Fitbit_Data_Analysis-Copy1

November 27, 2018

0.1 Bring the Avengers!

libraries that will ease our data analysis

```
In [1]: import matplotlib
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt # default library for making plots
        import seaborn as sns # for making prettier plots!
        import datetime
        import math

        import warnings
        warnings.filterwarnings("ignore")

        pd.set_option('display.max_columns', None) # to see all columns in the pd dataframe
        %config InlineBackend.figure_format = 'retina' # for the crispier version of plots for

        matplotlib.style.use('ggplot')
        plt.rcParams['figure.figsize'] = [12.0, 6.0]
```

0.2 Importing the Dataset

```
In [2]: filename = 'data/database_main.xls'

        data = pd.read_excel(filename, sheet_name="main", converters= {'Date': pd.to_datetime}
        data.set_index(pd.to_datetime(data.Date),inplace=True)

        print("Loaded db successfully!")
```

```
Loaded db successfully!
```

0.3 Building new columns and sanitising the data

Firstly we will be removing the activity where there were no steps recorded. For sleep data, we will clear out rows where there was no "Deep Sleep" entries

In [3]:
```
dayCodes = ['', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Su
minDayCodes = ['', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']

days = {1: 'Mon', 2: 'Tue', 3: 'Wed', 4: 'Thu', 5: 'Fri', 6: 'Sat', 7: 'Sun'}

def is_nan(x):
    return (x is np.nan or x != x)

def defineSleepBucket(row):
    sleepEntry = row['Sleep Start time']
    if not is_nan(sleepEntry):
        sleepTime0 = datetime.datetime.strptime(sleepEntry, '%Y-%m-%dT%H:%M:%S.000').ti
        if sleepTime0.minute > 30:
            return (sleepTime0.hour + 1)
            #return '{}:{}'.format(str((sleepTime0.hour + 1)), '00')
        elif sleepTime0.minute > 0:
            return sleepTime0.hour + 0.5
            #return '{}:{}'.format(str(sleepTime0.hour), ':30')
    else:
        return np.nan

def defineAwakeBucket(row):
    awakeEntry = row['Sleep End time']
    if not is_nan(awakeEntry):
        awakeEntry0 = datetime.datetime.strptime(awakeEntry, '%Y-%m-%dT%H:%M:%S.000').t
        if awakeEntry0.minute > 30:
            return (awakeEntry0.hour + 1)
            #return '{}:{}'.format(str((awakeEntry0.hour + 1)), '00')
        elif awakeEntry0.minute > 0:
            return awakeEntry0.hour + 0.5
            #return '{}:{}'.format(str(awakeEntry0.hour), ':30')
    else:
        return np.nan

#remove all entries where there was no steps recorded. i.e no activity
data = data[data['Steps'].notnull()]

data['Day Label'] = data['Day of Week'].apply(lambda x: days[x])
data['Active exercise'] = data['Minutes Very Active'] > 40

#Build additional Sleep columns
data['Sleep Bucket'] = data.apply(defineSleepBucket, axis=1)
data['Awake Bucket'] = data.apply(defineAwakeBucket, axis=1)

data['% Awake'] = 100 - (data['% Deep sleep'] + data['% REM sleep'] + data['% Light sle
data['% Restorative sleep'] = data['% Deep sleep'] + data['% REM sleep']
data['Restorative sleep mins'] = data['Minutes Deep sleep'] + data['Minutes REM sleep']
```

```
        #remove all entries where there was no Deep sleep recorded
        sleepData = data[data['% Deep sleep'].notnull()]
```

```
In [4]: print("Contains {} records ranging from {} to {}".format(str(len(data)), str(data.iloc
        print('Contains {} entries of Sleep data'.format(len(sleepData)))

        data.head()
```

```
Contains 116 records ranging from 2018-06-24 00:00:00 to 2018-10-17 00:00:00
Contains 76 entries of Sleep data
```

Out[4]:

Date	Date	Day of Week	Is Weekday	Is Weekend	Calories Burned
2018-06-24	2018-06-24	7	False	True	1996
2018-06-25	2018-06-25	1	True	False	2863
2018-06-26	2018-06-26	2	True	False	3398
2018-06-27	2018-06-27	3	True	False	3442
2018-06-28	2018-06-28	4	True	False	1701

Date	Calories BMR	Steps	Distance (Km)	Elevation (Ft)
2018-06-24	1690	3367	3.01	0.00
2018-06-25	1685	8394	5.55	36.58
2018-06-26	1683	13569	9.68	24.38
2018-06-27	1682	13337	9.33	33.53
2018-06-28	1682	559	0.36	18.29

Date	Resting Heart Rate	Floors	Minutes Sedentary
2018-06-24	59.0	0	1377
2018-06-25	58.0	12	709
2018-06-26	57.0	8	687
2018-06-27	57.0	11	665
2018-06-28	57.0	6	1025

Date	Minutes Lightly Active	Minutes Fairly Active
2018-06-24	47	2
2018-06-25	244	42
2018-06-26	231	33
2018-06-27	243	30
2018-06-28	8	0

Date	Minutes Very Active	Activity Calories	Active Score
2018-06-24	14	362	-1
2018-06-25	14	1361	-1

```
2018-06-26                102              2004           -1
2018-06-27                100              2035           -1
2018-06-28                  0                25           -1

            Cardio minutes  Cardio calories  Fat Burn minutes  \
Date
2018-06-24            11.0        133.10892             14.0
2018-06-25             2.0         12.40518             94.0
2018-06-26            10.0        103.59112            187.0
2018-06-27            11.0        114.28908            217.0
2018-06-28             0.0          0.00000              0.0

            Fat Burn calories  Peak minutes  Peak calories  \
Date
2018-06-24           89.20880           0.0            0.0
2018-06-25          514.11279           0.0            0.0
2018-06-26         1245.54876           0.0            0.0
2018-06-27         1310.46804           0.0            0.0
2018-06-28            0.00000           0.0            0.0

            Normal Cardio calories  Normal Cardio minutes  Sleep Efficiency  \
Date
2018-06-24               446.51352                  284.0              92.0
2018-06-25              2311.92765                 1326.0              95.0
2018-06-26              2034.05724                 1232.0              89.0
2018-06-27              1986.03570                 1187.0              97.0
2018-06-28               403.96356                  329.0               NaN

            Minutes Asleep  Minutes to fall asleep         Sleep Start time  \
Date
2018-06-24           379.0                     0.0  2018-06-24T22:17:00.000
2018-06-25           333.0                     0.0  2018-06-25T22:49:30.000
2018-06-26           351.0                     0.0  2018-06-26T22:25:30.000
2018-06-27           365.0                     0.0  2018-06-27T22:36:30.000
2018-06-28             NaN                     NaN                      NaN

                  Sleep End time  Time in bed  Minutes Deep sleep  \
Date
2018-06-24  2018-06-25T05:28:30.000        431.0                56.0
2018-06-25  2018-06-26T05:16:30.000        387.0                75.0
2018-06-26  2018-06-27T05:07:30.000        402.0                53.0
2018-06-27  2018-06-28T05:24:00.000        407.0                66.0
2018-06-28                     NaN          NaN                 NaN

            Deep sleep count  Minutes Light sleep  Light sleep count  \
Date
2018-06-24               2.0                244.0               26.0
2018-06-25               3.0                214.0               27.0
```

```
2018-06-26              3.0               223.0              28.0
2018-06-27              4.0               197.0              32.0
2018-06-28              NaN                 NaN               NaN

            Minutes REM sleep  REM sleep count  Minutes Awake  \
Date
2018-06-24              79.0             6.0           52.0
2018-06-25              44.0             7.0           54.0
2018-06-26              75.0             7.0           51.0
2018-06-27             102.0            10.0           42.0
2018-06-28               NaN             NaN            NaN

            Minutes Awake count  % Deep sleep  % Light sleep  % REM sleep  \
Date
2018-06-24                 25.0          13.0           57.0         19.0
2018-06-25                 30.0          20.0           56.0         12.0
2018-06-26                 29.0          14.0           56.0         19.0
2018-06-27                 32.0          17.0           49.0         26.0
2018-06-28                  NaN           NaN            NaN          NaN

           Day Label  Active exercise  Sleep Bucket  Awake Bucket  % Awake  \
Date
2018-06-24       Sun            False          22.5           5.5     11.0
2018-06-25       Mon            False          23.0           5.5     12.0
2018-06-26       Tue             True          22.5           5.5     11.0
2018-06-27       Wed             True          23.0           5.5      8.0
2018-06-28       Thu            False           NaN           NaN      NaN

            % Restorative sleep  Restorative sleep mins
Date
2018-06-24                 32.0                   135.0
2018-06-25                 32.0                   119.0
2018-06-26                 33.0                   128.0
2018-06-27                 43.0                   168.0
2018-06-28                  NaN                     NaN
```

In [5]:
```
#data['Active mins > 40'] = data['Minutes Very Active'] > 30
#data.resample('W').mean()
```

Let's take a quick glance how does does the numbers behave on weekdays vs weekends

In [6]:
```
dayGroupedData = data.groupby(['Day of Week']).mean()
dayTypeGroupedData = data.groupby(['Is Weekday']).mean()

dayTypeGroupedData
```

Out[6]:
```
            Day of Week  Is Weekend  Calories Burned  Calories BMR  \
Is Weekday
False          6.515152         1.0      2679.545455   1667.606061
```

```
True           2.963855         0.0      2742.457831   1610.048193

                    Steps  Distance (Km)  Elevation (Ft)  Resting Heart Rate  \
Is Weekday
False       8789.909091       5.716061       33.527879           61.666667
True        8127.807229       5.378795       23.098916           61.121622

                  Floors  Minutes Sedentary  Minutes Lightly Active  \
Is Weekday
False       11.000000         943.090909              215.515152
True         7.578313         788.963855              176.734940

            Minutes Fairly Active  Minutes Very Active  Activity Calories  \
Is Weekday
False                   20.181818            26.272727        1222.242424
True                    26.469880            41.626506        1247.108434

            Active Score  Cardio minutes  Cardio calories  Fat Burn minutes  \
Is Weekday
False               -1.0        7.037037        54.388412        136.666667
True                -1.0        8.184211        83.742450        146.105263

            Fat Burn calories  Peak minutes  Peak calories  \
Is Weekday
False              659.956047      0.074074       0.950886
True               785.508394      0.789474      10.055665

            Normal Cardio calories  Normal Cardio minutes  Sleep Efficiency  \
Is Weekday
False                 1774.602004            1025.777778         95.380952
True                  1739.526390            1089.631579         95.830508

            Minutes Asleep  Minutes to fall asleep  Time in bed  \
Is Weekday
False           376.450000                     0.0   414.428571
True            375.535714                     0.0   422.423729

            Minutes Deep sleep  Deep sleep count  Minutes Light sleep  \
Is Weekday
False                73.150000             3.400           219.250000
True                 73.660714             3.875           222.732143

            Light sleep count  Minutes REM sleep  REM sleep count  \
Is Weekday
False              28.900000          84.050000         7.650000
True               30.017857          79.142857         7.767857

            Minutes Awake  Minutes Awake count  % Deep sleep  % Light sleep  \
```

```
Is Weekday
False           51.900000            28.850000     17.750000      51.750000
True            53.178571            30.892857     17.678571      52.428571

            % REM sleep  Active exercise  Sleep Bucket  Awake Bucket  \
Is Weekday
False         19.750000         0.272727     16.452381      7.333333
True          18.964286         0.457831     20.931034      6.456140

              % Awake  % Restorative sleep  Restorative sleep mins
Is Weekday
False       10.750000            37.500000              157.200000
True        10.928571            36.642857              152.803571
```

