Importing library

In [60]:
```
import pandas as pd
import os
import datetime
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

1. Daily Activity data analysis

In [61]:
```
df_da = pd.read_csv(r"E:\Google Data Analytics\Data Analysis capestone project
```

In [62]:
```
df_da.head(50)
```

Out[62]:

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistanc
0	1503960366	04-12-2016	13162	8.50	8.50	0.(
1	1503960366	4/13/2016	10735	6.97	6.97	0.(
2	1503960366	4/14/2016	10460	6.74	6.74	0.(
3	1503960366	4/15/2016	9762	6.28	6.28	0.(
4	1503960366	4/16/2016	12669	8.16	8.16	0.(
5	1503960366	4/17/2016	9705	6.48	6.48	0.(
6	1503960366	4/18/2016	13019	8.59	8.59	0.(
7	1503960366	4/19/2016	15506	9.88	9.88	0.(
8	1503960366	4/20/2016	10544	6.68	6.68	0.(
9	1503960366	4/21/2016	9819	6.34	6.34	0.(
10	1503960366	4/22/2016	12764	8.13	8.13	0.(

In [63]:
```
len(df_da["Id"].unique())
```

Out[63]: 33

There is 33 users records in this data sets

In [64]:
```
df_da.dtypes
```

Out[64]:
```
Id                          int64
ActivityDate               object
TotalSteps                  int64
TotalDistance             float64
TrackerDistance           float64
LoggedActivitiesDistance  float64
VeryActiveDistance        float64
ModeratelyActiveDistance  float64
LightActiveDistance       float64
SedentaryActiveDistance   float64
VeryActiveMinutes           int64
FairlyActiveMinutes         int64
LightlyActiveMinutes        int64
SedentaryMinutes            int64
Calories                    int64
dtype: object
```

In [65]:
```
df_da.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 0 to 939
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Id                        940 non-null    int64  
 1   ActivityDate              940 non-null    object 
 2   TotalSteps                940 non-null    int64  
 3   TotalDistance             940 non-null    float64
 4   TrackerDistance           940 non-null    float64
 5   LoggedActivitiesDistance  940 non-null    float64
 6   VeryActiveDistance        940 non-null    float64
 7   ModeratelyActiveDistance  940 non-null    float64
 8   LightActiveDistance       940 non-null    float64
 9   SedentaryActiveDistance   940 non-null    float64
 10  VeryActiveMinutes         940 non-null    int64  
 11  FairlyActiveMinutes       940 non-null    int64  
 12  LightlyActiveMinutes      940 non-null    int64  
 13  SedentaryMinutes          940 non-null    int64  
 14  Calories                  940 non-null    int64  
dtypes: float64(7), int64(7), object(1)
memory usage: 110.3+ KB
```

There is 940 records available in the data sets

Data cleaning process

Transforming ActivityDate from object to datetime frame

In [66]:
```
df_da["ActivityDate"] = pd.to_datetime(df_da["ActivityDate"])
```

In [67]:
```
df_da.isnull().values.sum()
```

Out[67]: 0

As there is no null or missing value found so further data cleaning is not required

Finding statistical insights

In [68]:
```
df_da_description = df_da.describe()
```

In [69]:
```
df_da_description
```

Out[69]:

	Id	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	Ver
count	9.400000e+02	940.000000	940.000000	940.000000	940.000000	
mean	4.855407e+09	7637.910638	5.489702	5.475351	0.108171	
std	2.424805e+09	5087.150742	3.924606	3.907276	0.619897	
min	1.503960e+09	0.000000	0.000000	0.000000	0.000000	
25%	2.320127e+09	3789.750000	2.620000	2.620000	0.000000	
50%	4.445115e+09	7405.500000	5.245000	5.245000	0.000000	
75%	6.962181e+09	10727.000000	7.712500	7.710000	0.000000	
max	8.877689e+09	36019.000000	28.030001	28.030001	4.942142	

AS ID column and count row is not required I have deleted these values

In [70]:
```
df_da_description = df_da_description.drop(labels = ["Id"], axis = 1)
```

In [71]:
```
df_da_description = df_da_description.drop(labels = ["count"], axis = 0)
```

In [72]:
```
df_da_description
```

Out[72]:

	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance
mean	7637.910638	5.489702	5.475351	0.108171	1.502681
std	5087.150742	3.924606	3.907276	0.619897	2.658941
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	3789.750000	2.620000	2.620000	0.000000	0.000000
50%	7405.500000	5.245000	5.245000	0.000000	0.210000
75%	10727.000000	7.712500	7.710000	0.000000	2.052500
max	36019.000000	28.030001	28.030001	4.942142	21.920000

Note : As data is limited to only 33 users so I am using all 940 record for analysis

Some Findings :-

- Average total steps per day are 7638 which a little bit less for having health benefits for according to JAMA Neurology walking about 10,000 steps a day was linked to less cardiovascular disease (heart disease, stroke and heart failure), 13 types of cancer, and dementia.
