FitBit

June 19, 2020

1 Problem Statement

- We are trying to analyze activity behaviour of a user based on data collected by their fitbit. Look at the relation between various data available to you and try to find patterns with regards to the users excercise routines and daily activity.

1.1 Libraries used for plotting visualizations:

- Seaborn
- Matplotlib

1.2 Description of the data

Data	Description
Id	Id of log
ActivityDate	Date
TotalSteps	Number of steps
TotalDistance	distance in miles
TrackerDistance	Distane Tracked
LoggedActivitiesDistance	Distance logged
VeryActiveDistance	High activity distances
ModeratelyActiveDistance	Moderate activity distances
LightActiveDistance	Light activity distances
SedentaryActiveDistance	Little/no activity
VeryActiveMinutes	Time activity = High
FairlyActiveMinutes	Time activity = medium
LightlyActiveMinutes	Time activity = light
SedentaryMinutes	Time activity = none/little
Calories	Calories burnt in the day

[289]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[290]:
```
df = pd.read_csv('FitBit data.csv')
```

[291]:
```
df.head(5)
```

[291]:

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance	ModeratelyActiveDistance	LightActiveDistance	SedentaryActiveDistance	VeryActiveMinutes	FairlyActiveMinutes	LightlyActiveMinutes	SedentaryMinutes	Calories
0	1503960366	3/25/2016	11004	7.11	7.11	0.0	2.57	0.46	4.07	0.0	33	12	205	804	1819
1	1503960366	3/26/2016	17609	11.55	11.55	0.0	6.92	0.73	3.91	0.0	89	17	274	588	2154
2	1503960366	3/27/2016	12736	8.53	8.53	0.0	4.66	0.16	3.71	0.0	56	5	268	605	1944
3	1503960366	3/28/2016	13231	8.93	8.93	0.0	3.19	0.79	4.95	0.0	39	20	224	1080	1932
4	1503960366	3/29/2016	12041	7.85	7.85	0.0	2.16	1.09	4.61	0.0	28	28	243	763	1886

[292]:
```
df.drop(['Id'], axis = 1, inplace = True)
```

[293]:
```
#sns.pairplot(df)
```

[294]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 457 entries, 0 to 456
Data columns (total 14 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   ActivityDate              457 non-null    object
 1   TotalSteps                457 non-null    int64
 2   TotalDistance             457 non-null    float64
 3   TrackerDistance           457 non-null    float64
 4   LoggedActivitiesDistance  457 non-null    float64
 5   VeryActiveDistance        457 non-null    float64
```