In [7]:
```
data.groupby(['Day of Week']).mean()
```

Out[7]:
```
             Is Weekday  Is Weekend  Calories Burned  Calories BMR  \
Day of Week
1                   1.0         0.0      2766.764706   1679.764706
2                   1.0         0.0      2932.000000   1613.529412
3                   1.0         0.0      2848.588235   1611.411765
4                   1.0         0.0      2695.375000   1604.187500
5                   1.0         0.0      2449.562500   1536.687500
6                   0.0         1.0      2884.000000   1679.500000
7                   0.0         1.0      2487.117647   1656.411765

                     Steps  Distance (Km)  Elevation (Ft)  Resting Heart Rate  \
Day of Week
1              7474.117647       4.756471       35.142353           61.200000
2              9709.941176       6.495294       19.542941           60.625000
3              8856.588235       5.938235       22.770000           60.823529
4              7546.250000       5.081250       20.955000           61.500000
5              6948.562500       4.556875       16.574375           61.666667
6             10655.500000       6.877500       43.625000           62.076923
7              7034.058824       4.622941       24.024706           61.285714

                Floors  Minutes Sedentary  Minutes Lightly Active  \
Day of Week
1            11.529412         892.235294              197.647059
2             6.411765         695.588235              192.470588
3             7.470588         749.647059              179.705882
4             6.875000         806.750000              166.437500
5             5.437500         802.437500              144.937500
6            14.312500         955.437500              228.187500
7             7.882353         931.470588              203.588235

             Minutes Fairly Active  Minutes Very Active  Activity Calories  \
Day of Week
```

1	27.470588	24.470588	1188.058824
2	31.411765	62.705882	1517.058824
3	31.176471	51.411765	1370.058824
4	26.625000	38.187500	1184.937500
5	15.000000	30.500000	954.562500
6	26.875000	36.687500	1424.812500
7	13.882353	16.470588	1031.588235

Day of Week	Active Score	Cardio minutes	Cardio calories	Fat Burn minutes
1	-1.0	3.933333	39.324835	147.266667
2	-1.0	9.823529	102.797649	160.176471
3	-1.0	9.117647	90.933628	153.647059
4	-1.0	10.142857	101.458271	150.642857
5	-1.0	7.615385	81.592786	111.615385
6	-1.0	6.538462	55.732043	173.769231
7	-1.0	7.500000	53.140755	102.214286

Day of Week	Fat Burn calories	Peak minutes	Peak calories
1	723.268431	0.000000	0.000000
2	917.109046	1.058824	13.772624
3	835.312679	0.823529	10.401669
4	810.832673	0.571429	6.402520
5	592.829592	1.538462	20.279408
6	903.743161	0.076923	0.951635
7	433.582299	0.071429	0.950190

Day of Week	Normal Cardio calories	Normal Cardio minutes	Sleep Efficiency
1	1884.135575	1165.733333	95.875000
2	1782.886879	1125.058824	94.846154
3	1620.286221	1010.647059	96.769231
4	1711.505051	1085.357143	96.000000
5	1702.073737	1063.384615	95.500000
6	1759.771382	994.000000	95.500000
7	1788.373296	1055.285714	95.272727

Day of Week	Minutes Asleep	Minutes to fall asleep	Time in bed
1	374.200000	0.0	425.812500
2	377.416667	0.0	427.538462
3	371.230769	0.0	419.615385
4	386.363636	0.0	443.636364
5	362.400000	0.0	369.500000
6	361.100000	0.0	412.500000
7	391.800000	0.0	416.181818

Day of Week	Minutes Deep sleep	Deep sleep count	Minutes Light sleep
1	68.533333	3.533333	226.200000
2	72.666667	3.833333	225.333333
3	75.307692	4.153846	217.384615
4	83.000000	4.272727	224.454545
5	66.600000	3.400000	216.200000
6	66.600000	3.300000	207.700000
7	79.700000	3.500000	230.800000

Day of Week	Light sleep count	Minutes REM sleep	REM sleep count
1	29.600000	79.466667	8.133333
2	29.750000	79.416667	8.000000
3	30.384615	78.538462	6.923077
4	31.000000	78.909091	8.181818
5	28.800000	79.600000	7.400000
6	28.400000	86.800000	8.000000
7	29.400000	81.300000	7.300000

Day of Week	Minutes Awake	Minutes Awake count	% Deep sleep	% Light sleep
1	52.866667	30.866667	16.466667	53.400000
2	52.833333	31.166667	17.583333	52.750000
3	48.384615	30.769231	18.384615	52.230769
4	57.272727	31.545455	19.181818	51.272727
5	58.400000	29.200000	16.400000	51.800000
6	51.400000	27.600000	17.100000	51.200000
7	52.400000	30.100000	18.400000	52.300000

Day of Week	% REM sleep	Active exercise	Sleep Bucket	Awake Bucket
1	19.066667	0.117647	23.033333	6.218750
2	19.000000	0.705882	21.192308	6.153846
3	19.153846	0.705882	19.538462	6.083333
4	18.181818	0.437500	21.000000	6.550000
5	19.800000	0.312500	18.000000	8.333333
6	20.900000	0.437500	10.900000	8.100000
7	18.600000	0.117647	21.500000	6.636364

Day of Week	% Awake	% Restorative sleep	Restorative sleep mins
1	11.066667	35.533333	148.000000
2	10.666667	36.583333	152.083333
3	10.230769	37.538462	153.846154
4	11.363636	37.363636	161.909091
5	12.000000	36.200000	146.200000
6	10.800000	38.000000	153.400000

```
7            10.700000            37.000000            161.000000
```

0.4 Utilities

```
In [8]: def getDayLabel(dayNum):
            return dayCodes[dayNum]

        def plot_heatmap(corrmat, correlationOf, title, darkTheme=False):
            if darkTheme:
                sns.set(style='darkgrid', palette='deep') # Using Seaborn for making heatmap
                cmap="YlGnBu"
            else:
                sns.set(style = "white")
                cmap = sns.diverging_palette(220, 10, as_cmap=True)

            # Generate a mask for the upper triangle
            mask = np.zeros_like(corrmat, dtype=np.bool)
            mask[np.triu_indices_from(mask)] = True

            # Draw the heatmap with the mask and correct aspect ratio
            plt.figure(figsize=(10, 10))
            hm = sns.heatmap(corrmat, mask=mask, cbar=True, annot=True, square=True, fmt='.2f'
                             annot_kws={'size': 10}, cmap=cmap)
            hm.set_title(title)
            plt.yticks(rotation=0)
            plt.show()
```

---

1 Activity Analysis

1.1 1. Activity summary - Steps, Calories and Floor counts

```
In [9]: data[['Calories Burned', 'Steps', 'Minutes Sedentary', 'Minutes Fairly Active', 'Minute
```

```
Out[9]:                        count         mean          std    min      25%  \
        Calories Burned        116.0  2724.560345   653.985218  522.0  2521.75
        Steps                  116.0  8316.163793  4398.676997    0.0  6295.75
        Minutes Sedentary      116.0   832.810345   309.649982    0.0   682.75
        Minutes Fairly Active  116.0    24.681034    20.114714    0.0     4.75
        Minutes Very Active    116.0    37.258621    35.085268    0.0     2.00
        Cardio minutes         103.0     7.883495    10.061868    0.0     0.00
        Fat Burn minutes       103.0   143.631068    91.585917    0.0    74.00
        Resting Heart Rate     101.0    61.267327     3.036087   55.0    59.00

                                  50%       75%      max
```

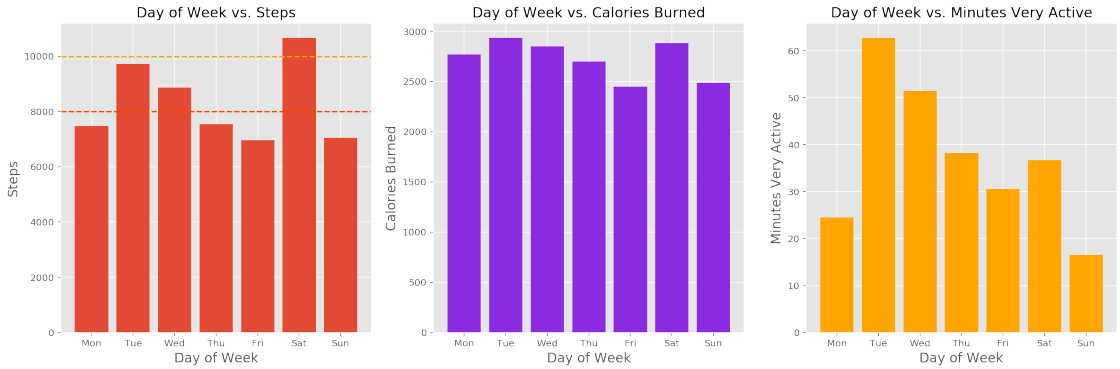
```
Calories Burned        2877.5   3116.75   4085.0
Steps                  8630.0  11275.00  25570.0
Minutes Sedentary       745.5   1049.25   1440.0
Minutes Fairly Active    23.0     38.25     78.0
Minutes Very Active      30.5     64.00    135.0
Cardio minutes            4.0     13.00     49.0
Fat Burn minutes        144.0    200.50    517.0
Resting Heart Rate       62.0     63.00     68.0
```

In [10]:
```
data[['Calories Burned', 'Steps', 'Minutes Sedentary', 'Minutes Fairly Active', 'Minut
```

Out[10]:
```
                        count         mean          std    min      25%  \
Calories Burned         116.0  2724.560345   653.985218  522.0  2521.75
Steps                   116.0  8316.163793  4398.676997    0.0  6295.75
Minutes Sedentary       116.0   832.810345   309.649982    0.0   682.75
Minutes Fairly Active   116.0    24.681034    20.114714    0.0     4.75
Minutes Very Active     116.0    37.258621    35.085268    0.0     2.00
Cardio minutes          103.0     7.883495    10.061868    0.0     0.00
Fat Burn minutes        103.0   143.631068    91.585917    0.0    74.00
Resting Heart Rate      101.0    61.267327     3.036087   55.0    59.00

                           50%       75%      max
Calories Burned         2877.5   3116.75   4085.0
Steps                   8630.0  11275.00  25570.0
Minutes Sedentary        745.5   1049.25   1440.0
Minutes Fairly Active     23.0     38.25     78.0
Minutes Very Active       30.5     64.00    135.0
Cardio minutes             4.0     13.00     49.0
Fat Burn minutes         144.0    200.50    517.0
Resting Heart Rate        62.0     63.00     68.0
```

In [11]:
```
fig = plt.figure(figsize = (20,6))

ax = plt.subplot(131)
plt.bar(dayGroupedData.index, dayGroupedData['Steps'])
plt.title('Day of Week vs. Steps', fontsize=15)
plt.xlabel('Day of Week', fontsize=14)
plt.ylabel('Steps', fontsize=14)
ax.axhline(8000, color="orangered", linestyle='--')
ax.axhline(10000, color="orange", linestyle='--')
ax.set_xticklabels(minDayCodes)

#############

ax2 = fig.add_subplot(132)
plt.bar(dayGroupedData.index, dayGroupedData['Calories Burned'], color='blueviolet')
plt.title('Day of Week vs. Calories Burned', fontsize=15)
plt.xlabel('Day of Week', fontsize=14)
```

```
plt.ylabel('Calories Burned', fontsize=14)
ax2.set_xticklabels(minDayCodes)

#############

ax3 = fig.add_subplot(133)
ax3.set_xticklabels(minDayCodes)
plt.bar(dayGroupedData.index, dayGroupedData['Minutes Very Active'], color='orange')
plt.title('Day of Week vs. Minutes Very Active', fontsize=15)
plt.xlabel('Day of Week', fontsize=14)
plt.ylabel('Minutes Very Active', fontsize=14)
plt.show()
```

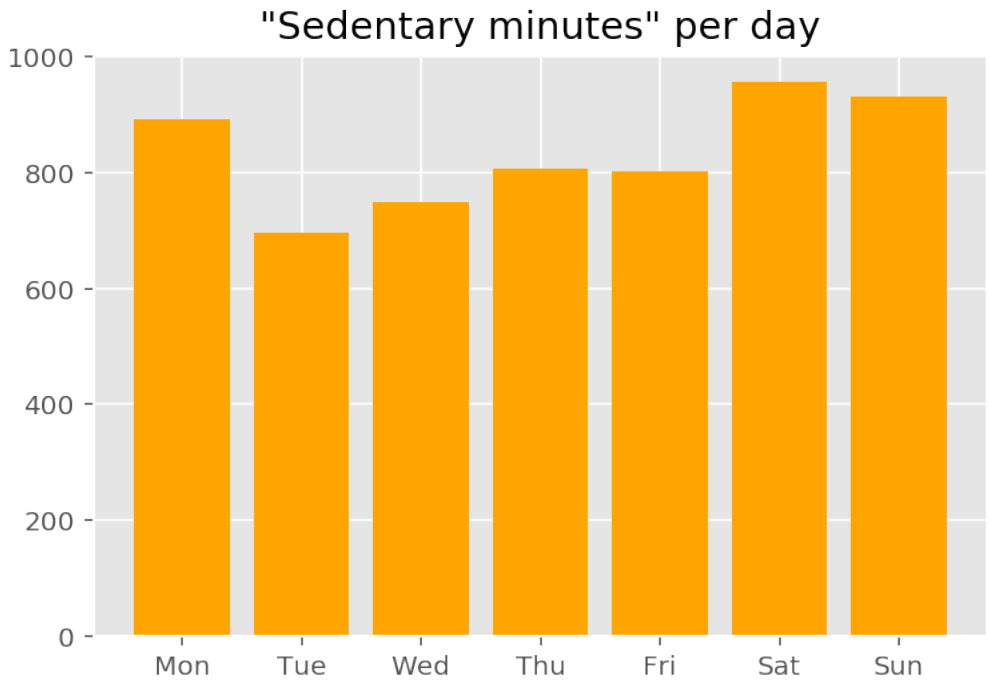


Not fussing more on the calories burned, I have kept a goal of attaining atleast 8000 steps day for my device.

From the graphs above, I am well in the range of steps>7500, so that's a good sign. There are some studies which suggest hitting 10000 steps per day.

1.2 2. Sedentary minutes

```
In [12]: plt.bar(dayGroupedData.index, dayGroupedData['Minutes Sedentary'], color='orange', ti
         plt.title('"Sedentary minutes" per day')

Out[12]: Text(0.5,1,'"Sedentary minutes" per day')
```

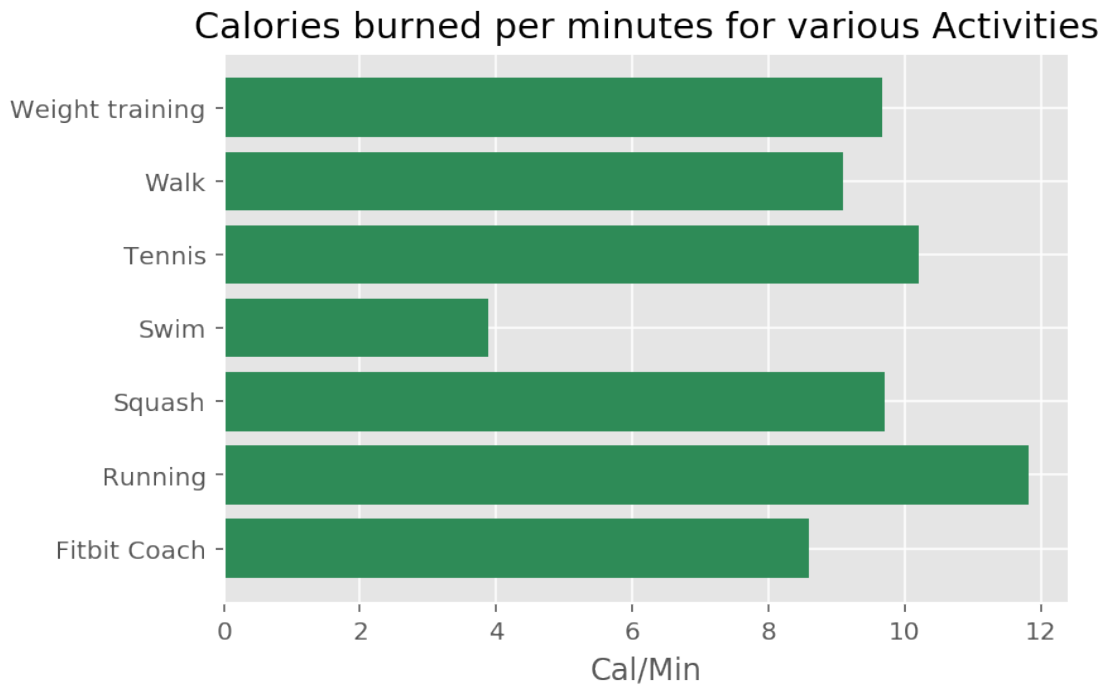


1.3 Average heart rate / calorie burn rate per min / vs exercise type

```
In [13]: activityDf = pd.read_excel(filename, sheet_name="activities")
         #skateboard,badminton, body weight ex, cyclinng
         activityDf['Cal/Min'] = activityDf['Calories burned']/activityDf['Time']
         groupedActivitiesDf = activityDf.groupby(['Activity Type']).mean()

         #Plot Data
         plt.barh(groupedActivitiesDf.index, groupedActivitiesDf['Cal/Min'], color='seagreen')
         plt.title('Calories burned per minutes for various Activities')
         plt.xlabel('Cal/Min')
         plt.plot()

         groupedActivitiesDf
```

Out[13]:

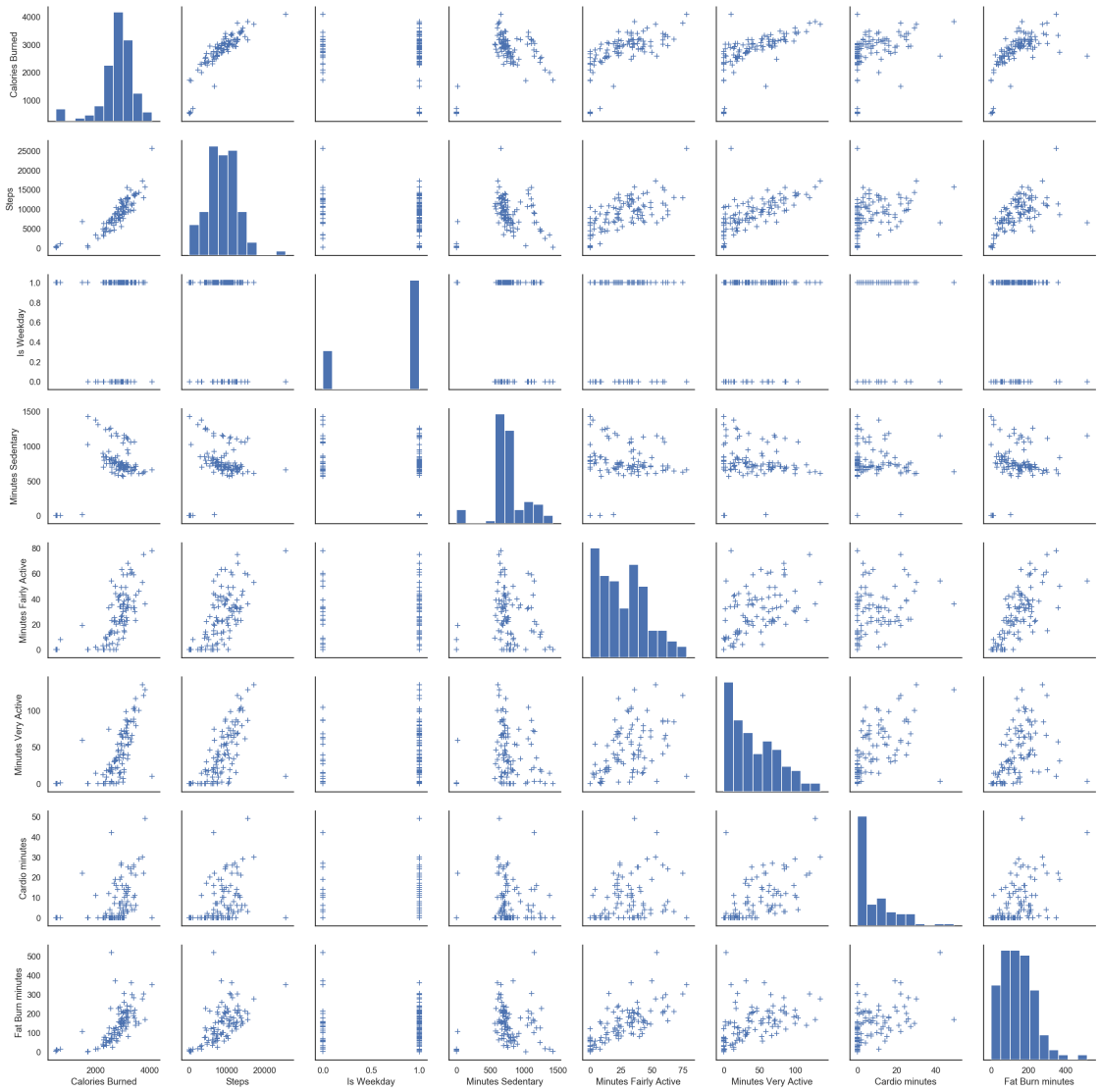
Activity Type	Time	avgBPM	maxBPM	Calories burned	Steps	Cal/Min
Fitbit Coach	10.000000	131.0	149.0	86.000000	5.0	8.600000
Running	9.000000	149.0	183.0	107.000000	1202.0	11.825000
Squash	44.000000	139.0	170.0	427.000000	4366.0	9.704545
Swim	20.333333	0.0	0.0	77.666667	97.0	3.876400
Tennis	35.000000	144.0	176.0	357.000000	2507.0	10.200000
Walk	20.000000	121.0	164.0	182.000000	2239.0	9.100000
Weight training	21.000000	136.0	166.0	203.000000	100.0	9.666667



1.4 3. Calorie burn coorelation

```
In [14]: correlationOf = 'Calories Burned'
         corrdf_calories = data[['Calories Burned','Steps', 'Is Weekday', 'Minutes Sedentary',
         plot_heatmap(corrdf_calories.corr(), correlationOf, '')
```

In [15]:
```
# Basic correlogram
sns.pairplot(corrdf_calories.dropna(), kind="scatter", markers="+", plot_kws=dict(s=5
plt.show()
```



---

2 Sleep Analysis

2.1 1. How regular is my sleeping habits?

- Am I getting the required hours of sleep? - Average sleep hours and the deviation
- Am I following a good sleep schedule? - Average sleep and wake up timings

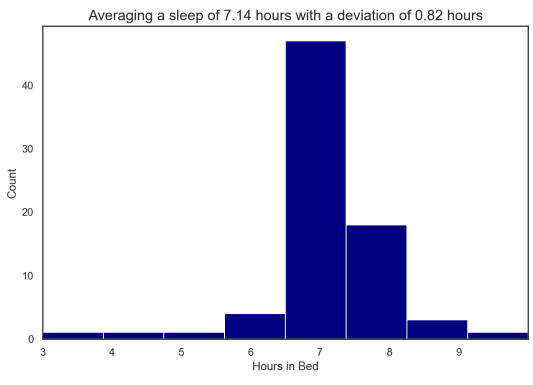
```
In [16]: import matplotlib.dates as mdates

         sleepDesc = pd.DataFrame(sleepData['Time in bed']/60).describe().transpose()
         avgSleepHours = round(sleepDesc.at['Time in bed','mean'],2)
```

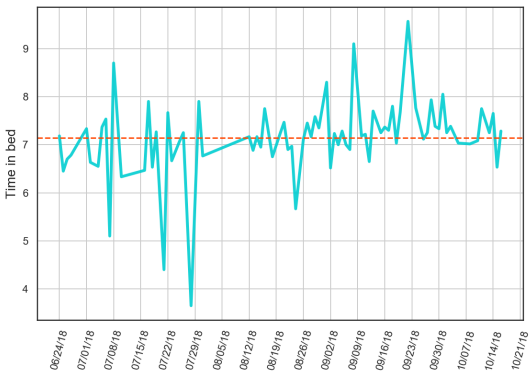
```
summary = 'Averaging a sleep of {} hours with a deviation of {} hours'.format(avgSlee
hoursInBed = sleepData['Time in bed']/60

fig = plt.figure(figsize = (20,6))

ax = plt.subplot(121)
plt.hist(hoursInBed, bins = 8, range = (3, 10), color="navy")
plt.xlim(3, 10)
plt.xticks(range(3, 10))
plt.xlabel('Hours in Bed')
plt.ylabel('Count');
plt.title(summary, fontsize=15)

#############

ax2 = fig.add_subplot(122)
plt.plot(sleepData['Date'],hoursInBed, linestyle='-',
         markersize=10, color='darkturquoise', label='% Light', linewidth=3.0, alpha=
plt.ylabel('Time in bed', fontsize=14)
ax2.axhline(avgSleepHours, color="orangered", linestyle='--')
ax2.xaxis.set_major_locator(mdates.WeekdayLocator(byweekday=6))
ax2.xaxis.set_major_formatter(mdates.DateFormatter('%D'))
ax2.grid(True)
plt.xticks(rotation=75)
plt.plot()

sleepDesc
```

Out[16]:

	count	mean	std	min	25%	50%	75%	\
Time in bed	76.0	7.14364	0.821759	3.65	6.895833	7.225	7.454167	

	max
Time in bed	9.566667



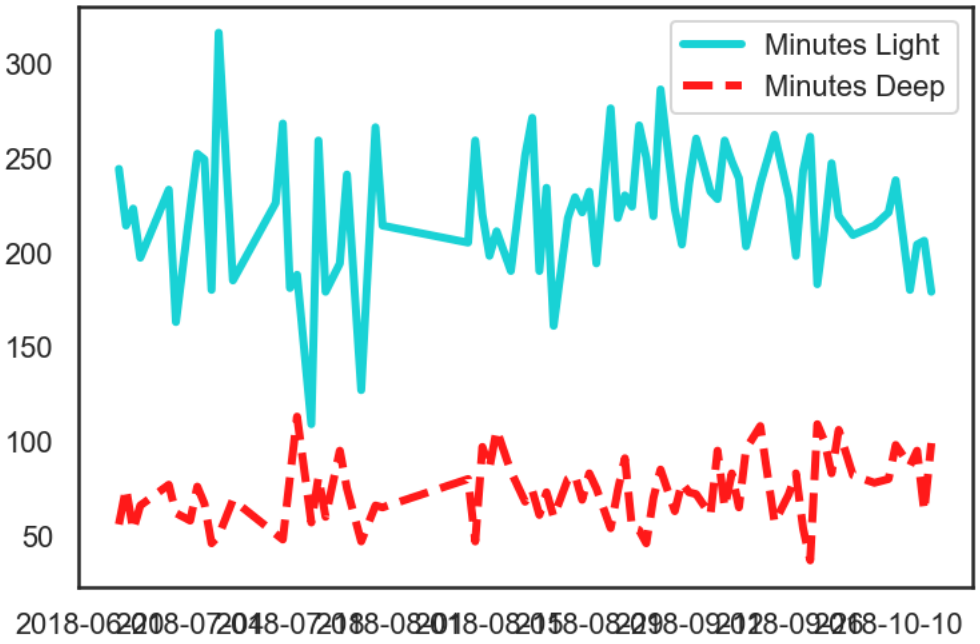


As far as behavioral data goes, this is reasonably well-behaved. Notice that this distribution doesn't vary much and is quite steep. The deviation is of less than an hour.

On the other note, according to National Sleep Foundation I should be getting sleep between 7-9 hours. Looks like barely scratching the mark here!

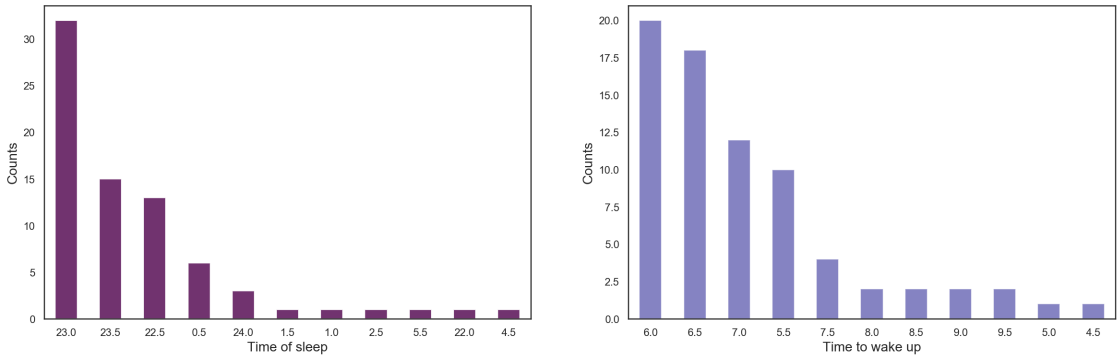
```
In [17]: plt.plot(sleepData['Date'],sleepData['Minutes Light sleep'], linestyle='-',
                  markersize=10, color='darkturquoise', label='Minutes Light', linewidth=3.0, a
         plt.plot(sleepData['Date'],sleepData['Minutes Deep sleep'], linestyle='--',
                  markersize=10, color='red', label='Minutes Deep', linewidth=3.0, alpha=0.9)
         plt.legend()
```

```
Out[17]: <matplotlib.legend.Legend at 0x258a343b710>
```



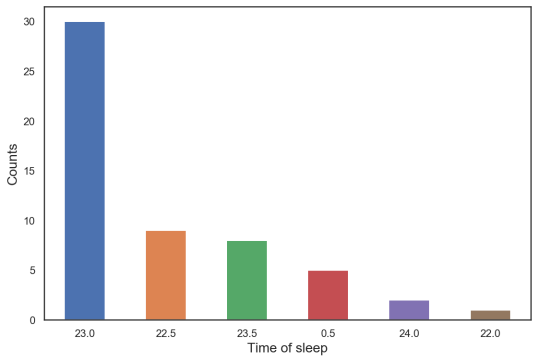

```
In [18]: sleepBDF = sleepData[['Sleep Bucket', 'Awake Bucket', 'Time in bed']]
         sleepBDF['Time in bed'] = sleepBDF['Time in bed']/60

         #sleepBDF.groupby(['Sleep Bucket']).mean()
         #sleepBDF.describe().transpose()

         ## plot the sleep and awake counts
         fig = plt.figure(figsize = (20,6))

         ax = plt.subplot(121)
         pd.value_counts(sleepData['Sleep Bucket']).plot.bar(cmap="BuPu_r", alpha=0.8)
         plt.xlabel('Time of sleep', fontsize=14)
         plt.ylabel('Counts', fontsize=14)
```

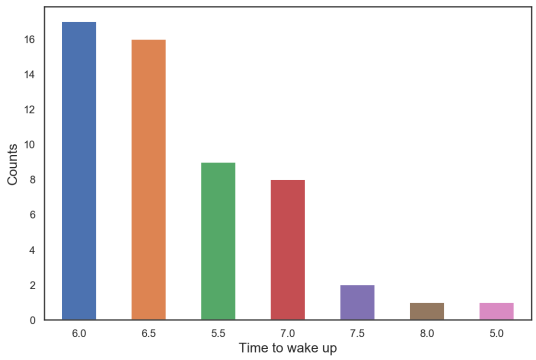
```
plt.xticks(rotation=0)

##############

ax2 = fig.add_subplot(122)
pd.value_counts(sleepData['Awake Bucket']).plot.bar(cmap="plasma", alpha=0.5)
plt.xlabel('Time to wake up', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.xticks(rotation=0)
plt.show()

#https://www.sleepfoundation.org/sleep-tools-tips/healthy-sleep-tips
```