- Average sedentary time is 991 minuted or 16 hours. so needs to reduce and as per data majority of participants are lightly active.

2. Daily Calories data analysis

In [73]:
```
df_dc = pd.read_csv(r"E:\Google Data Analytics\Data Analysis capestone project
```

Reading dailyActivity_merged.csv from local drive and gaining some insights

In [74]:
```
df_dc.head()
```

Out[74]:

	Id	ActivityDay	Calories
0	1503960366	4/12/2016	1985
1	1503960366	4/13/2016	1797
2	1503960366	4/14/2016	1776
3	1503960366	4/15/2016	1745
4	1503960366	4/16/2016	1863

In [75]:
```
len(df_dc["Id"].unique())
```

Out[75]:
```
33
```

There are 33 unique records in this datasets

In [76]:
```
df_dc.dtypes
```

Out[76]:
```
Id              int64
ActivityDay    object
Calories        int64
dtype: object
```

Data cleaning process

In ths datasets ActivityDay is object so needs to transform this to day

In [77]:
```
df_dc["ActivityDay"] = pd.to_datetime(df_dc["ActivityDay"])
```

In [78]:
```
df_dc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 940 entries, 0 to 939
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype         
---  ------       --------------  -----         
 0   Id           940 non-null    int64         
 1   ActivityDay  940 non-null    datetime64[ns]
 2   Calories     940 non-null    int64         
dtypes: datetime64[ns](1), int64(2)
memory usage: 22.2 KB
```

In [79]:
```
df_dc.isnull().values.sum()
```

Out[79]:
```
0
```

There are total 940 records available and there is no null value so further cleaning is not required

In [80]:
```
df_dc.describe()
```

Out[80]:

	Id	Calories
count	9.400000e+02	940.000000
mean	4.855407e+09	2303.609574
std	2.424805e+09	718.166862
min	1.503960e+09	0.000000
25%	2.320127e+09	1828.500000
50%	4.445115e+09	2134.000000
75%	6.962181e+09	2793.250000
max	8.877689e+09	4900.000000

Some findings:-

- 2,000 calories a day is used as a general guide for nutrition advice, but your calorie needs may be higher or lower depending on your age, sex, height, weight, and physical activity level. Eating too many calories per day is linked to overweight and obesity. As data shown on an average participants are taking aroung 2300 calories so participants needs to intake less calory to stay light, healthy and fit.

