Oura Ring Semi-Continuous Sleep Data Extraction

May 19, 2021

1 Oura Ring Semi-Continuous Data Extraction

1.1 Author Information

Author: Sjors Weggeman
Student number: s4799771
University: Radboud University Nijmegen

1.2 Imports

[1]:
```
#Importing the necessary packages
import json
import matplotlib.pyplot as plt
import re
import numpy
```

1.3 Loading the data

[2]:
```
#Opening the file
file=open('oura_try.json')

#Loading the file into a workable dataframe
data=json.load(file)
```

1.4 Data Inspection

[3]:
```
#Inspecting the data by printing the first entry of every section (bold) of the data
for i in data:
    counter=-1
    print('\033[1m'+i+'\033[0m')
    for j in data[i]:
        counter+=1
        if counter==0:
            for k in j:
                print(k+': '+repr(j[k]))
    print('counter=',counter,'\n')
```

activity

```
average_met: 1.28125
cal_active: 303
cal_total: 2426
class_5min: '000000000000000000000000000000000000000000000000000
0000002223222233332232222333332233332322233332032232232122233322322222222233213223
4442332222344222333233333223222223212222222232222122222322342222212222221222322 3
422221221111111111111111111111111111111112211111111111111'
daily_movement: 5588
day_end: '2019-08-16T03:59:59+02:00'
day_start: '2019-08-15T04:00:00+02:00'
high: 1
inactive: 500
inactivity_alerts: 0
low: 244
medium: 27
met_1min: [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
1.3, 1.6, 1.5, 1.9, 1.3, 1.2, 1.5, 1.2, 1.4, 1.2, 0.1, 0.1, 1.6, 3.3, 2, 2.1,
2.3, 3.2, 2, 1.4, 2.3, 1.4, 1.4, 1.2, 1.2, 1.4, 1.2, 1.3, 1.3, 1.2, 1.2, 1.2,
1.2, 1.2, 1.6, 1.2, 1.2, 1.6, 3, 2.6, 1.5, 2, 1.3, 1.2, 1.2, 1.3, 1.5, 1.2, 1.3,
3.2, 3.4, 2.9, 1.9, 1.3, 1.9, 3, 2.3, 2.7, 2.3, 1.3, 1.4, 1.4, 1.4, 1.2, 1.3,
1.2, 1.2, 1.3, 1.4, 1.9, 3.3, 2.7, 2, 1.2, 2.4, 1.3, 1.2, 1.3, 1.2, 1.2, 1.2,
1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.8, 1.3, 1.4, 1.1, 1.2,
3.9, 3, 1.4, 1.5, 1.8, 1.9, 2.5, 2.1, 2.3, 1.6, 1.5, 1.9, 1.6, 2.3, 1.7, 2, 1.5,
2.2, 1.5, 1.7, 1.5, 2.8, 2.5, 1.4, 1.4, 1.3, 1.3, 1.4, 1.4, 1.2, 1.4, 1.4, 1.4,
1.3, 1.5, 1.8, 2.1, 2.3, 3.2, 3.2, 1.4, 2.5, 1.8, 1.6, 1.3, 1.8, 1.9, 2, 1.5,
1.9, 1.3, 1.7, 1.3, 1.3, 1.7, 1.5, 1.2, 1.4, 1.3, 2.2, 2, 1.8, 1.7, 1.5, 1.3,
1.5, 1.2, 1.2, 1.2, 1.2, 1.2, 1.4, 1.2, 1.3, 1.3, 1.2, 1.2, 1.2, 1.6, 1.4, 1.3,
2.8, 2.1, 2, 1.5, 1.4, 1.3, 1.3, 1.2, 1.5, 2.1, 1.8, 1.6, 1.9, 4.5, 1.2, 0, 1.2,
1.1, 0.9, 0.9, 0.1, 0.1, 0.1, 0.1, 0.1, 1.6, 2, 1.7, 1.8, 1.8, 1.8, 1.8, 1.5,
1.2, 1.2, 1.5, 1.3, 1.2, 1.3, 1.3, 2.3, 2.2, 3.6, 2.4, 1.8, 4.2, 1.2, 1.7, 1.9,
1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.3, 1.8, 2.4, 1.8, 1.6, 1.7, 1.5, 1.3, 1.3, 1.4,
1.9, 1.4, 1.4, 1.3, 1.2, 1.3, 1.3, 1.9, 1.3, 1.3, 1.3, 2.6, 1.4, 1.6, 1.2, 1.2,
1.7, 1.4, 1.2, 1.2, 1.2, 1.3, 1.4, 2.1, 2.4, 2.6, 1.3, 2.2, 2.7, 1.7, 2.3, 3.4,
2.9, 1.7, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2, 1.3, 1.3, 1.3, 1.2, 1.3, 1.4, 4.9, 2.6,
1.3, 1.3, 4.1, 1.4, 2.1, 1.4, 1.2, 1.2, 1.4, 1.2, 1.2, 1.7, 1.6, 2.7, 0.1, 0.1,
0.1, 1.6, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.3, 1.4, 1.2, 1.2, 1.2,
1.2, 1.2, 1.2, 1, 1.2, 1.2, 1.4, 2.6, 2.1, 1.7, 2, 1.8, 3.3, 1.6, 1.2, 1.4, 1.3,
1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2, 1.3, 1.4, 1.5, 2.3, 2.7, 1.8, 1.6, 1.2, 1.3,
1.3, 1.3, 1.4, 1.2, 1.2, 1.2, 1.4, 1.3, 1.4, 1.3, 2.3, 2.4, 1.2, 1.2, 1.3, 1.4,
5.2, 5.1, 5.4, 4.7, 1.9, 2, 4, 5.2, 5.2, 4.9, 2.4, 1.2, 1.2, 1.2, 1.2, 1.6, 1.2,
1.2, 1.2, 1.2, 1.6, 3.2, 1.7, 2.1, 2.3, 1.7, 2.5, 1.7, 1.2, 1.2, 1.2, 0.9, 0.9,
0.9, 1, 1.8, 0.9, 0.1, 0.1, 1.2, 1, 1, 1.4, 1.3, 1.1, 1, 1.2, 1.3, 1.3, 2.3, 2,
2.5, 1.8, 1.4, 1.7, 4.8, 4.2, 5.7, 5.4, 8.3, 4.3, 2.1, 1.2, 1.2, 1.2, 1.2, 1.7,
1.7, 1.2, 1.2, 1.3, 1.2, 1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.9, 3.4, 3.9,
2.9, 2.8, 4.1, 3.9, 3, 3.9, 4.2, 2.5, 1.7, 1.6, 1.3, 1.4, 1.3, 1.4, 1.7, 1.7,
1.5, 1.9, 1.5, 2.5, 3.2, 3, 1.9, 3.4, 2.5, 2.4, 2.7, 2.3, 1.6, 2.7, 2.3, 1.8,
1.8, 1.8, 2.7, 2.2, 2, 2.8, 1.3, 1.4, 1.5, 1.3, 1.4, 1.3, 1.6, 2.2, 1.6, 1.2,
1.2, 1.2, 1.3, 1.2, 2, 2.6, 2.5, 1.4, 1.3, 1.3, 1.3, 1.5, 1.4, 1.4, 1.5, 1.7,
1.2, 1.2, 1.3, 1.4, 1.3, 1.4, 2.1, 1.6, 1.5, 1.3, 1.2, 1.2, 1.2, 1.2, 1.8, 3,
2.9, 2.9, 2, 1.3, 1.2, 1.1, 1.2, 1.2, 1.1, 0.9, 0.9, 0.9, 1.1, 1, 0.9, 0, 0,
1.3, 1.3, 1.2, 1.2, 1.2, 1.4, 1.3, 1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2,
1.2, 1.2, 1.1, 1.1, 1.2, 1, 1.2, 0.9, 0.9, 0.9, 1.1, 1.3, 1.5, 1.4, 1.3, 1.2,
1.3, 1.2, 1.3, 1.5, 1.3, 2.2, 2.3, 2.6, 1.3, 1.5, 1.8, 1.3, 1.5, 1.7, 1.5, 1.3,
1.3, 1.2, 1.3, 1.4, 1.2, 1.4, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 0.9, 1, 0.9,
1.1, 0.9, 1, 0.9, 0.9, 1, 1.1, 2.8, 1.3, 0.9, 1.3, 1.5, 1.3, 1.2, 1.3, 1.4, 1.3,
1.4, 1.6, 1.2, 1.2, 1.3, 2, 1.6, 1.4, 1.4, 1.9, 1.7, 1.6, 1.6, 1.9, 2.6, 1.3,
1.5, 1.4, 1.3, 1.5, 1.2, 1.2, 1.2, 1.3, 1.2, 1.4, 1.7, 1.5, 1.7, 3.7, 5.5, 2.4,
1.8, 5.2, 4.8, 1.6, 1.3, 1.4, 1.5, 1.2, 1.6, 1.6, 1.2, 1.2, 1.3, 1.4, 1.2, 1.4,
1.4, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2,
0.9, 0.9, 0.9, 1, 0.9, 1.2, 0.9, 1, 1.1, 1.1, 1.7, 1.4, 2.9, 1.5, 1.7, 1.5, 1.6,
1.3, 1.4, 1.6, 1.3, 1.5, 1.3, 1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.4, 1.3, 1.3, 1.2,
1.2, 1.2, 1.2, 1.3, 1.3, 1.3, 1.3, 1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.4, 1.3, 1.3,
1.3, 1.3, 1.4, 1.9, 1.7, 1.4, 2.3, 1.9, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2,
1.4, 1.5, 1.2, 2, 1.4, 1.2, 1.4, 2.9, 2.9, 4.9, 4.1, 1.2, 1.3, 1.4, 1.2, 1.2,
1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2, 1.2, 1.2,
1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.3, 1.2, 1.2, 1.2, 1.2, 1.2, 1.3, 1.2, 1.2, 1.1,
0.9, 0.9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 0.9, 0.9, 0.9,
0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 1, 0.9, 0.9, 0.9,
0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1.1, 0.9, 0.9, 0.9, 0.9, 0.9, 1.2, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9,
0, 0, 0, 0, 0, 0, 0, 1.2, 0.9, 1, 0.9, 1.1, 0.9, 0.9, 0.9, 0.9, 1.2, 0.9, 0.9,
0.9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 1, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9,
1, 0.9, 1, 0.9, 0.9, 0, 1.4, 3.2, 1.2, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9,
0.9, 0.9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 0.9, 1, 1, 0.9,
0.9, 0.9, 0.9, 0.9, 1.1, 0.9, 0.9, 0.9, 1.1, 0.9, 0.9, 0.9, 0.9, 0.9, 1.1, 1.3,
0.9, 0.9, 0.9, 0.9, 0.9, 0, 0, 0, 0, 0, 0, 0, 0, 1.1, 0.9, 0.9, 0.9, 0.9, 0.9,
0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0, 0, 0, 0, 0, 0, 0, 0, 0]
met_min_high: 7
met_min_inactive: 6
met_min_low: 152
met_min_medium: 78
non_wear: 378
rest: 290
score: 91
score_meet_daily_targets: 78
score_move_every_hour: 100
score_recovery_time: 98
score_stay_active: 82
score_training_frequency: 100
score_training_volume: 99
steps: 7237
summary_date: '2019-08-15'
target_calories: 450
target_km: 9
target_miles: 5.5
timezone: 120
to_target_km: 2.9
to_target_miles: 1.7
total: 272
counter= 584
```