In [19]:
```
sleepBDF_weekday = sleepData[['Sleep Bucket', 'Awake Bucket', 'Time in bed', 'Is Week
sleepBDF_weekday['Time in bed'] = sleepBDF_weekday['Time in bed']/60
sleepBDF_weekday = sleepBDF_weekday[sleepBDF_weekday['Is Weekday']]

#sleepBDF.groupby(['Sleep Bucket']).mean()
#sleepBDF.describe().transpose()

## plot the sleep and awake counts
fig = plt.figure(figsize = (20,6))

ax = plt.subplot(121)
pd.value_counts(sleepBDF_weekday['Sleep Bucket']).plot.bar()
plt.xlabel('Time of sleep', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.xticks(rotation=0)

##############

ax2 = fig.add_subplot(122)
pd.value_counts(sleepBDF_weekday['Awake Bucket']).plot.bar()
plt.xlabel('Time to wake up', fontsize=14)
```

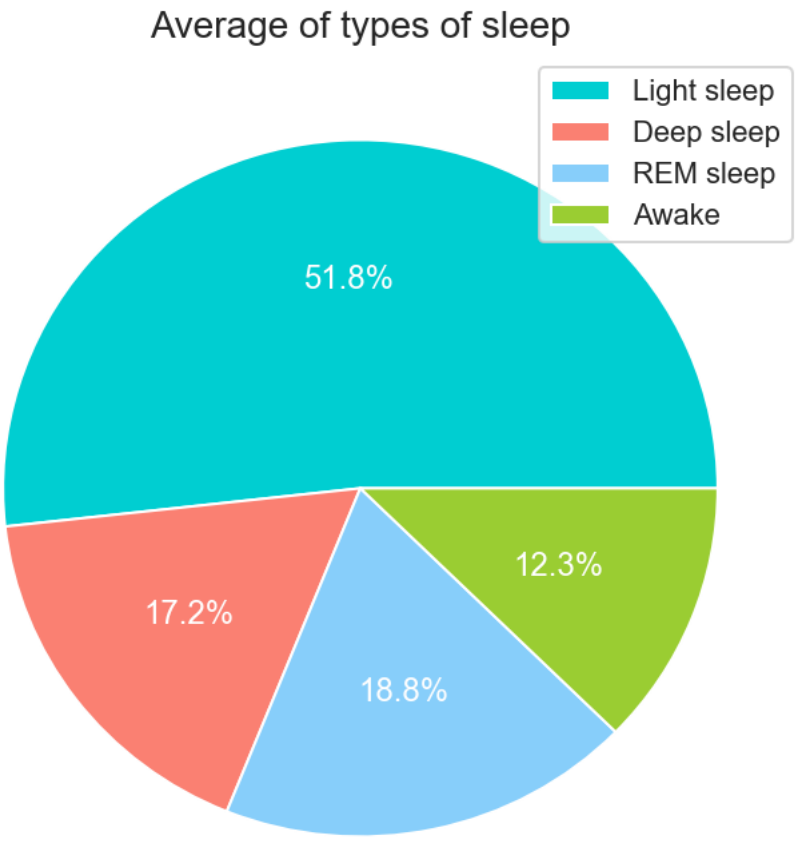
```
plt.ylabel('Counts', fontsize=14)
plt.xticks(rotation=0)
plt.show()

#https://www.sleepfoundation.org/sleep-tools-tips/healthy-sleep-tips
```





2.2 2. Types of sleep

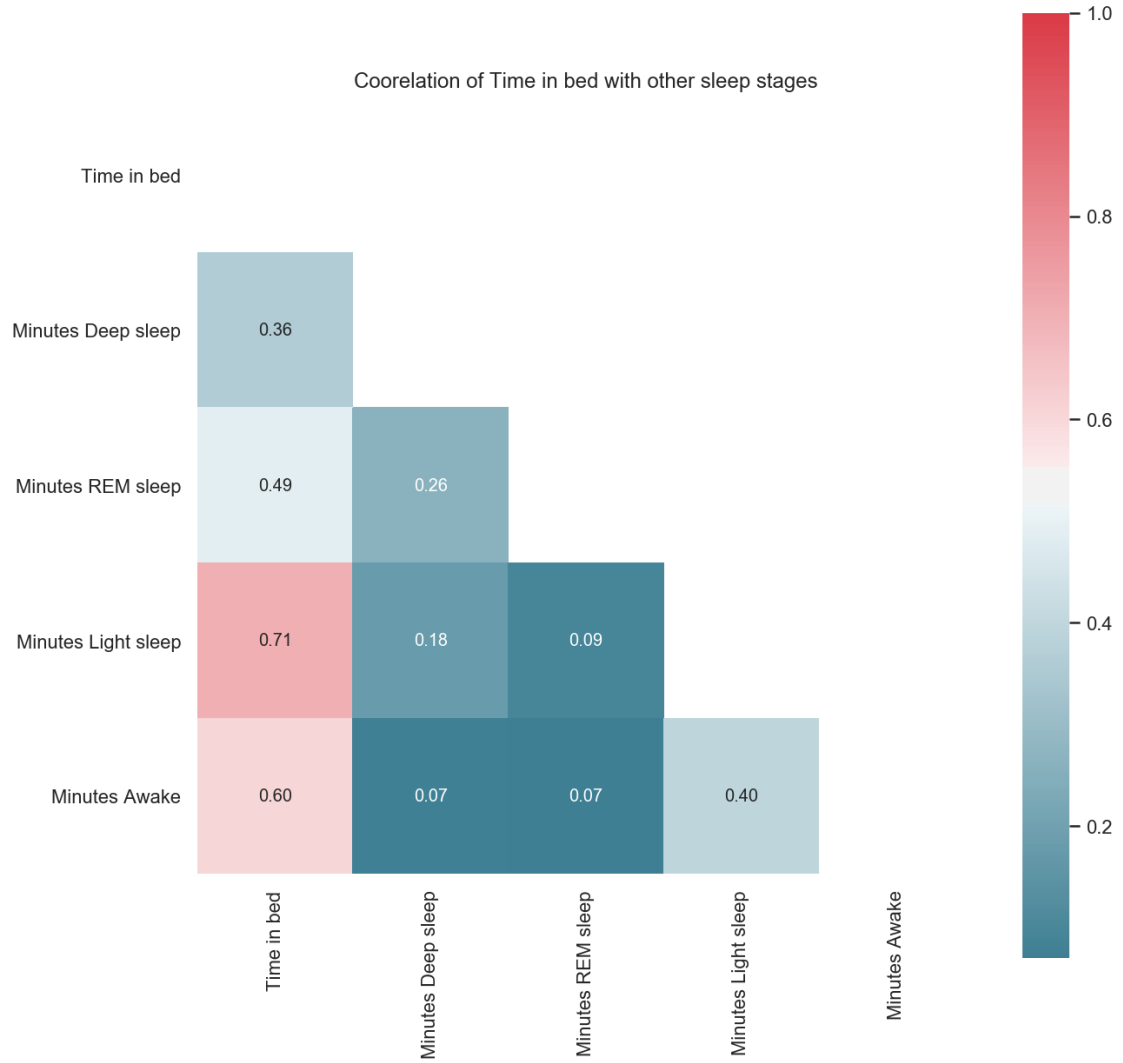
In [20]:
```
avgSleep = sleepData[['Minutes Light sleep', 'Minutes Deep sleep', 'Minutes REM sleep

fig = plt.figure(figsize = (6,6))
labels=['Light sleep', 'Deep sleep', 'REM sleep', 'Awake']
plt.pie(avgSleep, colors = ['darkturquoise', 'salmon', 'lightskyblue', 'yellowgreen']

# #carve the donut
# my_circle=plt.Circle( (0,0), 0.7, color='white')
# p=plt.gcf()
# p.gca().add_artist(my_circle)

plt.title('Average of types of sleep', fontsize=14)
plt.legend()
plt.show()
```



2.3 3. Correlation between amount of sleep and the sleep stages.

Do sleeping more will help me attain more deep or REM sleep?

```
In [21]: corrdf_sleep_types = sleepData[['Time in bed', 'Minutes Deep sleep', 'Minutes REM slee
         plot_heatmap(corrdf_sleep_types, correlationOf, 'Coorelation of Time in bed with other
         corrdf_sleep_types
```




```
Out[21]:                    Time in bed  Minutes Deep sleep  Minutes REM sleep  \
         Time in bed           1.000000            0.363047           0.485731
         Minutes Deep sleep    0.363047            1.000000           0.262515
         Minutes REM sleep     0.485731            0.262515           1.000000
         Minutes Light sleep   0.705513            0.178520           0.092427
         Minutes Awake         0.604619            0.074543           0.070540

                             Minutes Light sleep  Minutes Awake
         Time in bed                    0.705513       0.604619
         Minutes Deep sleep             0.178520       0.074543
         Minutes REM sleep              0.092427       0.070540
         Minutes Light sleep            1.000000       0.395523
         Minutes Awake                  0.395523       1.000000
```

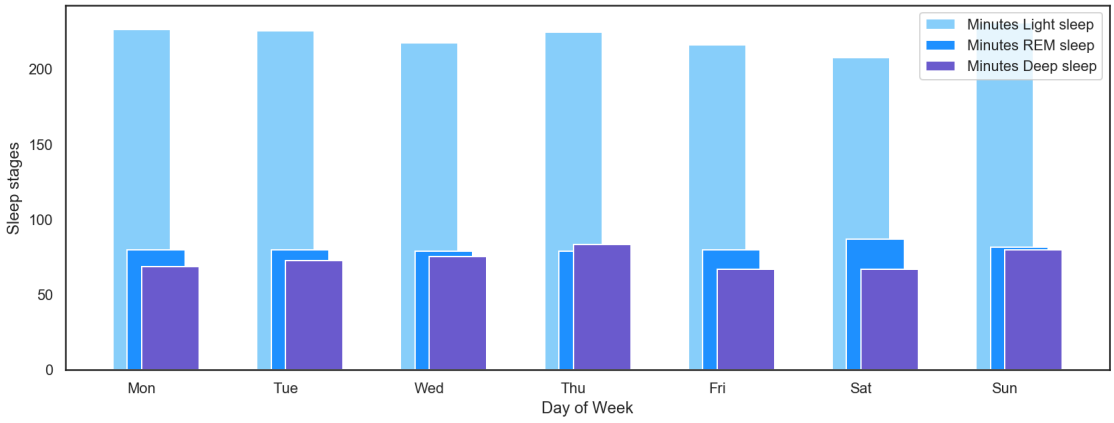
Notice that the "Deep sleep minutes" are not very coorelated with the time in bed. Which

shows that sleepinng more doesn't necessarily guarantee a good deep sleep.

2.4 4. Types of Sleep based on different days

```
In [22]: fig = plt.figure(figsize = (14,5))
         plt.bar((dayGroupedData.index), dayGroupedData['Minutes Light sleep'],width = 0.4, col
         plt.bar((dayGroupedData.index + 0.1), dayGroupedData['Minutes REM sleep'], width = 0.4
         plt.bar((dayGroupedData.index + 0.2), dayGroupedData['Minutes Deep sleep'], width = 0
         plt.xlabel('Day of Week')
         plt.ylabel('Sleep stages')
         plt.legend()

         print("Likely to get a more Restorative sleep on "+ str(getDayLabel(dayGroupedData['%
```

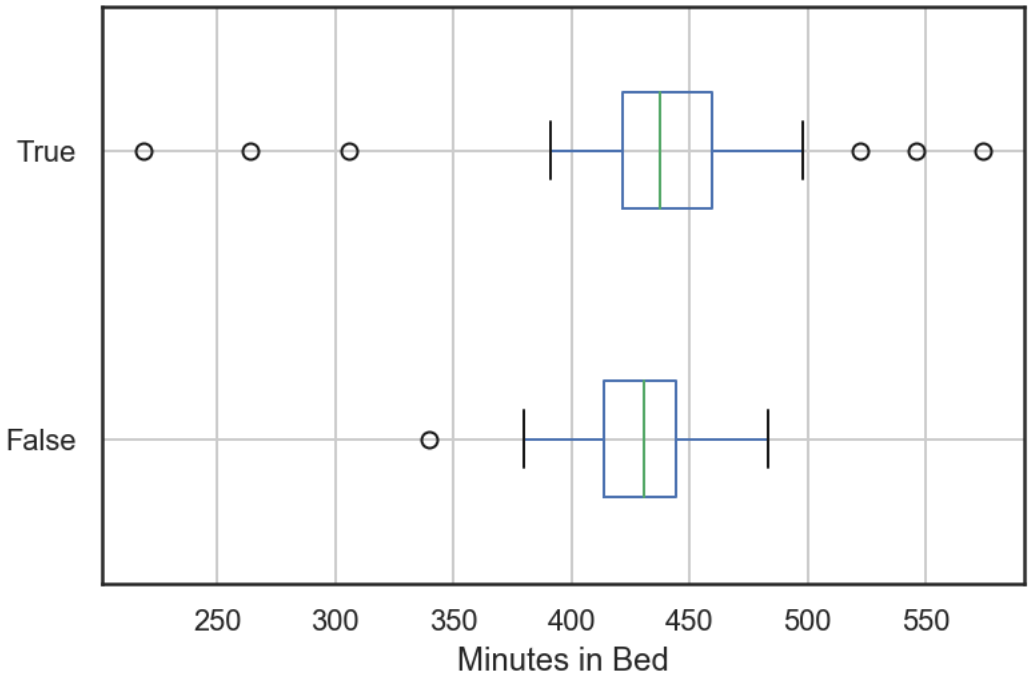
```
Likely to get a more Restorative sleep on Saturday
```



Let's now look at the different days of the week. Did I sleep more on weekends? (I certainly hope so.) What nights were the worst?

2.5 Effect of Sleep on Weekdays vs Weekends

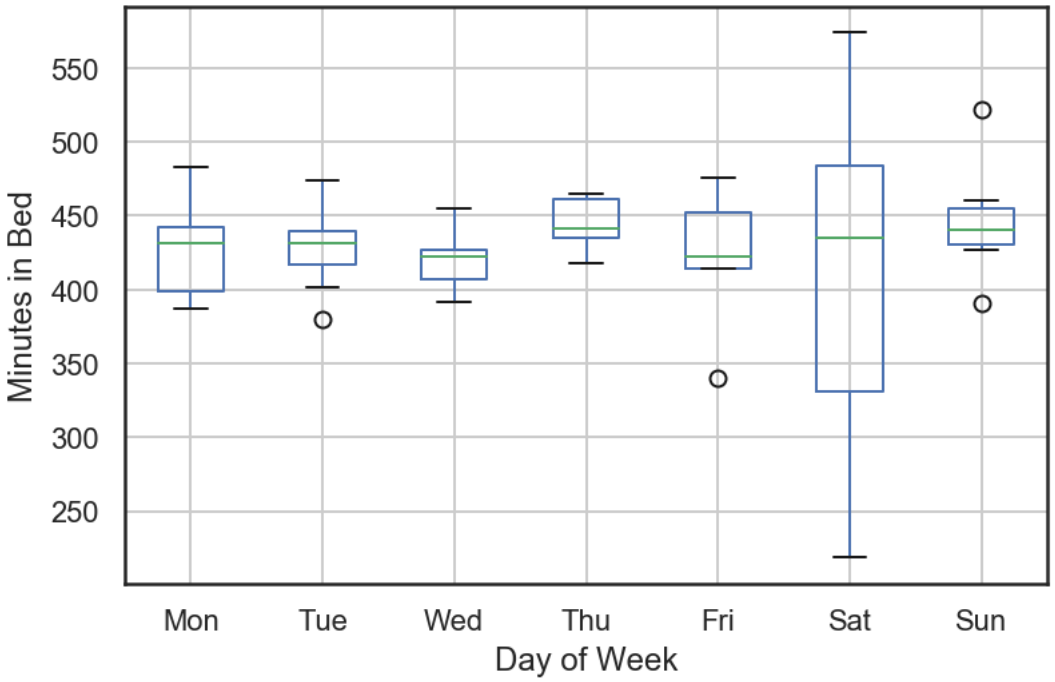
```
In [23]: ax = sleepData.boxplot(column = 'Time in bed', by = 'Is Weekend', vert = False, widths
         plt.xlabel('Minutes in Bed')
         plt.suptitle('')
         plt.title('');
```



The above plot shows that I tend to sleep a bit more on Weekends. The upper whisker is quite high for weekends indicatign varying sleep times.

let's check out how does the plots vary for indivisual days of the week.

```
In [24]: ax = sleepData.boxplot(column = 'Time in bed', by = 'Day of Week')
         ax.set_xticklabels(minDayCodes[1:])
         plt.ylabel('Minutes in Bed')
         plt.suptitle('')
         plt.title('');
```

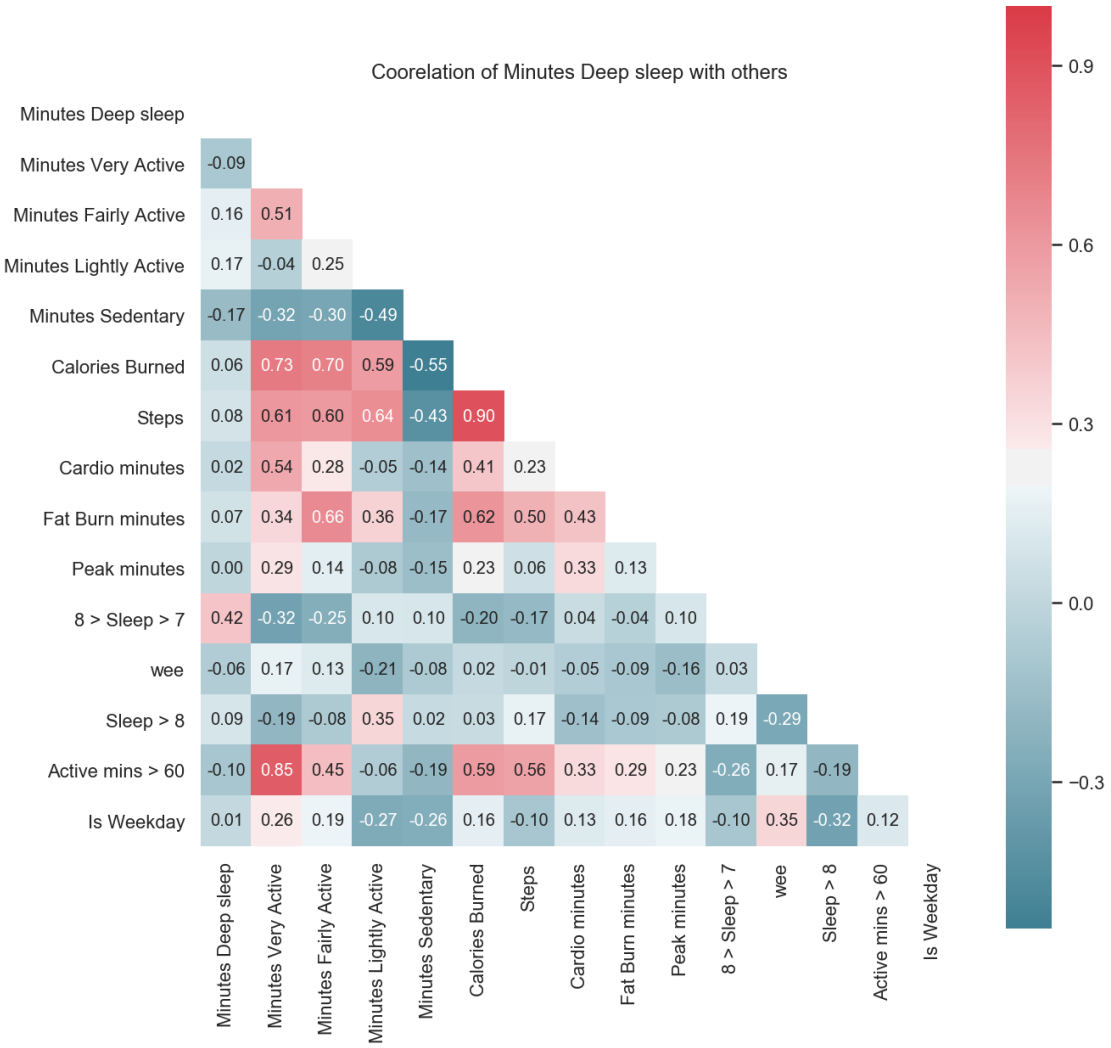


This is pretty interesting. The rest of the days are straighforward with lesser deviations but from the looks of it, I have exploited saturday night sleep for sure!

2.6 Average Sleep and Wake times

```
In [25]: sleepData['8 > Sleep > 7'] = sleepData['Time in bed'] > 7*60
         sleepData['Sleep > 7'] = sleepData['Time in bed'] > 7*60
         sleepData['Sleep > 8'] = sleepData['Time in bed'] > 8*60
         sleepData['Active mins > 50'] = sleepData['Minutes Very Active'] > 50
         sleepData['Active mins > 60'] = sleepData['Minutes Very Active'] > 60

         sleepData['wee'] = np.logical_and(sleepData['Sleep Bucket'] <= 23, sleepData['Awake Bu
         # slept before 11 and woke up by 6:30
         #sleepData
```

```
In [26]: correlationOf="Minutes Deep sleep"
         k = 15 #number of variables for heatmap
         corrmat = sleepData[['Minutes Deep sleep', 'Minutes Very Active', 'Minutes Fairly Acti
                   'Cardio minutes', 'Fat Burn minutes', 'Peak minutes',
                   '8 > Sleep > 7', 'wee', 'Sleep > 8', 'Active mins > 60', 'Is Weekday']].corr
         #corrmat = sleepData.drop(['% Restorative sleep', 'Minutes Light sleep', 'Minutes REM
         #cols = corrmat.nlargest(k, correlationOf)[correlationOf].index
         #corrdf_sleep = sleepData[cols]

         plot_heatmap(corrmat, correlationOf, 'Coorelation of {} with others'.format(correlatio
```



2.7 Machine Learning

Since we have some insights now on the activities and sleep. Let's see if we could use some of the basic ML techniques to see if there is a pattern to predict what are the ingredients for a decent sleep!

Though we have a handfulof data and vely less features, lets see what we can prdict.

```
In [27]: sleepData.head()
```

```
Out[27]:                  Date  Day of Week  Is Weekday  Is Weekend  Calories Burned  \
         Date
         2018-06-24 2018-06-24            7       False        True             1996
         2018-06-25 2018-06-25            1        True       False             2863
         2018-06-26 2018-06-26            2        True       False             3398
         2018-06-27 2018-06-27            3        True       False             3442
```

```
2018-07-01 2018-07-01            7       False        True             1718

           Calories BMR  Steps  Distance (Km)  Elevation (Ft)  \
Date
2018-06-24         1690   3367           3.01            0.00
2018-06-25         1685   8394           5.55           36.58
2018-06-26         1683  13569           9.68           24.38
2018-06-27         1682  13337           9.33           33.53
2018-07-01         1681    155           0.09            0.00

           Resting Heart Rate  Floors  Minutes Sedentary  \
Date
2018-06-24               59.0       0               1377
2018-06-25               58.0      12                709
2018-06-26               57.0       8                687
2018-06-27               57.0      11                665
2018-07-01               58.0       0               1427

           Minutes Lightly Active  Minutes Fairly Active  \
Date
2018-06-24                     47                     2
2018-06-25                    244                    42
2018-06-26                    231                    33
2018-06-27                    243                    30
2018-07-01                     13                     0

           Minutes Very Active  Activity Calories  Active Score  \
Date
2018-06-24                  14                362            -1
2018-06-25                  14               1361            -1
2018-06-26                 102               2004            -1
2018-06-27                 100               2035            -1
2018-07-01                   0                 46            -1

           Cardio minutes  Cardio calories  Fat Burn minutes  \
Date
2018-06-24           11.0        133.10892              14.0
2018-06-25            2.0         12.40518              94.0
2018-06-26           10.0        103.59112             187.0
2018-06-27           11.0        114.28908             217.0
2018-07-01            0.0          0.00000               1.0

           Fat Burn calories  Peak minutes  Peak calories  \
Date
2018-06-24          89.20880           0.0            0.0
2018-06-25         514.11279           0.0            0.0
2018-06-26        1245.54876           0.0            0.0
2018-06-27        1310.46804           0.0            0.0
```

```
2018-07-01           3.97018          0.0           0.0

            Normal Cardio calories  Normal Cardio minutes  Sleep Efficiency  \
Date
2018-06-24              446.51352                  284.0              92.0
2018-06-25             2311.92765                 1326.0              95.0
2018-06-26             2034.05724                 1232.0              89.0
2018-06-27             1986.03570                 1187.0              97.0
2018-07-01              233.77354                  171.0              96.0

            Minutes Asleep  Minutes to fall asleep         Sleep Start time  \
Date
2018-06-24           379.0                     0.0  2018-06-24T22:17:00.000
2018-06-25           333.0                     0.0  2018-06-25T22:49:30.000
2018-06-26           351.0                     0.0  2018-06-26T22:25:30.000
2018-06-27           365.0                     0.0  2018-06-27T22:36:30.000
2018-07-01           402.0                     0.0  2018-07-01T22:34:00.000

                    Sleep End time  Time in bed  Minutes Deep sleep  \
Date
2018-06-24  2018-06-25T05:28:30.000        431.0                56.0
2018-06-25  2018-06-26T05:16:30.000        387.0                75.0
2018-06-26  2018-06-27T05:07:30.000        402.0                53.0
2018-06-27  2018-06-28T05:24:00.000        407.0                66.0
2018-07-01  2018-07-02T05:54:30.000        440.0                77.0

            Deep sleep count  Minutes Light sleep  Light sleep count  \
Date
2018-06-24               2.0                244.0               26.0
2018-06-25               3.0                214.0               27.0
2018-06-26               3.0                223.0               28.0
2018-06-27               4.0                197.0               32.0
2018-07-01               5.0                233.0               34.0

            Minutes REM sleep  REM sleep count  Minutes Awake  \
Date
2018-06-24               79.0              6.0           52.0
2018-06-25               44.0              7.0           54.0
2018-06-26               75.0              7.0           51.0
2018-06-27              102.0             10.0           42.0
2018-07-01               92.0              8.0           38.0

            Minutes Awake count  % Deep sleep  % Light sleep  % REM sleep  \
Date
2018-06-24                 25.0          13.0           57.0         19.0
2018-06-25                 30.0          20.0           56.0         12.0
2018-06-26                 29.0          14.0           56.0         19.0
2018-06-27                 32.0          17.0           49.0         26.0
```

```
2018-07-01                34.0          18.0           53.0         21.0

           Day Label  Active exercise  Sleep Bucket  Awake Bucket  % Awake  \
Date
2018-06-24       Sun            False          22.5           5.5     11.0
2018-06-25       Mon            False          23.0           5.5     12.0
2018-06-26       Tue             True          22.5           5.5     11.0
2018-06-27       Wed             True          23.0           5.5      8.0
2018-07-01       Sun            False          23.0           6.0      8.0

            % Restorative sleep  Restorative sleep mins  8 > Sleep > 7  \
Date
2018-06-24                 32.0                   135.0           True
2018-06-25                 32.0                   119.0          False
2018-06-26                 33.0                   128.0          False
2018-06-27                 43.0                   168.0          False
2018-07-01                 39.0                   169.0           True

            Sleep > 7  Sleep > 8  Active mins > 50  Active mins > 60   wee
Date
2018-06-24       True      False             False             False  True
2018-06-25      False      False             False             False  True
2018-06-26      False      False              True              True  True
2018-06-27      False      False              True              True  True
2018-07-01       True      False             False             False  True
```

2.8 Regression

```
In [28]: from sklearn.model_selection import train_test_split
```

Let's build our training and test dataset

```
In [30]: y = sleepData['Restorative sleep mins']
         X = sleepData[['Day of Week', 'Calories Burned', 'Steps', 'Elevation (Ft)', 'Floors',
         X.fillna(X.mean(), inplace=True)

         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state
         print('X shape: {}'.format(X.shape))
         print('Y shape: {}'.format(Y.shape))
```

```
X shape: (76, 12)
Y shape: (76,)
```

2.8.1 Linear Regression

```
In [31]: from sklearn import linear_model
         from sklearn.metrics import mean_squared_error, r2_score
```

```
        # Create linear regression object
        regr = linear_model.LinearRegression()

        # Train the model using the training sets
        regr.fit(X_train, y_train)

        # Make predictions using the testing set
        lr_y_pred = regr.predict(X_test)

        print("Mean squared error: %.2f"% mean_squared_error(y_test, lr_y_pred))
        coefficients = pd.concat([pd.DataFrame(X.columns),pd.DataFrame(np.transpose(regr.coef
        coefficients
```

```
Mean squared error: 2186.12
```

Out[31]:

	0	0
0	Day of Week	2.645130
1	Calories Burned	-0.074257
2	Steps	-0.005408
3	Elevation (Ft)	320.089458
4	Floors	-975.794971
5	Minutes Sedentary	0.014429
6	Minutes Lightly Active	0.232111
7	Minutes Fairly Active	0.279348
8	Minutes Very Active	0.742517
9	Fat Burn minutes	0.186536
10	Active exercise	10.576876
11	Time in bed	0.310769