3. Sleep data analysis

In [81]:
```
df_sd = pd.read_csv(r"E:\Google Data Analytics\Data Analysis capestone project
```

In [82]:
```
df_sd.head()
```

Out[82]:

	Id	SleepDay	TotalSleepRecords	TotalMinutesAsleep	TotalTimeInBed
0	1503960366	4/12/2016 12:00:00 AM	1	327	346
1	1503960366	4/13/2016 12:00:00 AM	2	384	407
2	1503960366	4/15/2016 12:00:00 AM	1	412	442
3	1503960366	4/16/2016 12:00:00 AM	2	340	367
4	1503960366	4/17/2016 12:00:00 AM	1	700	712

In [83]:
```
len(df_da["Id"].unique())
```

Out[83]:
```
33
```

There is 33 unique records in this data sets

In [84]:
```
df_sd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 413 entries, 0 to 412
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Id                  413 non-null    int64 
 1   SleepDay            413 non-null    object
 2   TotalSleepRecords   413 non-null    int64 
 3   TotalMinutesAsleep  413 non-null    int64 
 4   TotalTimeInBed      413 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 16.3+ KB
```

In [85]:
```
df_sd.isnull().values.sum()
```

Out[85]:
```
0
```

Data cleaning process

- There is no null value in this data sets
- sleepDay column is object so no needs transform into datetime

In [86]:
```
df_sd["SleepDay"] = pd.to_datetime(df_sd["SleepDay"])
```

Description of data sets

In [87]:
```
df_sd.describe()
```

Out[87]:

	Id	TotalSleepRecords	TotalMinutesAsleep	TotalTimeInBed
count	4.130000e+02	413.000000	413.000000	413.000000
mean	5.000979e+09	1.118644	419.467312	458.639225
std	2.060360e+09	0.345521	118.344679	127.101607
min	1.503960e+09	1.000000	58.000000	61.000000
25%	3.977334e+09	1.000000	361.000000	403.000000
50%	4.702922e+09	1.000000	433.000000	463.000000
75%	6.962181e+09	1.000000	490.000000	526.000000
max	8.792010e+09	3.000000	796.000000	961.000000

Some insights:-

- As per data on an average participantstake around 40 munit to take sleep as total bed time is 413 and Minutes asleep is 419
- On an average participants sleep 1 time around 7 hours a day.

4. Working with weight data

In [88]:
```
df_wd = pd.read_csv(r"E:\Google Data Analytics\Data Analysis capestone project
```

In [89]:
```
df_wd.head()
```

Out[89]:

	Id	Date	WeightKg	WeightPounds	Fat	BMI	IsManualReport	
0	1503960366	5/2/2016 11:59:59 PM	52.599998	115.963147	22.0	22.650000	True	14622335
1	1503960366	5/3/2016 11:59:59 PM	52.599998	115.963147	NaN	22.650000	True	14623199
2	1927972279	4/13/2016 1:08:52 AM	133.500000	294.317120	NaN	47.540001	False	14605097
3	2873212765	4/21/2016 11:59:59 PM	56.700001	125.002104	NaN	21.450001	True	14612831
4	2873212765	5/12/2016 11:59:59 PM	57.299999	126.324875	NaN	21.690001	True	14630975

In [90]:
```
len(df_wd["Id"].unique())
```

Out[90]: 8

There is record of 8 perticipent

In [91]:
```
df_wd.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67 entries, 0 to 66
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Id              67 non-null     int64  
 1   Date            67 non-null     object 
 2   WeightKg        67 non-null     float64
 3   WeightPounds    67 non-null     float64
 4   Fat             2 non-null      float64
 5   BMI             67 non-null     float64
 6   IsManualReport  67 non-null     bool   
 7   LogId           67 non-null     int64  
dtypes: bool(1), float64(4), int64(2), object(1)
memory usage: 3.9+ KB
```

Data cleaning process

In [92]:
```
df_wd.isnull().values.sum()
```

Out[92]: 65

As clearly shown in Fat column out of 67, 65 having null value so removing the Fat column

In [93]:
```
df_wd = df_wd.drop(["Fat"], axis=1)
```

As date is object so transforming object into datetime

In [94]:
```
df_wd["Date"] = pd.to_datetime(df_wd["Date"])
```

Description of Data sets

In [95]:
```
df_wd.describe()
```

Out[95]:

	Id	WeightKg	WeightPounds	BMI	LogId
count	6.700000e+01	67.000000	67.000000	67.000000	6.700000e+01
mean	7.009282e+09	72.035821	158.811801	25.185224	1.461772e+12
std	1.950322e+09	13.923206	30.695415	3.066963	7.829948e+08
min	1.503960e+09	52.599998	115.963147	21.450001	1.460444e+12
25%	6.962181e+09	61.400002	135.363832	23.959999	1.461079e+12
50%	6.962181e+09	62.500000	137.788914	24.389999	1.461802e+12
75%	8.877689e+09	85.049999	187.503152	25.559999	1.462375e+12
max	8.877689e+09	133.500000	294.317120	47.540001	1.463098e+12

Some insights:-

- The formula is BMI = kg/m2 where kg is a person's weight in kilograms and m2 is their height in metres squared. A BMI of 25.0 or more is overweight, while the healthy range is 18.5 to 24.9. So participents are overweight and needs to reduce some weight.
