Fitbit Tracker Data analysis
Using python and SQL

Data visualization using Tableau Desktop
See PowerPoint file for the result

```python
import pandas as pd
import datetime as dt
dailyActivity =
pd.read_csv("Desktop/Data/dailyActivity_merged.csv", index_col
= "Id")
```

I will do some cleaning and manipulation in the data. First I will convert Date column from String to datetime format.

```
dailyActivity["ActivityDate"] =
pd.to_datetime(dailyActivity["ActivityDate"])
```

if very active minutes, fairly active minutes and lightly active minutes are zero we will delete these rows. Because it is not possible that someone be sedentary all day long. So it means the data is not complete. Probably the user didn't carry the device in that day.
Tracker distance and Total distance is actually the same. we can take one of them.
I want to know how the very active, fairly active and lightly active distinguished from each other in this dataset. 2.2 meter per second is the minimum speed for running. So we will see if the user is running during veryActiveMinutes or just walk faster.

I will define a funtion to calculate the speed. If i want this function to be usable for all three categories, I have to make temporary Data frame for each category that first column is the Distance and second column is the minutes.

```python
def speedCalculator(row):
    kmHour = row[0] / (row[1] / 60)
    mph = kmHour * 0.62137
    row['Speed in mph'] = mph
    return row


activitySpeed = dailyActivity[["VeryActiveDistance",
"VeryActiveMinutes"]]
activitySpeed = activitySpeed.apply(speedCalculator, axis = 1)
```

And we do this again for all the three categories with a temporary dataframe. Then we join it to activitySpeed Dataframe. i will copy the Date column from dailyActivity dataset to this new activitySpeed Dataframe.

```python
activitySpeed.insert(loc = 0, column = "ActivityDate", value =
dailyActivity["ActivityDate"])
```

Then I calculated the average speed of Active, Fair and Light movement of each person with groupby method. The question that i must ask here is that what is the measurement refrence here? because 0.35 m/s is for Someone a FairSpeed and 0.44 is for someone Light Speed. So this must be asked from the business owner and it has a great affect on the analysis of this data. But in this matter i must skip it.

```python
activitySpeed.groupby(["Id"]).mean()
minuteIntensity =
pd.read_csv("Desktop/Data/minuteIntensitiesNarrow_merged.csv",
index_col = "Id").copy()
minuteCalories =
pd.read_csv("Desktop/Data/minuteCaloriesNarrow_merged.csv",
index_col = "Id").copy()
minuteMET =
pd.read_csv("Desktop/Data/minuteMETsNarrow_merged.csv",
index_col = "Id").copy()
sleep = pd.read_csv("Desktop/Data/sleepDay_merged.csv",
index_col = "Id").copy()
secondsHeartRate =
pd.read_csv('Desktop/Data/heartrate_seconds_merged.csv',
index_col = 'Id').copy()
minuteStepsNarrow =
pd.read_csv("Desktop/Data/minuteStepsNarrow_merged.csv",
index_col = "Id").copy()

sleep['SleepDay'] = pd.to_datetime(sleep['SleepDay'])
secondsHeartRate['Time'] =
pd.to_datetime(secondsHeartRate['Time'])

TempDF = pd.DataFrame(index = pd.date_range(start = '2016-04-
12', end = '2016-05-13', freq = 'T'), columns = ['Id',
'Values'])
```

# I define a temporary Dataframe and create a list of 33 IDs to use in the for loop.

```python
minuteMovement = pd.DataFrame(columns = ['Id', 'Values'])
ids = minuteCalories.groupby('Id').head(1).index

for i in ids:
    TempDF['Id'] = i
    minuteMovement = pd.concat([minuteMovement, TempDF])
```

I have made a full dataframe for every minute from 2016-04-12 to 2016-05-12 for every Participant ID and I have brought all the data that I had in minute into this Data frame. This process wasn't so hard and I have done it with a loop and merge method. The toughest one was Heart Beat. Heart beat was in some random seconds.
So i Had to write a couple of line of codes to achieve what i want and i will write it here.

```python
secondsHeartRate.insert(loc = 0, column = "Day", value = secondsHeartRate['Time'].dt.date)
secondsHeartRate.insert(loc = 1, column = "Hour", value = secondsHeartRate['Time'].dt.hour)
secondsHeartRate.insert(loc = 2, column = "Minute", value = secondsHeartRate['Time'].dt.minute)
        #Group by id and minute
minutesHeartRate = secondsHeartRate.groupby(["Id", "Day", "Hour", "Minute"]).mean()
        #Reset index to have the groupby as column not index
minutesHeartRate.reset_index(inplace = True)
minutesHeartRate.insert(loc = 0, column = 'Date', value = 0)
        #Write a function to combine the columns together and apply it to the
        #minutesHeartRate Dataframe
def datecreator(row):
    row['Date'] = str(row["Day"]) + ' ' + str(row["Hour"]) + ':' + str(row['Minute'])
    return row
minutesHeartRate = minutesHeartRate.apply(datecreator, axis = 1)
        #Convert the string to the datetime again
minutesHeartRate['Date'] = pd.to_datetime(minutesHeartRate['Date'])
minutesHeartRate.head()
```