readiness

```
period_id: 0
score: 81
score_activity_balance: 100
score_previous_day: 64
score_previous_night: 70
score_recovery_index: 56
score_resting_hr: 100
score_sleep_balance: 80
score_temperature: 93
summary_date: '2019-08-16'
counter= 572
```

restful_periods

```
bedtime_end: '2020-08-12T16:50:02+02:00'
bedtime_start: '2020-08-12T16:37:02+02:00'
breath_average: 12.75
duration: 780
```

```
period_id: 1
summary_date: '2020-08-12'
timezone: 120
counter= 1
```

sleep

```
awake: 5550
bedtime_end: '2019-08-16T07:41:32+02:00'
bedtime_end_delta: 27692
bedtime_start: '2019-08-15T23:18:32+02:00'
bedtime_start_delta: -2488
breath_average: 12.75
deep: 5190
duration: 30180
efficiency: 82
hr_5min: [0, 57, 58, 58, 53, 52, 51, 51, 51, 51, 52, 55, 52, 50, 52, 52, 53, 58,
56, 56, 54, 51, 50, 51, 51, 51, 52, 51, 49, 50, 49, 52, 48, 47, 47, 49, 50, 47,
47, 0, 51, 49, 51, 50, 51, 50, 49, 48, 49, 48, 44, 45, 46, 47, 46, 45, 47, 46,
45, 44, 45, 45, 43, 44, 48, 46, 51, 48, 49, 54, 51, 51, 52, 51, 50, 52, 50, 0,
49, 50, 51, 52, 54, 48, 48, 48, 48, 48, 49, 49, 51, 53, 53, 52, 51, 46, 48, 48,
50, 46, 51]
hr_average: 52.375
hr_lowest: 43
hypnogram_5min: '44442111111121123111111222222222221122442232222222221122222222
22333444442224444432222222223332333344'
is_longest: 1
light: 15630
midpoint_at_delta: 12182
midpoint_time: 14670
onset_latency: 1080
period_id: 0
rem: 3810
restless: 42
rmssd: 61
rmssd_5min: [0, 40, 29, 27, 34, 46, 57, 50, 49, 53, 56, 56, 79, 67, 59, 68, 54,
41, 52, 40, 47, 54, 55, 55, 50, 49, 47, 67, 83, 67, 52, 68, 80, 84, 61, 49, 59,
78, 69, 0, 41, 55, 61, 66, 53, 58, 57, 50, 49, 61, 63, 68, 71, 60, 63, 75, 60,
65, 59, 78, 63, 82, 86, 96, 83, 80, 54, 78, 67, 40, 60, 53, 44, 52, 66, 58, 49,
0, 45, 42, 46, 39, 38, 52, 56, 55, 40, 43, 42, 60, 72, 63, 45, 47, 63, 89, 58,
73, 69, 86, 60]
score: 78
score_alignment: 88
score_deep: 96
score_disturbances: 67
score_efficiency: 76
score_latency: 91
score_rem: 60
score_total: 77
```

```
summary_date: '2019-08-16'
temperature_delta: -0.31
temperature_deviation: -0.31
timezone: 120
total: 24630
counter= 570
```

[4]:
```
#Visualizing the last entry in the sleep data
print(data['sleep'][570])
```

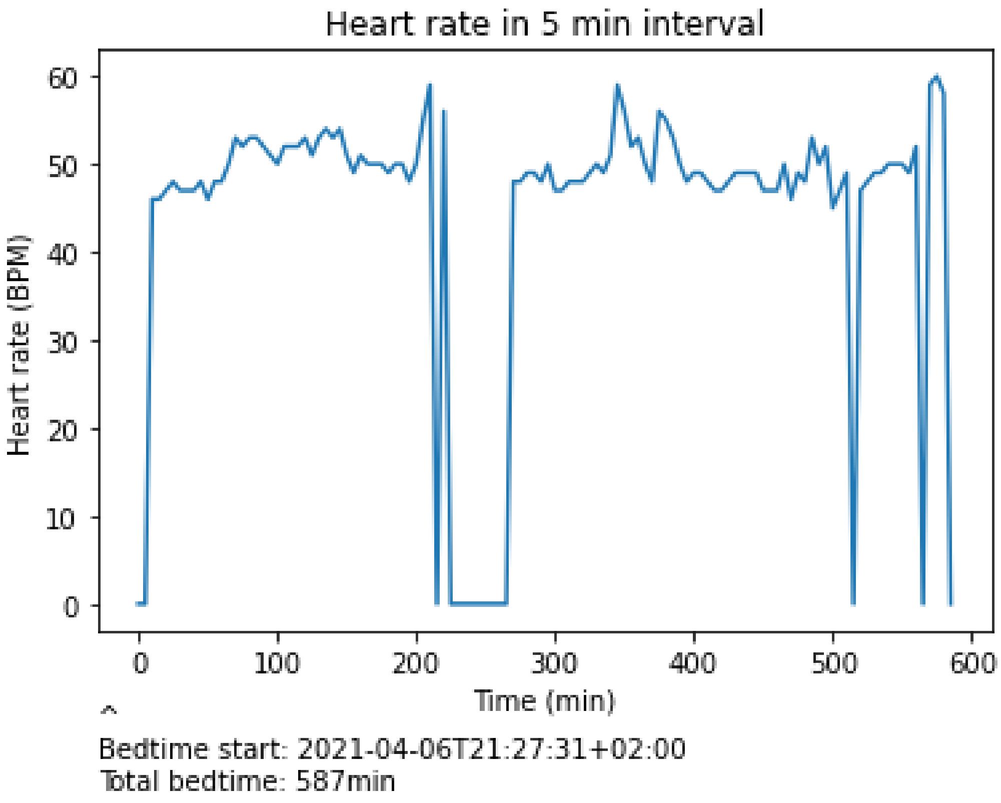
```
{'awake': 2130, 'bedtime_end': '2021-04-13T07:21:23+02:00', 'bedtime_end_delta':
26483, 'bedtime_start': '2021-04-12T21:15:23+02:00', 'bedtime_start_delta':
-9877, 'breath_average': 12.625, 'deep': 4110, 'duration': 36360, 'efficiency':
94, 'hr_5min': [0, 0, 0, 64, 61, 61, 59, 59, 60, 61, 61, 62, 61, 62, 59, 58, 56,
59, 64, 61, 60, 60, 59, 59, 59, 58, 58, 58, 57, 58, 57, 56, 53, 55, 52, 51, 51,
52, 52, 53, 53, 55, 53, 53, 55, 57, 55, 54, 54, 54, 53, 52, 53, 52, 52, 53, 51,
50, 51, 50, 50, 52, 52, 54, 53, 55, 55, 58, 53, 52, 53, 54, 52, 51, 53, 53, 54,
53, 53, 53, 55, 52, 54, 54, 53, 54, 53, 53, 54, 54, 55, 52, 51, 51, 52, 53, 52,
51, 51, 51, 50, 50, 48, 51, 54, 50, 49, 53, 53, 52, 53, 54, 54, 53, 54, 55, 56,
55, 52, 52, 60, 0], 'hr_average': 54.51, 'hr_lowest': 48, 'hypnogram_5min': '444
42211111112222233322111211222222222212222233222222222222222222233222222222222332
223333332222212222222233232222222223344', 'is_longest': 1, 'light': 23970,
'midpoint_at_delta': 8663, 'midpoint_time': 18540, 'onset_latency': 1320,
'period_id': 0, 'rem': 6150, 'restless': 50, 'rmssd': 45, 'rmssd_5min': [0, 0,
0, 28, 27, 32, 36, 30, 28, 29, 27, 30, 31, 30, 36, 34, 35, 31, 19, 21, 22, 36,
27, 23, 25, 39, 31, 34, 38, 66, 47, 55, 39, 47, 58, 54, 52, 46, 45, 57, 45, 32,
47, 56, 51, 59, 44, 46, 39, 46, 36, 50, 39, 45, 67, 52, 47, 50, 49, 56, 53, 51,
58, 57, 59, 52, 39, 45, 45, 47, 39, 40, 42, 44, 38, 36, 34, 53, 56, 43, 50, 72,
44, 53, 45, 41, 38, 48, 57, 50, 43, 44, 53, 47, 60, 41, 42, 55, 53, 51, 63, 69,
57, 81, 47, 59, 72, 60, 49, 50, 52, 52, 48, 42, 41, 34, 33, 64, 55, 51, 38, 0],
'score': 91, 'score_alignment': 100, 'score_deep': 92, 'score_disturbances': 63,
'score_efficiency': 99, 'score_latency': 81, 'score_rem': 95, 'score_total':
100, 'summary_date': '2021-04-13', 'temperature_delta': 0.32,
'temperature_deviation': 0.32, 'temperature_trend_deviation': 0.19, 'timezone':
120, 'total': 34230}
```

1.5 Visualizing the data

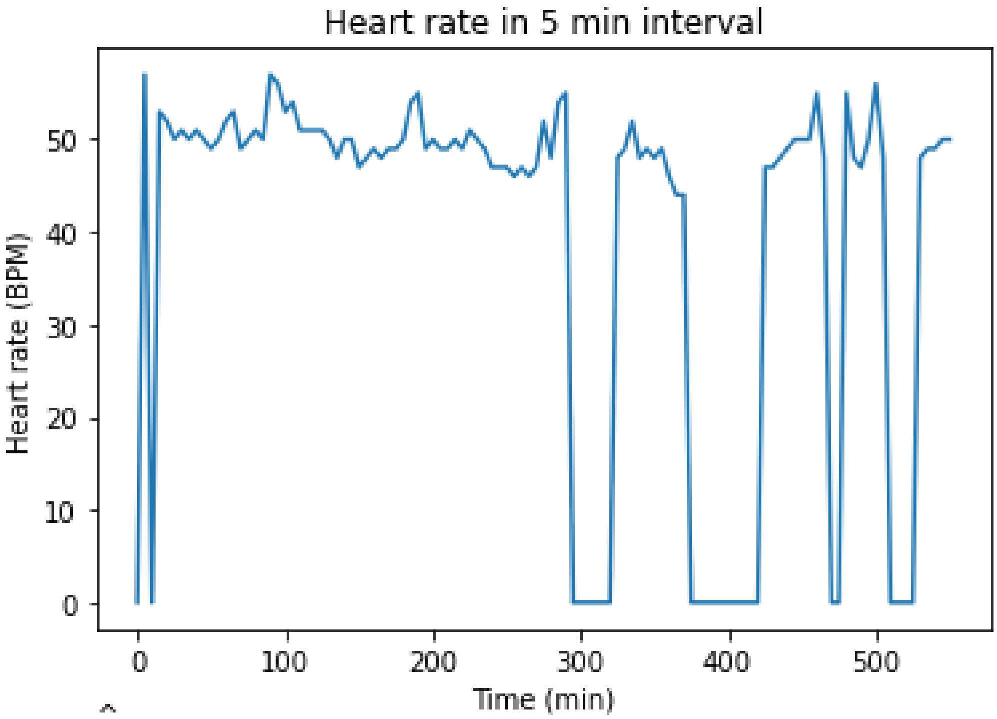
Below I will visualize the semi-continuous sleep data, with which I mean to refer to the data entries that are measured once every five minutes. Respectively the 'hr_5min', the 'hypnogram_5min' and the 'rmssd_5min' variables.