- Note: There is a limitaton of data as participents height is not available.

Marging Data

Marging Sleep and activity data

In [96]:
```
df_damarge = df_da
```

Puting two dataframe in two variable

In [97]:
```
df_sdmarge = df_sd
```

In [98]:
```
df_damarge.head()
```

Out[98]:

	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance
0	1503960366	2016-04-12	13162	8.50	8.50	0.0
1	1503960366	2016-04-13	10735	6.97	6.97	0.0
2	1503960366	2016-04-14	10460	6.74	6.74	0.0
3	1503960366	2016-04-15	9762	6.28	6.28	0.0
4	1503960366	2016-04-16	12669	8.16	8.16	0.0

Changing column ActivityDate as Date

In [99]:
```
df_damarge = df_damarge.rename({"ActivityDate": "Date"}, axis = 1)
```

In [100]:
```
df_sdmarge.head()
```

Out[100]:

	Id	SleepDay	TotalSleepRecords	TotalMinutesAsleep	TotalTimeInBed
0	1503960366	2016-04-12	1	327	346
1	1503960366	2016-04-13	2	384	407
2	1503960366	2016-04-15	1	412	442
3	1503960366	2016-04-16	2	340	367
4	1503960366	2016-04-17	1	700	712

Rename SleepDay as Date

In [101]:
```
df_sdmarge = df_sdmarge.rename({"SleepDay" : "Date"}, axis = 1)
```

In [102]:
```
df_merged = df_damarge.merge(df_sdmarge, on = ["Id", "Date"])
```

In [103]:
```
df_merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 413 entries, 0 to 412
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Id                         413 non-null    int64
 1   Date                       413 non-null    datetime64[ns]
 2   TotalSteps                 413 non-null    int64
 3   TotalDistance              413 non-null    float64
 4   TrackerDistance            413 non-null    float64
 5   LoggedActivitiesDistance   413 non-null    float64
 6   VeryActiveDistance         413 non-null    float64
 7   ModeratelyActiveDistance   413 non-null    float64
 8   LightActiveDistance        413 non-null    float64
 9   SedentaryActiveDistance    413 non-null    float64
 10  VeryActiveMinutes          413 non-null    int64
 11  FairlyActiveMinutes        413 non-null    int64
