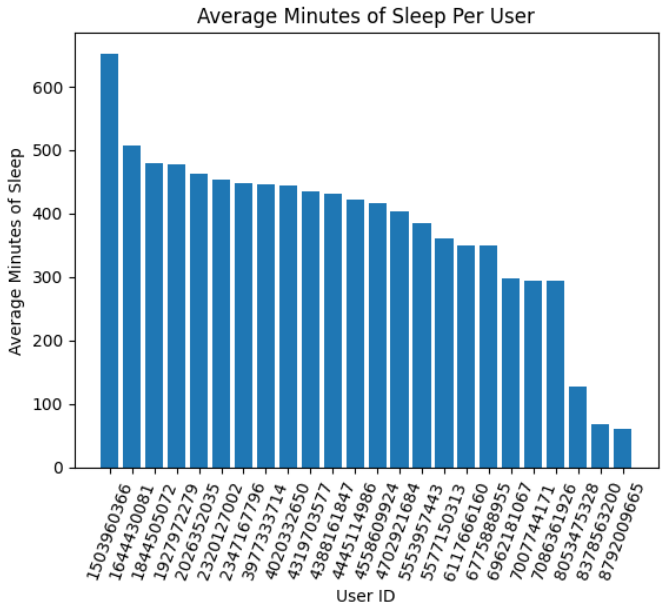

Figure 2: Average Minutes of Sleep Per User

As seen in figure 2, there are some very clear outliers. On the lower side of things, users 4558609924, 7007744171, and 23247167796 were all below 200 minutes on average. Then on the higher side user 1844505072 was well above 600 minutes. I'm not sure about the likelihood of these values being correct, but they are certainly out of the norm. It could be as simple as the fitbit malfunctioning and failing to gather correct information. Aside from those 4 users, everyone else appeared to be sleeping a normal amount. Between about 300 and 500 minutes, which is between 5 and ~8 hours.

c) Daily Steps

Taking a look at daily steps per user, it's a very similar process as the average minutes of sleep. The only difference was that we were using the dailySteps_merged dataframe which just gave us a user id, date, and step total. Similar to the minutes of sleep, we read in the dataset,

found the list of unique users, and found the mean value for their steps. Once again, the list of mean values was plotted with users on the x-axis and average steps on the y-axis.

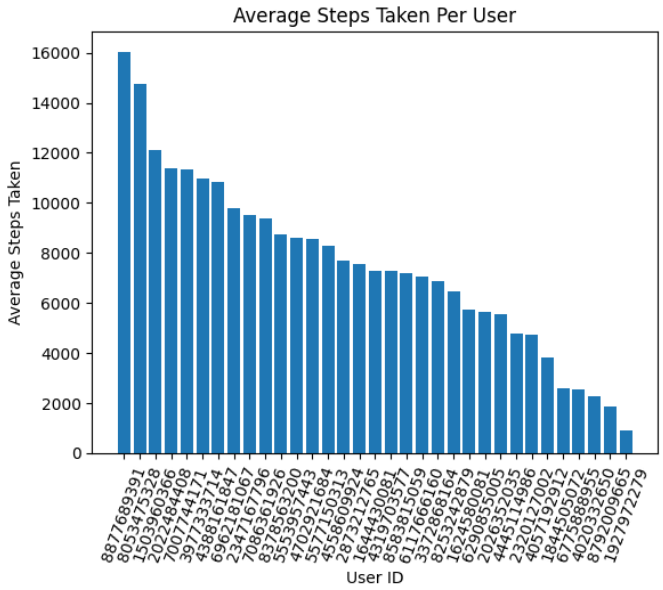

Figure 3: Average Steps Take Per User

In figure 3 there are still a decent number of outliers. There are two users on the high end that are both above 14,000 steps on average per day. That could be because they are marathon runners, have super active jobs, or are just active people in general. On the other end of the spectrum there appears to be five users that are below 4,000 steps per day. There could also be a wide range of answers for this, too. Regardless, it does not appear that everyone is getting their recommended 10,000 steps per day.

d) Weight Change

Lastly for this section, we wanted to look at the weight change of the user that recorded their weight most often (most weight records). The weightLogInfo_merged dataset was used for this. We begun by finding a frequency table of the user ID's and isolating the most frequent one. That's because the user ID that appears the most has the most weight records. After that we were able to plot a line graph showing the user's weight over the course of the month of data that was present.