[5]:
```
#Plotting the heart rate as measured in 5 minute intervals for the last week␣
 ↪covered by the dataset
counter=-1
for i in data['sleep']:
    counter+=1
    if counter>(570-7):
        hr5=i['hr_5min']
        fig,ax=plt.subplots()
        ax.plot(hr5)
        ax.set_title('Heart rate in 5 min interval')
        ax.set_ylabel('Heart rate (BPM)')
        ax.set_xlabel('Time (min)')
        ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
        fig.show()
        fig.text(0.125,-0.08,'^\nBedtime start: {}\nTotal bedtime: {}min'.
↪format(i['bedtime_start'],int(i['duration']/60)))
```

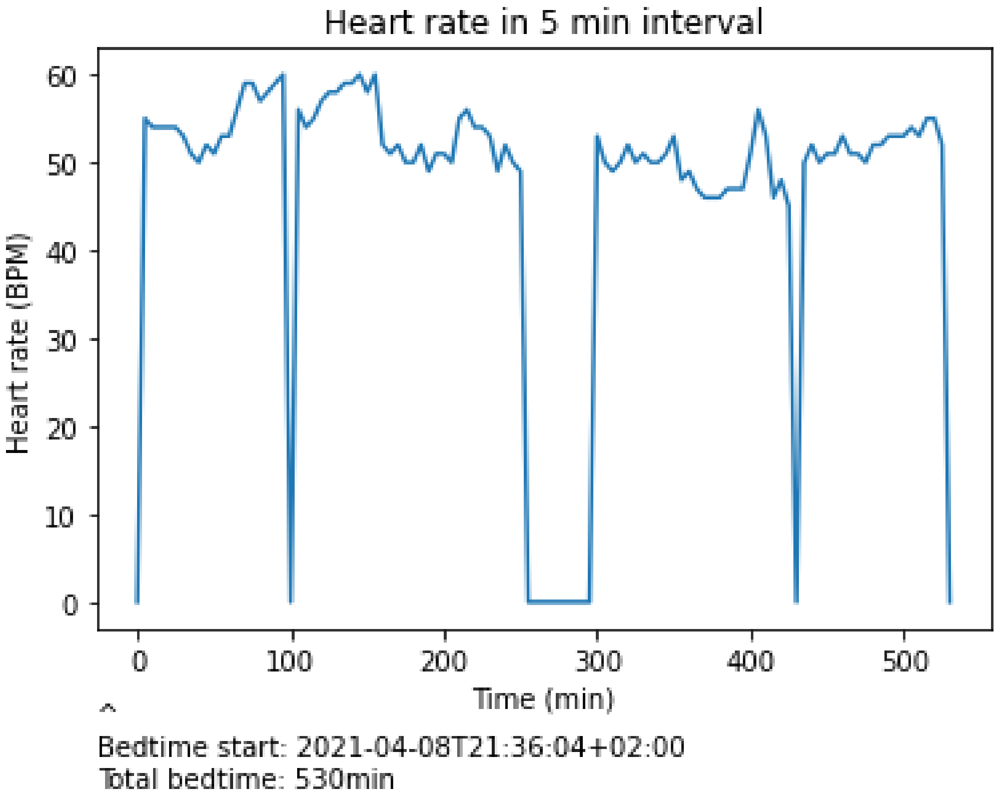
```
<ipython-input-5-dbf3d1b710a0>:12: UserWarning: FixedFormatter should only be
used together with FixedLocator
  ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
<ipython-input-5-dbf3d1b710a0>:13: UserWarning: Matplotlib is currently using
module://ipykernel.pylab.backend_inline, which is a non-GUI backend, so cannot
show the figure.
  fig.show()
```



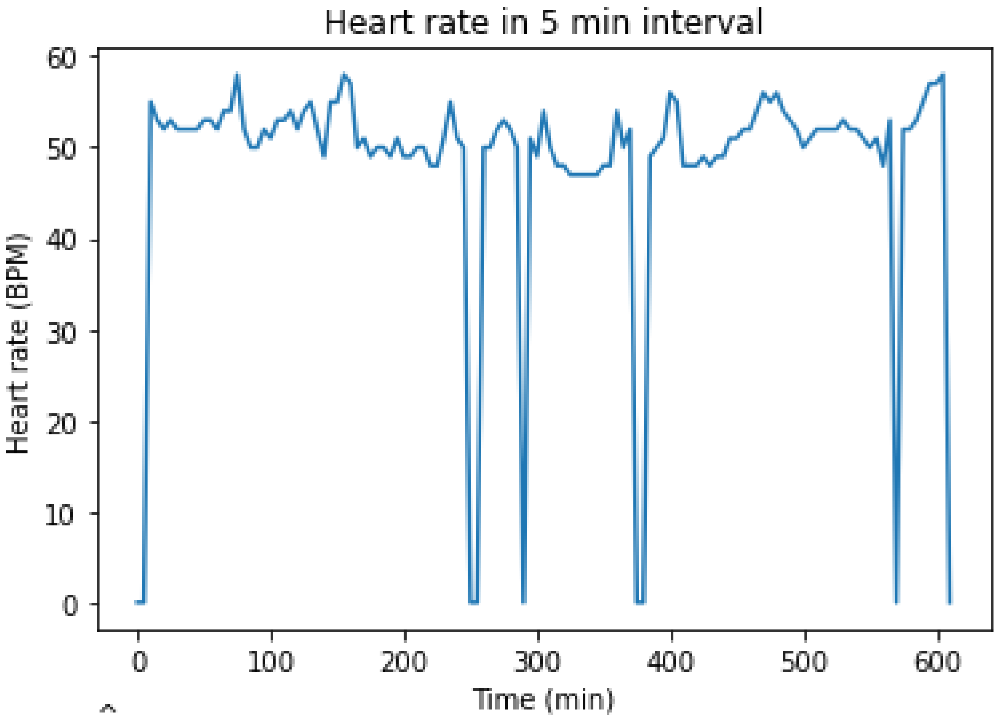
^
Bedtime start: 2021-04-06T21:27:31+02:00
Total bedtime: 587min



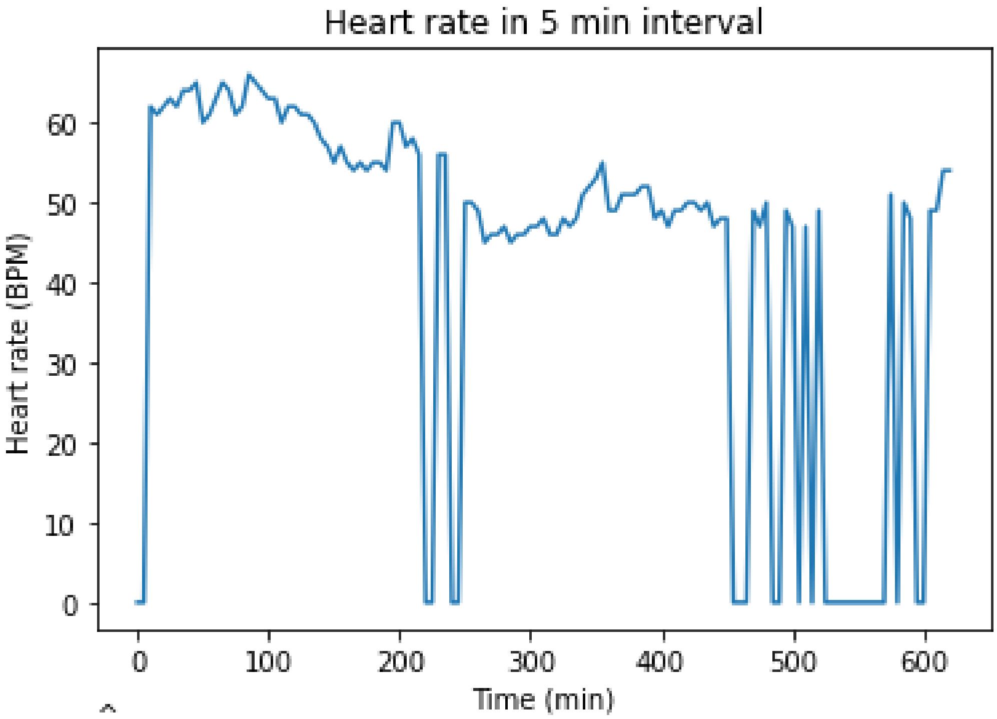
```
^
Bedtime start: 2021-04-07T21:20:06+02:00
Total bedtime: 551min
```



```
^
Bedtime start: 2021-04-08T21:36:04+02:00
Total bedtime: 530min
```

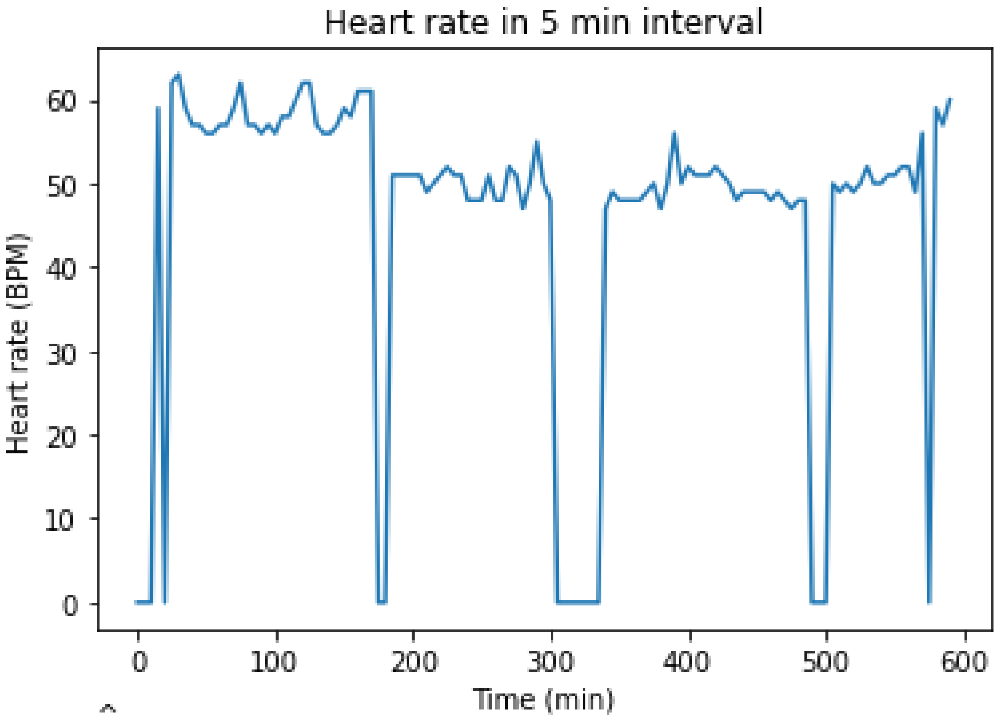


```
^
Bedtime start: 2021-04-09T21:21:30+02:00
Total bedtime: 614min
```



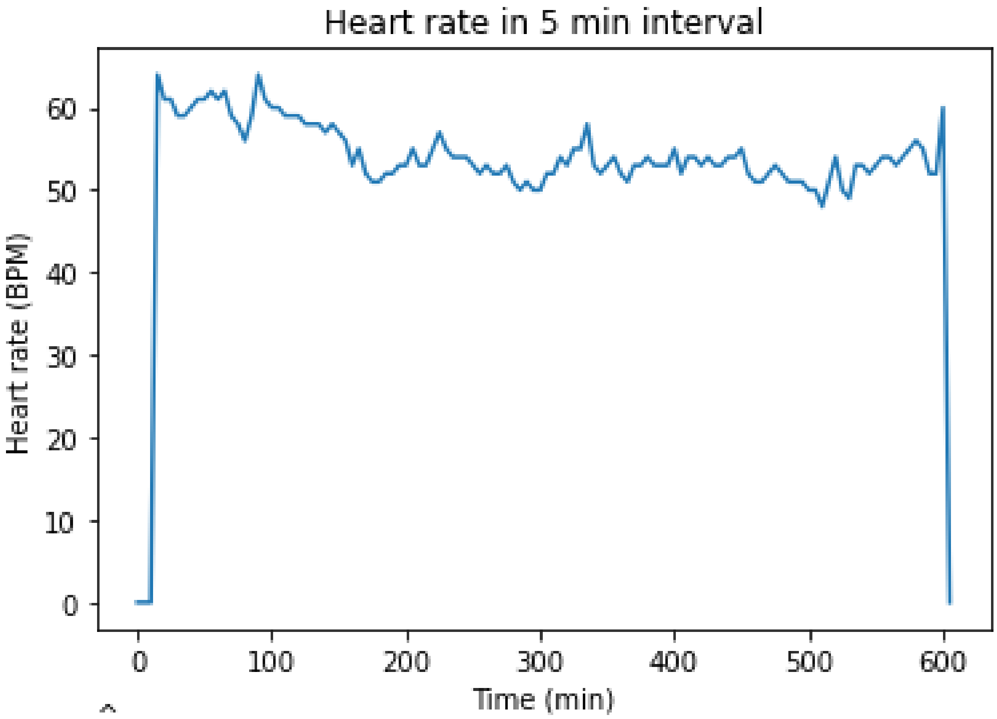
^

```
Bedtime start: 2021-04-10T20:46:44+02:00
Total bedtime: 622min
```



^

Bedtime start: 2021-04-11T21:02:59+02:00
Total bedtime: 593min



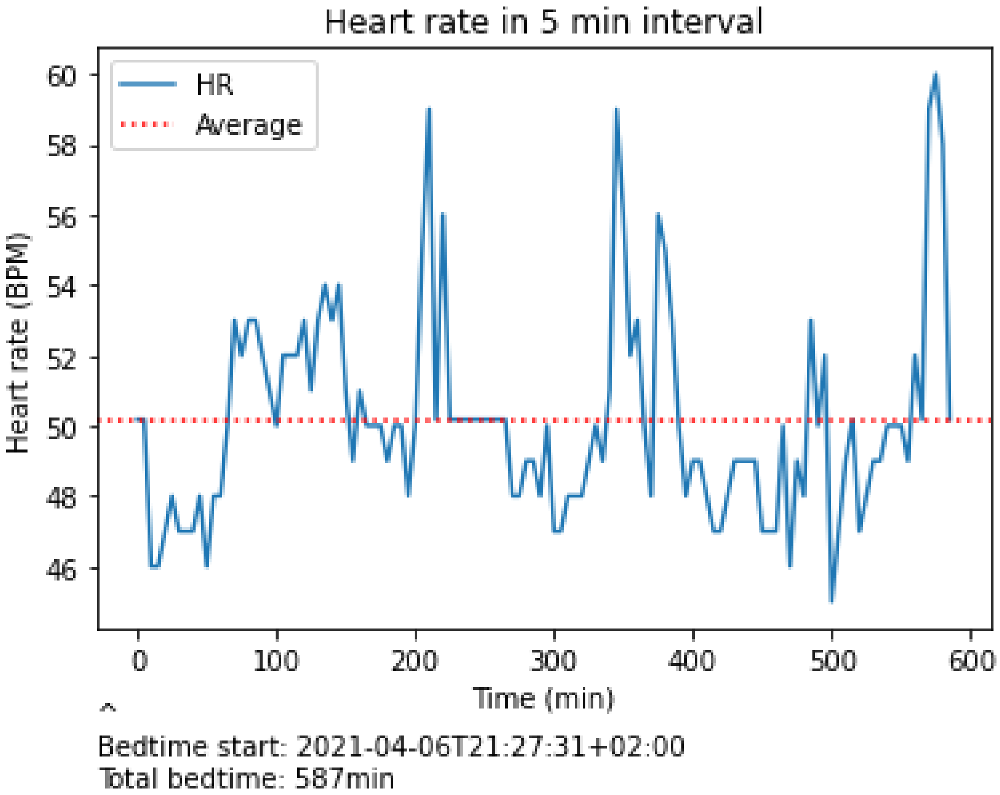
^

Bedtime start: 2021-04-12T21:15:23+02:00
Total bedtime: 606min

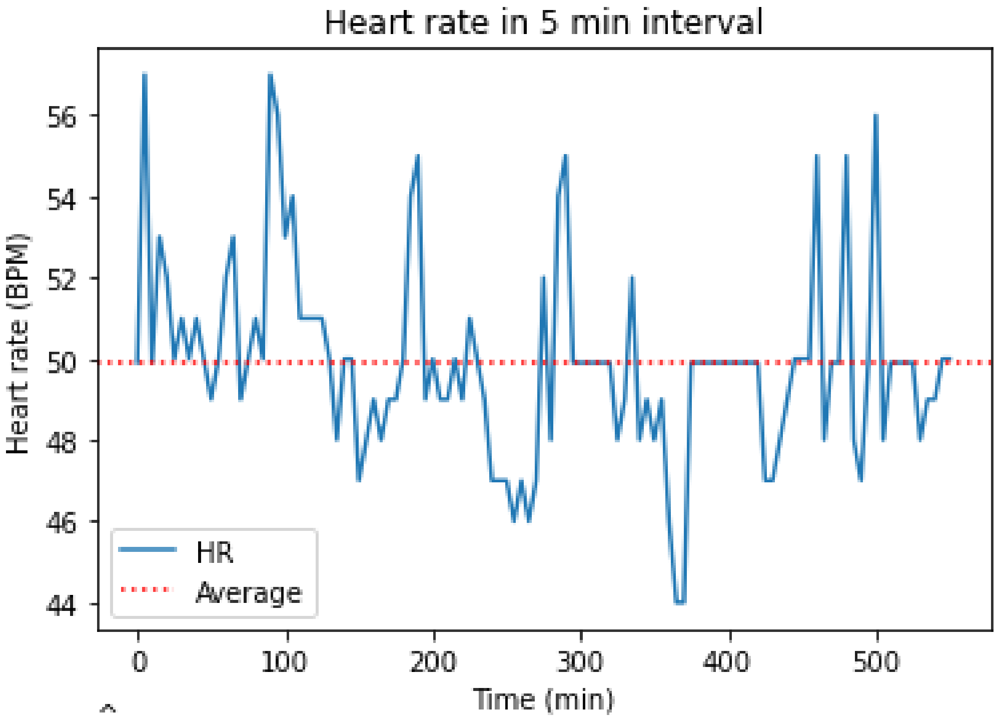
In this data we can see that the heart rate sometimes drops to 0. Assuming that the person didn't suddenly die multiple times during these nights, I decided to resolve this by taking the average heart rate of the entire measurement period and change all the zero values to this average. Since this value is most likely a non-integer value like the heartbeats, it should be easy to see in the plots where the values were zero. In order to make this even more clear, I also plotted the average heart rate in each plot. I decided to use the average heart rate for this, because then the average isn't influenced by this outlier removal.

[6]:
```
#Removing 0 values from the heart rate data as measured in 5 minute intervals␣
 ↪for the last week covered by the dataset and plotting it again
counter=-1
for i in data['sleep']:
    counter+=1
    if counter>(570-7):
        hr5=i['hr_5min']
        n_nonzero=len(numpy.nonzero(hr5)[0])
        nonzero_sum=0
        for j in numpy.nonzero(hr5)[0]:
            nonzero_sum+=hr5[j]
        nonzero_avg=nonzero_sum/n_nonzero
        for k in range(0,len(hr5)):
            if hr5[k]==0:
                hr5[k]=nonzero_avg
        fig,ax=plt.subplots()
        ax.plot(hr5)
        ax.axhline(y=nonzero_avg,c='r',linestyle='dotted')
        ax.set_title('Heart rate in 5 min interval')
        ax.set_ylabel('Heart rate (BPM)')
        ax.set_xlabel('Time (min)')
        ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
        ax.legend(['HR','Average'])
        fig.text(0.125,-0.08,'^\nBedtime start: {}\nTotal bedtime: {}min'.
 ↪format(i['bedtime_start'],int(i['duration']/60)))
```

```
<ipython-input-6-173f3a853cfb>:21: UserWarning: FixedFormatter should only be
used together with FixedLocator
  ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
```