 12  LightlyActiveMinutes       413 non-null    int64
 13  SedentaryMinutes           413 non-null    int64
 14  Calories                   413 non-null    int64
 15  TotalSleepRecords          413 non-null    int64
 16  TotalMinutesAsleep         413 non-null    int64
 17  TotalTimeInBed             413 non-null    int64
dtypes: datetime64[ns](1), float64(7), int64(10)
memory usage: 61.3 KB
```

Now records in merged cell is 413

Visualization:-

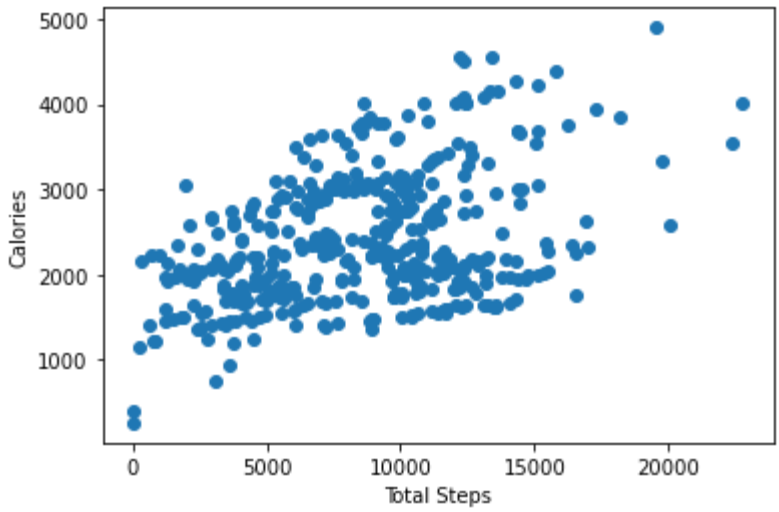
In [104]:
```
df_merged.head()
```

Out[104]:

	Id	Date	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance
0	1503960366	2016-04-12	13162	8.50	8.50	0.0
1	1503960366	2016-04-13	10735	6.97	6.97	0.0
2	1503960366	2016-04-15	9762	6.28	6.28	0.0
3	1503960366	2016-04-16	12669	8.16	8.16	0.0
4	1503960366	2016-04-17	9705	6.48	6.48	0.0

In [105]:
```
%matplotlib inline
x = df_merged["TotalSteps"]
y = df_merged["Calories"]
plt.xlabel("Total Steps")
plt.ylabel("Calories")
plt.scatter(x, y)
```

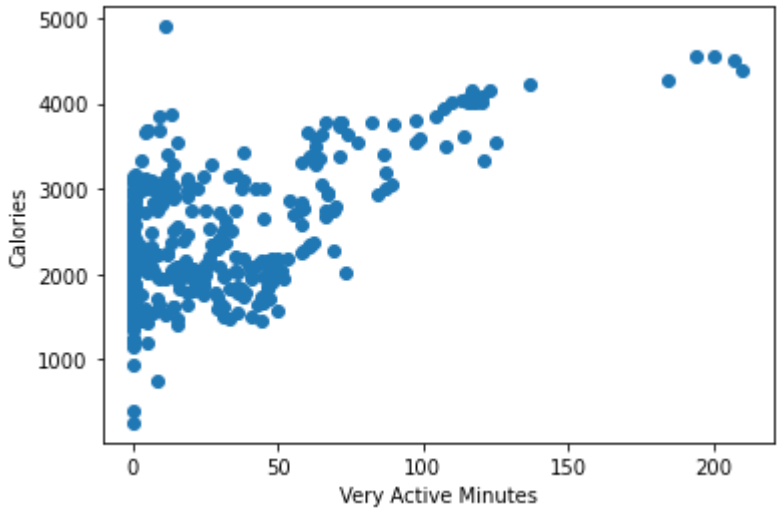
Out[105]: <matplotlib.collections.PathCollection at 0x1cb21df4910>



The above plot clearly shown that the more you take steps the more you burn Calori

In [106]:
```
%matplotlib inline
x = df_merged["VeryActiveMinutes"]
y = df_merged["Calories"]
plt.xlabel("Very Active Minutes")
plt.ylabel("Calories")
plt.scatter(x, y)
```

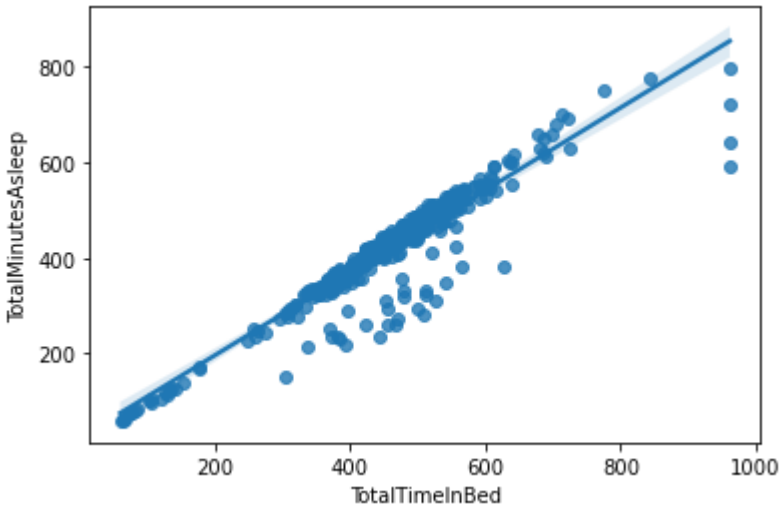
Out[106]: <matplotlib.collections.PathCollection at 0x1cb21e51bb0>



This above plot shows active minute has more positive correlaton with Calories then steps

In [107]:
```
sns.regplot(x = "TotalTimeInBed", y = "TotalMinutesAsleep", data = df_merged)
```

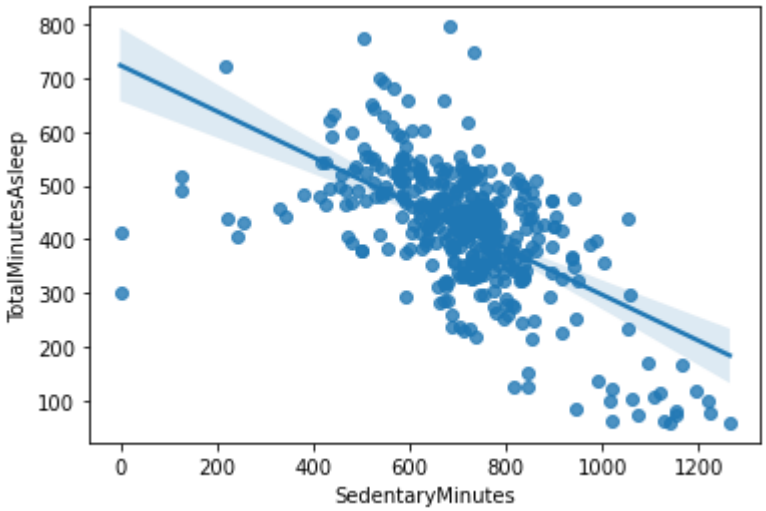
Out[107]:
```
<AxesSubplot:xlabel='TotalTimeInBed', ylabel='TotalMinutesAsleep'>
```



- The relationship between Total Minutes Asleep and Total Time in Bed looks linear. So if the Bellabeat users want to improve their sleep, we should consider using notification to go to sleep.