Now I join Calories, Intensity, MET, HeartRate to this empty DataFrame one by one.

```python
pd.merge(left = minuteMovement, right = minutesHeartRate, left_on = 'Date', right_on = 'ActivityMinute', how = 'left')
```

I delete the rows with all NaN values. It narrows down our Dataframe from 1453320 rows to 1324885 rows.
The total minutes of these 30 days are 42600. I want to calculate the percentage of the carrying device in the day. I excluded the users fewer tham 0.5 % usage per day. It's 7 minutes

per day! So, it is completely conservative. This has excluded around 100 days from 940 days from different users.

```python
activitySpeed["%Usage"] = (activitySpeed["VeryActiveMinutes"]
+ activitySpeed["FairlyActiveMinutes"] +
activitySpeed["lightlyActiveMinutes"]) / 1440 * 100
activitySpeed.to_csv("Desktop/Data2/activitySpeed.csv")
activitySpeed = activitySpeed[activitySpeed["%Usage"] > 0.5]
minuteMovement.dropna(subset = ['Steps', 'Calories',
'Intensity', 'MET', 'HeartRate'], how = 'all', inplace = True)
```

I want to build this analysis with half an hour of intervals because if i want to get an average of heart beat or steps, 30 minutes makes more sense in the matter of running and heavy intensity workout.

```python
minuteMovement.groupby(minuteMovement.index // 30)
```

Here I want to clean some not useful data from this minuteMovement dataframe using SQL. I deleted the rows from any ID that have HeartRate = 0 and MET = 1 for more than two third of the rows from that ID.

```MySQL
SELECT * FROM minutemovement WHERE Id in
(SELECT df1.Id
FROM (SELECT Id, count(MET) MET, count(HeartRate) hrt
FROM minutemovement
WHERE HeartRate = 0 AND MET = 1
GROUP BY Id) df1
JOIN (SELECT Id, (count(Id) * 2 / 3) cnt FROM minutemovement
GROUP BY Id) df2
ON df1.Id = df2.Id
WHERE df1.MET < df2.cnt)
```

Because there is not any full data about the users, I have created a Data frame for the weight of the users using their Calories at night and during sleep. I don't know the age and the height of the users but i know their Calories at night. So, I calculated their BMR using their Weight. It is not completely precise but it is 90% accurate. Then i will categorize them in 3 groups of weights using my defined function.

```
Users = minutemovement_opt[(minutemovement_opt['Steps'] == 0)
& (minutemovement_opt['Intensity'] == 0)].groupby('Id').min()
```

```
def weight(row):
    if row['Calories'] > 36:
        row['Weight'] = 'Heavy'
    elif row['Calories'] > 28:
        row['Weight'] = 'Normal'
    else:
        row['Weight'] = 'Light'
    return row

Users = Users.apply(weight, axis = 1)
```

My Suggestions:

1. I guess most of the customers in Fitbit won’t use their devices during sleep. Probably because it is not comfortable enough or they plug it in charger. BellaBeat should notice this and find a solution. In order to know the user better, data like Heartbeat, deepness and stability during sleep is very important. They can even introduce a small and very light device just for the track of sleep.

2. The light weight users, who aren't normally active should get a notification every two or three days to do an intense activity. I suggest once in two or three days because their body is in good shape and if they get too much notifications for the activity they probably turn it off or ignore it after a while.

3. The heavy weight users should get a notification the second and third day, when they start to get unmotivated. This notification should be just motivational and not comparative, because it can make them completely discouraged.

Ambiguous Data:

1. The average of steps and Intensity are zero in some days. But the burnt Calories number is not. I don't know if the device is on them and they didn't move. Or the Calories burned are coming from their minimum burnt calories for their BMI and they don't carry the device. So there should be a new column in the dataset to track if the device is on the user or not. This would help a lot for analysis.

2. Each person consume approximately 70 Calories each hour as average. Slow walking have MET = 2.8 so if the user walk slow for 20 minutes in a day, will consume 70/3 * 2.8 = 70 Cal for 20 minutes. So Calories below 80 for 30 minutes must be considered light movement. But for some users it is considered fairly active, or even Sedentary. And I have found someone with 270 Calories in 30 minutes that has been considered Fairly Active. This affects the analysis and should be explained.