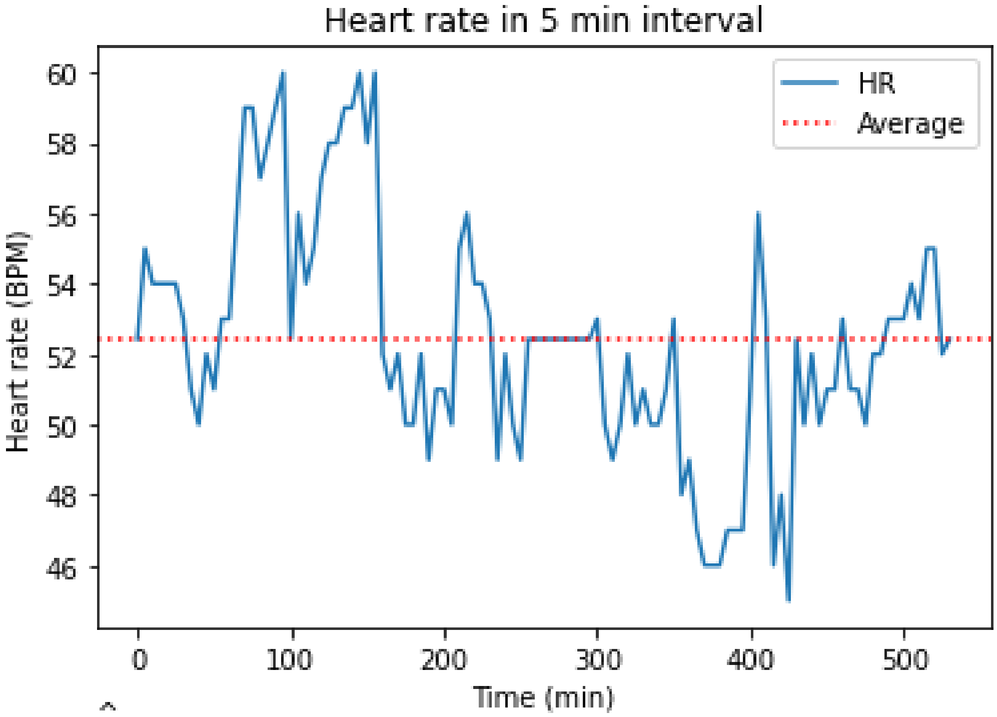


^
Bedtime start: 2021-04-06T21:27:31+02:00
Total bedtime: 587min

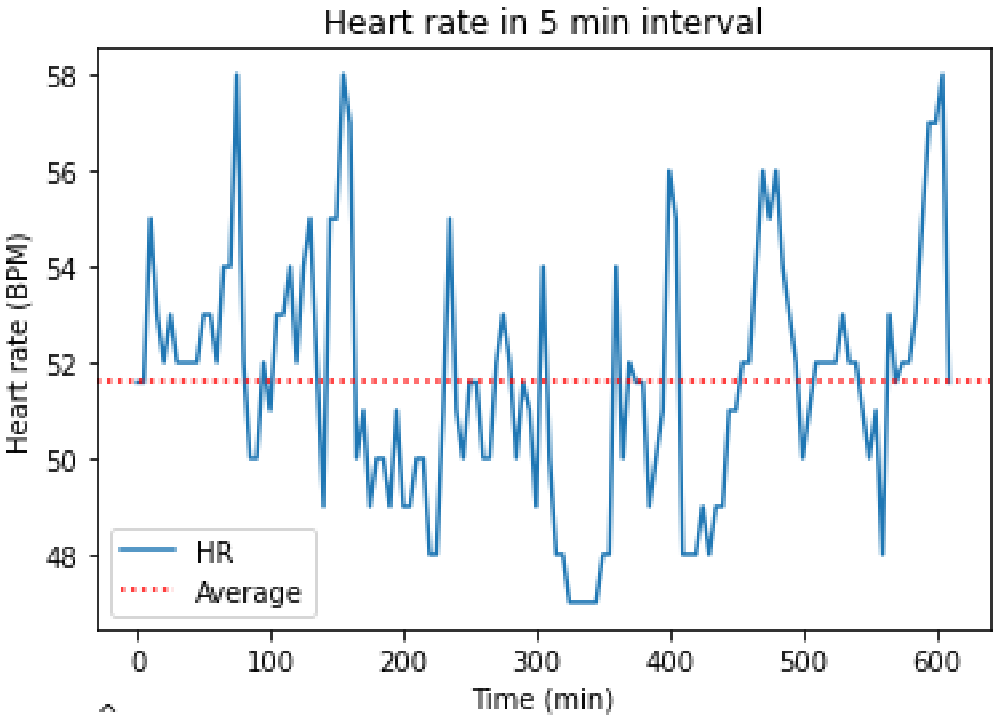


^

Bedtime start: 2021-04-07T21:20:06+02:00
Total bedtime: 551min

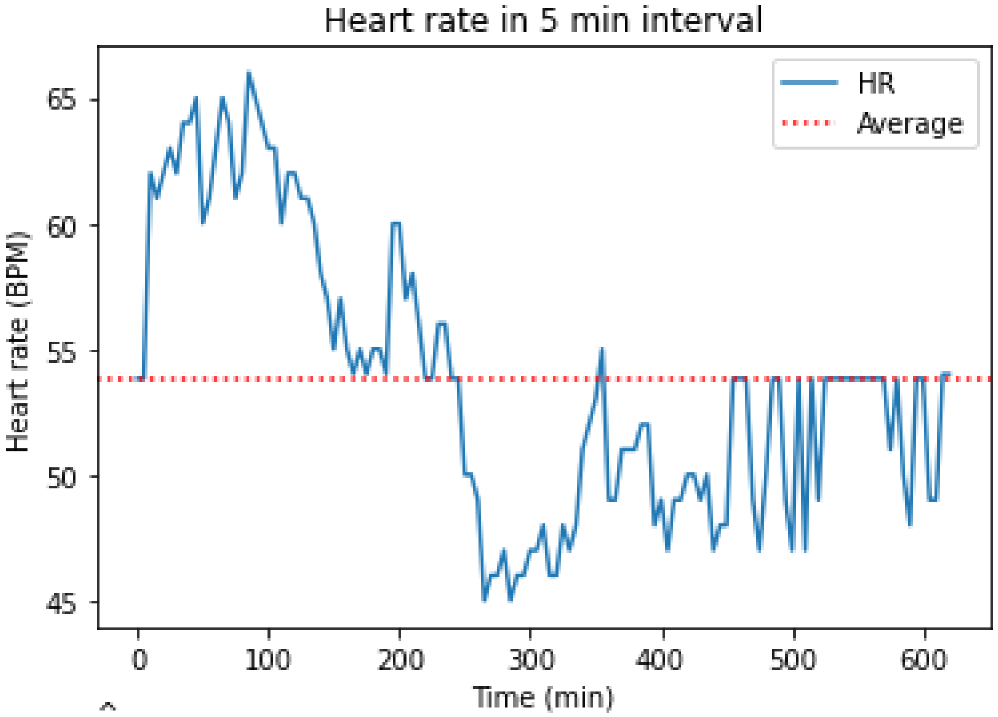


```
^
Bedtime start: 2021-04-08T21:36:04+02:00
Total bedtime: 530min
```



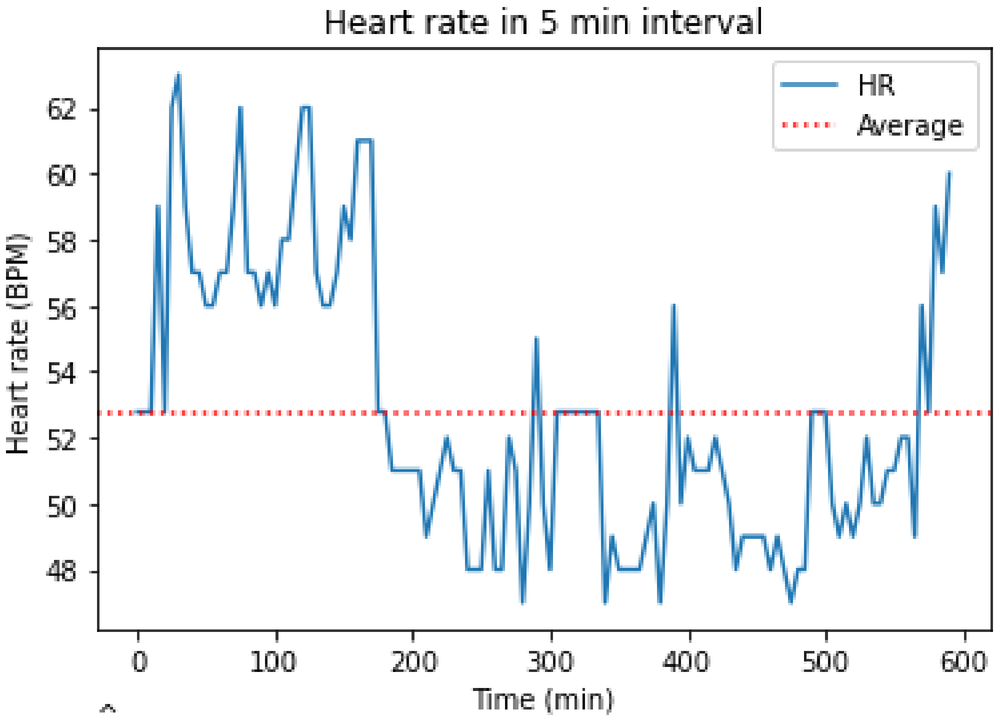
^

Bedtime start: 2021-04-09T21:21:30+02:00
Total bedtime: 614min



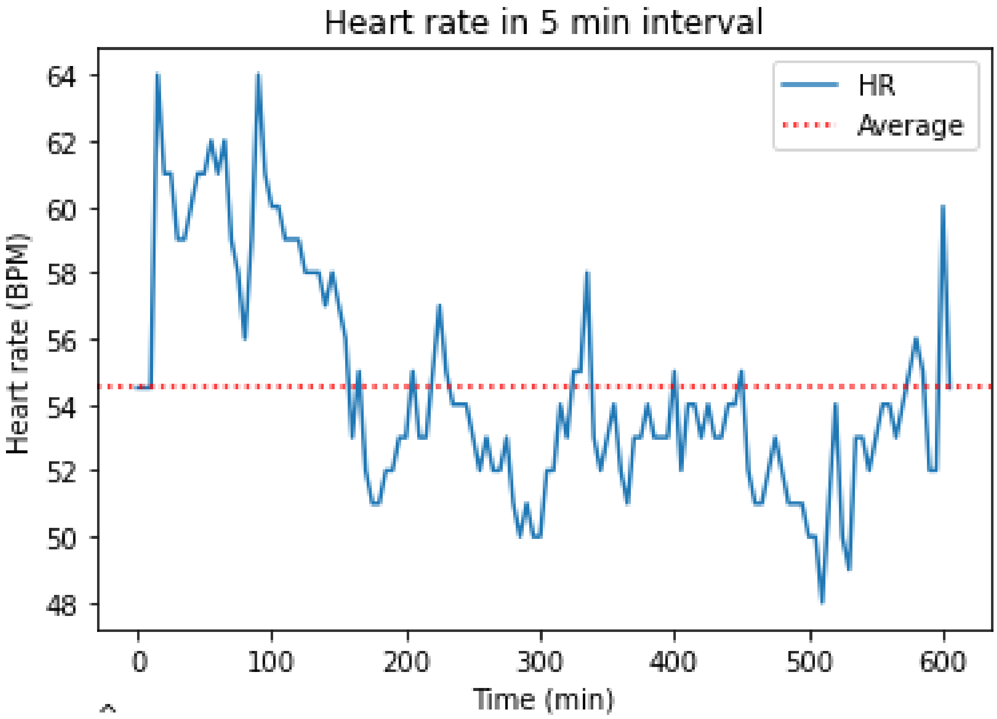
^

Bedtime start: 2021-04-10T20:46:44+02:00
Total bedtime: 622min



^

Bedtime start: 2021-04-11T21:02:59+02:00
Total bedtime: 593min



```
^
Bedtime start: 2021-04-12T21:15:23+02:00
Total bedtime: 606min
```

To interpret the following hypnograms it is necessary to know which sleepstage number corresponds to which sleepstage. This information can be found on: https://cloud.ouraring.com/docs/sleep under the header sleep.hypnogram_5min.

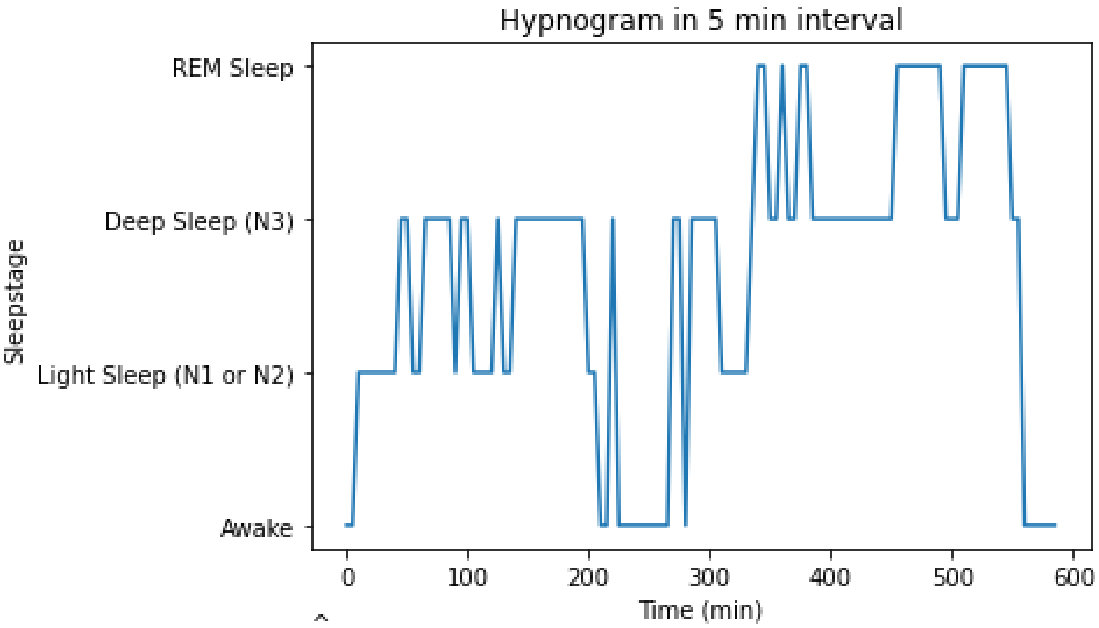
The measurements are taken at the start of every 5 minutes since bedtime_start, which is also a variable in the sleep data.

The sleep stage numbers correspond to:
1. Deep sleep (N3)
2. Light sleep (N1 or N2)
3. REM sleep
4. Awake

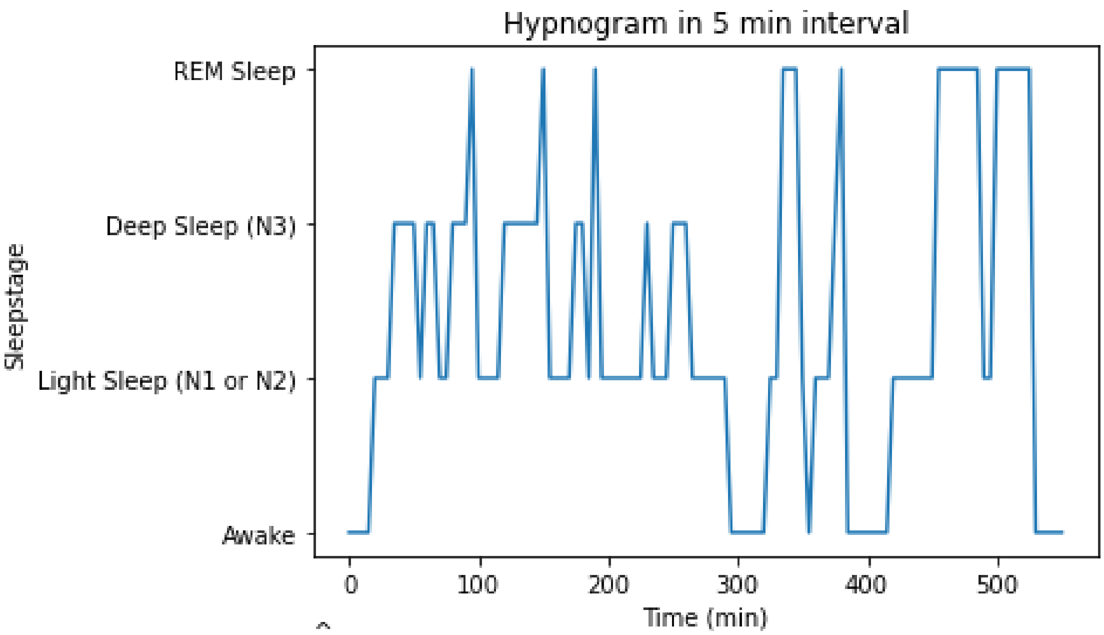
I have adapted the tick labels on the y-axis to match these names for clarity. Furthermore, I have also printed the 'bedtime_start' and 'duration' variables, for completeness.

[7]:
```
#Plotting the hypnograms for the last week
counter=-1
for i in data['sleep']:
    counter+=1
    hg5_conc=i['hypnogram_5min']
    hg5_split=[]
    if counter>(570-7):
        for j in hg5_conc:
            hg5_split.append(j)
        fig, ax=plt.subplots()
        ax.plot(hg5_split)
        ax.set_title('Hypnogram in 5 min interval')
        ax.set_ylabel('Sleepstage')
        ax.set_yticklabels(['Awake','Light Sleep (N1 or N2)','Deep Sleep␣
↪(N3)','REM Sleep'])
        ax.set_xlabel('Time (min)\n')
        ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
        fig.text(0.125,-0.08,'^\nBedtime start: {}\nTotal bedtime: {}min'.
↪format(i['bedtime_start'],int(i['duration']/60)))
```

```
<ipython-input-7-54356b036468>:14: UserWarning: FixedFormatter should only be
used together with FixedLocator
  ax.set_yticklabels(['Awake','Light Sleep (N1 or N2)','Deep Sleep (N3)','REM
Sleep'])
<ipython-input-7-54356b036468>:16: UserWarning: FixedFormatter should only be
used together with FixedLocator
  ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
```

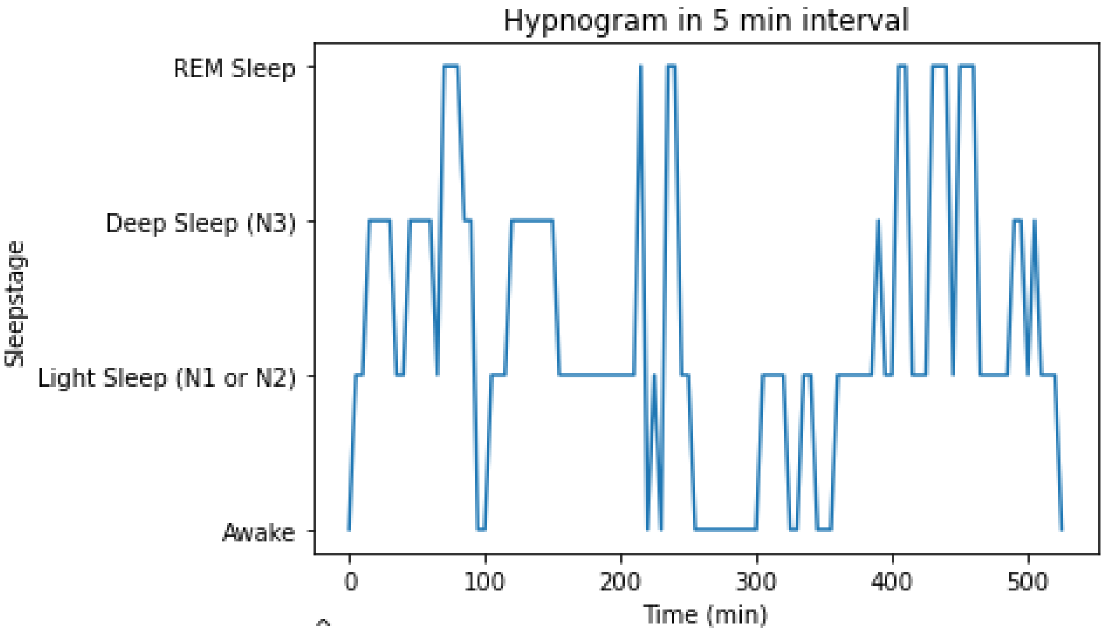


^
Bedtime start: 2021-04-06T21:27:31+02:00
Total bedtime: 587min



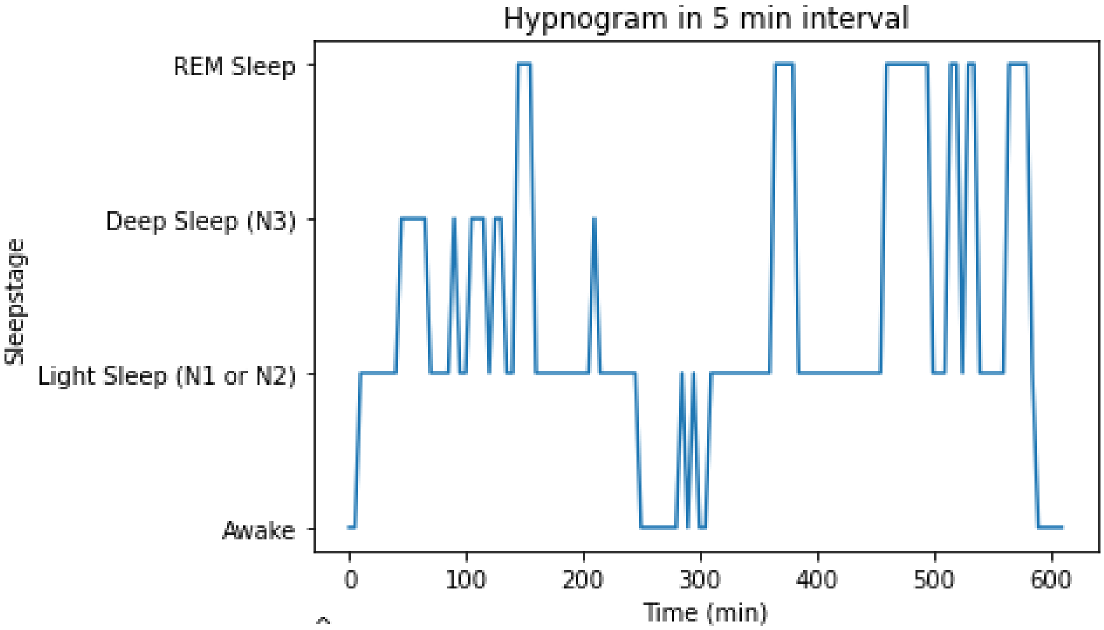
^

Bedtime start: 2021-04-07T21:20:06+02:00
Total bedtime: 551min

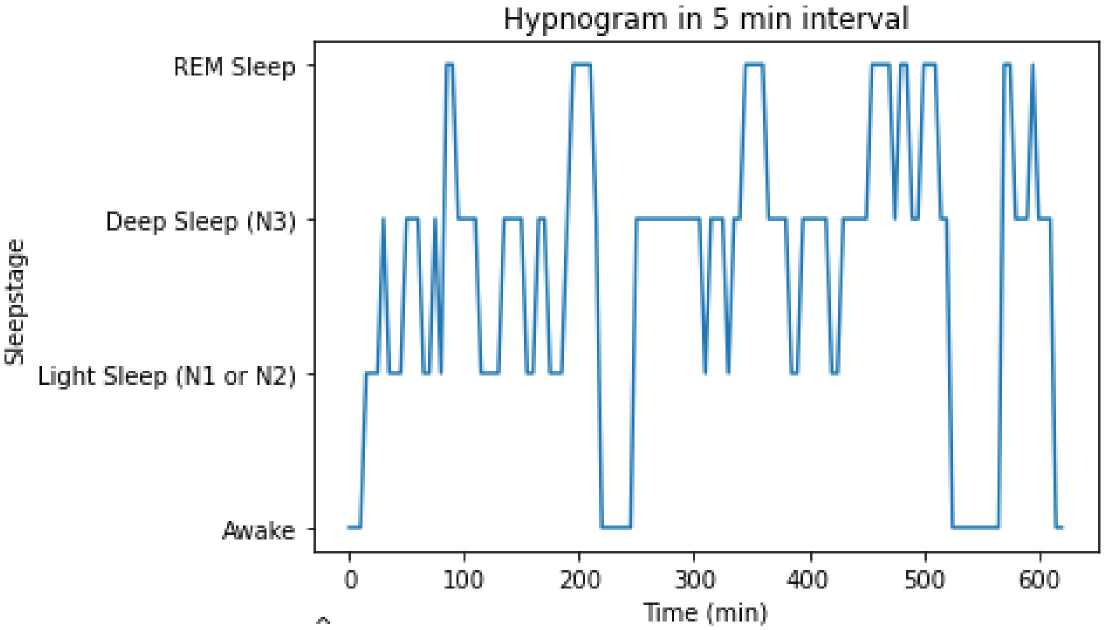


^

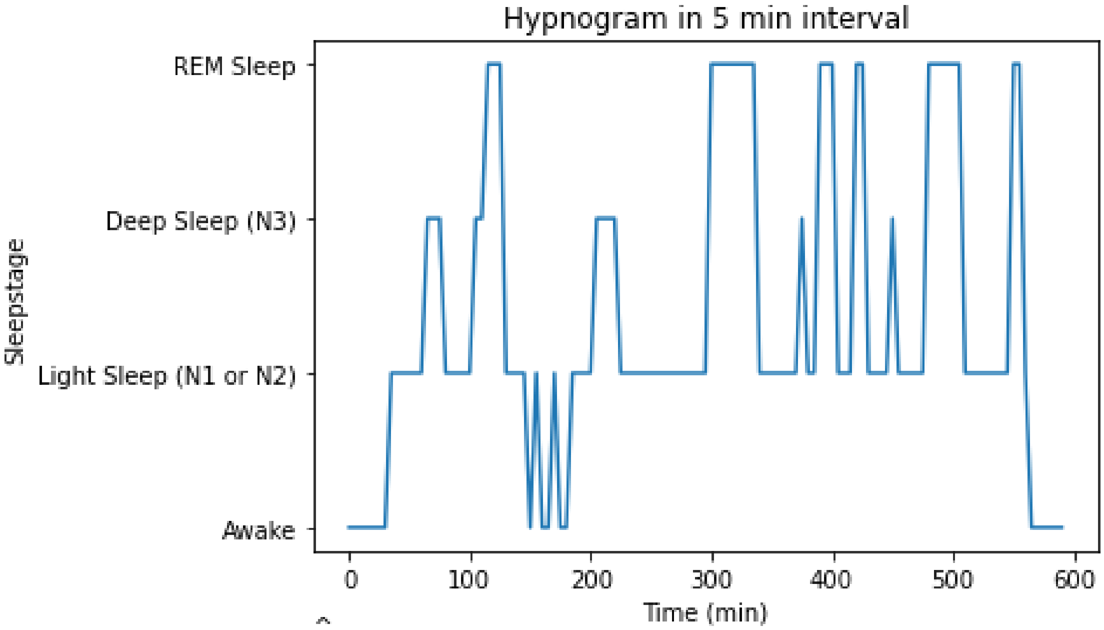
Bedtime start: 2021-04-08T21:36:04+02:00
Total bedtime: 530min



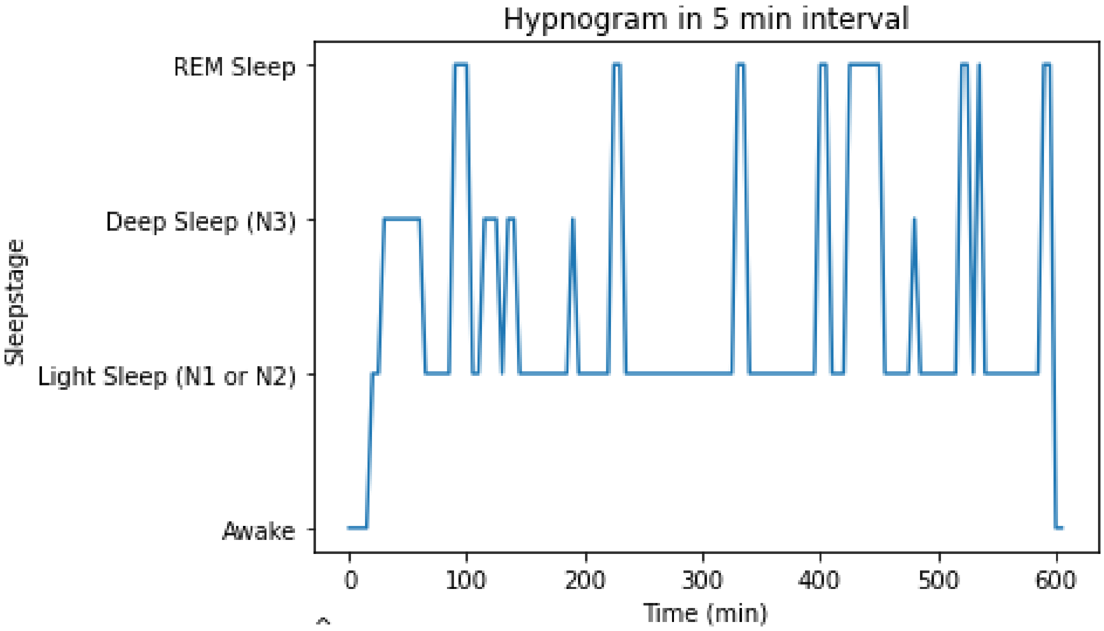
^
Bedtime start: 2021-04-09T21:21:30+02:00
Total bedtime: 614min



^
Bedtime start: 2021-04-10T20:46:44+02:00
Total bedtime: 622min



Bedtime start: 2021-04-11T21:02:59+02:00
Total bedtime: 593min



Bedtime start: 2021-04-12T21:15:23+02:00
Total bedtime: 606min

In order to be able to understand the following section, some prior knowledge is required:

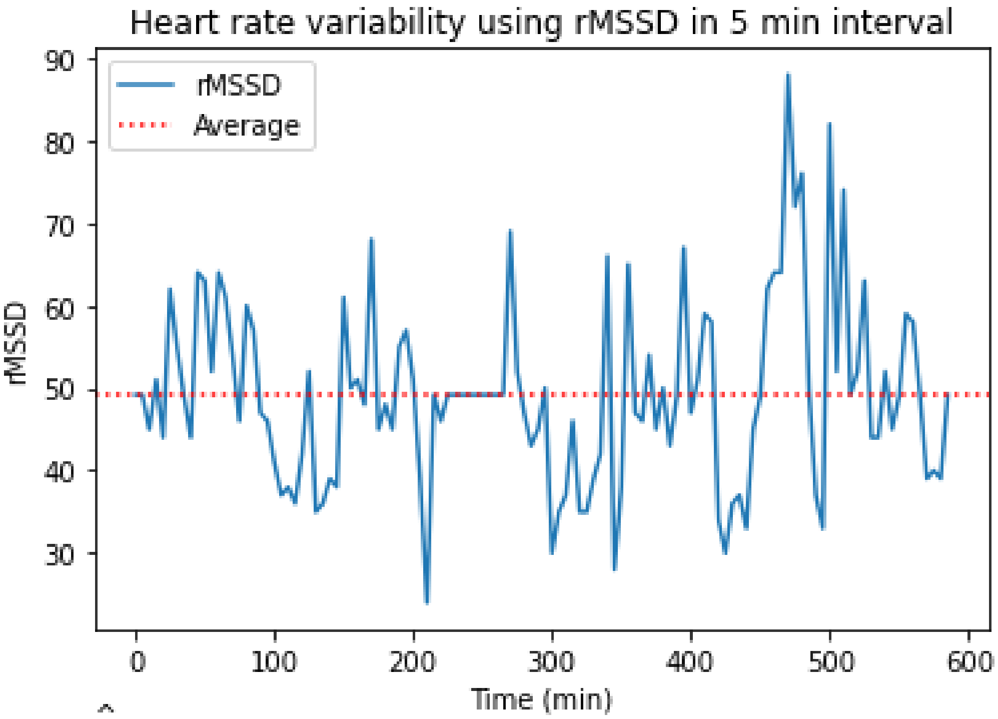
The root mean square of successive differences (rMSSD) is a method to calculate the heart rate variability. '[It] reflects the beat-to-beat variance in the heart rate'. (Shaffer & Ginsberg; 2017)

Furthermore, since this data is based on the heart rate, it will also contain the presumed to be faulty zero measurements, hence I will apply the same outlier removal as described earlier: replacing the zero values by the average over the entire measurement period, visualized for clarity by an additional average line which is most likely located on a non-integer value.

[8]:

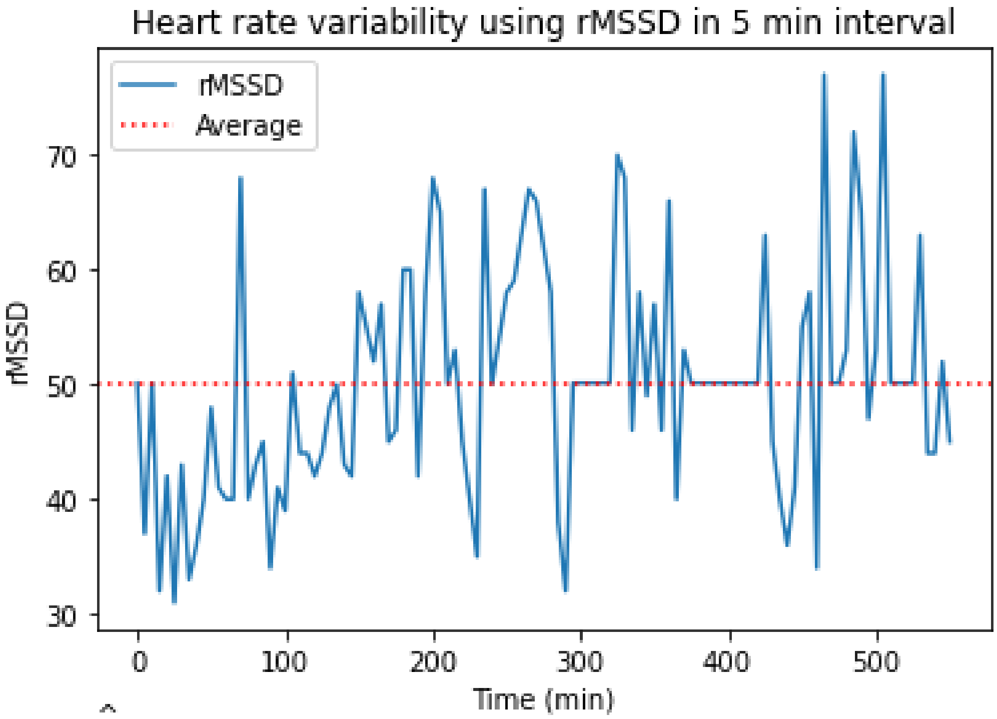
```
#Removing 0 values in the continuous data and plotting it for the last week
counter=-1
for i in data['sleep']:
    counter+=1
    if counter>(570-7):
        rmssd5=i['rmssd_5min']
        n_nonzero=len(numpy.nonzero(rmssd5)[0])
        nonzero_sum=0
        for j in numpy.nonzero(rmssd5)[0]:
            nonzero_sum+=rmssd5[j]
        nonzero_avg=nonzero_sum/n_nonzero
        for k in range(0,len(rmssd5)):
            if rmssd5[k]==0:
                rmssd5[k]=nonzero_avg
        fig,ax=plt.subplots()
        ax.plot(rmssd5)
        ax.axhline(y=nonzero_avg,c='r',linestyle='dotted')
        ax.set_title('Heart rate variability using rMSSD in 5 min interval')
        ax.set_ylabel('rMSSD')
        ax.set_xlabel('Time (min)')
        ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
        ax.legend(['rMSSD','Average'])
        fig.text(0.125,-0.08,'^\nBedtime start: {}\nTotal bedtime: {}min'.
 ↪format(i['bedtime_start'],int(i['duration']/60)))
```

```
<ipython-input-8-286b1dcdfc56>:21: UserWarning: FixedFormatter should only be
used together with FixedLocator
  ax.set_xticklabels(['-1','0','100','200','300','400','500','600'])
```

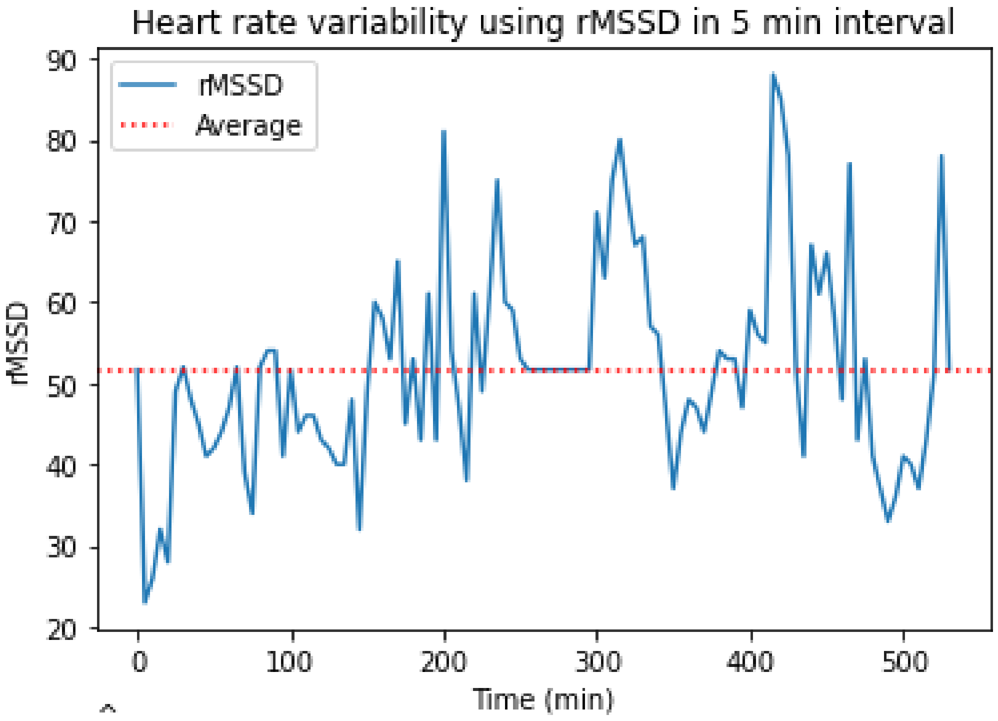


^

Bedtime start: 2021-04-06T21:27:31+02:00
Total bedtime: 587min

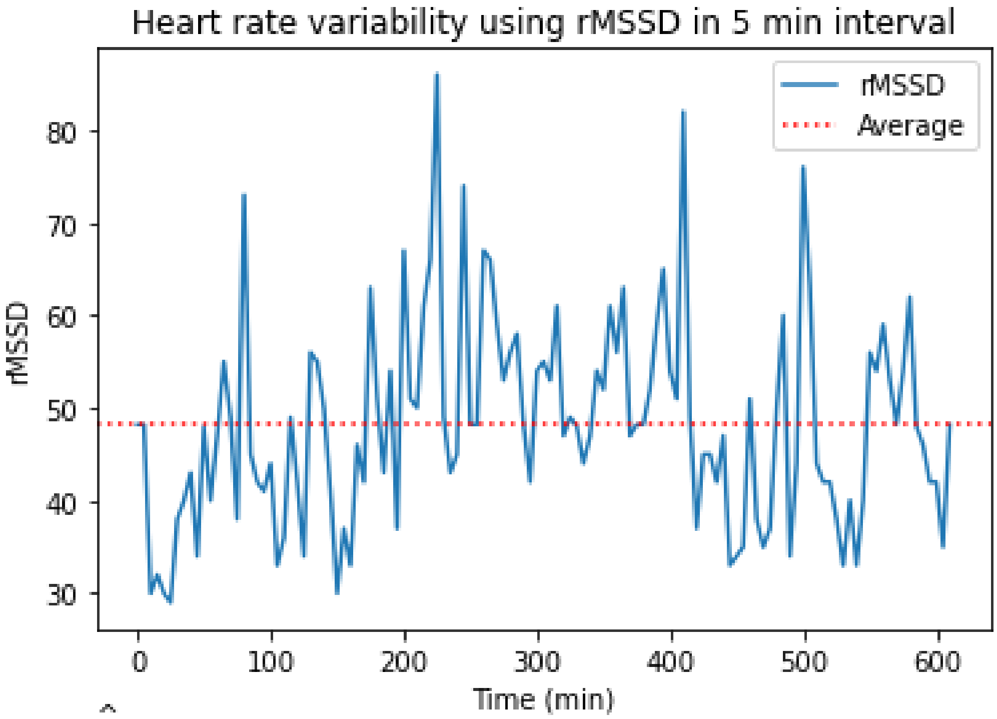


```
^
Bedtime start: 2021-04-07T21:20:06+02:00
Total bedtime: 551min
```

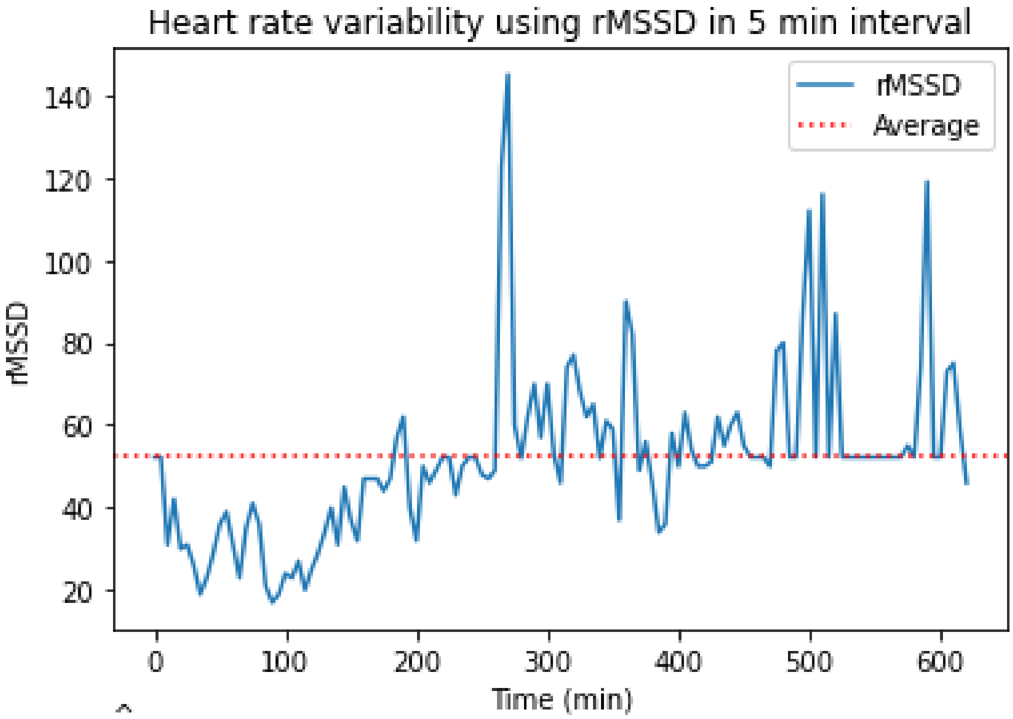


Heart rate variability using rMSSD in 5 min interval

Bedtime start: 2021-04-08T21:36:04+02:00
Total bedtime: 530min

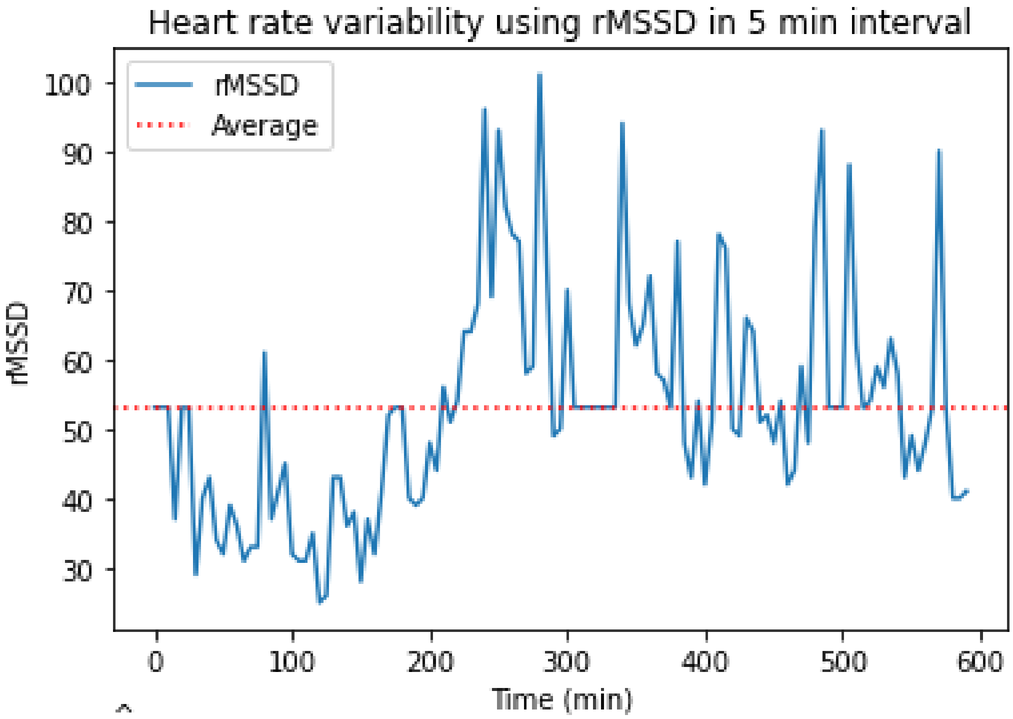


```
^
Bedtime start: 2021-04-09T21:21:30+02:00
Total bedtime: 614min
```



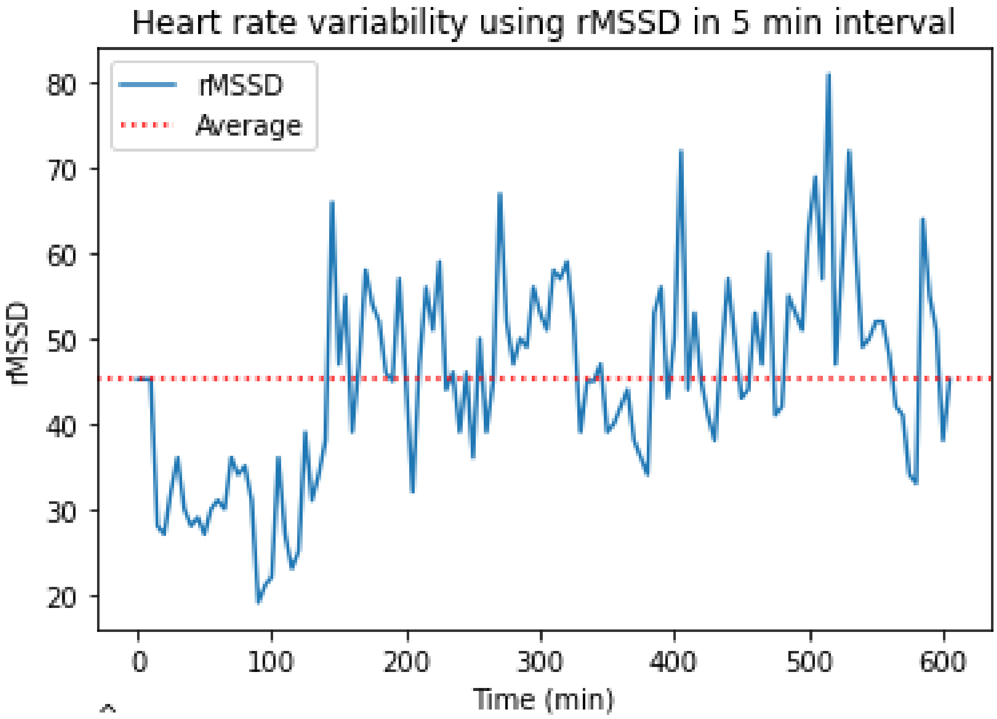
^

Bedtime start: 2021-04-10T20:46:44+02:00
Total bedtime: 622min



^

Bedtime start: 2021-04-11T21:02:59+02:00
Total bedtime: 593min



^

Bedtime start: 2021-04-12T21:15:23+02:00
Total bedtime: 606min