```
 6   ModeratelyActiveDistance  457 non-null    float64
 7   LightActiveDistance       457 non-null    float64
 8   SedentaryActiveDistance   457 non-null    float64
 9   VeryActiveMinutes         457 non-null    int64
 10  FairlyActiveMinutes       457 non-null    int64
 11  LightlyActiveMinutes      457 non-null    int64
 12  SedentaryMinutes          457 non-null    int64
 13  Calories                  457 non-null    int64
dtypes: float64(7), int64(6), object(1)
memory usage: 50.1+ KB
```

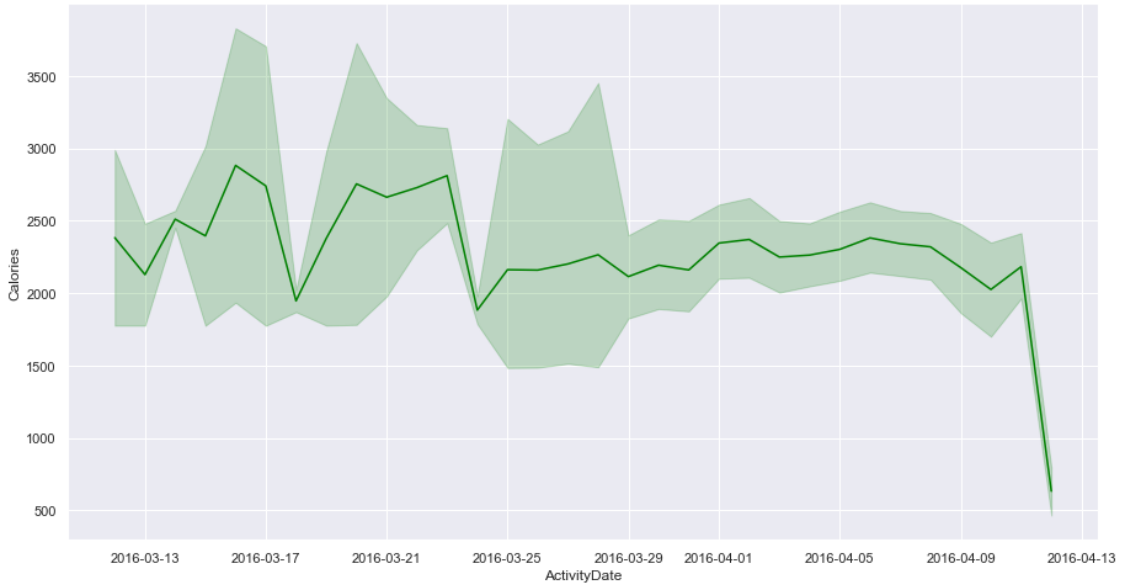
[295]:
```
df['ActivityDate'] = pd.to_datetime(df['ActivityDate'])
```

[296]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 457 entries, 0 to 456
Data columns (total 14 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   ActivityDate              457 non-null    datetime64[ns]
 1   TotalSteps                457 non-null    int64
 2   TotalDistance             457 non-null    float64
 3   TrackerDistance           457 non-null    float64
 4   LoggedActivitiesDistance  457 non-null    float64
 5   VeryActiveDistance        457 non-null    float64
 6   ModeratelyActiveDistance  457 non-null    float64
 7   LightActiveDistance       457 non-null    float64
 8   SedentaryActiveDistance   457 non-null    float64
 9   VeryActiveMinutes         457 non-null    int64
 10  FairlyActiveMinutes       457 non-null    int64
 11  LightlyActiveMinutes      457 non-null    int64
 12  SedentaryMinutes          457 non-null    int64
 13  Calories                  457 non-null    int64
dtypes: datetime64[ns](1), float64(7), int64(6)
memory usage: 50.1 KB
```

[297]:
```
plt.figure(figsize = (15,8))
sns.lineplot(x = 'ActivityDate', y = 'Calories', data = df, color = 'green')
sns.set(style= "darkgrid", color_codes=True)
plt.xticks()
plt.yticks()
```

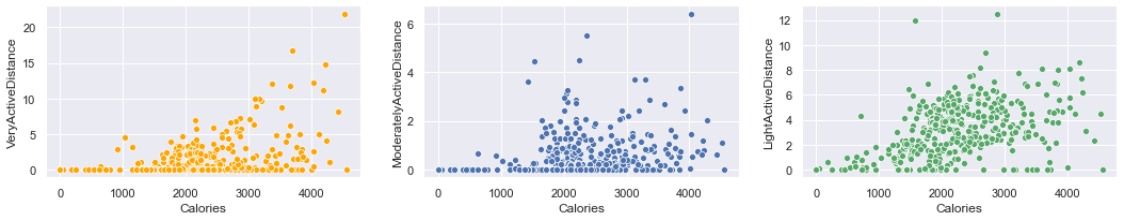
[297]:
```
(array([   0.,  500., 1000., 1500., 2000., 2500., 3000., 3500., 4000.]),
 <a list of 9 Text major ticklabel objects>)
```



I feel initially when the customer bought the fitbit he was very enthusiastic, worked out a lot harder and stuck to a plan. But as time went on he reduced his activity rate and currently is excercising at a much lesser rate. The values of calories burnt here are a mean of the original value. The upper and lower ranges are based on the 25th and 75th percentile values around the mean.

