linear_regression_XYZ_MET

January 11, 2021

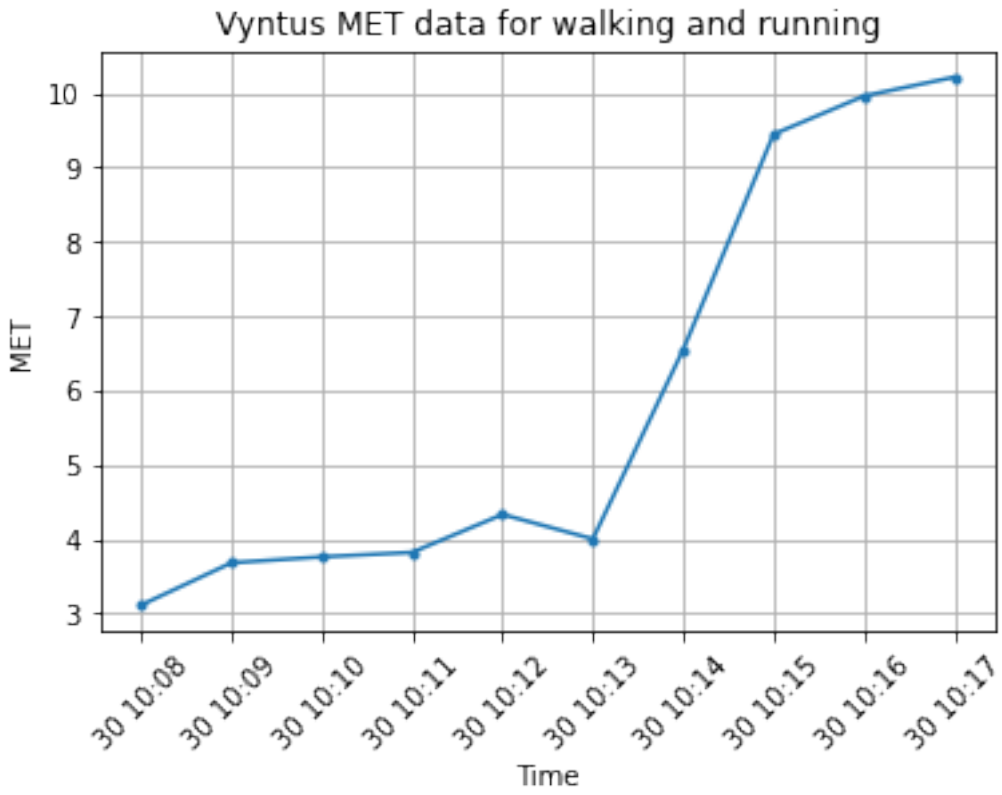
[1]:
```
from helpers import pandas_helper as pdh
from helpers import math_helper as mth
from sensors.activpal import *
from utils import read_functions
from scipy.stats import linregress
import math
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import datetime

activpal = Activpal()
```

[44]:
```
def get_regression_df(correspondent):
    start, stop = get_timestamps(correspondent)

    vyntus_df, min_index, max_index = get_vyntus_df(correspondent, start, stop)
    raw_df = get_raw_df(correspondent, min_index, max_index)

    new_df = pd.DataFrame(index=raw_df.index)
    new_df['mean_met'] = vyntus_df['met']
    new_df['sum_mag_acc'] = raw_df['mag_acc']

    return new_df

def get_timestamps(correspondent):
    activities_df = read_functions.read_activities(correspondent)
    start = activities_df.loc['lopen'].start
    stop = activities_df.loc['rennen'].stop

    return (start, stop)

def get_vyntus_df(correspondent, start, stop):
    vyntus_df = pdh.read_csv_vyntus(correspondent)
    mask = (vyntus_df.index >= start) & (vyntus_df.index < stop)
    vyntus_df = vyntus_df.loc[mask]
```

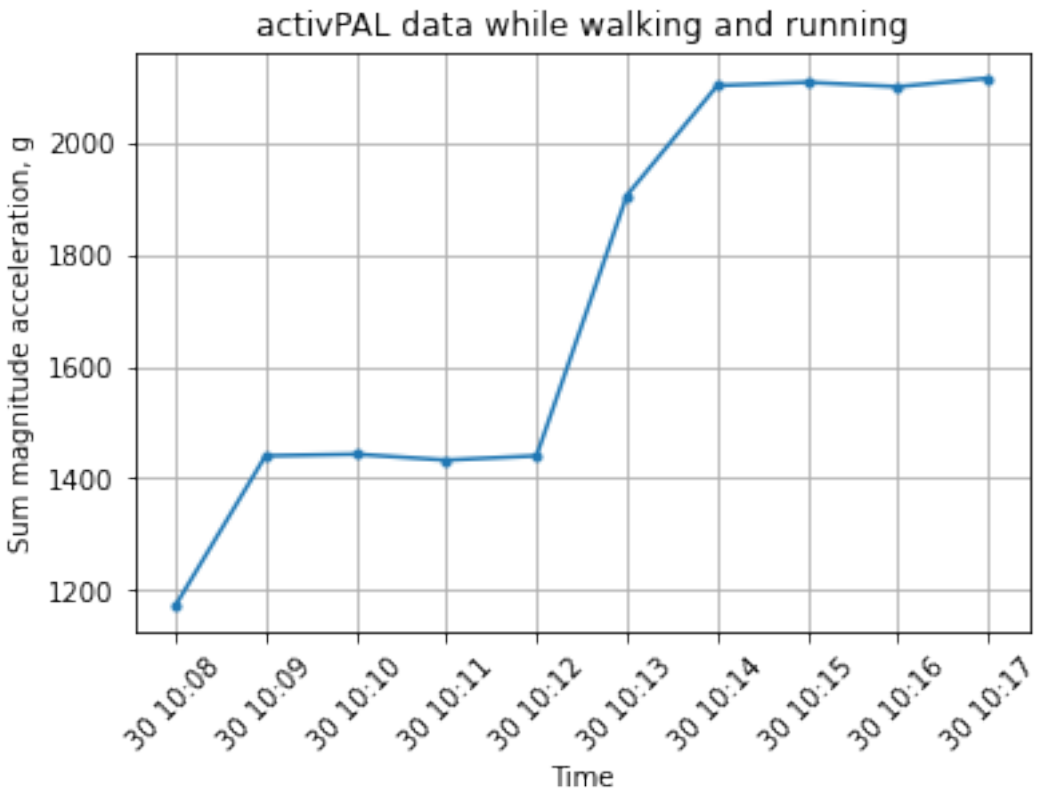
```
    min_index = vyntus_df.index.min()
    max_index = vyntus_df.index.max()

    respondents_df = pdh.read_csv_respondents()
    corr_number = int(correspondent.replace('BMR0', ''))
    weight = respondents_df['gewicht'][corr_number]

    vyntus_df['vyn_VO2'] = [float(vo2.replace(',', '.')) if type(vo2) == str␣
 ↪else vo2 for vo2 in vyntus_df['vyn_VO2']]
    vyntus_df['met'] = mth.calculate_met(vyntus_df['vyn_VO2'], weight)

    vyntus_df = vyntus_df.resample('60s').mean()[:-1]

    return vyntus_df, min_index, max_index

def get_raw_df(correspondent, start, stop):
    df = activpal.read_data(correspondent, start, stop)
    mask = (df.index >= start) & (df.index < stop)
    df = df.loc[mask]

    df = df[['pal_accX', 'pal_accY', 'pal_accZ']].apply(mth.convert_value_to_g)
    df['mag_acc'] = mth.to_mag_acceleration(df['pal_accX'], df['pal_accY'],␣
 ↪df['pal_accZ'])

    df = df.resample('60s').sum()[:-1]

    return df

def plot_met(met, title = 'Vyntus MET data for walking and running'):
    plt.title(title)
    plt.xlabel('Time')
    plt.xticks(rotation=45)
    plt.ylabel('MET')
    plt.plot(met, marker='.')
    plt.grid()

def plot_mag_acc(mag_acc, title = 'activPAL data while walking and running'):
    plt.title(title)
    plt.xlabel('Time')
    plt.xticks(rotation=45)
    plt.ylabel('Sum magnitude acceleration, g')
    plt.plot(mag_acc, marker='.')
    plt.grid()

def plot_lin_reg(x, y, title='Linear regression walking and running', xlabel =␣
 ↪'Sum magnitude acceleration, g', ylabel = 'MET'):
    linreg = linregress(x, y)
```

```
    fx = np.array([x.min(), x.max()])
    fy = linreg.intercept + linreg.slope * fx

    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.plot(x, y, 'o')
    plt.plot(fx, fy, '-')
    plt.grid()
    plt.show()
```