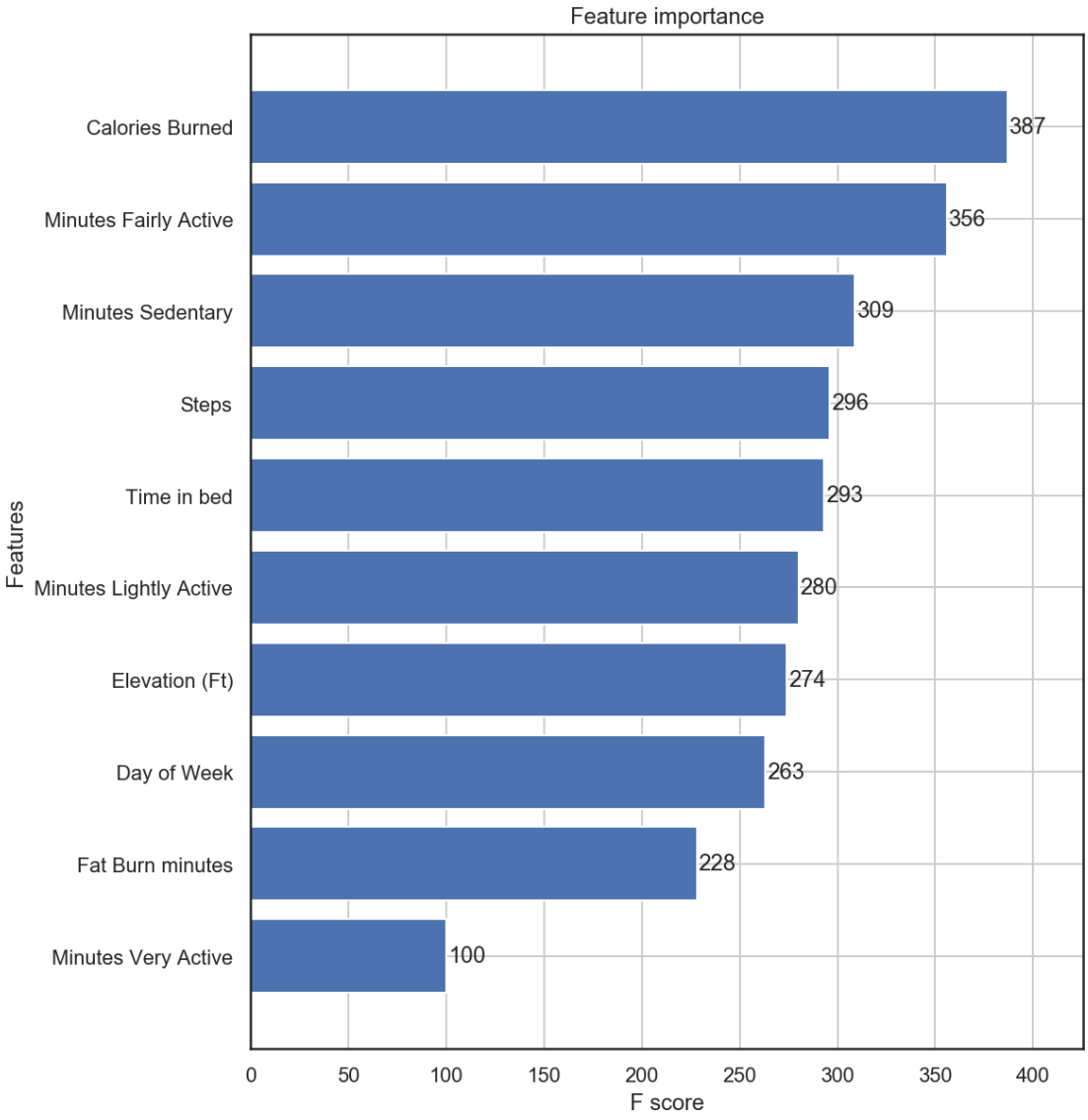
2.8.2 XGboost

In [32]:
```
        import xgboost as xgb
        from xgboost.sklearn import XGBRegressor
        from sklearn.model_selection import GridSearchCV

        xgbR = XGBRegressor()
        parameters = {'objective':['reg:linear'],
                      'learning_rate': [.03, 0.05, .07],
                      'max_depth': [3, 4, 5, 6],
                      'min_child_weight': [1, 5, 10],
                      'silent': [1],
                      'n_estimators': [500],
                      'seed': [1212]
                     }

        xgb_grid = GridSearchCV(xgbR, parameters, n_jobs = 5, verbose=True)
        xgb_grid.fit(X_train, y_train)
```

```
        print(xgb_grid.best_score_)
        print(xgb_grid.best_params_)

        y_pred = xgb_grid.predict(X_test)
        print ("RMSE : ",mean_squared_error(y_test,y_pred))
```

```
Fitting 3 folds for each of 36 candidates, totalling 108 fits


[Parallel(n_jobs=5)]: Using backend LokyBackend with 5 concurrent workers.
[Parallel(n_jobs=5)]: Done  40 tasks      | elapsed:    3.1s


-0.230838665936
{'learning_rate': 0.03, 'max_depth': 3, 'min_child_weight': 1, 'n_estimators': 500, 'objective
RMSE :  2224.62265833


[Parallel(n_jobs=5)]: Done 108 out of 108 | elapsed:    4.5s finished
C:\Anaconda\envs\YAO\lib\site-packages\sklearn\model_selection\_search.py:841: DeprecationWarni
  DeprecationWarning)
```

```
In [33]: fig, ax = plt.subplots(figsize=(8,10))
         xgb.plot_importance(xgb_grid.best_estimator_, height=0.8, ax=ax)
         plt.show()
         #y_pred = xgb_grid.predict(X_test)
```



2.8.3 Classification

```
In [34]: from sklearn.tree import DecisionTreeClassifier
         from sklearn import tree

         from sklearn.metrics import accuracy_score
         from sklearn.metrics import classification_report
         from sklearn.metrics import confusion_matrix

         import graphviz
```

```
        def trainAndPredict(classifier):
            classifier.fit(X_train, y_train)
            y_pred = classifier.predict(X_test)

            print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
            print("Report : ",classification_report(y_test, y_pred))
            #print("Confusion Matrix: ",confusion_matrix(y_test, y_pred))
            return

        def plotGraph(classifier):
            dot_data = tree.export_graphviz(classifier, out_file=None,
                                feature_names=X.columns.values,
                                class_names=['Yes', 'No'],
                                filled=True, rounded=True,
                                special_characters=True)
            graph = graphviz.Source(dot_data)
            return graph
```

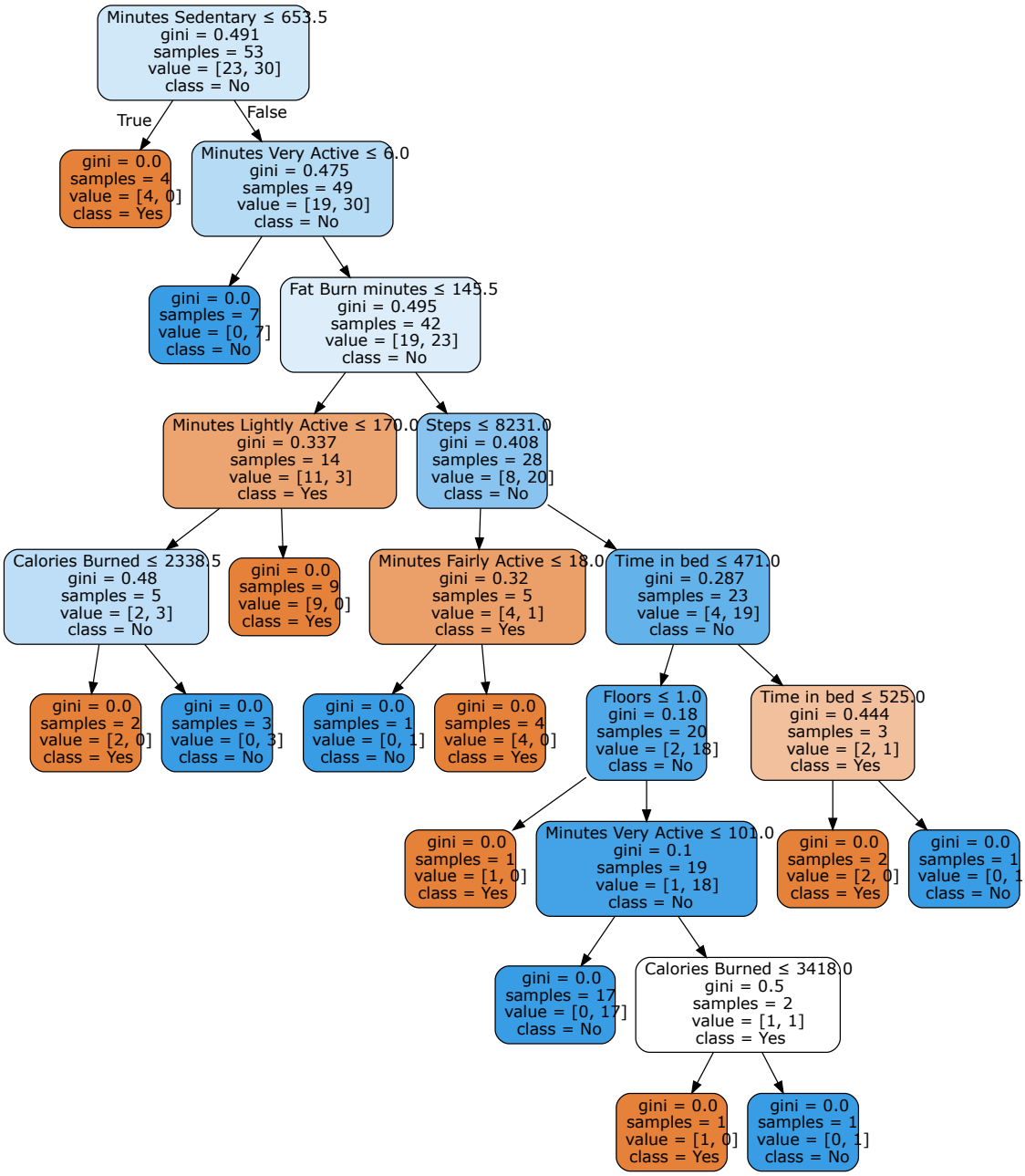
In [35]:
```
Y = sleepData['% Restorative sleep'] > 35

print('X shape: {}'.format(X.shape))
print('Y shape: {}'.format(Y.shape))

X_train, X_test, y_train, y_test = train_test_split( X, Y, test_size = 0.3, random_sta
print('X_train shape: {}. X_test shape: {}'.format(X_train.shape, X_test.shape))
```

```
X shape: (76, 12)
Y shape: (76,)
X_train shape: (53, 12). X_test shape: (23, 12)
```

In [36]:
```
clf_gini_default = DecisionTreeClassifier(criterion = "gini")
trainAndPredict(clf_gini_default)
plotGraph(clf_gini_default)
```

```
Accuracy :  30.4347826087
Report :                precision    recall  f1-score   support

       False       0.36      0.42      0.38        12
        True       0.22      0.18      0.20        11

   micro avg       0.30      0.30      0.30        23
   macro avg       0.29      0.30      0.29        23
weighted avg       0.29      0.30      0.30        23
```

Out[36]:


Minutes Sedentary ≤ 653.5
gini = 0.491
samples = 53
value = [23, 30]
class = No
True
False
gini = 0.0
samples = 4
value = [4, 0]
class = Yes
Minutes Very Active ≤ 6.0
gini = 0.475
samples = 49
value = [19, 30]
class = No
gini = 0.0
samples = 7
value = [0, 7]
class = No
Fat Burn minutes ≤ 145.5
gini = 0.495
samples = 42
value = [19, 23]
class = No
Minutes Lightly Active ≤ 170.0
gini = 0.337
samples = 14
value = [11, 3]
class = Yes
Steps ≤ 8231.0
gini = 0.408
samples = 28
value = [8, 20]
class = No
Calories Burned ≤ 2338.5
gini = 0.48
samples = 5
value = [2, 3]
class = No
gini = 0.0
samples = 9
value = [9, 0]
class = Yes
Minutes Fairly Active ≤ 18.0
gini = 0.32
samples = 5
value = [4, 1]
class = Yes
Time in bed ≤ 471.0
gini = 0.287
samples = 23
value = [4, 19]
class = No
gini = 0.0
samples = 2
value = [2, 0]
class = Yes
gini = 0.0
samples = 3
value = [0, 3]
class = No
gini = 0.0
samples = 1
value = [0, 1]
class = No
gini = 0.0
samples = 4
value = [4, 0]
class = Yes
Floors ≤ 1.0
gini = 0.18
samples = 20
value = [2, 18]
class = No
Time in bed ≤ 525.0
gini = 0.444
samples = 3
value = [2, 1]
class = Yes
gini = 0.0
samples = 1
value = [1, 0]
class = Yes
Minutes Very Active ≤ 101.0
gini = 0.1
samples = 19
value = [1, 18]
class = No
gini = 0.0
samples = 2
value = [2, 0]
class = Yes
gini = 0.0
samples = 1
value = [0, 1]
class = No
gini = 0.0
samples = 17
value = [0, 17]
class = No
Calories Burned ≤ 3418.0
gini = 0.5
samples = 2
value = [1, 1]
class = Yes
gini = 0.0
samples = 1
value = [1, 0]
class = Yes
gini = 0.0
samples = 1
value = [0, 1]
class = No