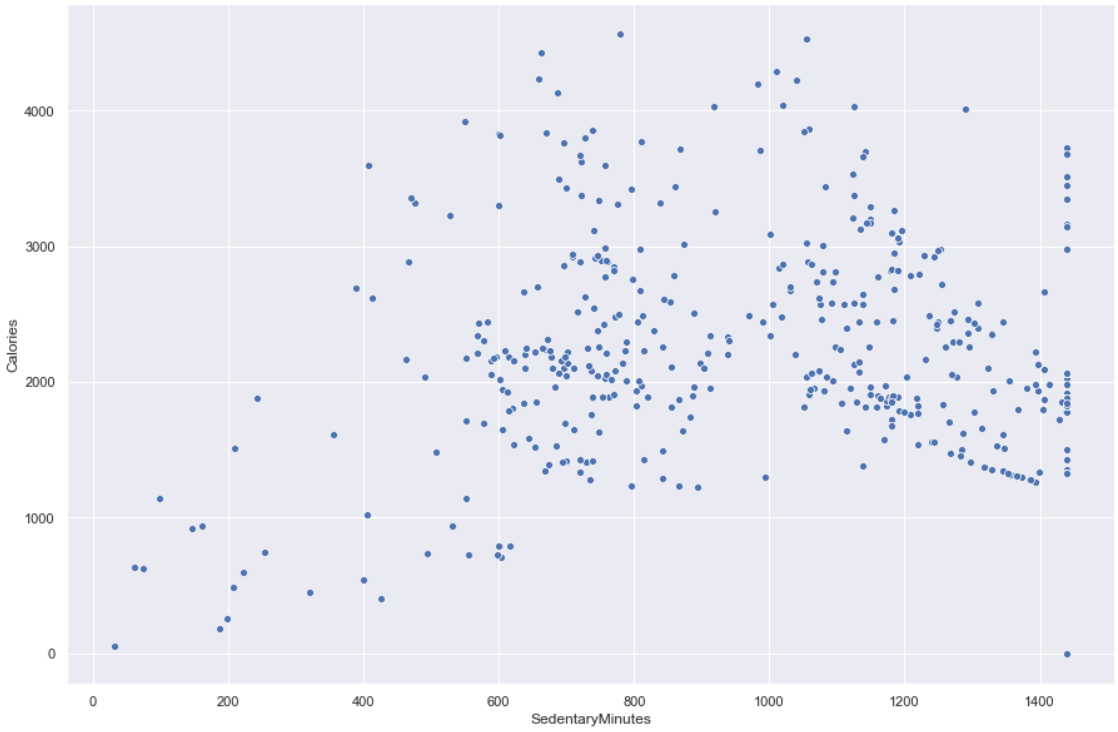
[298]:
```
plt.figure(figsize = (18,10))
plt.subplot(3,3,1)
sns.scatterplot(x = 'Calories', y = 'VeryActiveDistance',data = df, color =␣
↪'orange')
plt.subplot(3,3,2)
sns.scatterplot(x = 'Calories', y = 'ModeratelyActiveDistance',data = df, color␣
↪= 'b')
plt.subplot(3,3,3)
sns.scatterplot(x = 'Calories', y = 'LightActiveDistance',data = df, color =␣
↪'g')
```

[298]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x245a8d3b408>
```



These graphs are just telling us that the user has usually run more 'lightly' than being very active. We cannot conclude from these graphs that the fitbit was giving faulty readings because on each day a user would go through all 3 stages of activity.

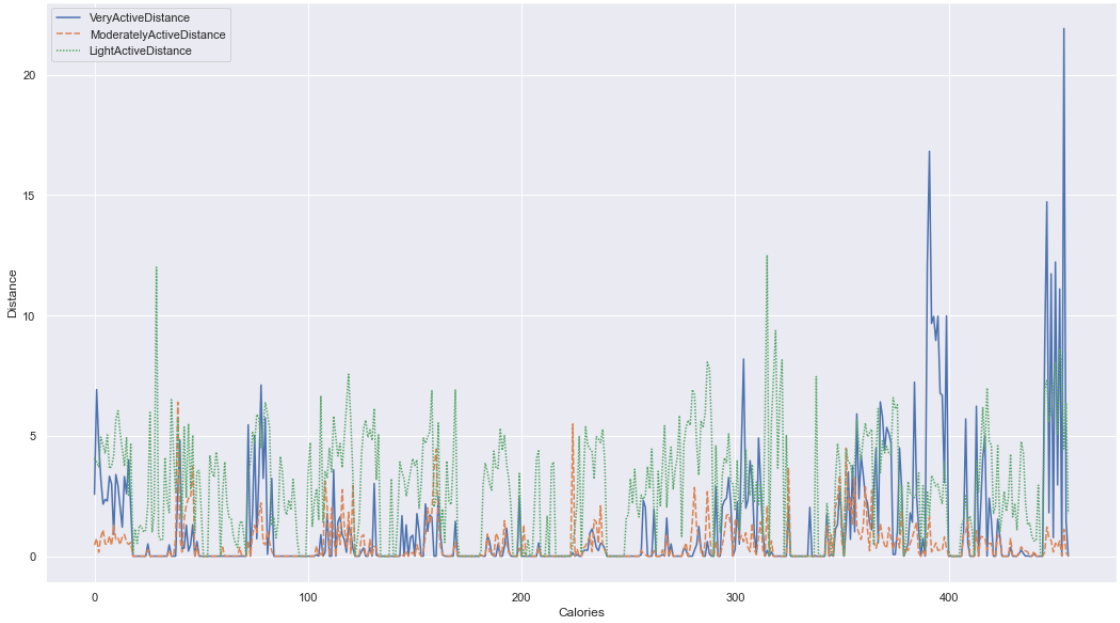
[306]:
```
plt.figure(figsize = (15,10))
sns.scatterplot(x = 'SedentaryMinutes', y = 'Calories', data = df)
sns.set(style= "darkgrid", color_codes=True)
```



Sedentary minutes also are not directly correlated with Calories. This is probably because after a heavy workout, the person might take rest for longer periods.

[300]:
```
plt.figure(figsize = (18,10))
sns.lineplot(data = df[['VeryActiveDistance',
       'ModeratelyActiveDistance', 'LightActiveDistance']])
plt.xlabel('Calories')
plt.ylabel('Distance')
```

[300]:
```
Text(0, 0.5, 'Distance')
```



The user has either been mostly lightlyactive while running or Very active.

[301]:
```
plt.figure(figsize = (15,10))
sns.lineplot(data = df[[ 'VeryActiveMinutes', 'FairlyActiveMinutes',
       'LightlyActiveMinutes', 'SedentaryMinutes', 'Calories']])
```

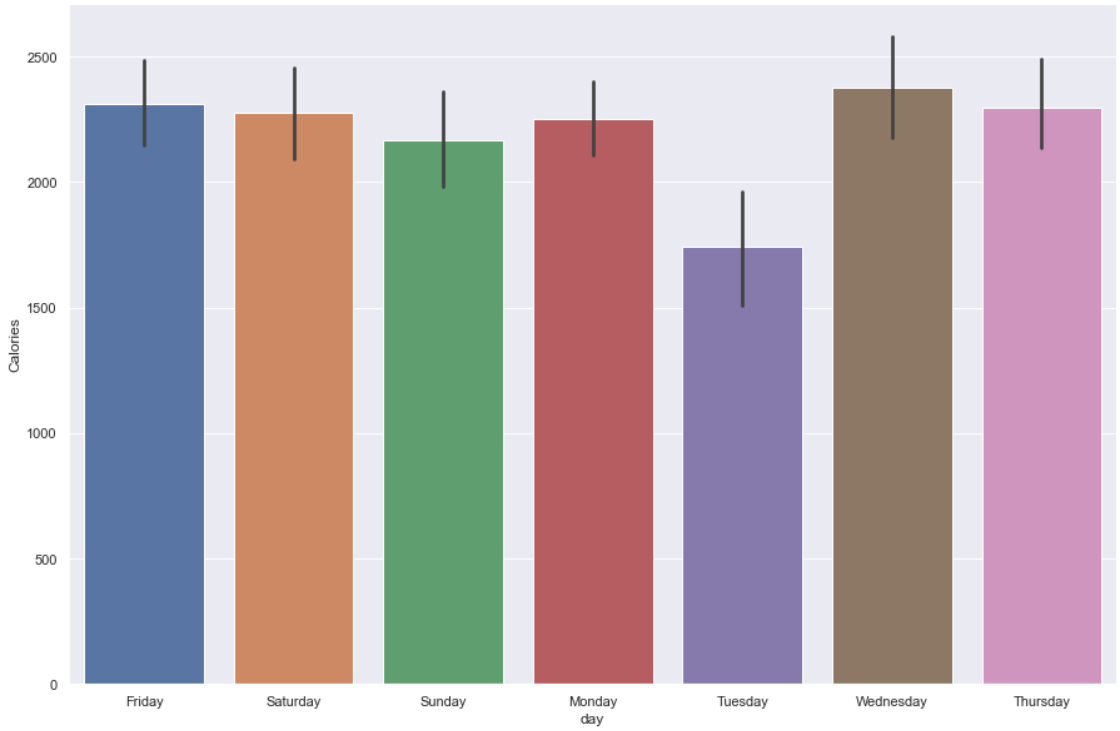
[301]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2456908f648>
```

The user of this bit has not constantly burnt calories but there his rate of calorie burn is extremely eratic. The user does not seem to remain consistant with his activity. This shows that the user is doing extra activities every day to remain healthy

[302]:
```
df['day'] = df['ActivityDate'].dt.day_name()
```

[303]:
```
plt.figure(figsize = (15,10))
sns.barplot(x = 'day', y = 'Calories', data = df)
```

[303]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x245a98a9248>
```

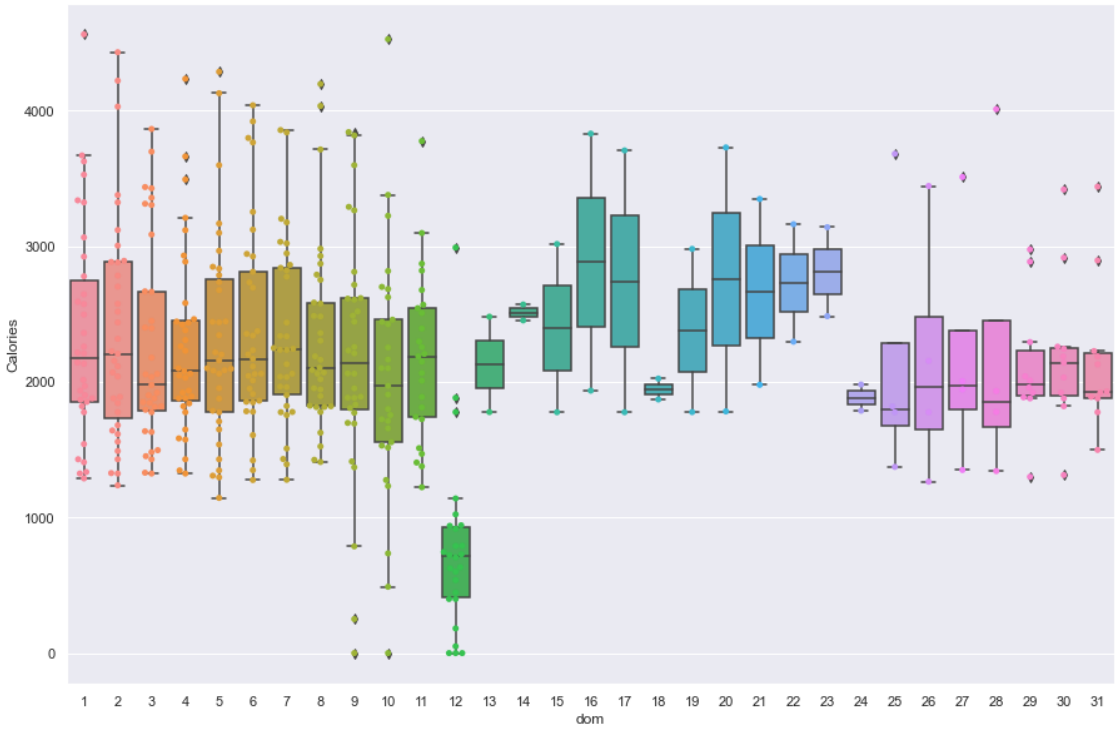


The user seems to be most active on wednesday.. Looking at this graph we can conclude that the user is consistant with burning calories. Tuesday seems to be rest day.

[304]:
```
df['dom'] = df['ActivityDate'].dt.day
```

[305]:
```
plt.figure(figsize = (15,10))
sns.boxplot(x = 'dom', y = 'Calories', data = df)
sns.swarmplot(x = 'dom', y = 'Calories', data = df)
```

[305]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x245a9253b88>
```



The users behaviour on a monthly basis is pretty consistant. He semes to burn more calories in the 3rd week of any month. Also the user is more consistant in the 3rd week since the range of the boxplot seems to be much lesser compared to the first 2 weeks of the month