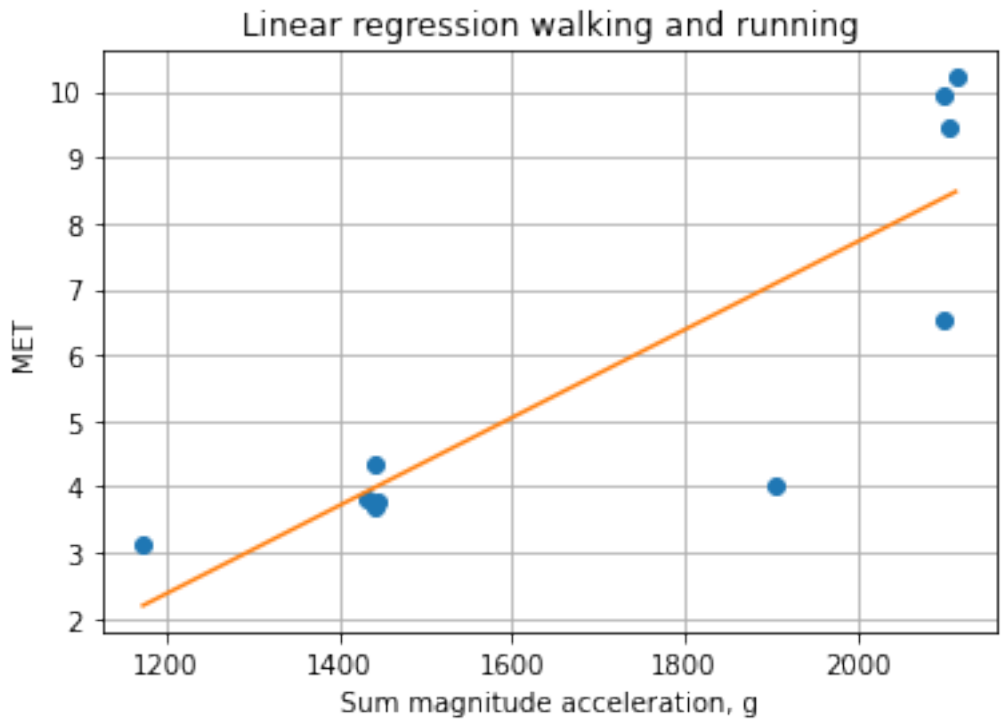
[39]:
```
corr30 = 'BMR030'
df30 = get_regression_df(corr30)
plot_met(df30['mean_met'])
```



[40]:
```
plot_mag_acc(df30['sum_mag_acc'])
```



[45]:
```
plot_lin_reg(df30['sum_mag_acc'], df30['mean_met'])
print('Pearson: ' + str(df30['sum_mag_acc'].corr(df30['mean_met'])))
```




```
Pearson: 0.8536325746177618
```

[42]:
```
#correspondents = ['BMR002', 'BMR004']
correspondents = ['BMR002', 'BMR004', 'BMR008', 'BMR012', 'BMR014', 'BMR015',␣
 ↪'BMR018', 'BMR030', 'BMR031', 'BMR032', 'BMR033', 'BMR034','BMR036',␣
 ↪'BMR040', 'BMR041', 'BMR042', 'BMR043', 'BMR044', 'BMR052', 'BMR053',␣
 ↪'BMR055', 'BMR058', 'BMR064', 'BMR097', 'BMR098', 'BMR099']

all_df = pd.DataFrame(index=pd.to_datetime([]))

for cor in correspondents:
    df = get_regression_df(cor)
    print(cor + ' - ' + str(df['sum_mag_acc'].corr(df['mean_met'])))
    all_df = pd.concat([all_df, df])

all_df.sort_index(inplace=True)
print(all_df['sum_mag_acc'].corr(all_df['mean_met']))
```

```
BMR002 - 0.741261072333766
BMR004 - 0.8179003485927083
BMR008 - 0.8953234039790643
BMR012 - 0.8130162815106681
BMR014 - 0.805273271826391
BMR015 - 0.7849473342234234
```

```
BMR018 - 0.6697739552417848
BMR030 - 0.8536325746177618
BMR031 - 0.9163778941815249
BMR032 - 0.7357734375912933
BMR033 - 0.5016810661831773
BMR034 - 0.8316776383794161
BMR036 - 0.8345204587637988
BMR040 - 0.8272072370732724
BMR041 - 0.7585043487469627
BMR042 - 0.7757419633281704
BMR043 - 0.8464378316244615
BMR044 - 0.7708898841000122
BMR052 - 0.7593112501548085
BMR053 - 0.7933490363189672
BMR055 - 0.9059407366169736
BMR058 - 0.871517588640081
BMR064 - 0.7164644988282575
BMR097 - 0.7013132380741743
BMR098 - 0.8971233953599903
BMR099 - 0.6568990694021114
0.7182204891264288
```

[ ]: