Fitbit Data Analysis

Problem

Fitbit is an electronics company that sells devices that track the user's physical activity and sleep. A few of the common features of Fitbit devices include step counts, calories burnt, heart rate, and quality of sleep. Millions of devices have been sold worldwide, proving the popularity of Fitbit and its products, but how effective, or rather, impactful, are the devices? Fitbit watches do a decent job tracking sleep patterns and physical activities, and users have the option to download their data to view. The data collected is meaningless, though, if they are only represented as just numbers on a spreadsheet.

Effectively transitioning to a healthier lifestyle and achieving fitness goals comes with knowing the appropriate changes to make. Having a thorough analysis and visual representations of the collected data provide insights to an individual's fitness and sleeping habits. Having this piece of knowledge can help an individual gauge what habits they should maintain and what they can change.

Client

As the data used for this project belongs to a particular Fitbit user, the analysis would be of great interest to that user. Another audience this study may attract are fitness enthusiasts.

It is reasonable to assume users who regularly track their physical activities and/or sleep strive for healthy living and have fitness goals. By translating the data into graphs or models, the targeted audience will be provided with an intuitive understanding of Tracy's physical and sleep pattern. The subject of this project, Tracy, is a group exercise instructor and a home health aide. As someone who is constantly up on her feet and moving, it is expected that her activity levels will be higher than that of an average person's.

The outcomes of the data analysis provide details to Tracy on her current fitness and sleeping habits. Based on her reactions to the findings of this report, she can make the appropriate changes to help herself achieve new health/fitness goals.

Data

The data that will be analyzed in this project belongs to a Fitbit user. She has been using her Fitbit Alta to track her activity and sleep for nearly four years.

The data are grouped into two different categories: "Activities" and "Sleep". Personal body information (e.g., weight, BMI) will be excluded in this analysis.

Approach

The goal of this project is to provide a comprehensive report that will intuitively inform Tracy of her current habits. Topics of particular interest are Tracy's activity levels, sleep quality, and how age has affected Stacey. Various graphs (e.g., bar graphs, scatter plots) will be used to visualize her physical activity and sleep over four years.

Part 1: Data Wrangling

My capstone project focuses on analysis of an individual's Fitbit data. Data was exported via Fitbit's website, and only up to 31 days of data can be collected at a time. Instinctively, I exported the data by month, all of which are Excel files. Activity, Sleep, and Food data were collected for all months, and are saved as separate sheets within an Excel file. In addition, daily food logs are saved as its own sheets, meaning each Excel file has at least 30 sheets. Towards the later half of the project, the heart rate data was also gathered using a source outside of Fitbit.

Fitbit's activity and sleep data are saved in a traditional database format, where each column is a variable and each row is an instance of the data. To consolidate each of the monthly data into one dataframe, the activity and sleep data were first parsed separately and stored into two lists of dataframes. Second, we concatenate the lists of dataframes to get two big dataframes. Fitbit displays their data in a user-friendly manner, so commas are used for numbers when needed. In Python, unfortunately, numeric data types are strictly just numbers, so the presence of a comma automatically renders the value to be treated as a string. To convert columns of strings into numeric values, commas were removed using regular expressions then converted using Python's handy "pd.to_numeric" function.

	Start Time	End Time	Minutes Asleep	Minutes Awake	Number of Awakenings	Time in Bed	Date
0	2015-10-22 00:00:00	2015-10-22 05:07:00	292	15	1	307	2015-10-21
1	2015-10-22 21:29:00	2015-10-23 04:17:00	401	7	1	408	2015-10-22
2	2015-10-23 21:47:00	2015-10-24 06:43:00	514	22	2	536	2015-10-23
3	2015-10-24 23:24:00	2015-10-25 07:16:00	459	13	1	472	2015-10-24
4	2015-10-24 14:40:00	2015-10-24 16:05:00	80	5	0	85	2015-10-24

Sleep dataframe

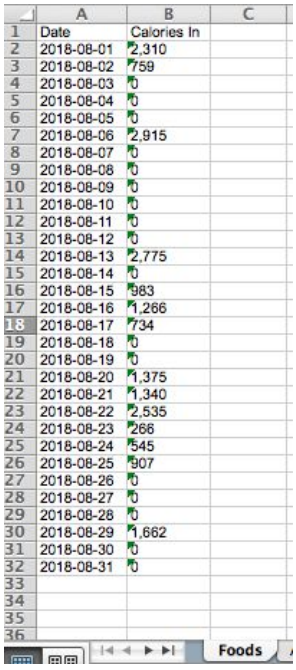
Upon analysis of the datasets, it was discovered that on days when Tracy does not use her Fitbit, data will still be displayed for that day with default values (e.g., minimum calories burned value based on Tracy's BMI). A check was performed to determine how many missing values (days Tracy doesn't use her Fitbit) were present in the dataset. It turns out only a small portion of the data contained missing values, so instead of filling in missing values, they were removed instead. The default value for Steps is 0, so we remove instances of data where the Steps

column contains 0. Outliers were determined to be any values 2 standard deviations away from the mean. Outliers were easily handled by being replaced with the mean value.

Date	Calories Burned	Steps	Distance	Floors	Minutes Sedentary	Minutes Lightly Active	Minutes Fairly Active	Minutes Very Active	Activity Calories	Weekday
2015-10-21	2150.000000	14061.0	5.71	17.0	531.452381	324.060524	0.0	0.0	1588.812105	Wednesday
2015-10-22	2274.000000	13617.0	5.46	12.0	596.000000	300.000000	17.0	69.0	1344.000000	Thursday
2015-10-23	2174.000000	16530.0	6.57	20.0	639.000000	361.000000	15.0	35.0	1275.000000	Friday
2015-10-24	2161.000000	14710.0	5.88	11.0	550.000000	278.000000	36.0	52.0	1227.000000	Saturday
2015-10-25	2479.197832	5077.0	2.02	8.0	869.000000	324.060524	9.0	14.0	1588.812105	Sunday

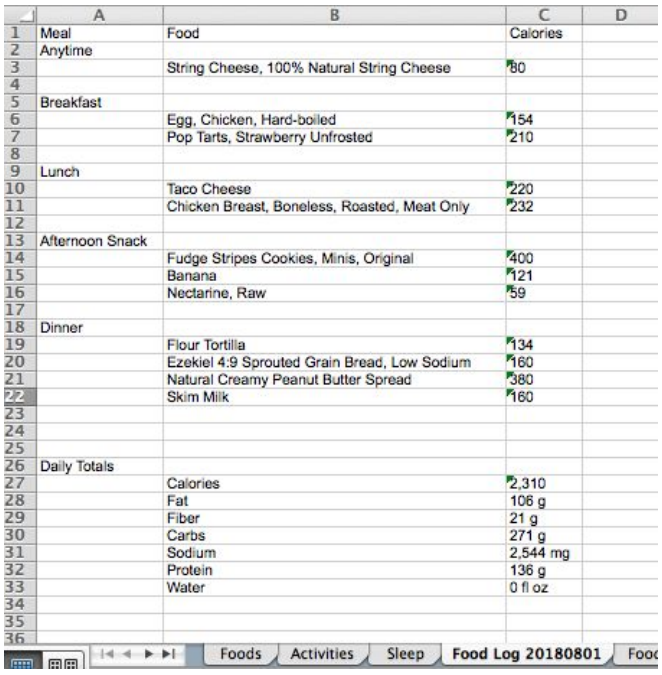
Activities dataframe

The trickiest dataset to work with was the Food data.



	A	B	C
1	Date	Calories In	
2	2018-08-01	2,310	
3	2018-08-02	759	
4	2018-08-03	0	
5	2018-08-04	0	
6	2018-08-05	0	
7	2018-08-06	2,915	
8	2018-08-07	0	
9	2018-08-08	0	
10	2018-08-09	0	
11	2018-08-10	0	
12	2018-08-11	0	
13	2018-08-12	0	
14	2018-08-13	2,775	
15	2018-08-14	0	
16	2018-08-15	983	
17	2018-08-16	1,266	
18	2018-08-17	734	
19	2018-08-18	0	
20	2018-08-19	0	
21	2018-08-20	1,375	
22	2018-08-21	1,340	
23	2018-08-22	2,535	
24	2018-08-23	266	
25	2018-08-24	545	
26	2018-08-25	907	
27	2018-08-26	0	
28	2018-08-27	0	
29	2018-08-28	0	
30	2018-08-29	1,662	
31	2018-08-30	0	
32	2018-08-31	0	
33			
34			
35			
36			

Foods



	A	B	C	D
1	Meal	Food	Calories	
2	Anytime			
3		String Cheese, 100% Natural String Cheese	80	
4				
5	Breakfast			
6		Egg, Chicken, Hard-boiled	154	
7		Pop Tarts, Strawberry Unfrosted	210	
8				
9	Lunch			
10		Taco Cheese	220	
11		Chicken Breast, Boneless, Roasted, Meat Only	232	
12				
13	Afternoon Snack			
14		Fudge Stripes Cookies, Minis, Original	400	
15		Banana	121	
16		Nectarine, Raw	59	
17				
18	Dinner			
19		Flour Tortilla	134	
20		Ezekiel 4:9 Sprouted Grain Bread, Low Sodium	160	
21		Natural Creamy Peanut Butter Spread	380	
22		Skim Milk	160	
23				
24				
25				
26	Daily Totals			
27		Calories	2,310	
28		Fat	106 g	
29		Fiber	21 g	
30		Carbs	271 g	
31		Sodium	2,544 mg	
32		Protein	136 g	
33		Water	0 fl oz	
34				
35				
36				

Foods | Activities | Sleep | Food Log 20180801 | Fooc

This kind of data layout is not in a traditional dataset format. In order to transform the data to achieve the desired structure, extensive manipulation of data was involved.

In the “Foods” sheet, it should be intuitive that no food data is entered for a particular day if the value in “Calories In” is 0. Since there is a sheet for each day of the month (for daily food log), we are only interested in the sheets where food data is entered--when “Calories In” is not 0. We store the dates of interest in a list and use it to get the corresponding daily food log sheets. Each sheet is converted into a dataframe, where further manipulation is done. We associate

each food item with the type of meal it was eaten as (e.g., Breakfast, Lunch, etc.) by taking the existing “Meal” column and forward filling each value. Any rows with missing values are removed.

Fitbit includes daily food composition totals within each sheet, which has sufficient information to be a separate dataframe itself. These data were extracted out of the food dataframe and put into its own dataframe. As a result of manipulating the food data, two dataframes were created.

	Meal	Food	Calories	Date	Weekday
0	Breakfast	American Cheese	61	2015-11-09	Monday
1	Breakfast	Bagel thins, Everything	110	2015-11-09	Monday
2	Breakfast	Egg, Chicken, Fried	184	2015-11-09	Monday
3	Breakfast	Ham Steak, Traditional	30	2015-11-09	Monday
4	Morning Snack	Dark Chocolate Dreams	170	2015-11-09	Monday
5	Morning Snack	Banana	90	2015-11-09	Monday
6	Morning Snack	Rice Cakes, Salt Free	70	2015-11-09	Monday
7	Breakfast	English Muffin, Original	129	2015-11-11	Wednesday
8	Breakfast	Egg, Chicken, Fried	184	2015-11-11	Wednesday
9	Breakfast	Bacon Pre-Cooked (S)	75	2015-11-11	Wednesday
10	Breakfast	American Cheese	79	2015-11-11	Wednesday

Food dataframe

Date	Food	Calories	Carbs	Fat	Fiber	Protein	Sodium	Water	Weekday
2015-11-09		715	72	34	8	35	943	0	Monday
2015-11-11		797	74	39	4	37	1064	0	Wednesday
2015-11-12		1049	108	45	11	53	1216	0	Thursday
2015-11-30		90	20	0	1	1	2	0	Monday
2015-12-02		240	29	6	3	17	152	0	Wednesday
2015-12-09		860	101	35	8	37	1105	0	Wednesday
2015-12-10		1054	135	40	26	58	1210	0	Thursday
2015-12-11		1157	155	35	23	68	679	0	Friday
2015-12-15		1162	142	44	13	57	1402	0	Tuesday

Macros dataframe

As mentioned in the beginning, Tracy’s intraday heart heart data was obtained by a source outside of Fitbit.

The data comes in the format of a CSV file, and contains simply two columns of data--a column for minutes and another for her heart rate at the time. To be able to make this data more useful, a few features were added (e.g., date, time of day). Since the Fitbit devices are constantly tracking your heart rate, there are no instances of missing data.

	A	B
1	Time	Heart Rate
2	0:00:00	57
3	0:01:00	55
4	0:02:00	50
5	0:03:00	51
6	0:04:00	51
7	0:05:00	51
8	0:06:00	52
9	0:07:00	51
10	0:08:00	51
11	0:09:00	51
12	0:10:00	52
13	0:11:00	56
14	0:12:00	57
15	0:13:00	59

	Time	Heart Rate	Date	Time_of_Day
0	00:00:00	57	2017-02-22	Morning
1	00:01:00	55	2017-02-22	Morning
2	00:02:00	50	2017-02-22	Morning
3	00:03:00	51	2017-02-22	Morning
4	00:04:00	51	2017-02-22	Morning
5	00:05:00	51	2017-02-22	Morning
6	00:06:00	52	2017-02-22	Morning
7	00:07:00	51	2017-02-22	Morning
8	00:08:00	51	2017-02-22	Morning
9	00:09:00	51	2017-02-22	Morning
10	00:10:00	52	2017-02-22	Morning

Before and after wrangling heart rate data

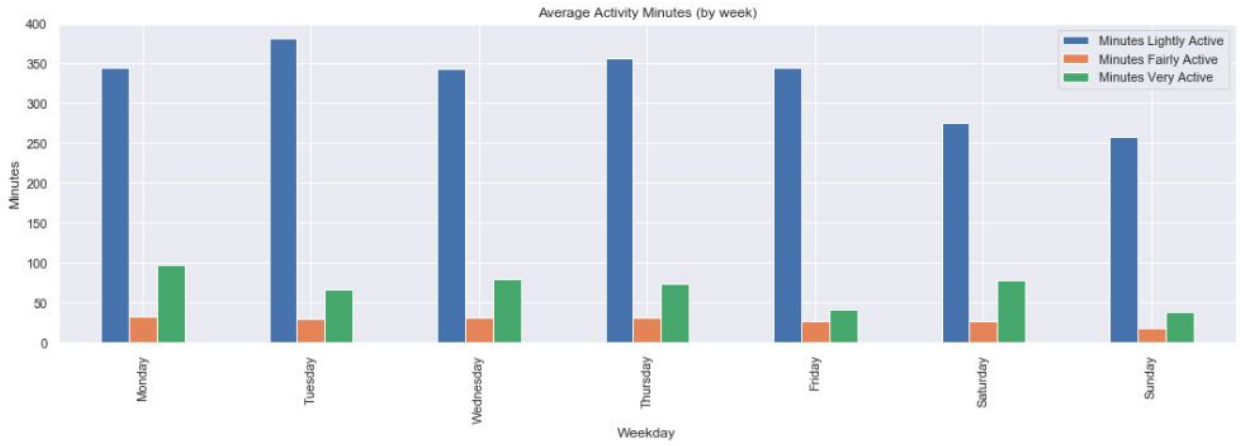
Part 2: Data Analysis

Now comes the fun of making use of the data! Starting off with some simple analysis, it seems that Tracy works out the most (determined by looking only at days she works out for at least an hour) on Monday's, Saturday's, Thursday's and Wednesday's.

```
activities[activities['Minutes Very Active'] >= 60].groupby('Weekday').size()\
                                                   .reset_index(name='Counts')\
                                                   .sort_values(by = 'Counts', \
                                                                ascending = False)
```

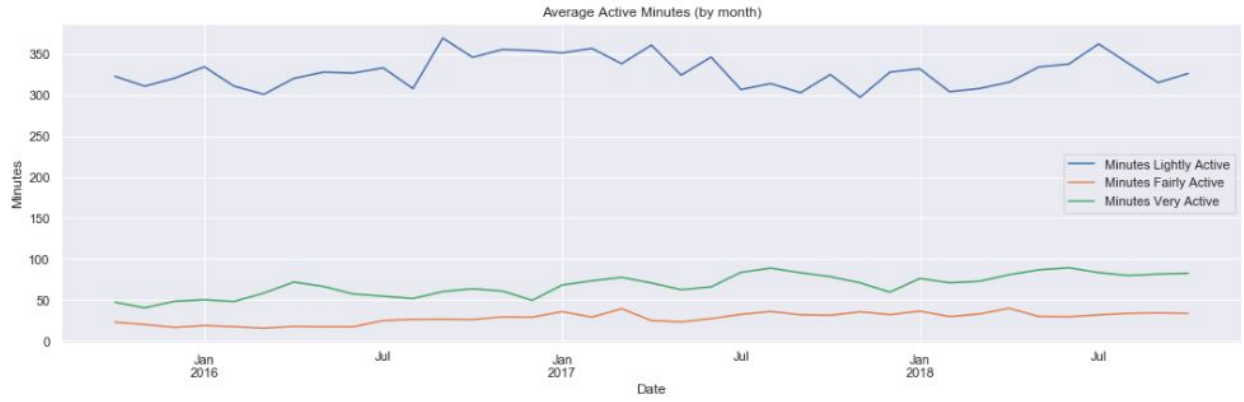
	Weekday	Counts
1	Monday	138
2	Saturday	106
4	Thursday	104
6	Wednesday	104
5	Tuesday	93
3	Sunday	37
0	Friday	29

A bar chart is provided for visualization to support the above claim. For the most part, Tracy is more active during the week, with the exception of Saturday.



Activity minutes by day of week

Shifting focus to the bigger picture, there seems to be a slight incline in 'Minutes Very Active' (green line) over time. To further expand on this, the average calories Tracy has burned over time has been graphed as well. There is an obvious upward trend in amount of calories Tracy has burned, which supports the initial observation that Tracy has been exercising more throughout the years.



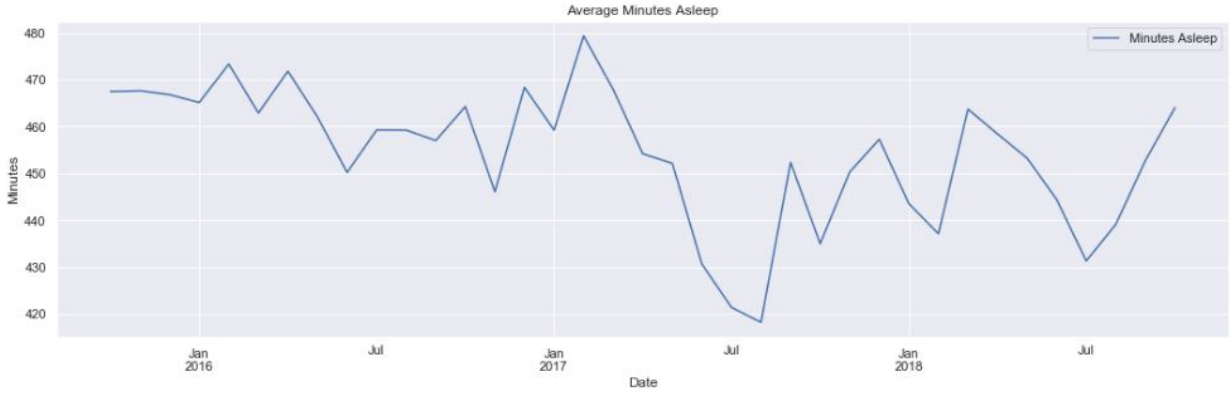
Average active minutes from October 2015 to November 2018

Average calories burned from October 2015 to November 2018

Moving on to analyzing her sleep data, since October 2015, she only has been getting at least 7 hours of sleep 69.6% of the time.

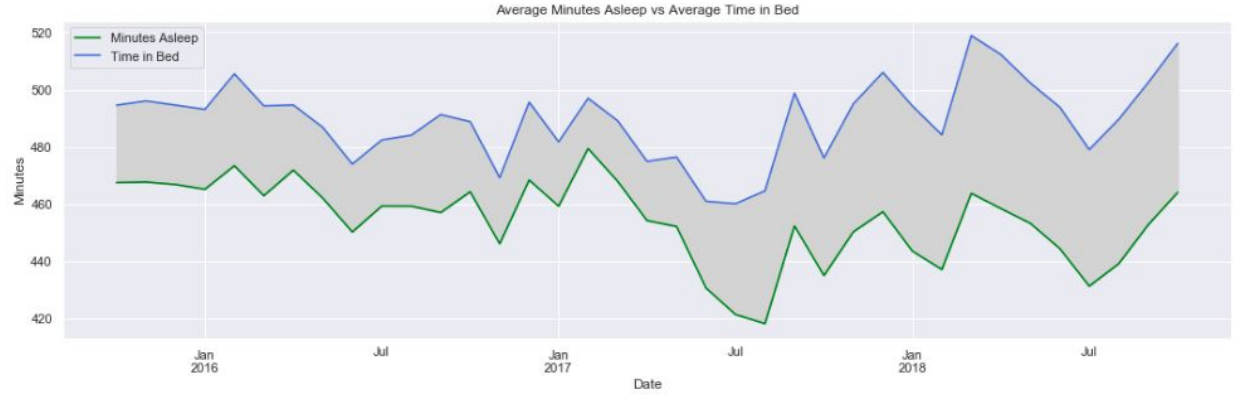
```
In [16]: seven_hours = len(daily_sleep[daily_sleep['Minutes Asleep'] >= 420])
         total = len(daily_sleep)
         seven_hours / total

Out[16]: 0.6960873521383075
```

This raises concerns regarding the amount of sleep Tracy gets. The recommended amount is usually within the range of 7-9 hours. First, to get a better understanding of her sleep overall, her average minutes asleep is graphed.



Average minutes asleep from October 2015 to November 2018

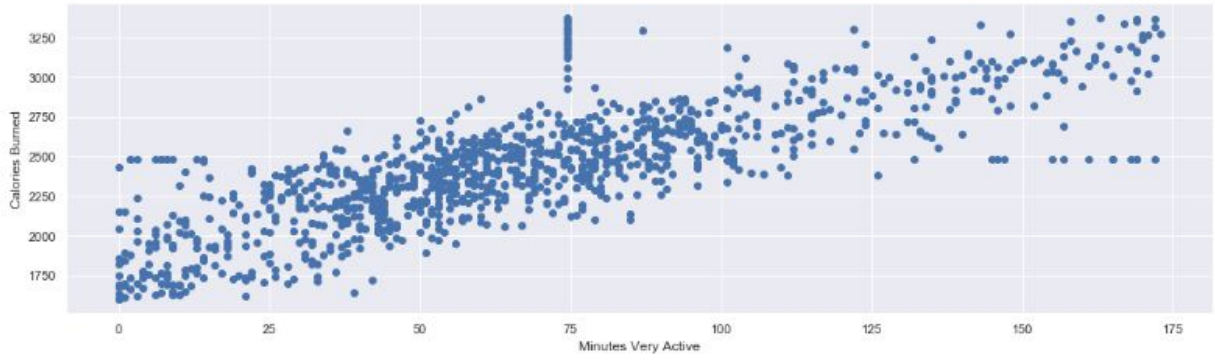
From February 2017 to July 2017, there is an evident downward trend, with the steepest negative slope from May 2017 to July 2017. This clearly informs us that Tracy has had less sleep during that time frame. From then till December 2017, we see a jagged, but upward trend, indicating a recovery of sleep time. However, despite the recovery in sleep time, Tracy seems to be experiencing more restless nights, as depicted by the graph below.



The space between the two lines represent the amount of time Tracy is awake during her sleep. From looking at her sleep data, specifically the ‘Number of Awakenings’ feature, it is not unusual for Tracy to wake 10+ times in the middle of the night. Not only does it seem like Tracy has trouble staying asleep, but it seems to be getting worse.

Part 3: Inferential Stats

Two variables that obviously have a strong correlation are “Minutes Very Active” and “Calories Burned”, as exercising burns calories. Creating a scatter plot with “Minutes Very Active” as the independent variable and “Calories Burned” as the dependent variable, an upward trend is formed. This means that the longer Tracy is active, the more calories she burns. It is important to note that there is a separate variable called “Activity Calories” in the dataset, but was not chosen as it is more interesting to see if there are other factors that contribute to calorie expenditure. The calculated pearson correlation coefficient is 0.786, indicating a moderately strong linear relationship between active minutes and daily burned calories. Since the coefficient is not enough to represent an almost linear relationship, there are still other factors that contribute to the total amount of calories Tracy burns in a day.



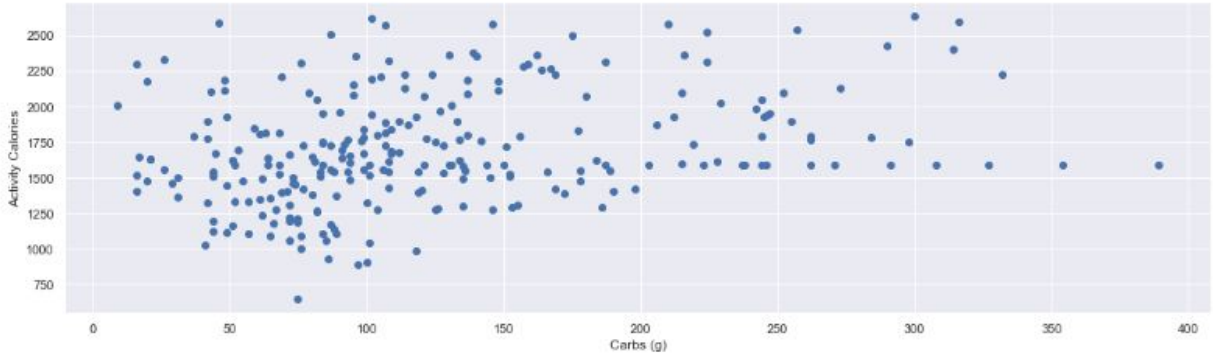

```
scipy_r, scipy_p = stats.pearsonr(activities['Minutes Very Active'], \
                                  activities['Calories Burned'])

print("Scipy's correlation coefficient:", scipy_r)
print("Scipy's p-value:", scipy_p)
```

```
Scipy's correlation coefficient: 0.7858467996449217
Scipy's p-value: 7.142083581912041e-233
```

Scatter plot depicting correlation between ‘Minutes Very Active’ and ‘Calories Burned’, with pearson correlation & p-value

Other variables that were tested for correlation include “Activity Calories” with “Carb (g)” and “Number of Awakenings” with “Minutes Asleep”. With the former, the correlation was tested to determine if Tracy is an athlete who tends to load up on carbs on her activity-heavy days. The r-value is calculated to be 0.3, which is a somewhat weak correlation, but is still a positive one nonetheless. One thing to note is that the food data is incomplete (e.g., there are instances

where dinner is not recorded). It is reasonable to assume that the r-value might actually be greater if all food intake is logged, but with the data we have available, it is still plausible to say there is a weak positive relationship between how many calories are burned from exercising and how much carbs Tracy eats.

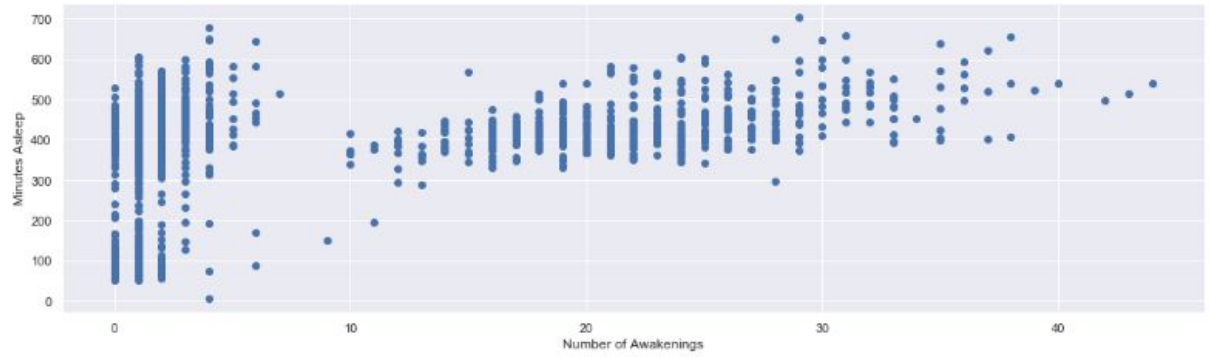



```
scipy_r, scipy_p = stats.pearsonr(df1['Activity Calories'], df1['Carbs (g)'])

print("Scipy's correlation coefficient:", scipy_r)
print("Scipy's p-value:", scipy_p)
```

```
Scipy's correlation coefficient: 0.301696397595359
Scipy's p-value: 9.62446731823555e-07
```

Scatter plot depicting correlation between 'Carbs (g)' and 'Activity Calories', with pearson correlation & p-value

When looking at Tracy's sleep data, it was intriguing -- maybe even shocking -- to see nights where she wakes up more than 30 times. According to a study, the average number of awakenings per night for an adult is around 6. Tracy's number is quintuple the average, which is astounding. However, the r-value for "Number of Awakenings" and "Minutes Asleep" is 0.381. This indicates a weak, but nonetheless positive correlation. The more awakenings Tracy experiences, the longer her sleeps. It would be more worrying if the upward trend is absent because that would indicate a low quantity in sleep, which would be a health concern.



```
scipy_r, scipy_p = stats.spearmanr(sleep['Number of Awakenings'], sleep['Minutes Asleep'])

print("Scipy's correlation coefficient:", scipy_r)
print("Scipy's p-value:", scipy_p)
```

```
Scipy's correlation coefficient: 0.38107089058809945
Scipy's p-value: 1.0564244602304605e-46
```

Scatter plot depicting correlation between 'Number of Awakenings' and 'Minutes Asleep', with pearson correlation & p-value

Only the pearson correlation test was performed, as the above three examples asks how strongly related the two variables are to each other. As an afterthought, it is possible to separate her data into winter and summer and see how her stats compare. For now, we are able to assume that Tracy is an active individual who has trouble staying asleep.

Part 4: In-depth Analysis

As mentioned above, the idea of predicting calories burned for a day using heart rate data was tested. In a typical predictive model, several feature variables are gathered to predict a target variable, all the variables belonging to an instance. The first problem that arose was the structure of the data. The idea is to use one day's worth of heart rate data to predict how many calories was burned, which in terms of the data means using all values of one column to predict a single row value. As a workaround, heart rates were aggregated by summation, and a new dataframe containing the sum of heart rates by day, calories burned, and date was created.

```
hr_daily_sum.head()
```

Date	Heart Rate	Calories Burned
2017-02-22	96265	2474.0
2017-02-23	100505	2963.0
2017-02-24	93022	2449.0
2017-02-25	94511	2649.0
2017-02-26	93240	2640.0

Supervised Learning

With the new dataframe, a model was tested using only the 'Heart Rate' feature to predict 'Calories Burned'. Both a linear regression and random forest regressor were used. Increased heart rate burns more calories, so intuitively, the greater the sum of heart rates, the more calories are burned. This suggests that there is probably a somewhat linear relationship between sum of heart rates and calories burned, which is why a Linear Regression model was chosen. However, the relationship between the two variables might not be completely linear, so a Random Forest Regressor is a good alternative to test out.

The datafame containing heart rate sums and calories burned is split into training and test sets (70-30 ratio), where the training set is used to train the model and test set to evaluate the performance of the model. To evaluate the performance, both the mean absolute error and mean error (square root of mean squared error) were calculated. The following are the results of the predictive models:

```
LINEAR REGRESSION
-----------------
Mean absolute error: 234.7799212462273
Mean error: 297.1541601692999

Cross validation results: [0.33977675 0.45372836 0.53440749 0.52467865 0.04831244]


RANDOM FOREST REGRESSOR
-----------------------
Mean absolute error: 231.14511290322582
Mean error: 303.53668597908666

Cross validation results: [-0.2258274   0.3819131   0.41860917  0.51159263  0.22444065]
```

The mean absolute errors suggests that on average, the predictive models will be roughly 230 calories off. The error is quite substantial--an individual would have to spend 15-20 minutes jump roping or 30 minutes jogging to burn off roughly 200 calories. The models are, to a certain degree, underfitting the training data.

While still utilizing just the heart rate data, new features were created in an attempt to create better models. Heart rate data were split by time of day (morning, afternoon, and evening) and averaged. Now three predictor variables are used to predict amount of calories burned.

Date	(Heart Rate, Afternoon)	(Heart Rate, Evening)	(Heart Rate, Morning)	Calories Burned
2017-02-22	67.763889	63.197222	69.084388	2474.0
2017-02-23	78.405556	58.660167	72.344633	2963.0
2017-02-24	70.273239	61.425714	69.206538	2449.0
2017-02-25	71.941341	60.113889	69.388807	2649.0
2017-02-26	69.534483	60.157233	69.322222	2640.0

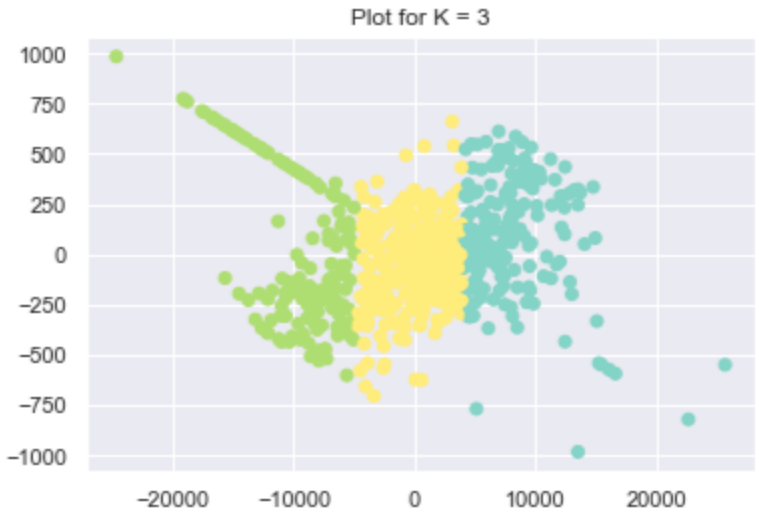
Both Linear Regression and Random Forest Regression models were used again to compare performance against the first two models.

```
LINEAR REGRESSION
-----------------
Mean absolute error: 212.88384694797185
Mean error: 265.31942766617124

Cross validation results: [ 0.1959427   0.46651428  0.66728062  0.5962532  -0.09702459]

RANDOM FOREST REGRESSOR
-----------------------
Mean absolute error: 181.615986245884
Mean error: 224.7414679986225

Cross validation results: [0.05213838 0.56110478 0.73685765 0.6782538  0.4419277 ]
```

With more features, we can see an improvement in predictive power. The new Random Forest Regressor model will have an average (absolute) predictive error of less than 200 calories.

Unsupervised Learning

In addition to supervised learning, unsupervised learning, specifically K-Means Clustering, was also utilized on the Fitbit data. Using the elbow method, it was determined the optimal number of clusters is 3, which yields the plot below:



Analyzing deeper into each cluster, many variables were investigated, but one of the more interesting ones to look at is ‘Weekday’.

```
CLUSTER 1:
Monday       51
Wednesday    46
Saturday     22
Tuesday      14
Friday        7
Thursday      6
Sunday        2
Name: Weekday, dtype: int64
```

```
CLUSTER 2:
Thursday     63
Tuesday      62
Friday       36
Saturday     34
Wednesday    34
Monday       30
Sunday       13
Name: Weekday, dtype: int64
```

```
CLUSTER 0:
Sunday       72
Friday       35
Saturday     30
Thursday     15
Tuesday      12
Wednesday     7
Monday        5
Name: Weekday, dtype: int64
```

Cluster 0 has many occurrences of Sunday's, Cluster 1 has Monday's and Wednesday's, and Cluster 2 has Tuesday's and Thursday's. Having made graphs pertaining to the day of week, something peculiar is noticed.

Looking at the graph depicting 'Minutes Very Active', Monday's and Wednesday's have the highest mean values, which in turn means those two days are days where Tracy is most active. The assumption is confirmed by Tracy herself--she has, for an extended period of time, taught at least two Insanity classes on both days. Sunday's, overall, are Tracy's most laid back days, ranking last in all three activity level categories except 'Minutes Sedentary'. Tuesday's and Thursday's have highest mean values in the 'Minutes Lightly Active' category, and relatively high mean values for 'Minutes Very Active'. This could be an indication that she spends more time at her home health aide profession and spends less time at the gym. To analyze the activity data by clusters, they first needed to be merged into the dataframe.

```
MINUTES SEDENTARY
-----------------
Cluster 0: 536.8065476190476
Cluster 1: 355.45454545454544
Cluster 2: 454.8181818181818

MINUTES LIGHTLY ACTIVE
----------------------
Cluster 0: 291.50119590211057
Cluster 1: 338.4547474303572
Cluster 2: 343.1639898506828

MINUTES FAIRLY ACTIVE
---------------------
Cluster 0: 21.869318181818183
Cluster 1: 40.34820186039698
Cluster 2: 34.39059261916149

MINUTES VERY ACTIVE
-------------------
Cluster 0: 40.26136363636363
Cluster 1: 114.07935984765253
Cluster 2: 81.99275665550773
```

Looking at the ‘Minutes Sedentary’ category, it seems that data points in Cluster 0 are representative of Tracy’s laid back/rest days--cluster 0 has the highest mean value under the ‘Minutes Sedentary’ category. Cluster 1 contains many occurrences of Monday’s and Wednesday’s, and looking at the ‘Minutes Very Active’ chart, Cluster 1 is in fact representative of Tracy’s workout-intensive days.

Overall, there were some limitations with how well the predictive models can perform due to lack of independence between some variables (data pertaining to activity are, to a degree, all dependent of one another). Breaking up the heart rate data by time of day and aggregating them by average increased the performance of both the Linear Regression and Random Forest Regression models. By performing unsupervised learning, we are able to determine which days Tracy teaches/trains and which days she uses as rest days.

Conclusion

- Tracy’s Fitbit data shows that she is a lot more active than the average person--burning an average of 2,442 calories per day (via activities and basal metabolic rate, the amount of calories she burns when her body is at rest)
- Her busiest days are Monday’s, Wednesday’s, Thursday’s, and Saturday’s

- From October 2015, she is getting at least 7 hours of sleep 70% of the time, but her restless nights are becoming increasingly worse

Recommendations

- Tracy should seek out ways to alleviate her restless nights
- Fueling her body with carbs seems to help her with slightly more increased activity levels. She should keep this in mind on days she wants to work out extra hard

Next Steps

Now that we know Tracy is a very light sleeper and has trouble staying asleep, it would be intriguing to see how the days she has little problem staying asleep differs from the days she is restless. Perhaps a pattern can be uncovered and can be brought to her attention to help her fix her sleeping problem.