In [108]:
```
sns.regplot(x = "SedentaryMinutes", y = "TotalMinutesAsleep", data = df_merged
```

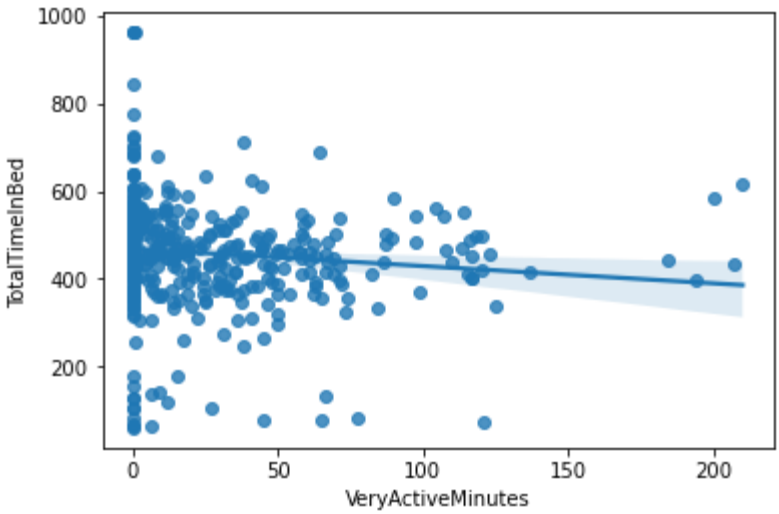
Out[108]:
```
<AxesSubplot:xlabel='SedentaryMinutes', ylabel='TotalMinutesAsleep'>
```



- The above plot clearly shows that SedentaryMinutes is negatively correlated to TotalMinutesAsleep so, if Bellabeat users want to improve their sleep, Bellabeat app can recommend reducing sedentary time.

In [109]:
```
sns.regplot(x = "VeryActiveMinutes", y = "TotalTimeInBed", data = df_merged)
```

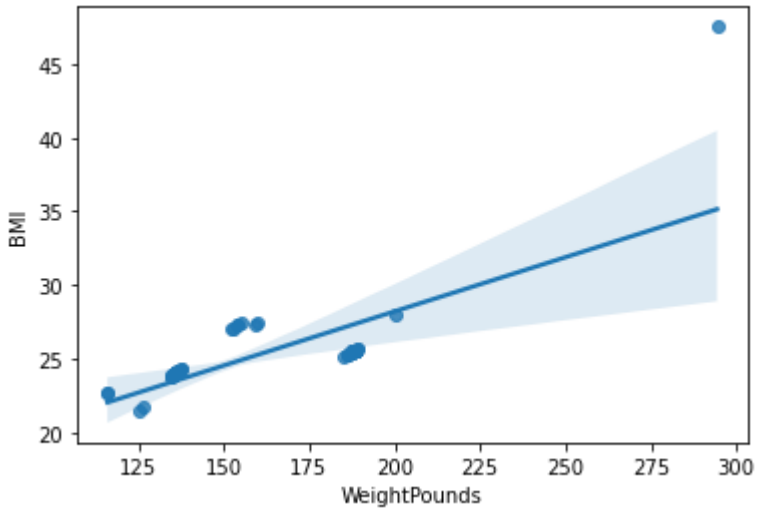
Out[109]:
```
<AxesSubplot:xlabel='VeryActiveMinutes', ylabel='TotalTimeInBed'>
```



- The plot shows that there is less or no correlation between TotalTimeInBed and VeryActiveMinutes

In [110]:
```
sns.regplot(x= "WeightPounds",y = "BMI", data = df_wd)
```

Out[110]:
```
<AxesSubplot:xlabel='WeightPounds', ylabel='BMI'>
```



- World Health Organisation (WHO) also recommends BMI as the most useful population level measure of overweight and obesity, and is used as the same for both sexes and in all ages of adults. So BMI of >25 kg/m2 and >30 kg/m2 are considered to be overweight and obese in adults irrespective of gender and age.