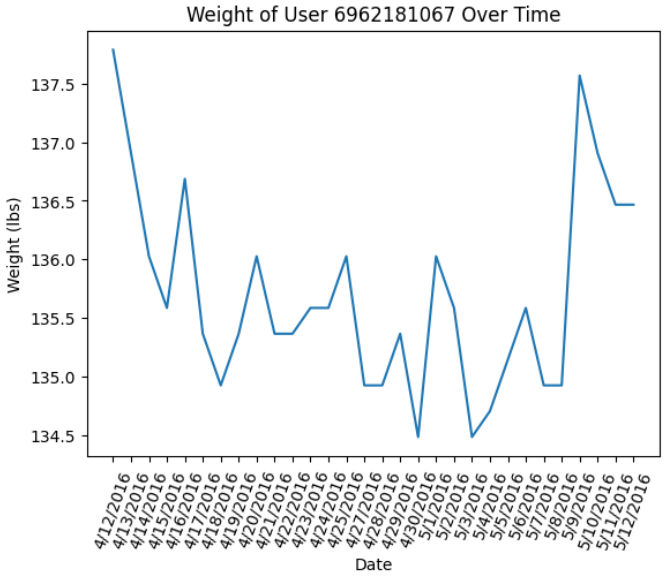

Figure 4: Weight of User 6962181067 Over Time

As you can see, it appears the user dropped weight at the beginning then began to climb back up as time went on. The first weigh in is pretty extreme as it appears the user lost 1.5 pounds over the next 2 days. Then, they weighed in at 135 and gained 2.5 pounds over a day at another point. These outliers might be due to the scale, or they could honestly just be natural. It's normal for the body to fluctuate weight, however these swings are the most extreme we can see here.

Part 2: Data Processing

1) Hourly and Minutely DataFrames

We wanted to make a pair of new dataframes, one that combines a bunch of elements that were recorded by the hour and another that does the same for variables recorded by the minute. In the dataframes we wanted to keep elements such as the user ID and the date/time. We wanted to add in StepTotal, TotalIntensity, AverageIntensity, and Calories as new columns to the hourly dataframe. An inner merge on the ID and date/time columns was done on hourlySteps and hourlyIntensities dataframes to create a temporary dataframe. Then, the same was done with the temporary dataframe and hourlyCalories to create our hourly_df dataframe. Shown below is what the first 5 elements in hourly_df look like.

	Id	ActivityHour	StepTotal	TotalIntensity	AverageIntensity	Calories
0	1503960366	4/12/2016 12:00:00 AM	373	20	0.333333	81
1	1503960366	4/12/2016 1:00:00 AM	160	8	0.133333	61
2	1503960366	4/12/2016 2:00:00 AM	151	7	0.116667	59
3	1503960366	4/12/2016 3:00:00 AM	0	0	0.000000	47
4	1503960366	4/12/2016 4:00:00 AM	0	0	0.000000	48

For the minutely dataframe we did the same thing to get an inner merge on the user ID and the date/time. However, the new columns introduced were Calories, Intensity, and METs. Below is what the first 5 elements of the minutely_df look like.

	Id	ActivityMinute	Calories	Intensity	METs
0	1503960366	4/12/2016 12:00:00 AM	0.7865	0	10
1	1503960366	4/12/2016 12:01:00 AM	0.7865	0	10
2	1503960366	4/12/2016 12:02:00 AM	0.7865	0	10
3	1503960366	4/12/2016 12:03:00 AM	0.7865	0	10
4	1503960366	4/12/2016 12:04:00 AM	0.7865	0	10

2) New Date Column

This is a fairly simple step. We really just wanted to add a new column to four dataframes containing datetime objects instead of a string with the date and time. This will make future processing much easier. Those four dataframes are the dailyActivity_merged, sleepDay_merged, hourly_df, and minutely_df dataframes. The easiest solution was to use the to_datetime() function. Below is what the first five elements of the minutely_df look like one the operation was applied.

	Id	ActivityMinute	Calories	Intensity	METs	Date
0	1503960366	4/12/2016 12:00:00 AM	0.7865	0	10	2016-04-12 00:00:00
1	1503960366	4/12/2016 12:01:00 AM	0.7865	0	10	2016-04-12 00:01:00
2	1503960366	4/12/2016 12:02:00 AM	0.7865	0	10	2016-04-12 00:02:00
3	1503960366	4/12/2016 12:03:00 AM	0.7865	0	10	2016-04-12 00:03:00
4	1503960366	4/12/2016 12:04:00 AM	0.7865	0	10	2016-04-12 00:04:00

3) Average Heart Rate Per Minute

We are given heart rate data in the file heartrate_seconds_merged. This file contains the heart rate measured every 5 seconds. We want to find the average heart rate for each minute and store that in a new dataframe. To do that, we read in our dataframe and convert the time

column to an actual datetime object. Once that's done, we can group our data by the user ID (so we don't have rows containing data from multiple ID's) and extract each value that has the same minute. We take the mean of those values as add it to a new dataframe with the user ID and time. The first 5 rows of that new dataframe, titled minuteAvgHeartRate can be seen below.

	Id	Time	Mean
0	2022484408	2016-04-12 07:21:00	101.600000
1	2022484408	2016-04-12 07:22:00	87.888889
2	2022484408	2016-04-12 07:23:00	58.000000
3	2022484408	2016-04-12 07:24:00	58.000000
4	2022484408	2016-04-12 07:25:00	56.777778

Part 3: Correlation and Plots

1) Steps & Intensity vs Calories

So, for this, we want to find a relationship between steps taken and calories burned. This can be done pretty easily, especially with the hourly_df we made earlier. We simply use the scatter function with steps on our x-axis and calories on the y-axis to look at this. We can also set the color to change based on the user ID and decrease the opacity so you can see more of the points. Despite all of that, there's still a lot of datapoints so it's tough to tell them apart. The same exact process can be done for plotting intensity and calories, too.

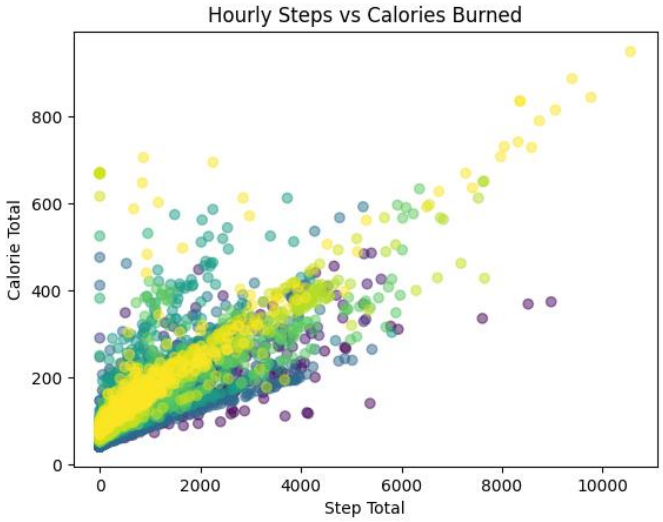

Figure 5: Hourly Steps vs Calories Burned

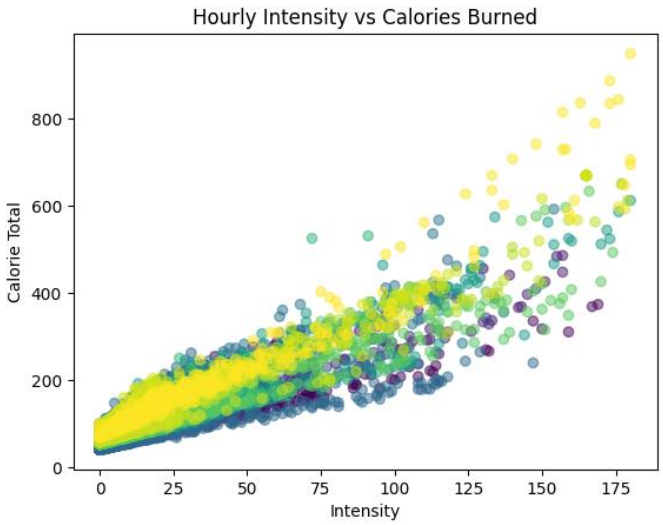

Figure 6: Hourly Intensity vs Calories Burned

As you may be able to tell, there is a pretty linear relationship on both graphs. They both have a handful of outliers but the general theme is that there is a clear and obvious correlation between steps and calories, and intensity and calories. With that being said, I do think that the intensity plot is a bit more relevant in terms of a measure for calories burned. Calories can be burned through a vast array of activities, and a lot of those don't necessarily involve walking/taking steps. In figure 5 we can see that there are a number of points where the calories burned is super high but the number of steps is nearly zero. We don't see anything like that on figure 6. That could be because the person was swimming, for example. That is a super intense activity that burns a lot of calories. That would push it towards the top right on figure 6. In figure 5 it would likely hover around the left edge since you don't walk while you swim.

2) TotalMinutesAsleep vs TotalTimeInBed

We can plot a graph looking at this nearly the exact same way as in figures 5 and 6. The only difference here is that we're pulling our data from sleepDay_merged instead.

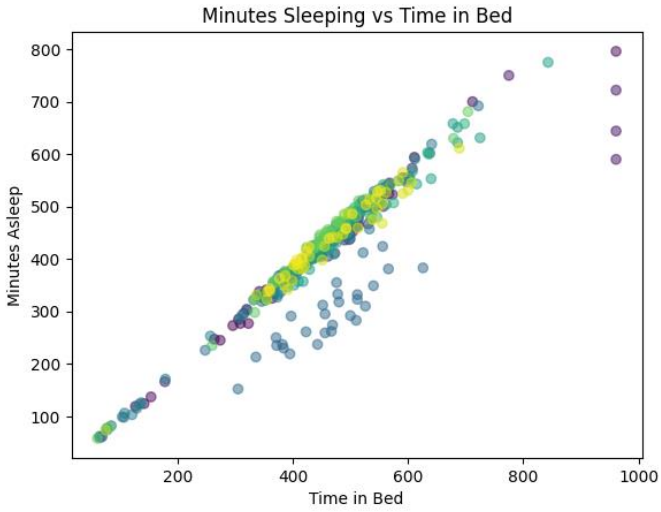

Figure 7: Minutes Sleeping vs Time in Bed

As you can imagine, there's a very clear and defined correlation between these two things as well. The interesting thing is that we can see one user who clearly isn't able to sleep as well as the others. Their minutes asleep are consistently below the area where all the other users are. I'm not sure what might cause this. It could be some medical condition, it could be that they have a kiddo at home who's always waking them up. Either way, it's odd to see just one person who so consistently strays so far from the group. Moving on from that, it's actually pretty incredible how linear the relationship is. There's a total of 414 datapoints in the dataframe and there appear to only be about 30 points that move away from the line. Other than that, it is a clear relationship.

3) Intensity During a Day

Now we want to see different users' intensities over the course of a day. In order to do this, I decided to just look at the first four users in the list (despite there being 33 different users in the hourly_df). I went ahead and narrowed down the dataframe to only one day (4/12/2016). Once that was done, I went ahead and split up each of the four users into their own dataframes. I then made a plot with 4 subplots. On each subplot a bar chart was made for that user displaying the intensity over the course of the day.

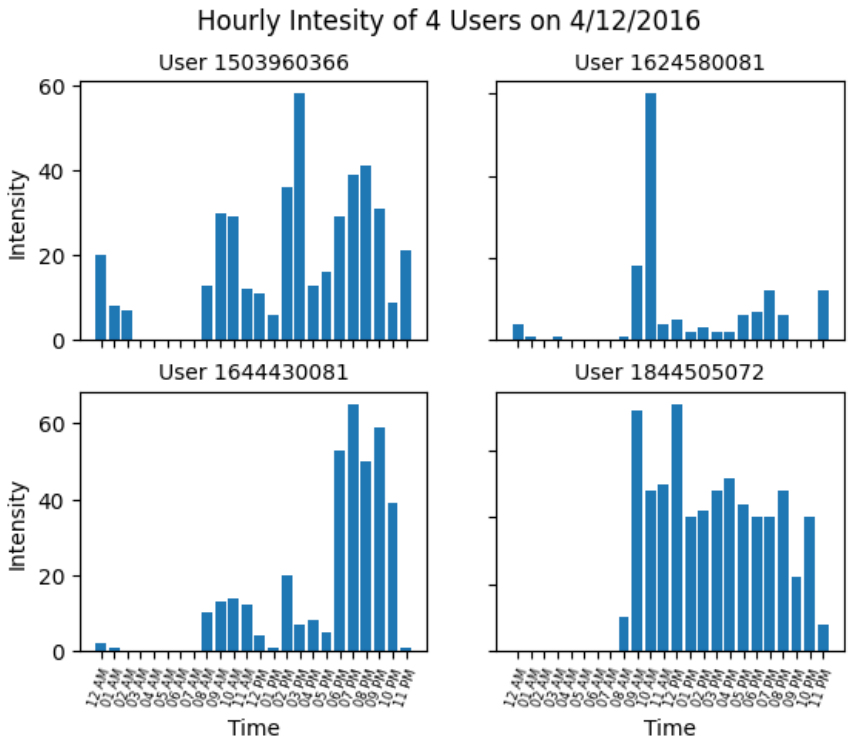

Figure 8: Hourly Intensity of 4 Users on 4/12/2016

As you may be able to tell, even the four users we're looking at here have a massive difference in the distribution of their intensities. In the top left, they stayed up late doing things and are were active right after lunch and dinner. In the top right they were very mellow all day except for right before noon. In the bottom left they were pretty mellow up until after dinner when they were intense for 5 hours. The bottom right was super intense nearly all day but did absolutely nothing after midnight. That goes to show that there is not one way to stay active. Different people are able to do different things depending on what works for them.

4) TotalMinutesAsleep vs SedentaryMinutes

Here is another thing relating the user's activity (or lack-there-of) to sleeping time. The best way I figured to do this was to merge sleepDay_merged, which had our sleeping minutes data with dailyActivity_merged, which had out sedentary time. I did another inner merge on user ID and date, similar to the merges we did earlier. This gave me a new dataframe with all of the information needed in one location. That was then plotted as a scatter plot with user ID's changing the color like before.

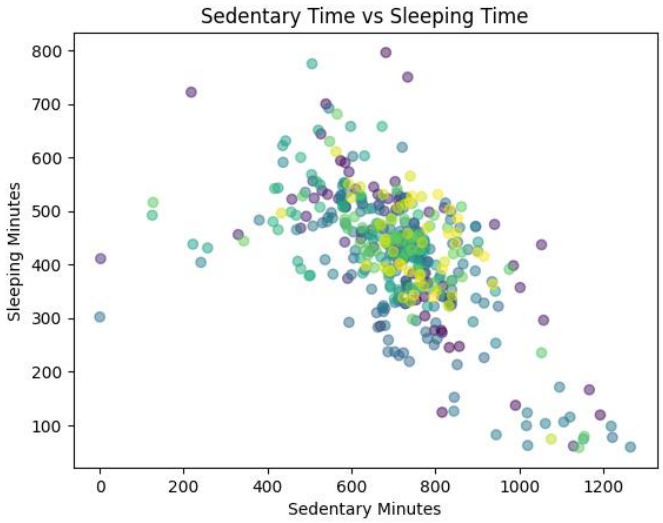

Figure 9: Sedentary Time vs Sleeping Time

Interestingly enough, there is not much of a correlation between these two things. If you asked me who would sleep more, someone who's sedentary more or less, I would probably tell you the person who is sedentary more. That doesn't appear to be the case here, though. In fact, it almost appears like there's a very loose correlation between being more sedentary and getting less sleep. I'm not sure why this is the case. I guess I would have to ask Fitbit what counts as being sedentary. If lying in bed doesn't count then it would make sense because that means these people are spending more time doing sedentary activities and giving themselves less time to sleep. They could be watching movies, shows, playing video games, or a number of other things. If lying in bed does count then I have no idea where this negative correlation is coming from.

Part 4: T-tests

1) Most Significant

Alright so I think the variable that is most significant for calories is VeryActiveMinutes. We can do a two-sample t test using a significance level of 0.05 to test this. Our null is going to be: there is no significant difference between the average number of calories burned between people with less than 20 active minutes versus more than 20 active minutes. The alternative would be: There is a significant difference between the average number of calories burned between people with less than 20 active minutes versus more than 20 active minutes. To test this, we can split our dailyActivity_merged dataframe in half at the 20 minute mark. This gives us our two samples. We then plug our 2 lists into the test and get a p-value. That p-value came

out to be 9.21e-44. This is much lower than our alpha of 0.05, so we can reject the null. Since we reject the null we are able to say that there is a significant difference in the average number of calories burned between people who have more than 20 very active minutes versus less than 20.

2) Least Significant

For the least significant variable, I think that would have to be SedentaryMinutes. After seeing the graph from figure 9, I don't necessarily believe that SedentaryMinutes has much to do with anything related to fitness, especially calories. We will use a two-sample t test and a significance level of 0.05 again for this one. The null would be: there is no significant difference between the average number of calories burned between people with less than 990 sedentary minutes versus more than 990 sedentary minutes. The alternative would be: there is a significant difference between the average number of calories burned between people with less than 990 sedentary minutes versus more than 990 sedentary minutes. Similar to the last test, we split the dailyActivity_merged dataframe and plugged them into our test. The p-value came out to be 0.0694. This is greater than our alpha of 0.05 and so we fail to reject the null. We cannot determine if there is a significant difference between the average number of calories burned between people with less than 990 sedentary minutes versus more than 990 sedentary minutes.

Conclusion

This is a really long report, so I'll keep this brief. We were able to take a look at all sorts of Fitbit data. Things like distributions and outliers where we discovered that there are very extreme ends to the spectrum of how much people sleep. We also got to look at data processing where we got to mess around with datetime objects and merging dataframes. Next, we looked at correlation and plots, allowing us to see that there is a negative correlation between sedentary time and sleep time. Lastly, we got to see that we cannot find a relationship between sedentary time and calories burned. Thanks, have a lovely day :)