Create pie chart:-

In [111]:
```
df_pie = [df_merged["VeryActiveMinutes"], df_merged["FairlyActiveMinutes"], df
```

In [112]:
```
pie_df = pd.DataFrame(data = df_pie)
```

In [113]:
```
pie_df["Results"] = pie_df.sum(axis=1)
```

In [114]:
```
pie_df
```

Out[114]:

	0	1	2	3	4	5	6	7	8	9	...	404	405	406	407
VeryActiveMinutes	25	21	29	36	38	50	28	19	41	39	...	8	0	0	6
FairlyActiveMinutes	13	19	34	10	20	31	12	8	21	5	...	45	0	0	14
LightlyActiveMinutes	328	217	209	221	164	264	205	211	262	238	...	232	112	310	380

3 rows × 414 columns

In [115]:
```
pie_dflist = pie_df["Results"]
pie_dflist = pd.DataFrame(data=pie_dflist)
```

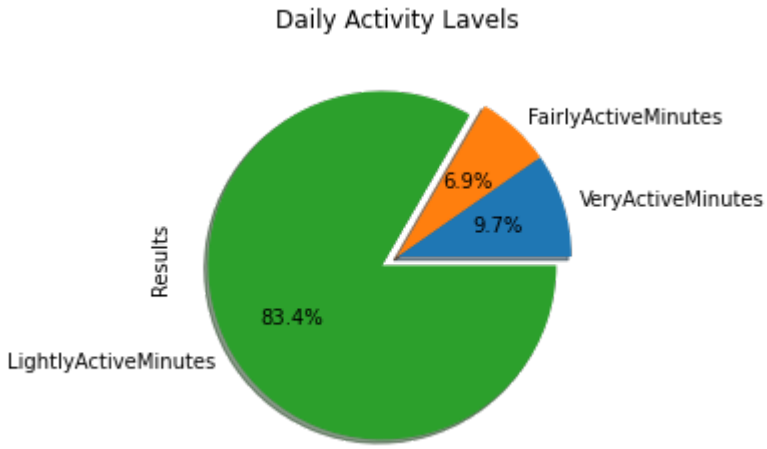
In [116]:
```
pie_dflist
```

Out[116]:

	Results
VeryActiveMinutes	10403
FairlyActiveMinutes	7450
LightlyActiveMinutes	89561

In [117]:
```
pie_dflist.plot(kind = "pie",y='Results', title="Daily Activity Lavels", legen
                    autopct='%1.1f%%', explode=(0, 0, 0.1), \
                    shadow=True, startangle=0)
```

Out[117]:
```
<AxesSubplot:title={'center':'Daily Activity Lavels'}, ylabel='Results'>
```



- This above pie chart is clearly showing that 83% participants are lightly active which is

more then majority. So, At least 150 minutes a week of moderate intensity activity such as brisk walking. At least 2 days a week of activities that strengthen muscles. Activities to improve balance such as standing on one foot. Aim for the recommended activity level but be as active as one is able.

Description of statistical summary :-

- Mean value is average value
- Higher standerd daviation meand maximum value is not closest to the mean
- 25% data is lesthen given value
- 50% data is less then given value
- 75% data is less then given value
- max is the maximum value
- min is the minimum value

In [ ]: