Question 1

Homework 1

(MAX SCORE: 100 points)

The task is to develop a PPG system using a smartphone device to capture imagery blood flow related imagery data and post process such data to estimate the pulse rate of the subject in hand.

Task 1: A Survey of relevant PPG applications [MAX SCORE: 15 points]

Navigate to the Google Play Store and shortlist a few pulse rate/SpO_2 sensing apps (3 – 5) that you find interesting. Submit a brief writeup (3-4 lines) for each of the apps you have shortlisted, as mentioned below.

a. Why do you find them interesting?
b. Is it an outcome of an academic research lab. Are there any research papers or articles linked to it. Is it a startup doing it?
c. Anything associated with the app that you find unique.
d. State a couple of good things the app does and are missing in others.
e. State a couple of glitches that you feel the app has.
For points d & e, restrict yourself to just the sensing, performance, latency, stability, robustness of the sensing. Do not judge based on fancy UIs, nice animations etc.

Solution

Solution:
App 1: Heart Rate Monitor

This app is interesting because :
• Its interface is simple and attractive.
• It is also good because it has some additional ways of doing breathing , like to do 6 breaths in 1 minute, 6 breathe in 2 minute . This can be helpful in measuring the breathing rate, or do breathing exercises.After that it gives , the chart analysis of the bpm.
• Also, it is good as we can store the reading by some labels, and can check it out later in the history option.
• Also, there is stats corner where our weekly, monthly stats are saved labelwise. This gives a pretty good analysis meter.

Owner and Research Paper related: This app is owned by REPS. This app was mentioned in article in Journal of Medical Systems. The article name is "Reviewing Mobile Apps To Control Heart Rate in Literature and Virtual Stores". Also there is a screenshot posted of this app in that.

Uniqueness of the app:
• This app is good and user friendly due to its interface. I pretty much liked it.

• ALso, the management of the stats , charts, weekly monthly stats are simply mind blowing.

Things that this app has while others did not:
• Breathing stuff as we can regulate breathing exercise by this application.

Glitches that the app has:
• The heartbeat is not that consistent as expected. Sometimes it gives very large resting heartbeat.
• It shows the heartbeat even I put some things like red cloth or something else.

App 2: Cardiograph - Heart Rate Meter

This app is interesting because :
• It is easy to use.
• The reading are almost accurate.
• Profile option is available. Multiple profiles can be created and records can be stored separately.
• Their is record storing option available. Also, the records of different users are stored separately.

Owner and Research Paper related: This app is offered by MacroPinch (a company formed by industry veterans). This app was mentioned in article in Journal of Innovation in Health Informatics . The article name is "Reliability of heart rate mobile apps in young healthy adults: exploratory study and research directions".

Uniqueness of the app:
• This app has a unique feature that,it has multiple profile feature. We can create as many profiles as we want. The records of different person will be saved in different locations. We just neeed to simply switch between the users and choose our profile and then start whatever we wanted to do.
• When saving the measurement, we can associate it with the type of activity . It shows some actvities like : Rest, Warm up, Cardio and Xtreme with their logo. That seems attractive.

Things that this app has while others did not:
• very less latency and good sensing capability.
•

Glitches that the app has:
• The heartbeat is not that consistent as expected. Sometimes it gives very large resting heartbeat.
• Sometimes, it feels like , it has large latency than expected. When I removed my finger from the camera, then it still keeps on increasing/decreasing fluctuating, rather than fall down to 0, and then also happens with latency.
• It shows the heartbeat even I put some things like red cloth or something else.

App 3: Heart Rate Monitor: Pulse Checker & Step Counter

This app is interesting because :
• Its interface is simple and attractive.
• It has some labels (icons) showing the status of the activity(like resting, stretching, standing, exercise). We can simply check them and then continue with the bpm measurement.
It has statistics corner, where it shows the stats of the measurement we had taken. There it also displays the average BPM , the minimumBPM and the maximum BPM .

Owner and Research Paper related: This app is owned by MobilMinds Studios(Android developer active since 2020).

Uniqueness of the app:
• It has some feature like unique way of presenting stats, also it has also pedometer feature apart from hear rate measurement.

Things that this app has while others did not:
• The statistics part of this app is nice. It shows the average ,minimum , maximum BPM which none of the other app shows.

Glitches that the app has:
• The readings it gave are very much high than expected.Even at resting state, it was showing above 100 BPM.
• The sensing also does not seem to be that smooth. Their seems to be latency.
• IT also shows the heart beat if some object is put in front of camera.

App 4: Welltory - ECG Heart Rate Monitor and Stress Check

This app is interesting because :
• Its interface is good.
• There are lots of feature in this app , like It can take data from other apps like samsun health, google fit, rescue time, fitbit, Garmin and show your health results in Welltory.
• The graph analysis after every meaure of heartbeat is good. As they appears immediately after the measurement.
• Measurement History is managed amazingly.
• It also suggest Recommendation of some other fitness apps.

Owner and Research Paper related: This app is owned by Weltory team. This app was mentioned in article in medium as "How Welltory's Phone Camera Measurements Stack Up Against Professional Heart Rate Monitors".

Uniqueness of the app:
• This app has lot of unique things. Like they provide stress checker along with bpm measurement, the storage of records are managed effectively, the recommendataion of other health fitness app is simply awesome and the link with other fitness apps to take data from their sources is good.

Things that this app has while others did not:
• There are lots of things, it if fast responding, sensor responding well, it seems to be more robust of fluctuating in the bpm.

Performance of the app:
• It seems to be the more robust app among all the apps I mentioned here or checked . It is more robust.
• It was robust to measure the heartbeat when a cloth(let say red in color) is put in front of the camera. Most of the time , it was not giving heartbeat measure.
• It has less latency.
• Sensing Capability is just awesome.
• Also, it is stable , as it give final stable result after taking some time (the bar moves to 100 %).

App 5: Instant Heart Rate

This app is interesting because :
- It is an Indian app.
- It is simple.
- Apart from the bpm, it has some additional feature of bmi . Hence Body mass index can also be calculated using it.
- It has the bpm vs day chart for every month. So, we can track our heartbeat day by day, month by month.

Owner and Research Paper related: This app is owned by Igost Technologies (a Mobile Application and Game Development company that specializes in developing Android, iOS,).

Uniqueness of the app:
- BMI feature other than BPM.
- Monthly stats records storing and the way they are presented is good. i.e good visualization.

Things that this app has while others did not:
- BMI feature is an addition in the app.

Glitches that the app has:
- The heartbeat is not that consistent as expected.
- It shows the heartbeat even I put some objects like red cloth or something else in front of the camera.

Question 2

Task 2: Generating test video/datasets [MAX SCORE: 25 points]

Use your smartphone's camera to capture the blood-flow video. Turn on the flash while recording and turn it off once done. Capture the video using the maximum frame rate that is available in your phone (e.g., 60 fps). The same holds for image resolution. The captured videos are saved on the device's SD card. (***Don't move your finger/hand randomly or press too hard against the camera or flash while recording***.)

Copy the video on to your computer and post process the video using your custom application (use C/C++/Python). Set configuration parameters for the processed video: Frames per second Resolution or Dimension.

Solution

Solution:
The code is included in the file.

Question 3

Task 3: Sensing Algorithm [MAX SCORE: 25 points]
Input -> Proceeded Video
Output -> BPM (sliding window), reported say every 5 seconds.
Remove noise, make attempts to make the algorithm stable and robust so that it has the same desired output for data collected under various conditions.

(a) **Time-series data** [15 points]
Find the BPM using your algorithm that leverages the time series of the RGB pixels.

(b) **Use DFT** [10 points]
Use a fourier transform to obtain the pulse frequency directly from the time series.

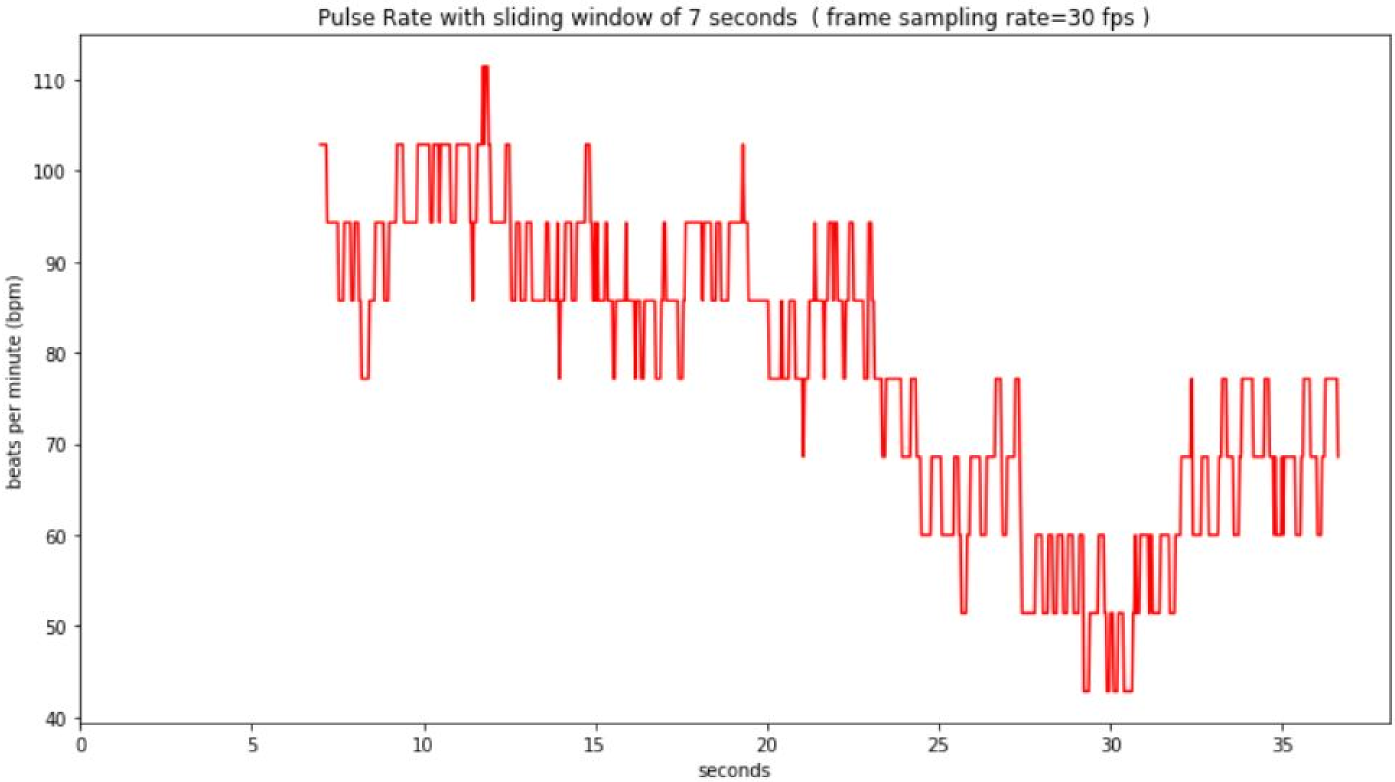
Solution

Solution:
The code is in the file. The results are written below:

The parameters taken are as:
1. SLiding window size = 7 seconds
2. Standard frame rate = 30 fps (frame rate of my smartphone)
3. Window size for running mean for smoothing of intensity of frames : 4 frames

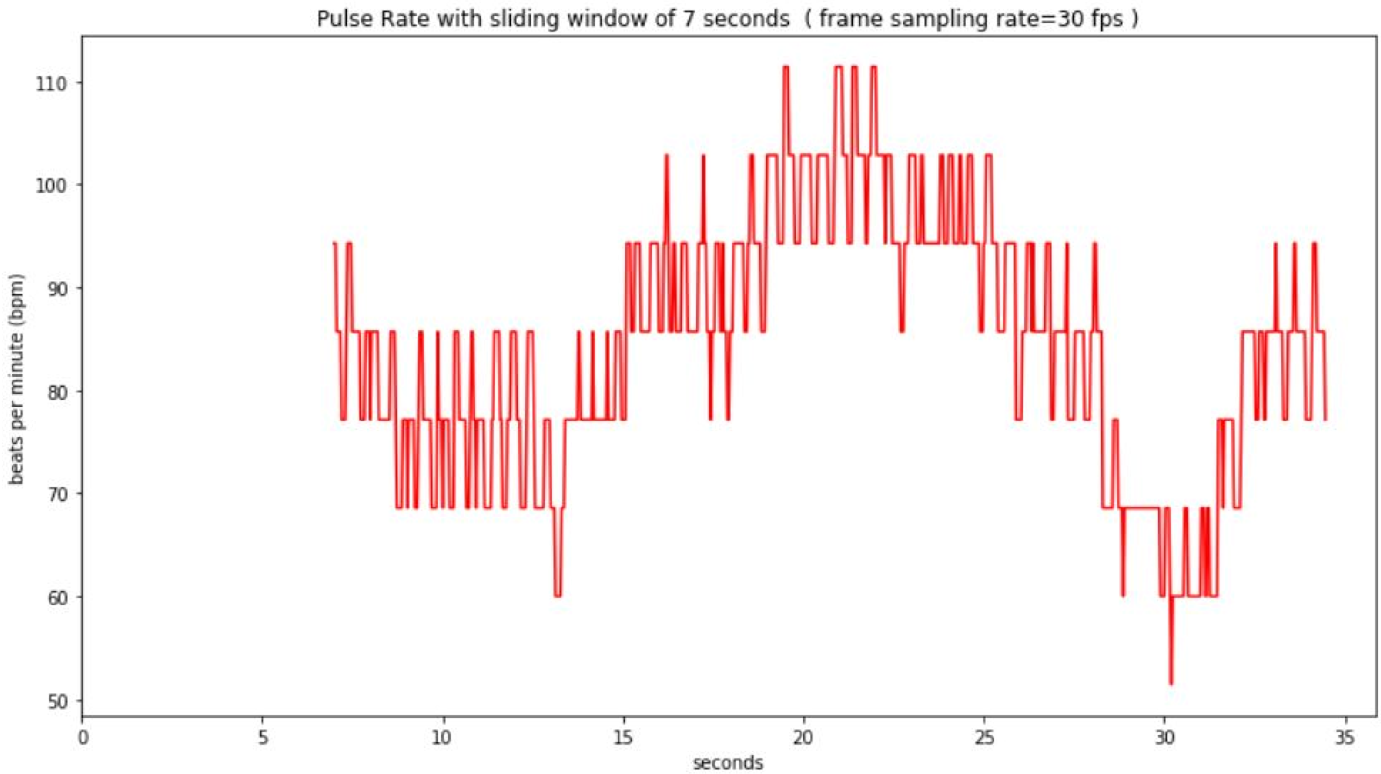
(a) Time - series data

The results of the bpm of different activities and lights are as follows :
1. Normal Resting condition in a dark room :



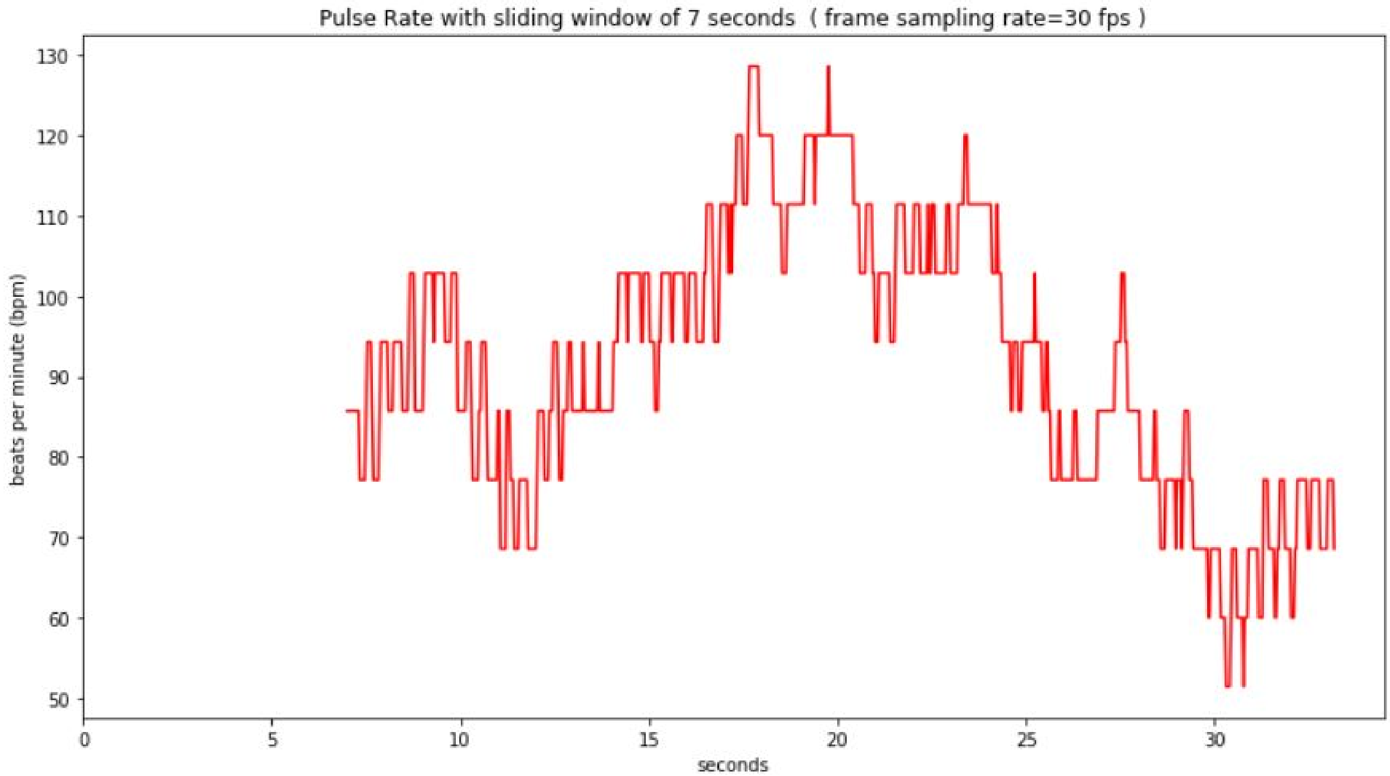
The average pulse rate (in BPM) by time series analysis 77.84590690208657

2. Normal Resting condition in a Led(7W) lighted room :



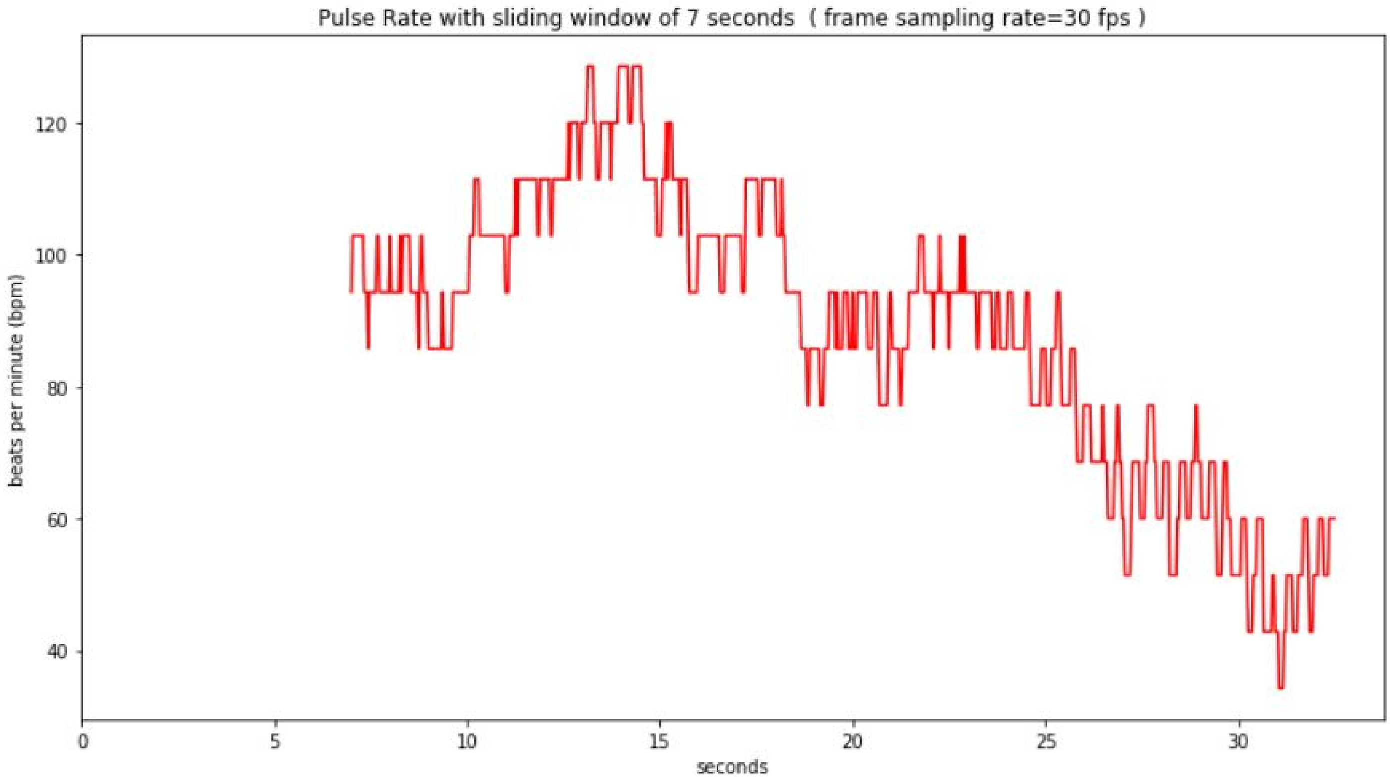
The average pulse rate (in BPM) by time series analysis 84.43636363636386

3. Push up exercise in a dark room :



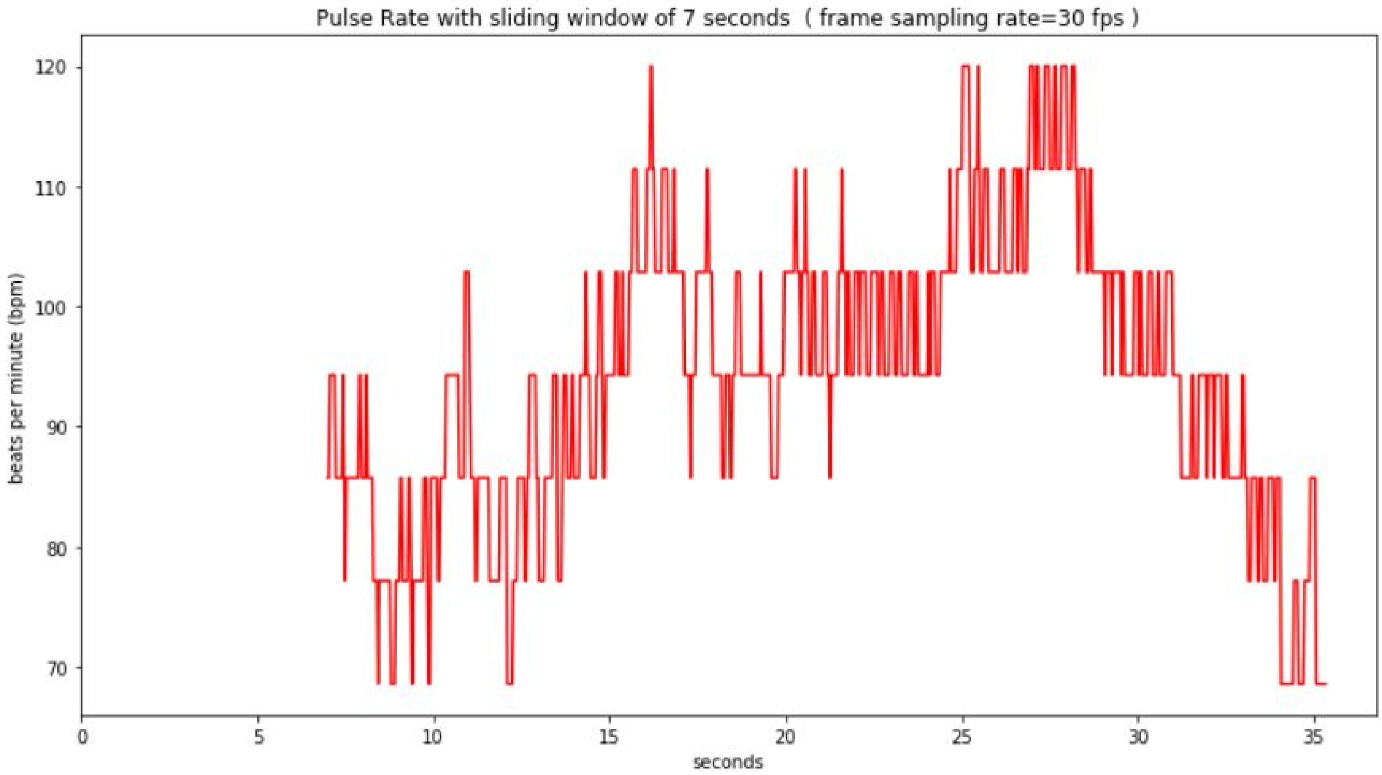

```
The average pulse rate (in BPM) by time series analysis  91.89749182115615
```

4. Push up exercise in a Led(7W) lighted room :



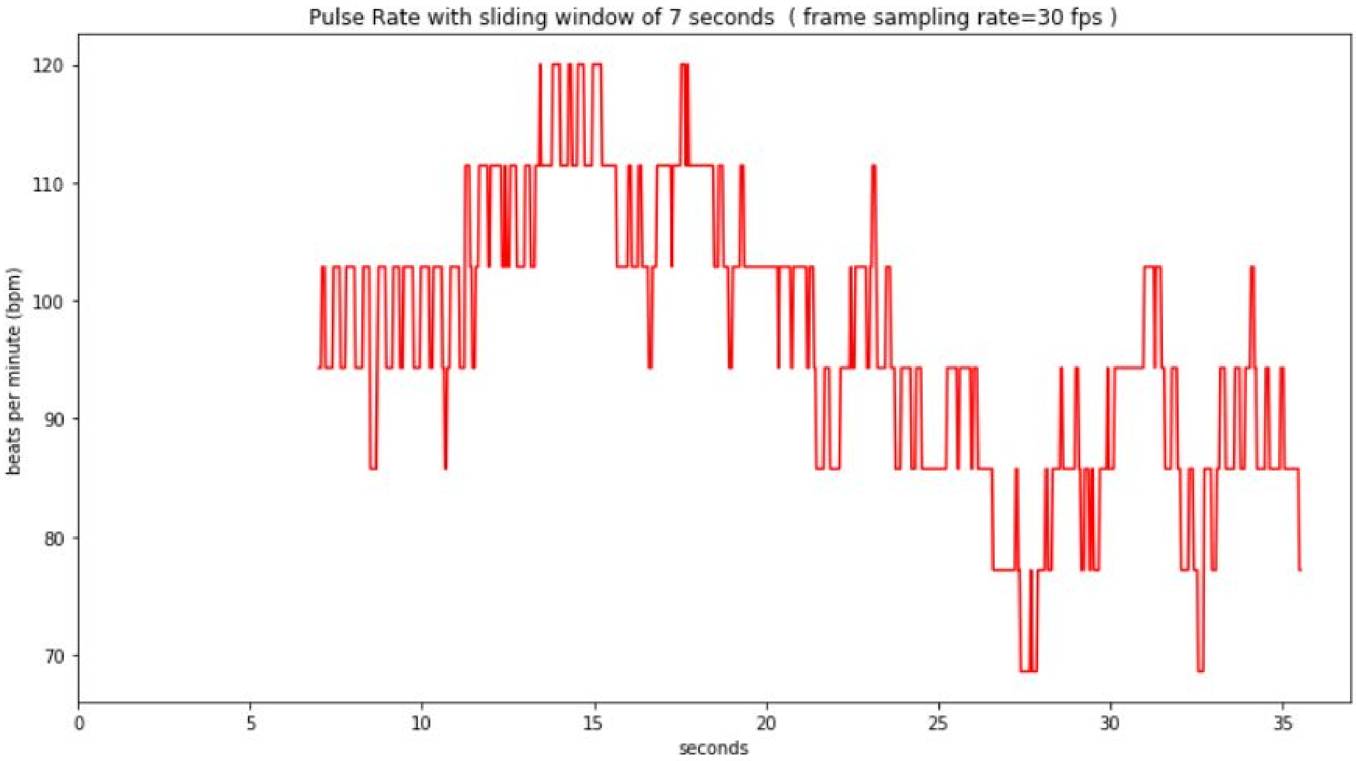

```
The average pulse rate (in BPM) by time series analysis  88.59007832898152
```

5. Crunches exercise in a dark room :



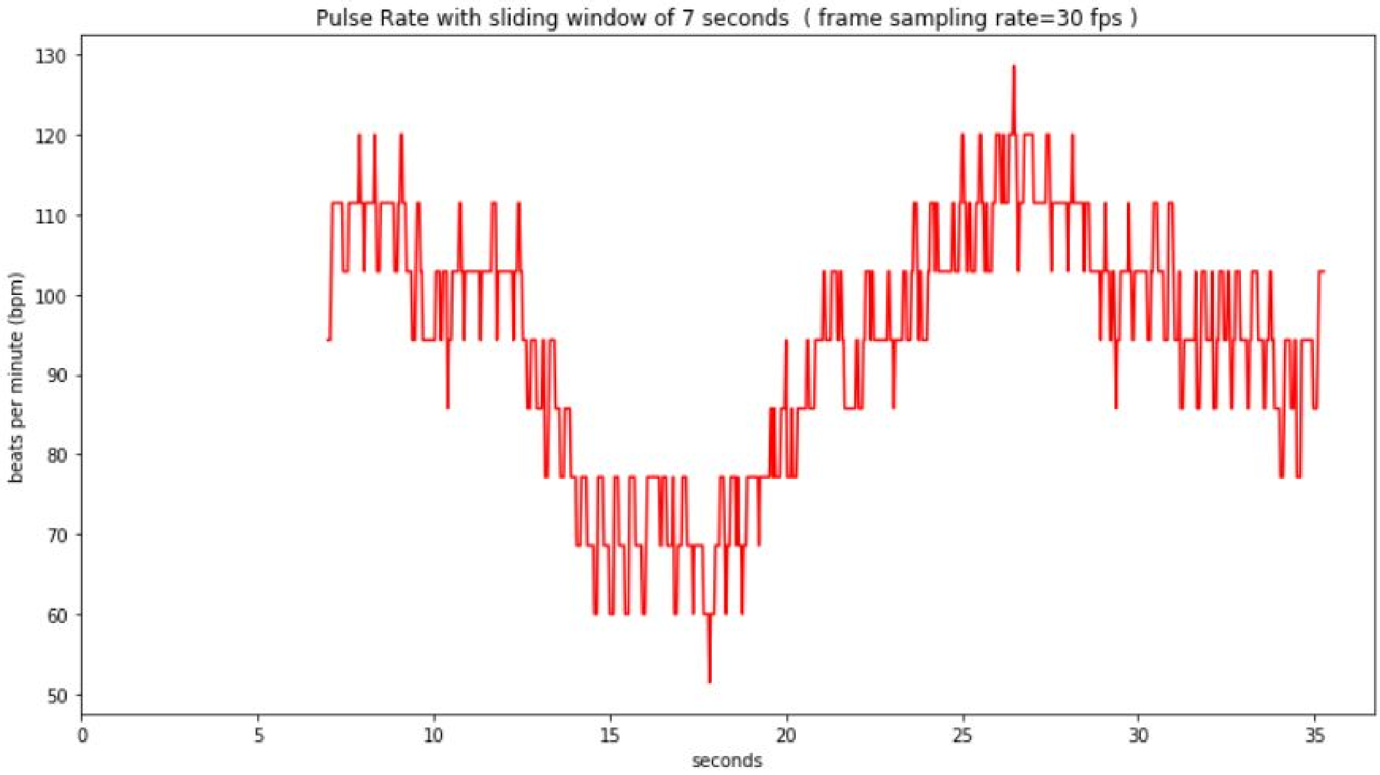

```
The average pulse rate (in BPM) by time series analysis  94.31593083767076
```

6. Crunches exercise in a Led(7W) lighted room :



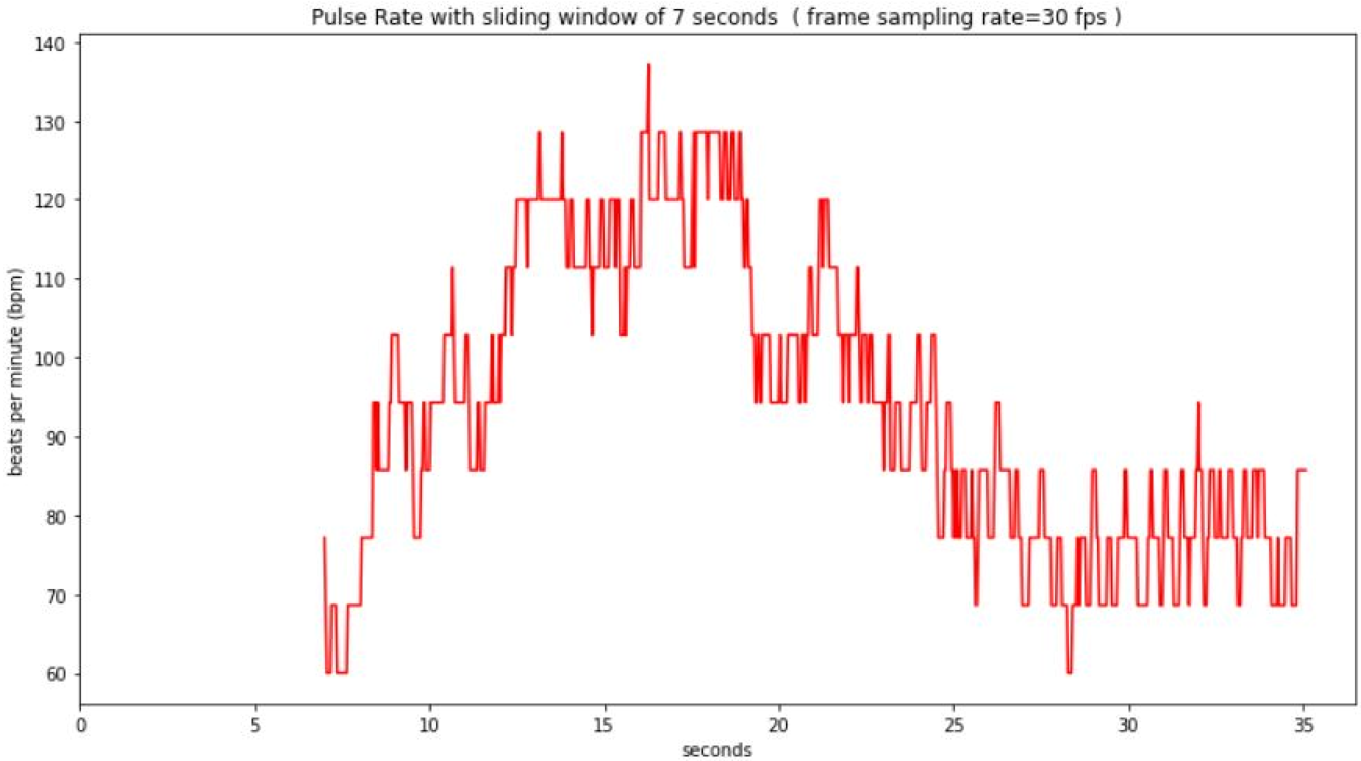

```
The average pulse rate (in BPM) by time series analysis  96.66611101850336
```

7. Normal Resting condition in sunlight :



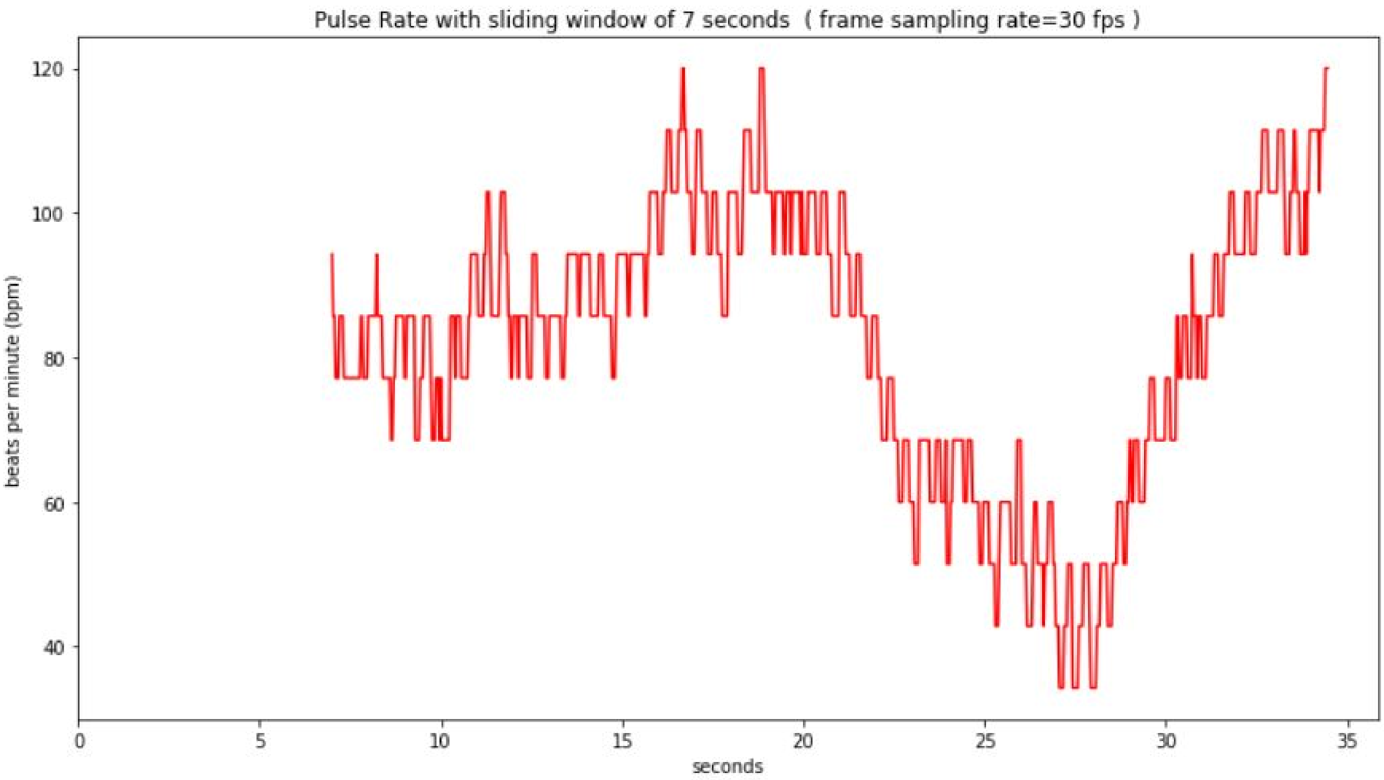
The average pulse rate (in BPM) by time series analysis 94.20494699646669

8. Exercise - Arm Rotation in a dark room :



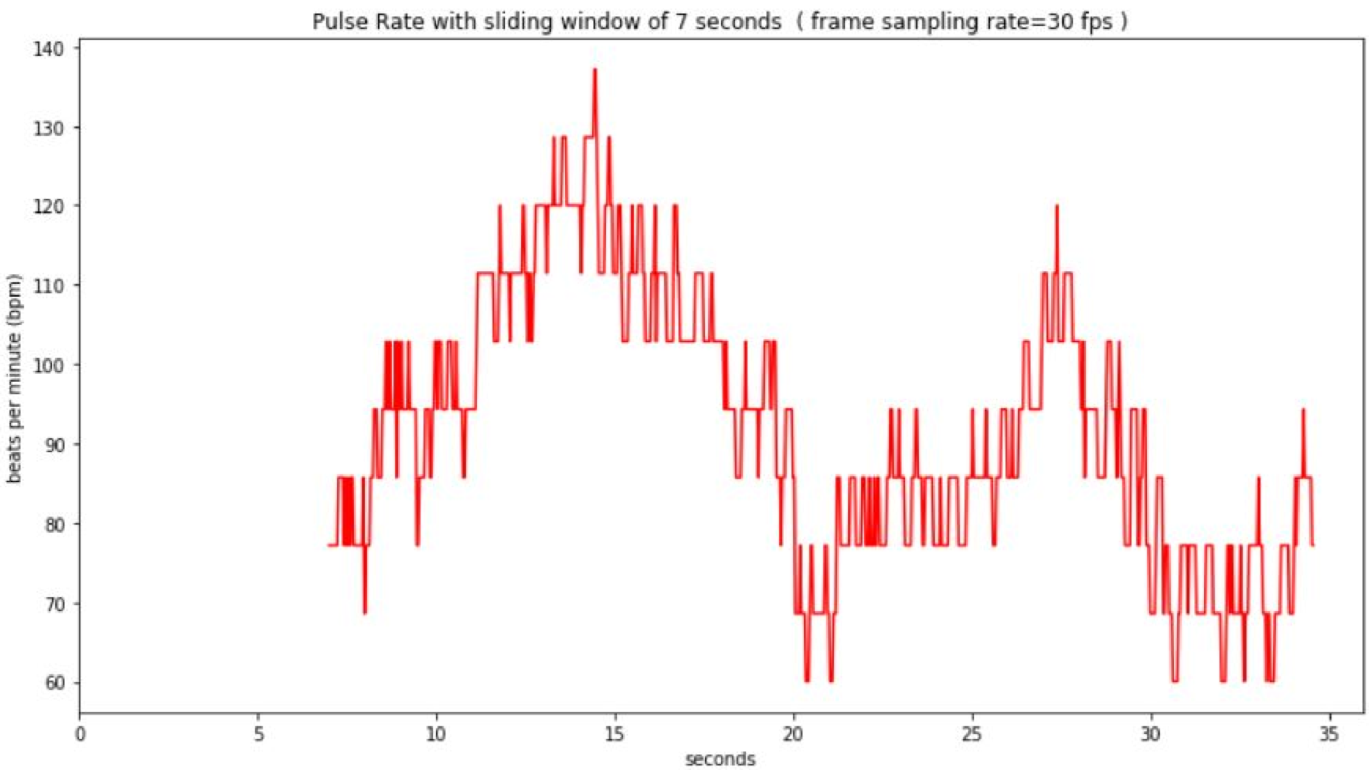
The average pulse rate (in BPM) by time series analysis 93.69598373157132

9. Exercise - Arm Rotation in a Led(7W) lighteed room :




```
The average pulse rate (in BPM) by time series analysis  82.12987012987021
```

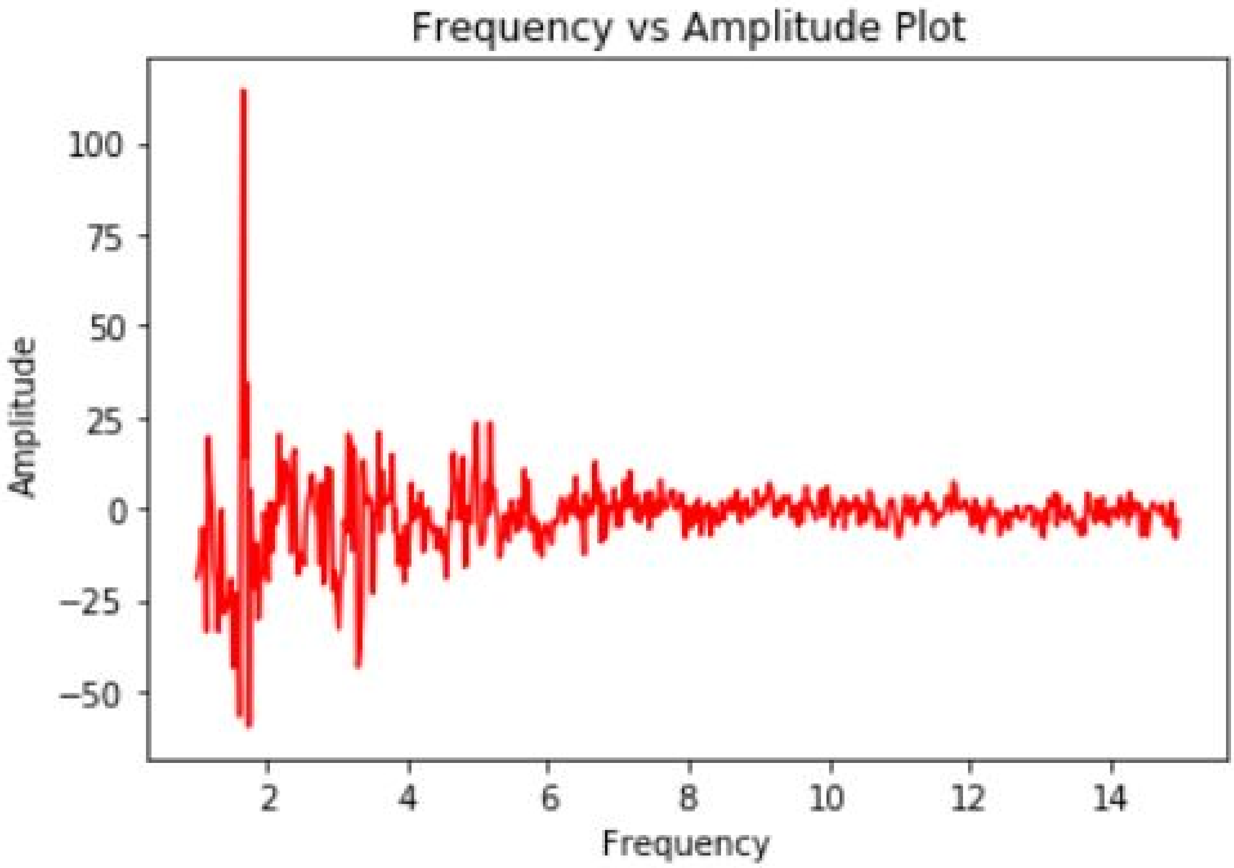
10. Waking after a 6 hours sleep :




```
The average pulse rate (in BPM) by time series analysis  91.84265010351983
```

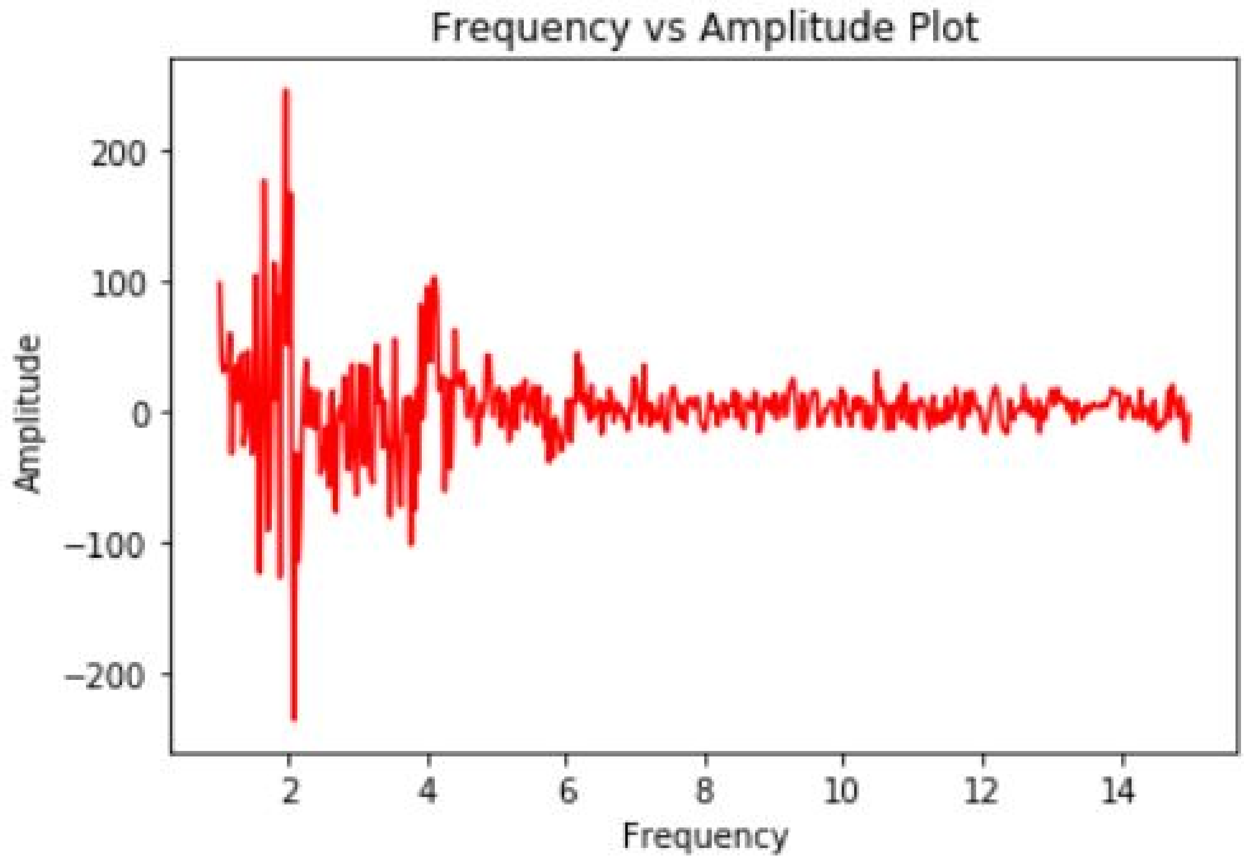
(b) Using DFT

1. Normal Resting condition in a dark room :



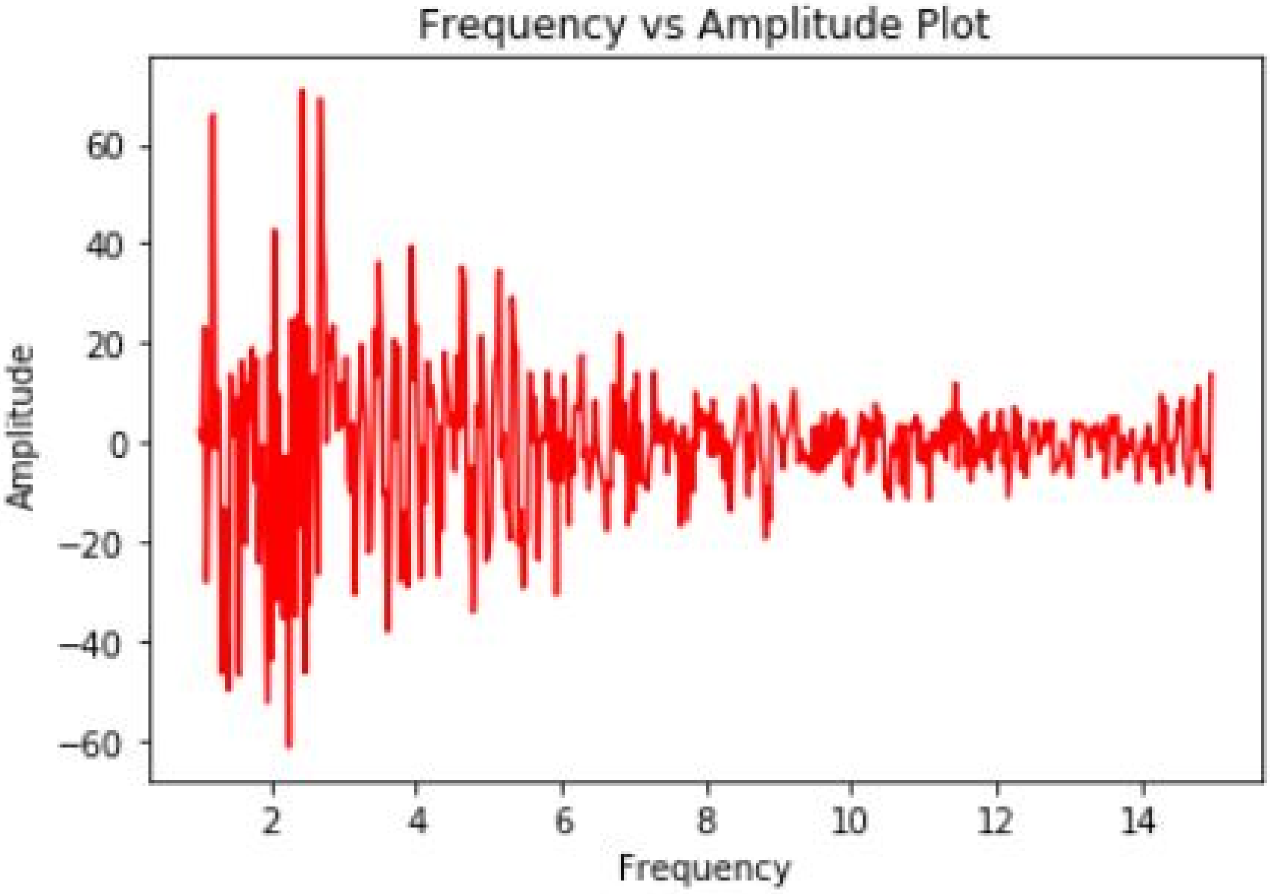

```
The pulse rate  in BPM (by DFT) is:  99.81818181818183
```

2. Normal Resting condition in a Led(7W) lighted room :



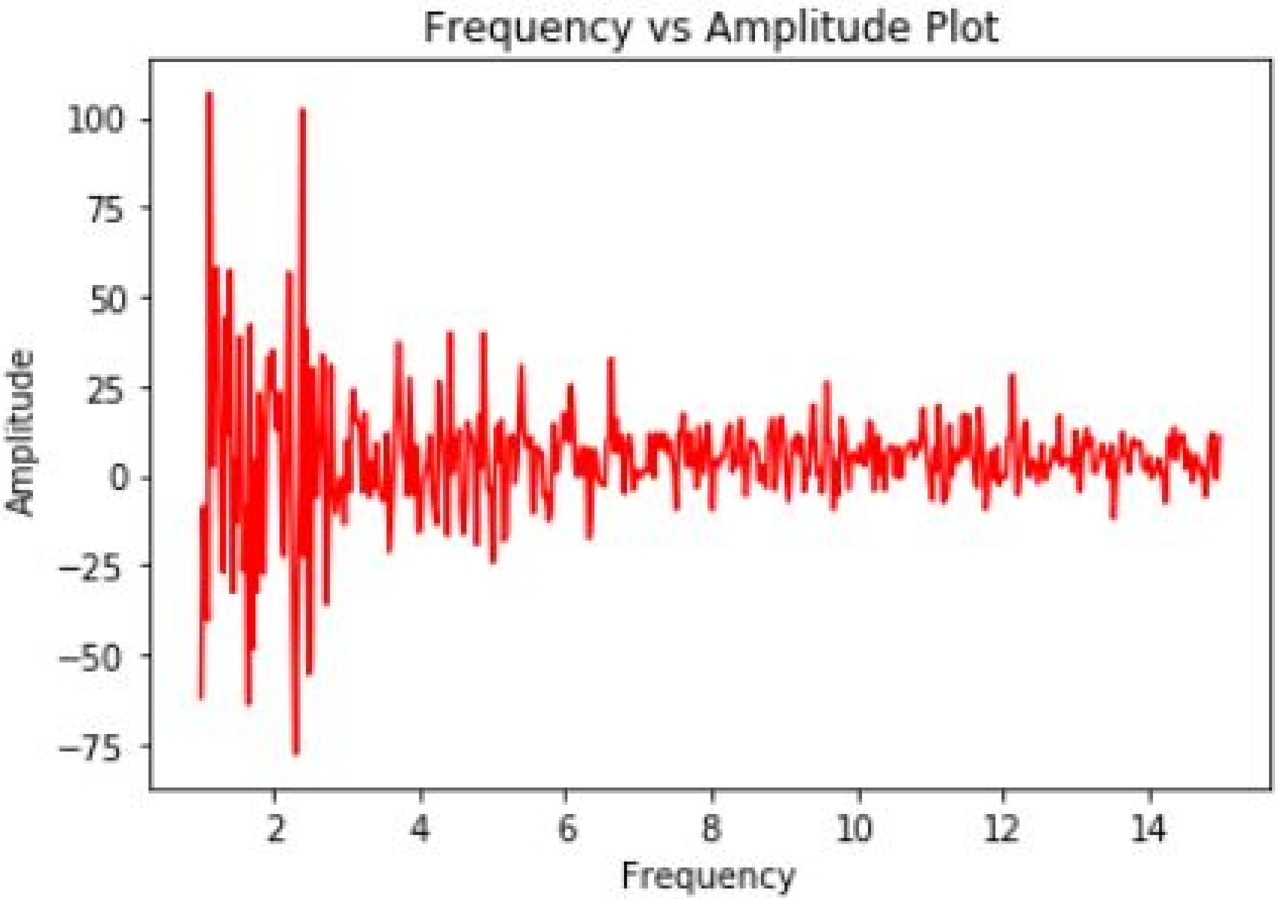
The pulse rate in BPM (by DFT) is: 118.2608695652173

3. Push up exercise in a dark room :



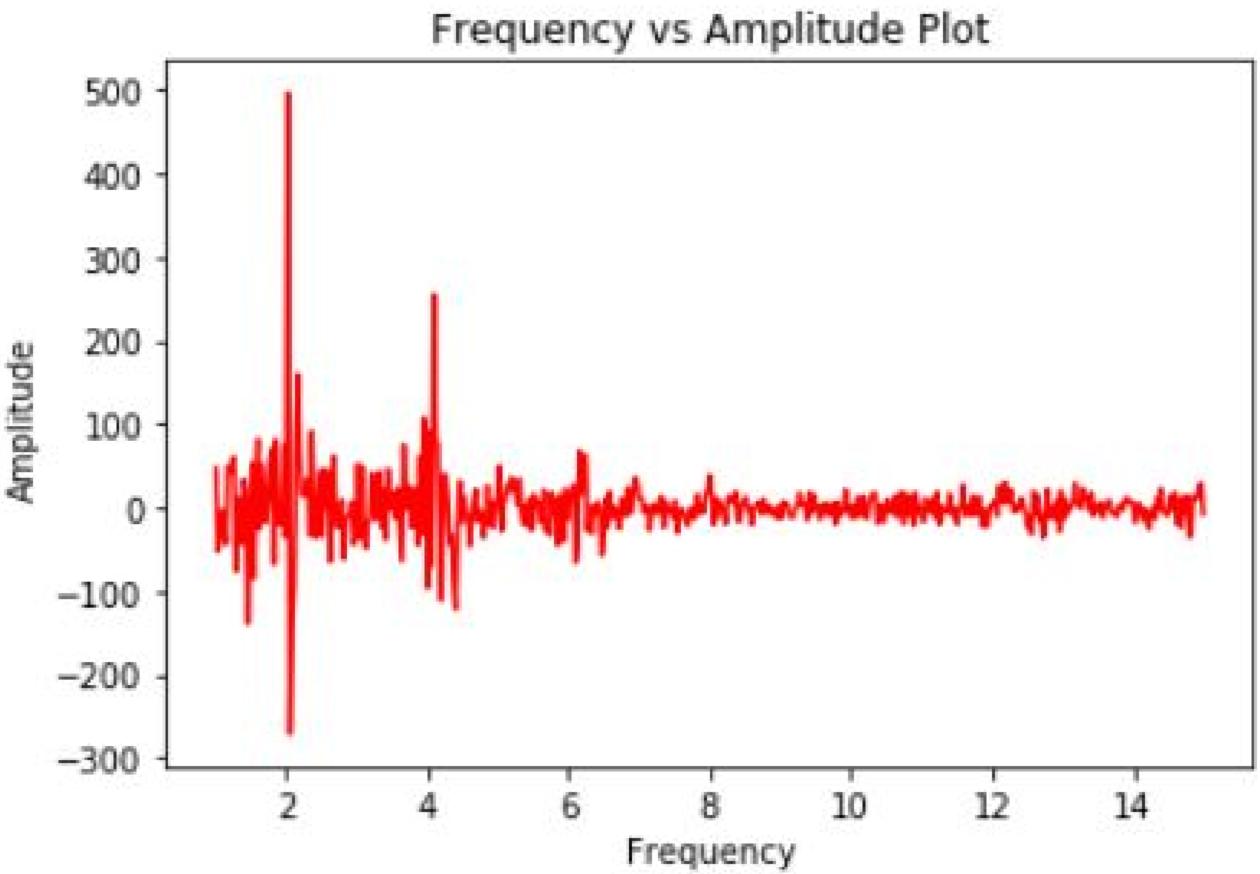
The pulse rate in BPM (by DFT) is: 146.3855421686747

4. Push up exercise in a Led(7W) lighted room :



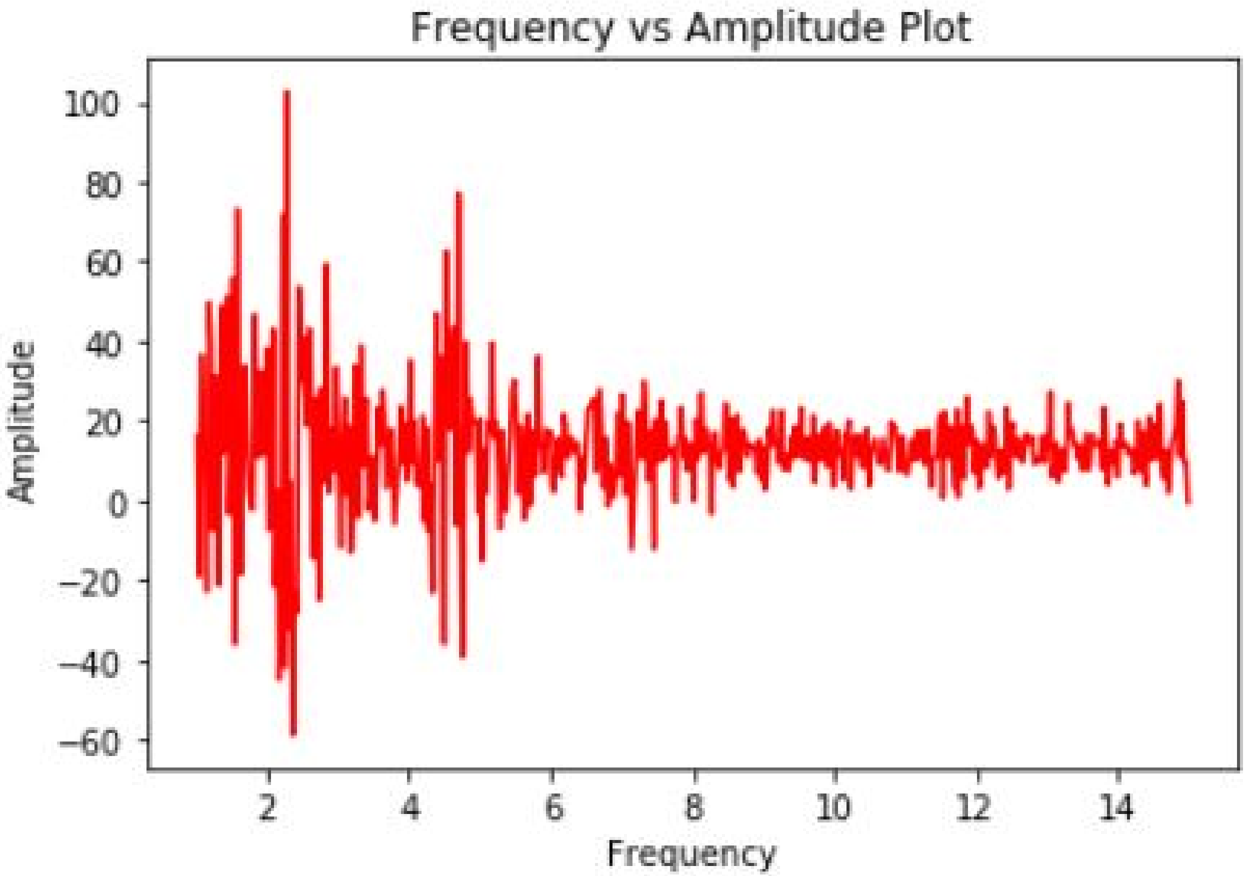

```
The pulse rate  in BPM (by DFT) is:  68.23770491803279
```

5. Crunches exercise in a dark room :



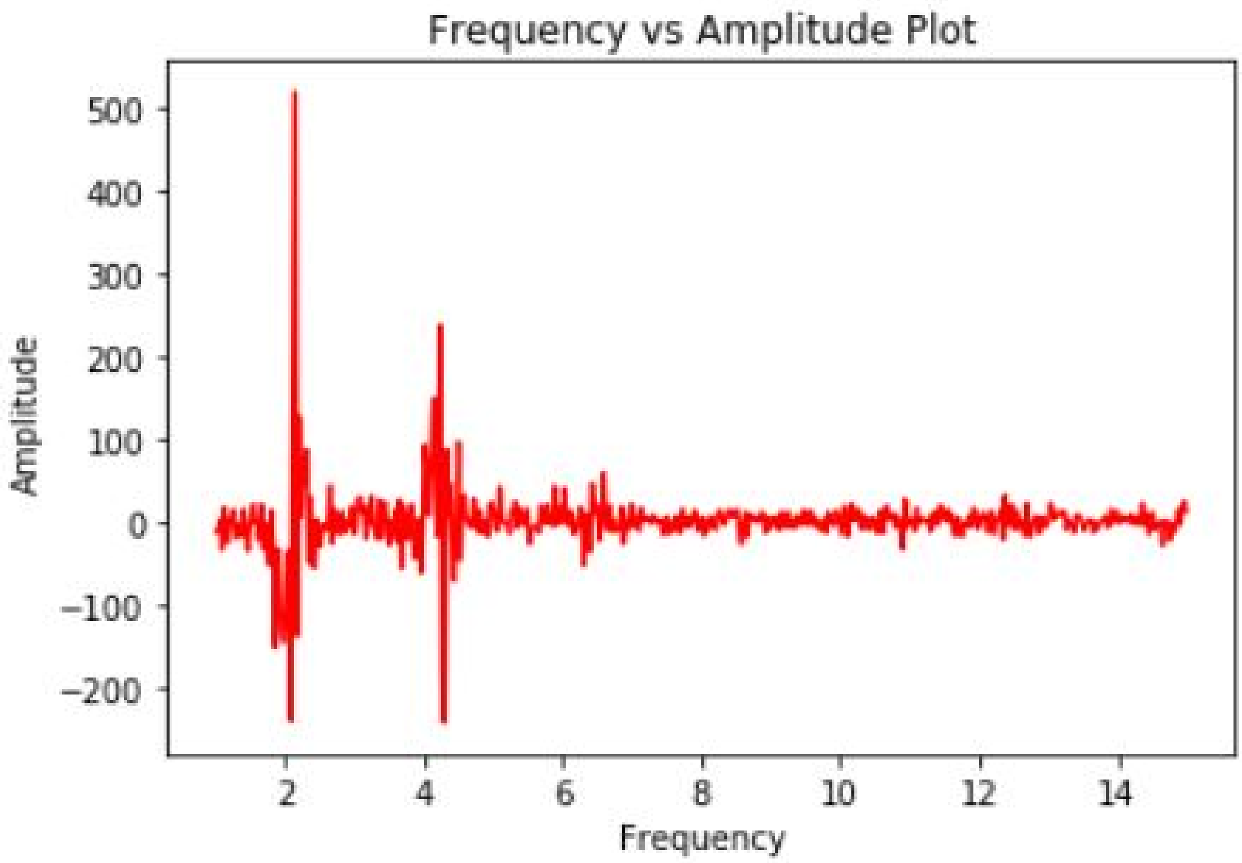
The pulse rate in BPM (by DFT) is: 122.14891611687088

6. Crunches exercise in a Led(7W) lighted room :



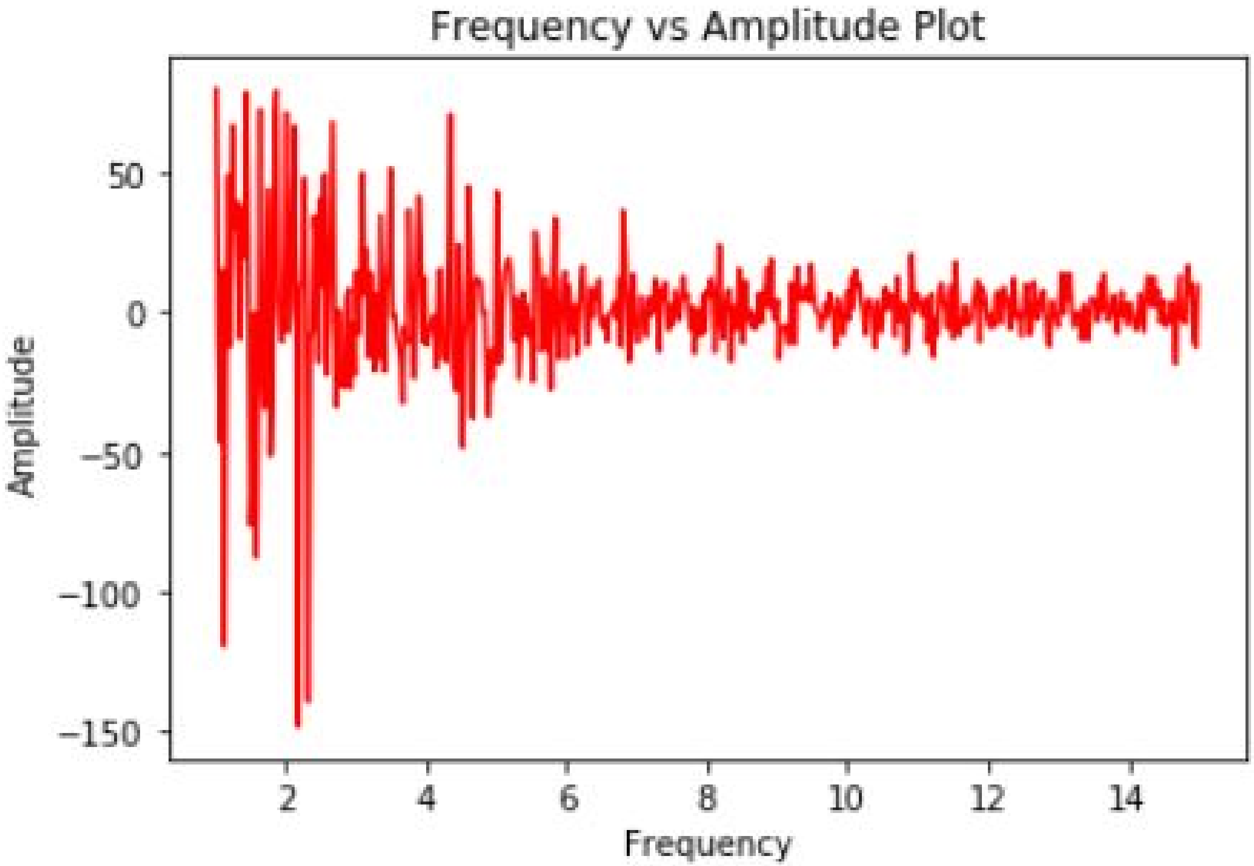
The pulse rate in BPM (by DFT) is: 136.6447985004686

7. Normal Resting condition in sunlight :



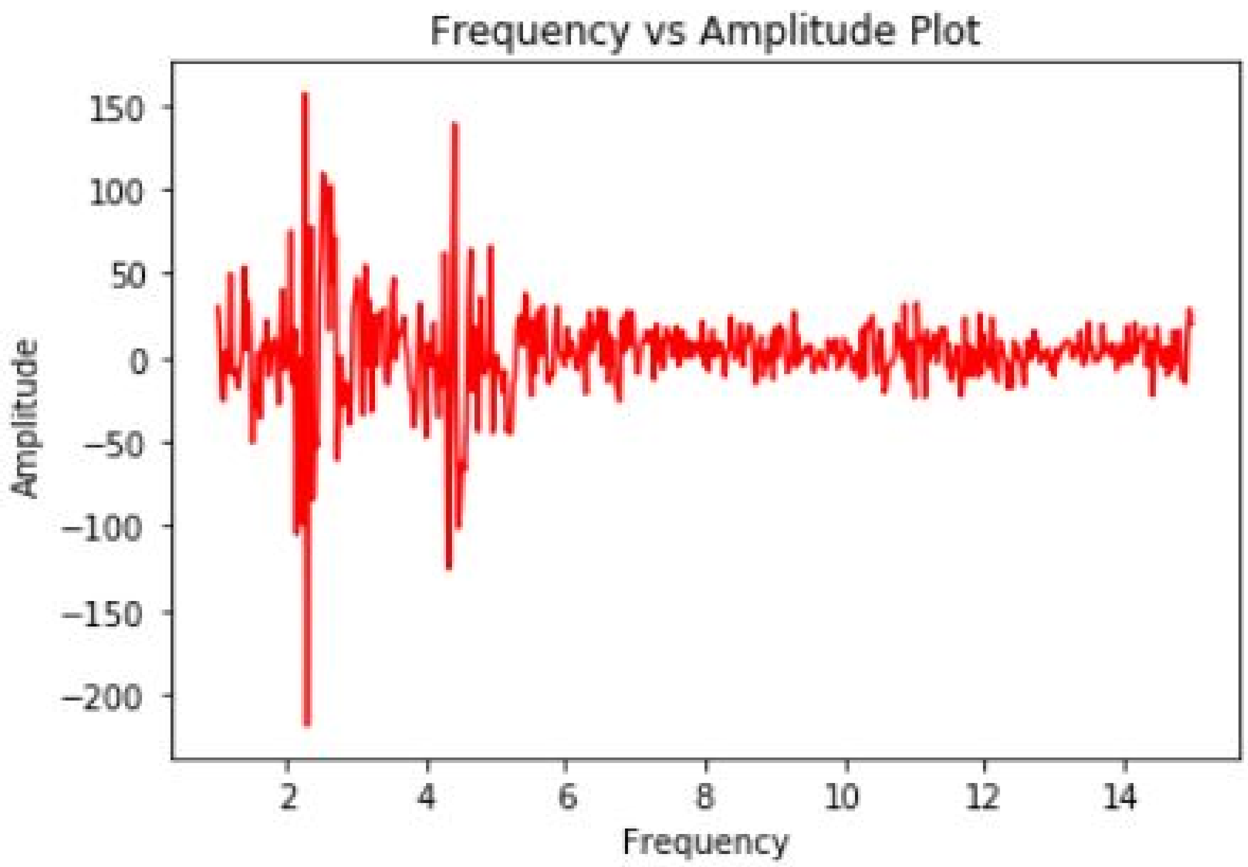

```
The pulse rate  in BPM (by DFT) is:  129.17847025495752
```

8. Exercise - Arm Rotation in a dark room :



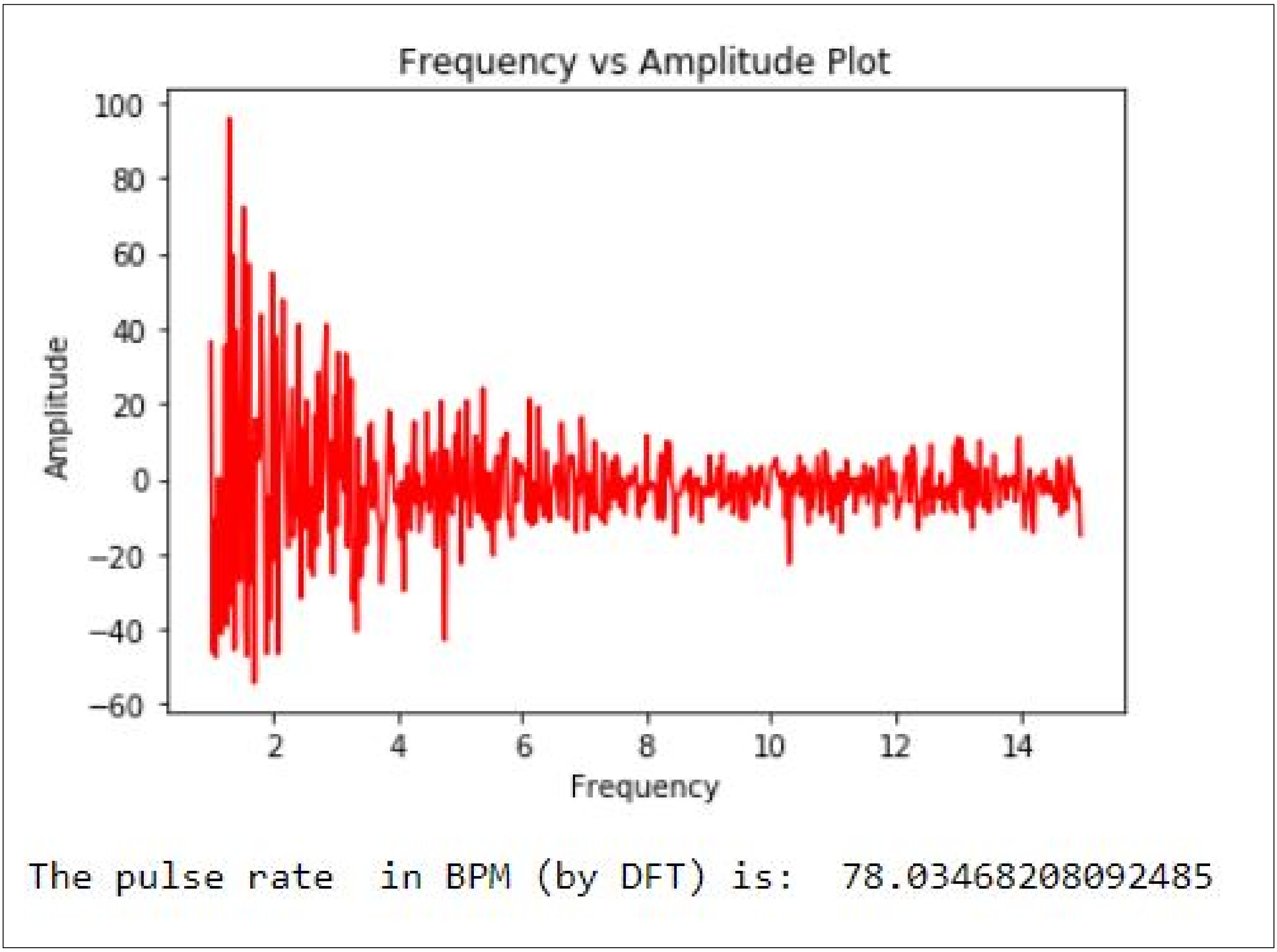

```
The pulse rate  in BPM (by DFT) is:  61.53846153846155
```

9. Exercise - Arm Rotation in a Led(7W) lighteed room :




```
The pulse rate  in BPM (by DFT) is:  135.6521739130435
```

10. Waking after a 6 hours sleep :



Question 4

Task 4: Evaluate the performance of your algorithm [MAX SCORE: 35 points]

Change the frame rate of the video (choose 1 every N frames, to downsample).
Similarly, change the resolution of the video (choose 1 every K pixels along X and Y axis).

Run your algorithm again on this video.

a. How much can you come down in terms of FPS, any thoughts on this/why? Show whether it works/doesn't work. [5 points]

b. What about low resolution? Any portions of the video that do not change? Remove them to bump up the SNR? Describe your technique. [15 points]

c. Is there a trade-off involved here? Low resolution vs Low FPS? Discuss based on your data. [15 points]

Draw conclusions using a data set (at least 10 original videos, each about 30 seconds long)

Solution

Solution:
(a) How much I can come down in term of FPS

Theoretical Explaination

Frame rate of the video recoded from my smartphone = 30fps
The bpm of a human being can range from 40 bpm to 200 bpm .
$\implies 40 \leq \text{bpm} \leq 200$
$\implies \frac{40}{60} \leq \text{Beat per second} \leq \frac{200}{60}$
$\implies 0.66 \leq \text{Beat per second} \leq 3.33$
Maximum frequency that could be present = 3.33 Hz

Using Shannon Nyquist sampling rate ,to avoid aliasing
sampling sampling frequency $\geq$ 2 * Maximum frequency component in the signal

$\implies$ Sampling frequency $\geq 2 * 3.33Hz$
$\implies$ Sampling frequency $\geq 6.66Hz$
$\implies$ Minimum fps possible = 7

Experimental explaination:
Pulse rate with different frame sampling rates and with different activities

Note: standard frame rate = 30fps :

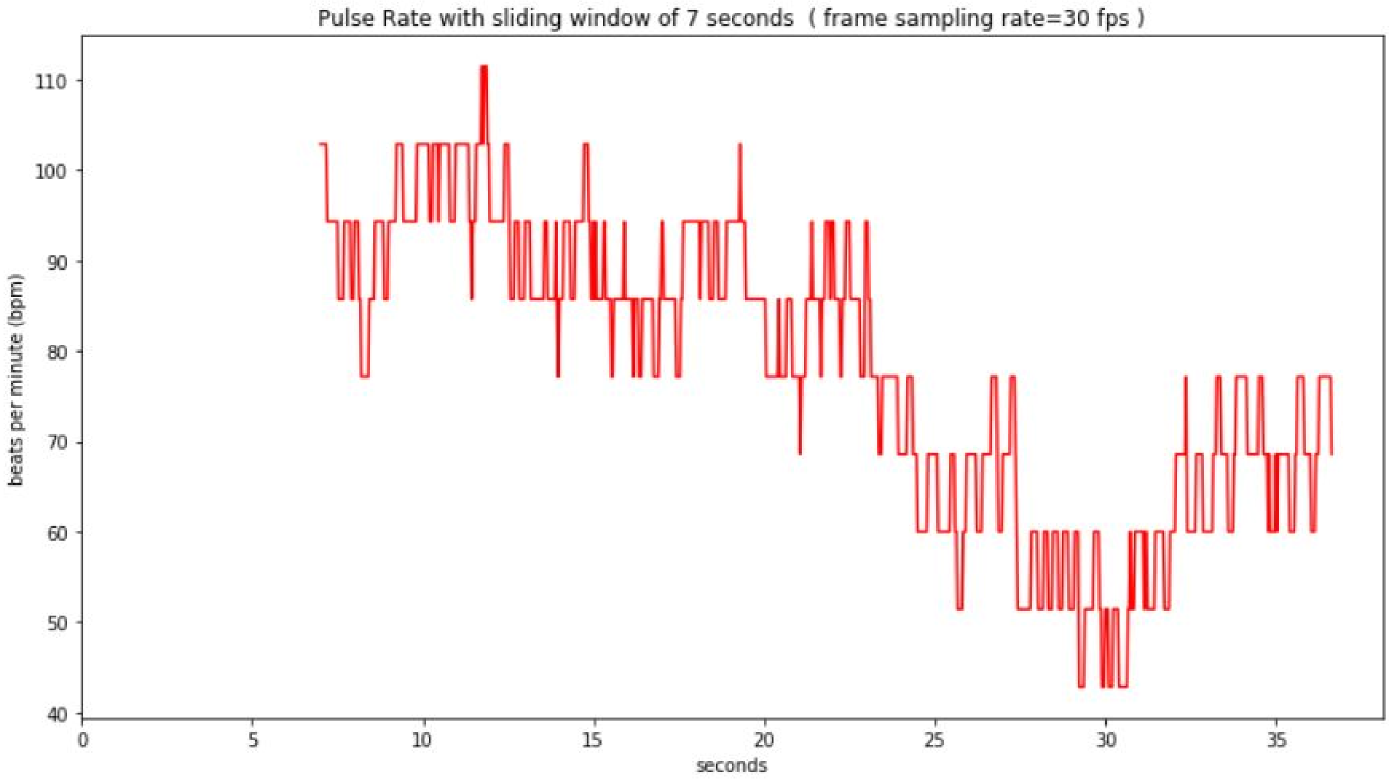
Activity : Resting mode (In the dark

- **Video name : "resting_dark.mp4"**
- **Video duration : "00:00:46" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)**
- **Sliding window used for bpm = 7 sec (last 7 seconds)**

(Every pixel of a frame is taken into consideration)

1. **Frame sample rate : 30 fps**

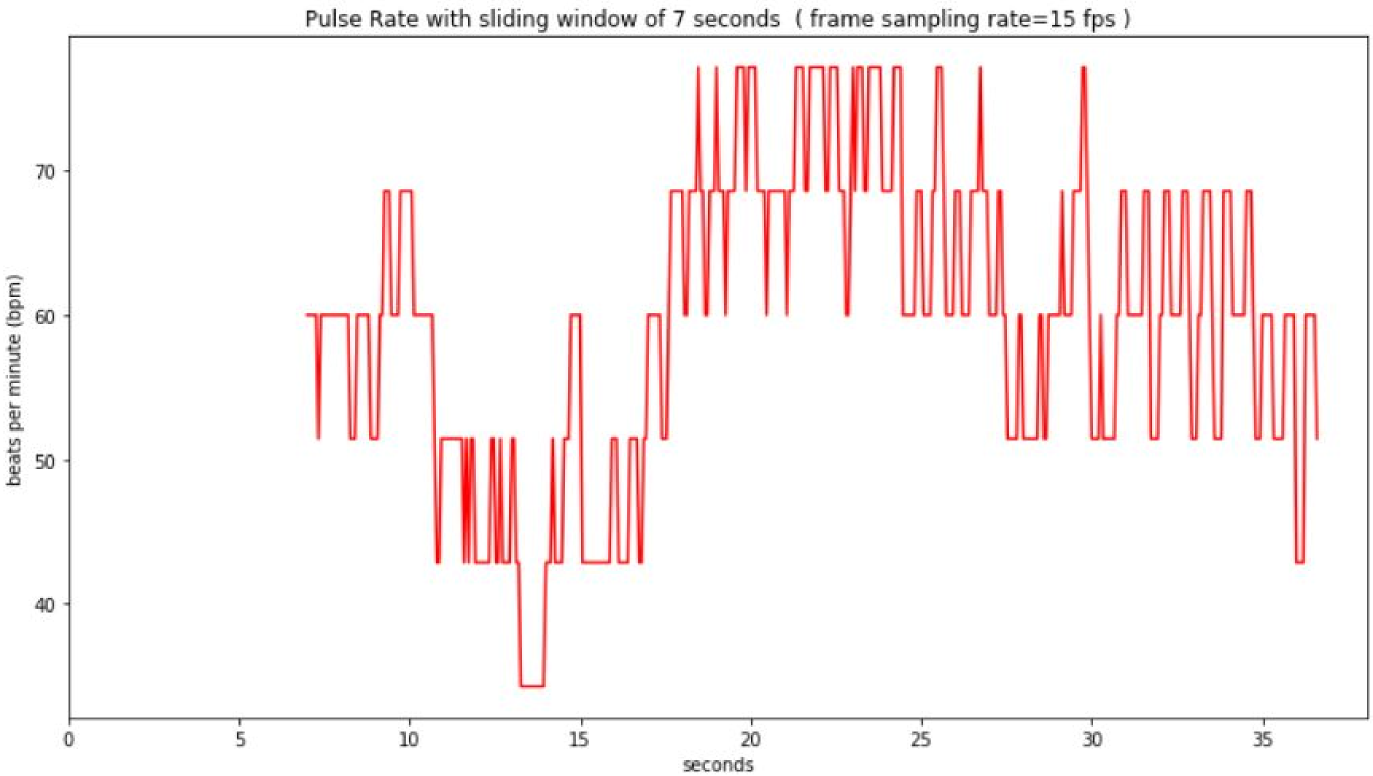
- **Average pulse rate = 77.84590690208657 bpm**




```
The average pulse rate (in BPM) by time series analysis  77.84590690208657
```

1. **Frame sample rate : 15 fps**

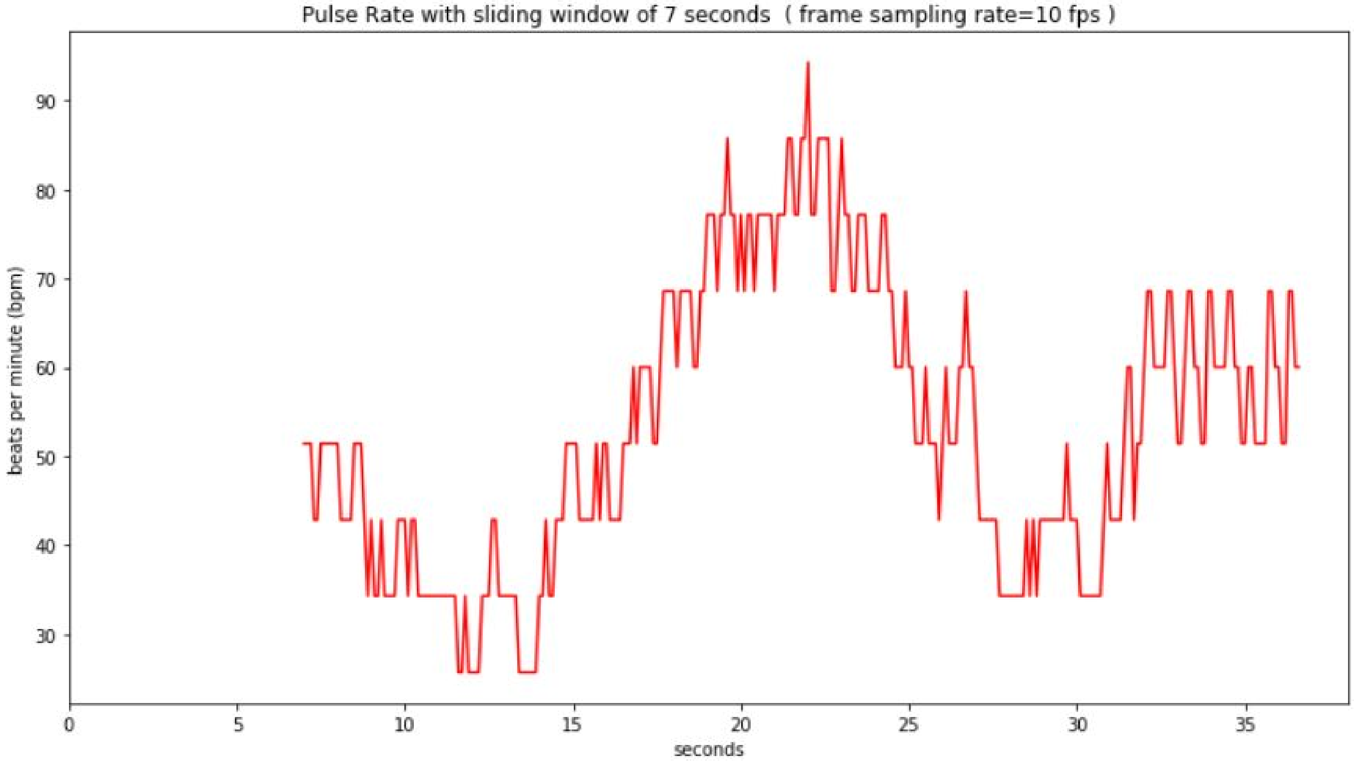
- **Average pulse rate = 59.61476725521682 bpm**




```
The average pulse rate (in BPM) by time series analysis  59.61476725521682
```

1. **Frame sample rate : 10 fps**

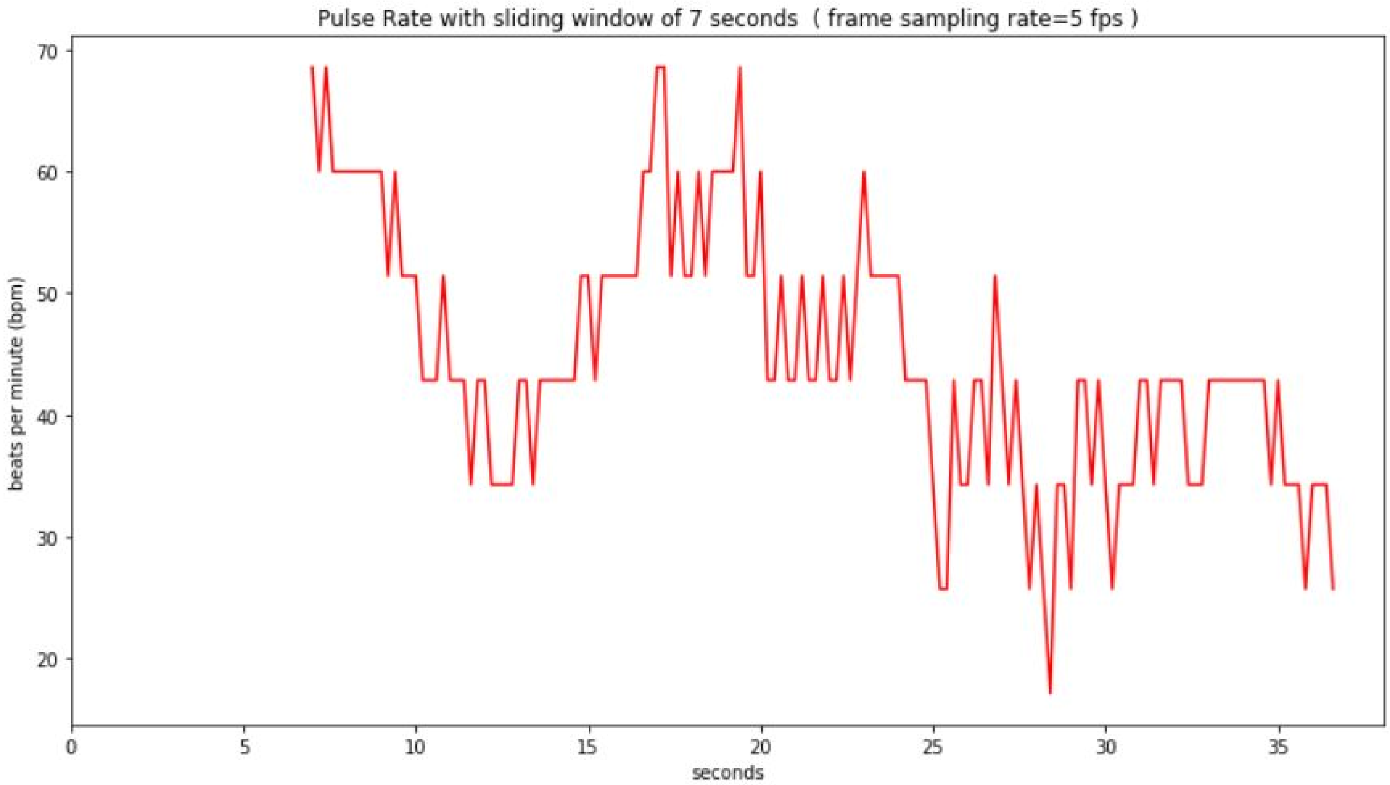
- **Average pulse rate = 53.33333 bpm**




```
The average pulse rate (in BPM) by time series analysis  53.33333333333327
```

1. **Frame sample rate : 5 fps**

- **Average pulse rate = 44.8705656759349 bpm**




```
The average pulse rate (in BPM) by time series analysis  44.8705656759349
```

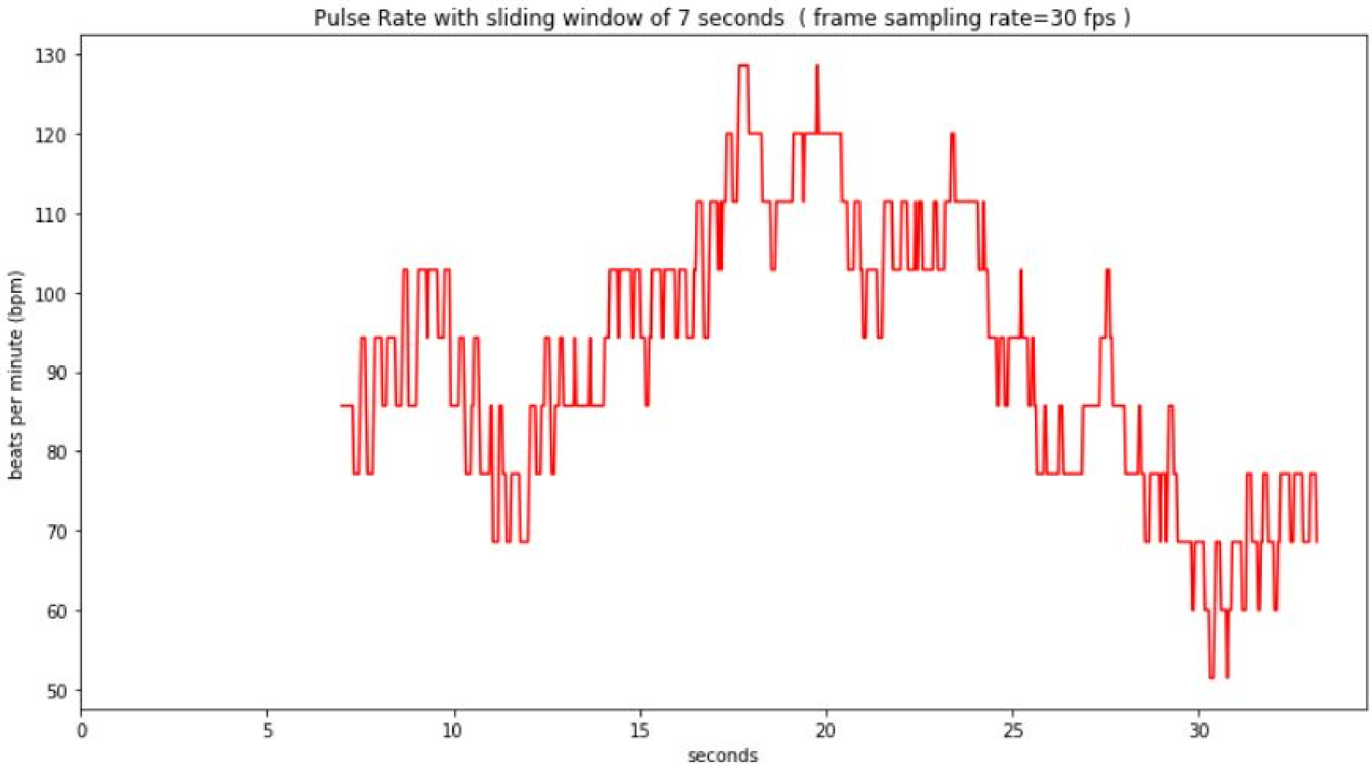
Activity : exercise-push up (In the dark)

- **Video name : "pushup_dark.mp4"**
- **Video duration : "00:00:43" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)**
- **Sliding window used for bpm = 7 sec (last 7 seconds)**

(Every pixel of a frame is taken into consideration)

1. **Frame sample rate : 30 fps**

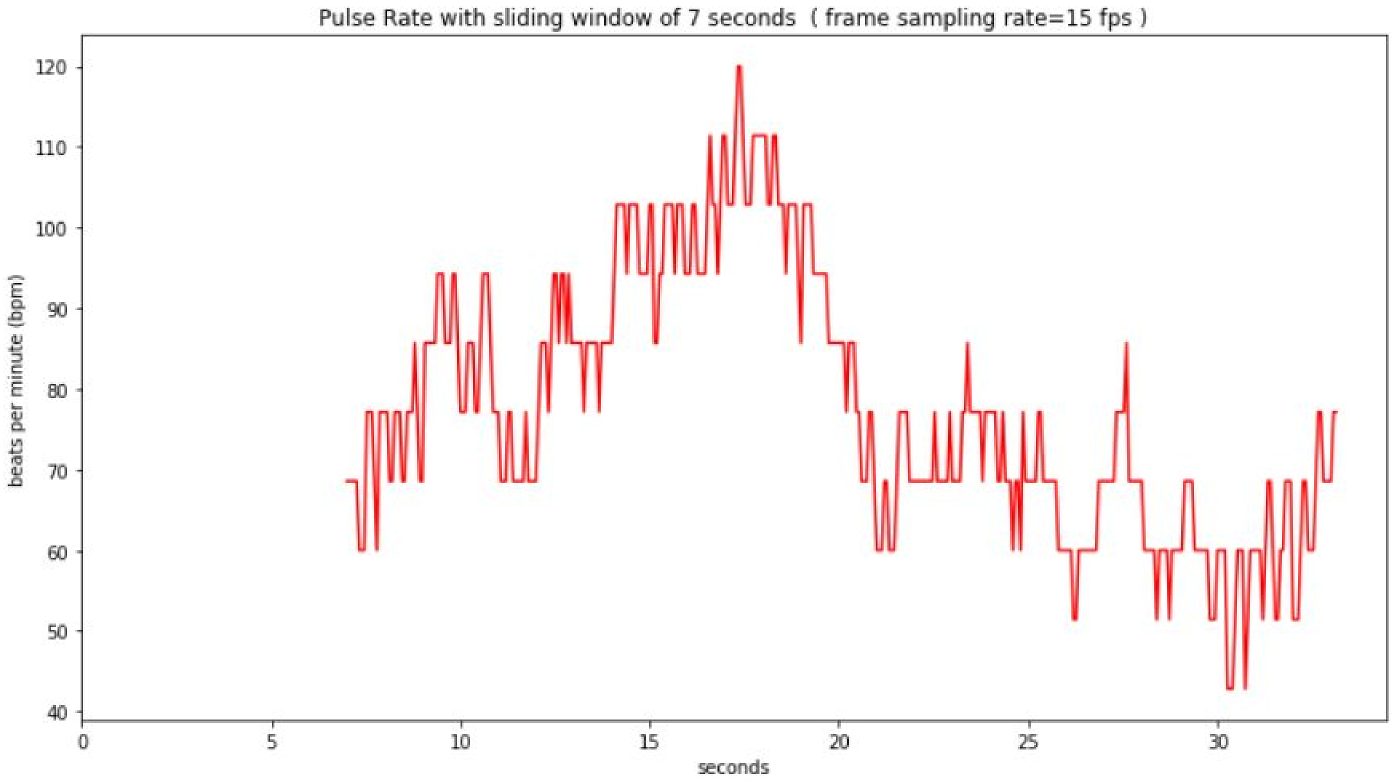
- **Average pulse rate = 91.89 bpm**




```
The average pulse rate (in BPM) by time series analysis  91.89749182115615
```

1. **Frame sample rate : 15 fps**

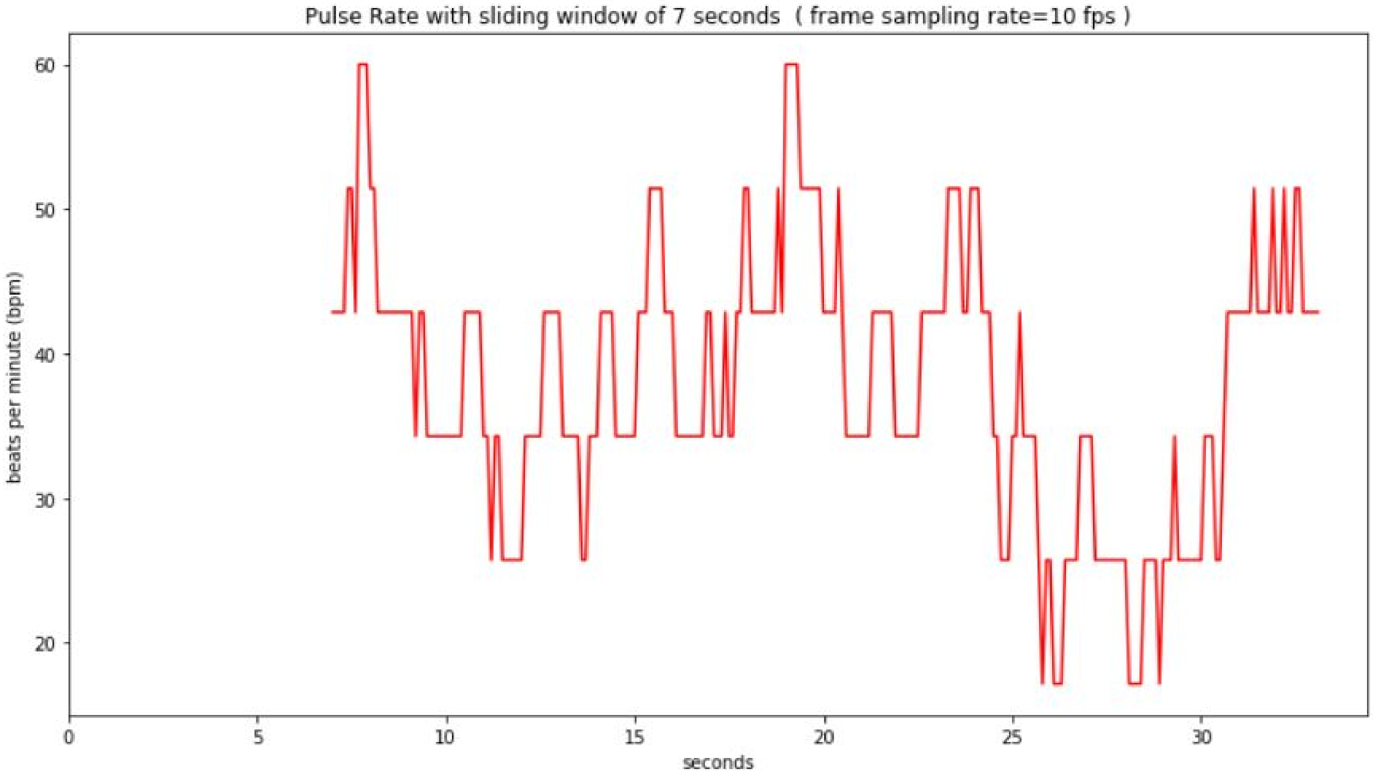
- **Average pulse rate = 77.666 bpm**




```
The average pulse rate (in BPM) by time series analysis  77.6663031624863
```

1. **Frame sample rate : 10 fps**

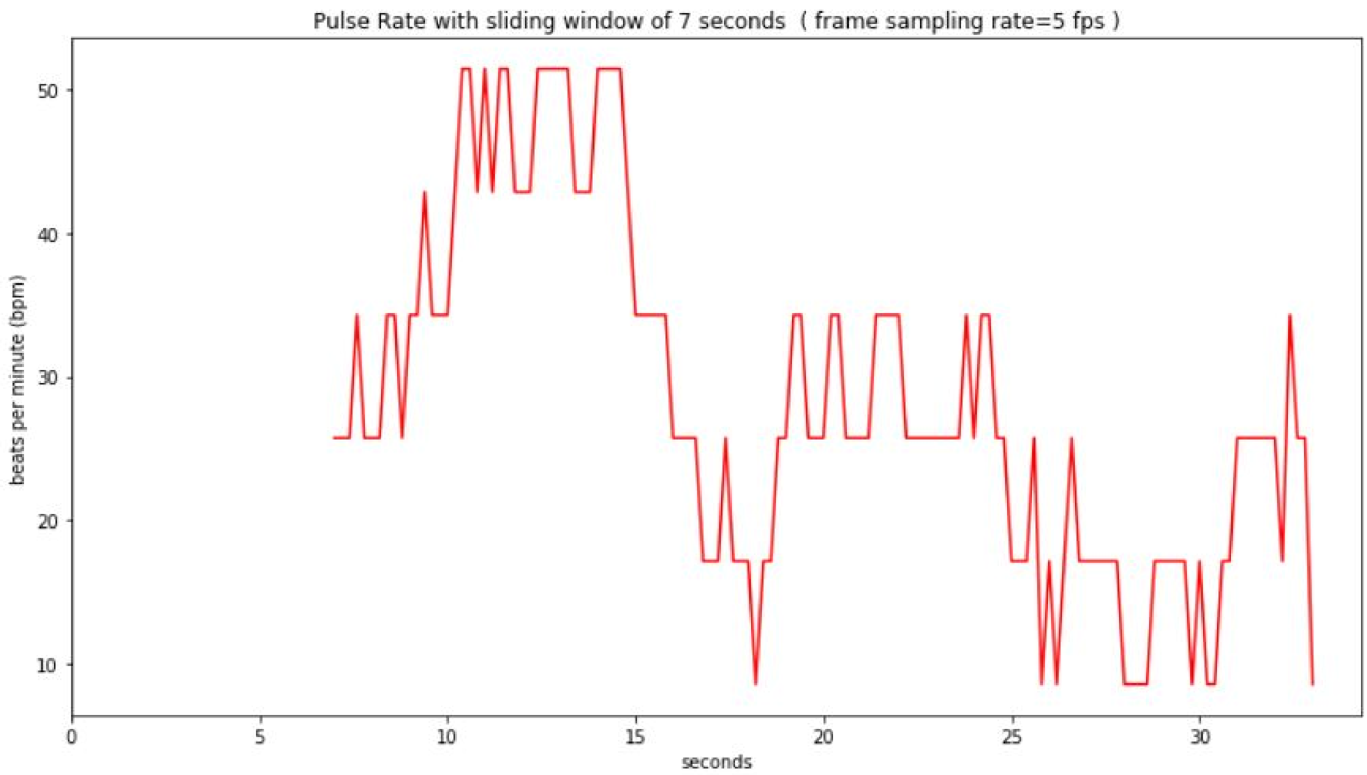
- **Average pulse rate = 37.9825 bpm**




```
The average pulse rate (in BPM) by time series analysis  37.98255179934575
```

1. **Frame sample rate : 5 fps**

- **Average pulse rate = 28.2660 bpm**




```
The average pulse rate (in BPM) by time series analysis  28.266085059978217
```

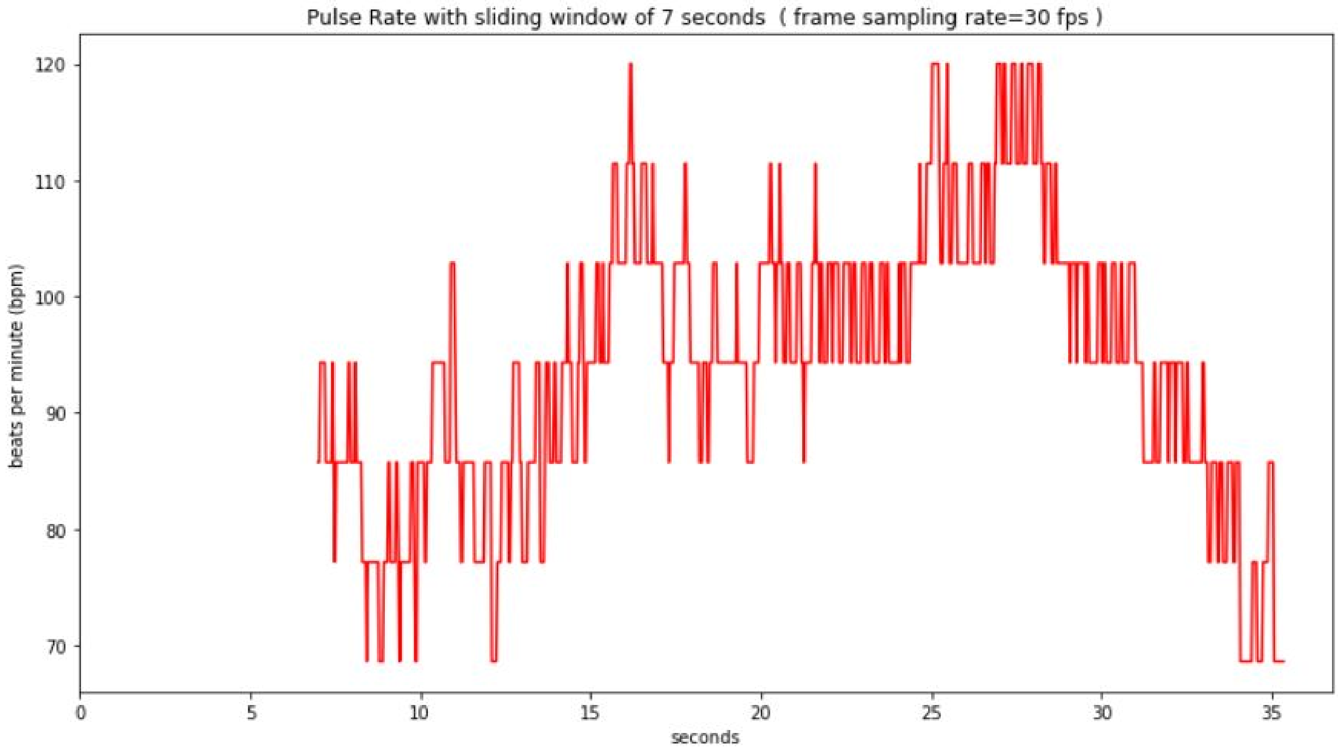
Activity : exercise - crunches (In the dark)

- **Video name : "crunches_dark.mp4"**
- **Video duration : "00:00:45" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)**
- **Sliding window used for bpm = 7 sec (last 7 seconds)**

(Every pixel of a frame is taken into consideration)

1. **Frame sample rate : 30 fps**

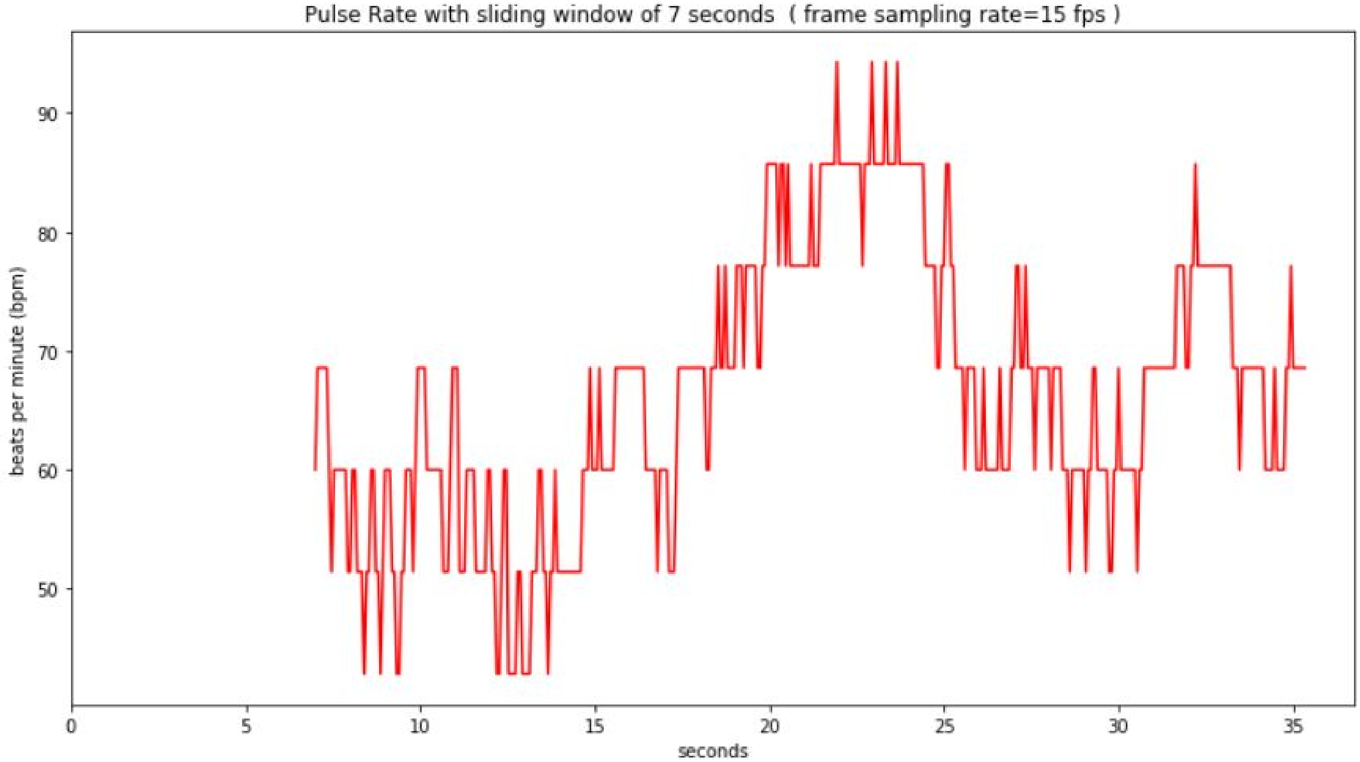
- **Average pulse rate = 94.31593 bpm**




```
The average pulse rate (in BPM) by time series analysis  94.31593083767076
```

1. **Frame sample rate : 15 fps**

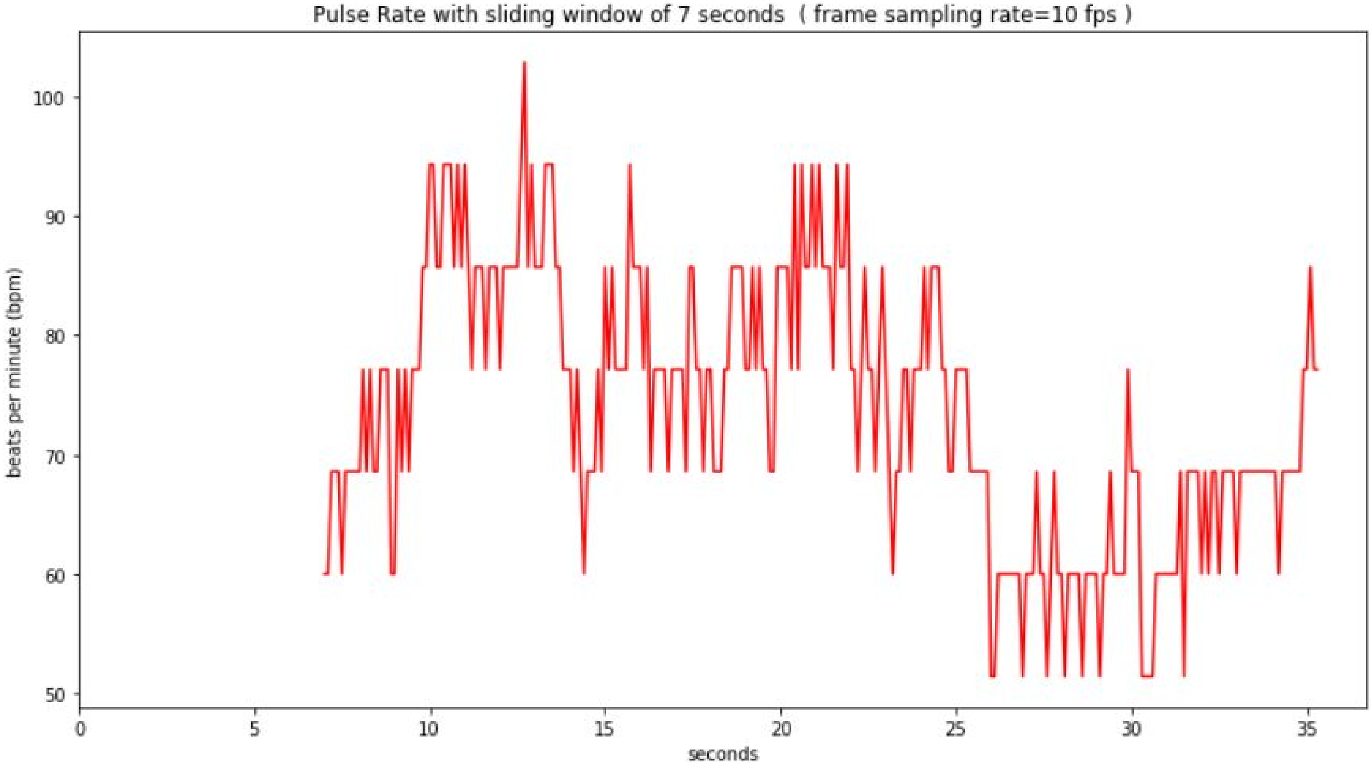
- **Average pulse rate = 66.559356 bpm**




```
The average pulse rate (in BPM) by time series analysis  66.55935613682098
```

1. **Frame sample rate : 10 fps**

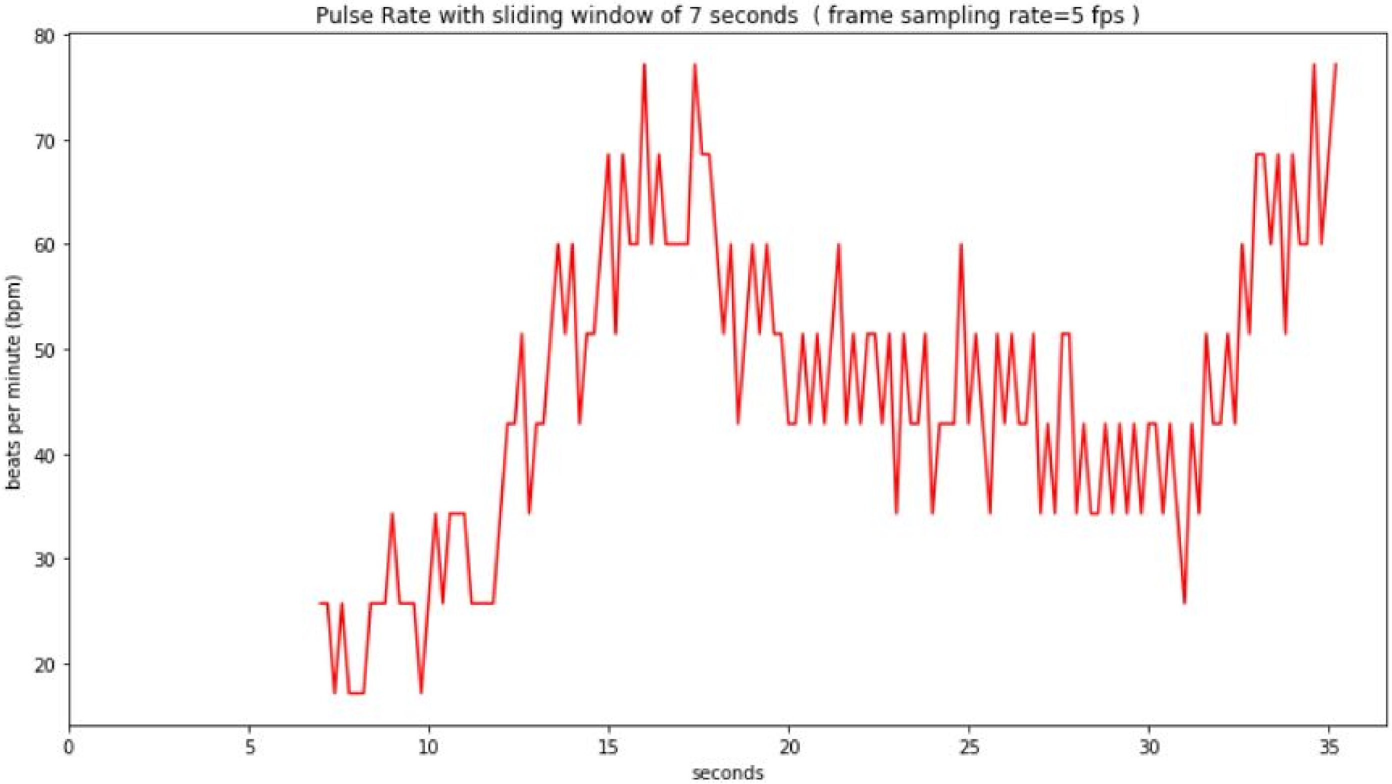
- **Average pulse rate = 73.792756 bpm**




```
The average pulse rate (in BPM) by time series analysis  73.79275653923538
```

1. **Frame sample rate : 5 fps**

- **Average pulse rate = 45.87525 bpm**



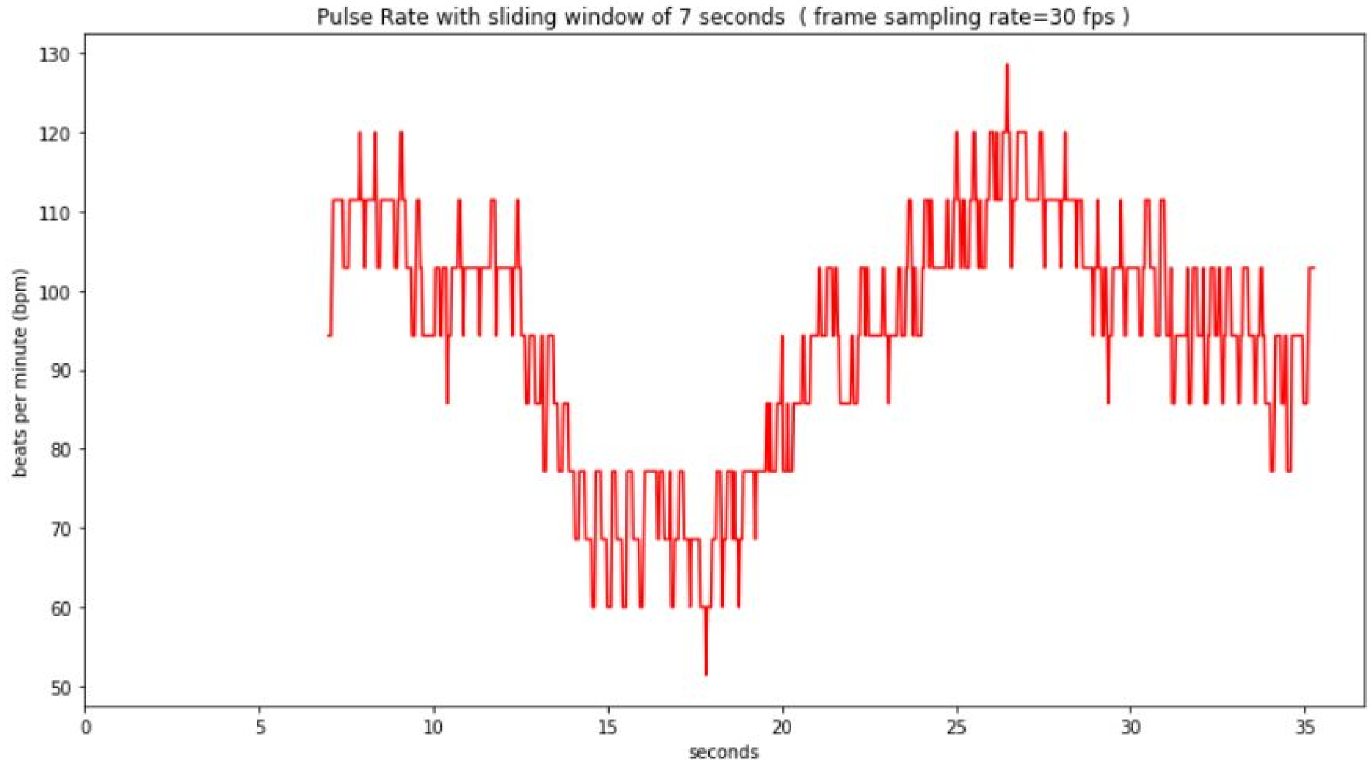

```
The average pulse rate (in BPM) by time series analysis  45.87525150905437
```

Activity : exercise - Resting in Sunlight

• Video name : "resting_sunlight.mp4"
• Video duration : "00:00:45" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)
• Sliding window used for bpm = 7 sec (last 7 seconds)
(Every pixel of a frame is taken into consideration)

1. **Frame sample rate : 30 fps**

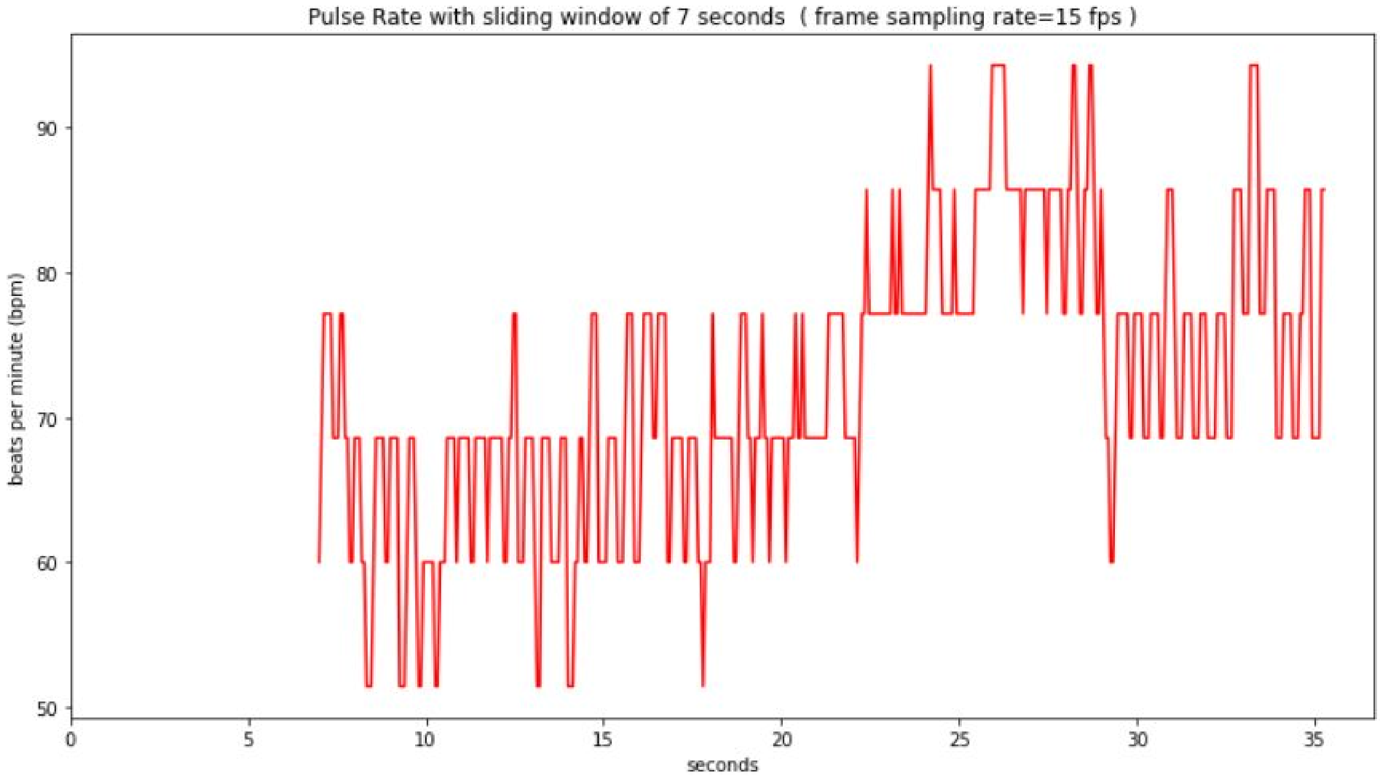
• **Average pulse rate =94.20494 bpm**




```
The average pulse rate (in BPM) by time series analysis  94.20494699646669
```

1. **Frame sample rate : 15 fps**

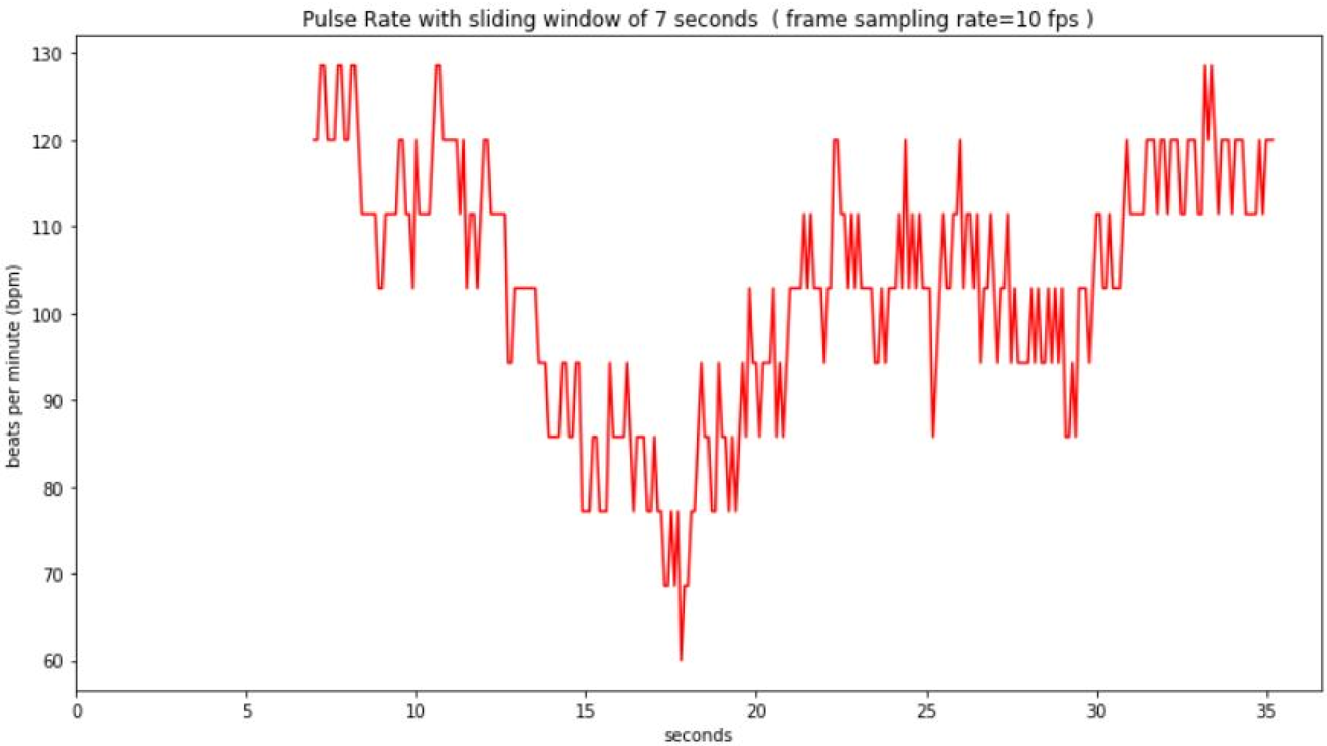
• **Average pulse rate = 72.4638 bpm**




```
The average pulse rate (in BPM) by time series analysis  72.4638655462188
```

1. **Frame sample rate : 10 fps**

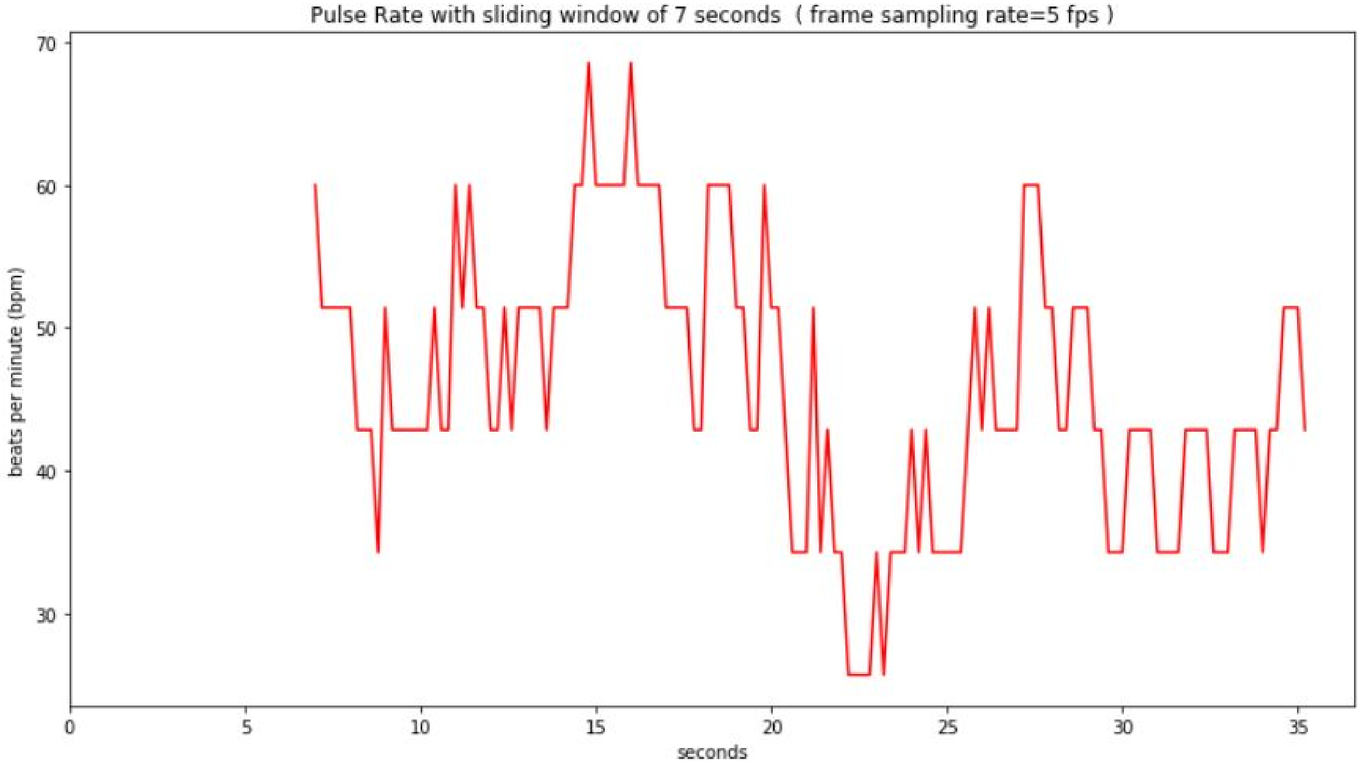
- **Average pulse rate = 102.948 bpm**




```
The average pulse rate (in BPM) by time series analysis  102.94800605754662
```

1. **Frame sample rate : 5 fps**

- **Average pulse rate = 45.8148 bpm**




```
The average pulse rate (in BPM) by time series analysis  45.814889336016186
```

Conclusion: It can be clearly observed that as the fps decreases , the pulse rate also goes down . This is due to the fact on decreasing the samples, the number of samples might not be enough to capture the full features of the wave. It is in accordance to the shannon nyquist sampling rate ,as minimum sample rate should be greater than or equal to the twice the maximum frequency component . Also, we might have noise in the data , so the degradation in the results might be visble very frequently.

Solution: (b) **Effect of Low resolution:**

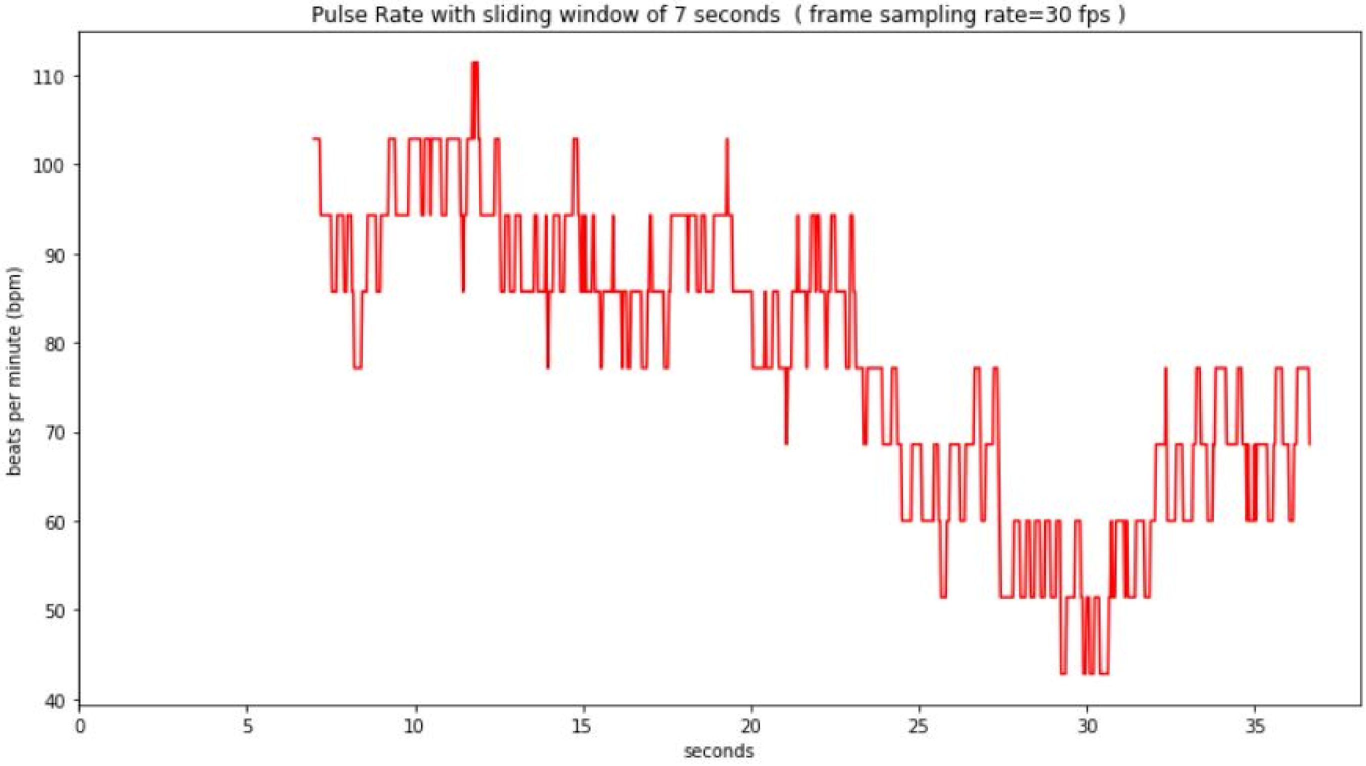
Sampling Frame rate is keeped as fixed as standard 30fps:

Activity : Resting mode(In the dark)

- Video name : "resting_dark.mp4"
- Video duration : "00:00:45" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)
- Sliding window used for bpm = 7 sec (last 7 seconds)

1. Every pixel is taken into consideration

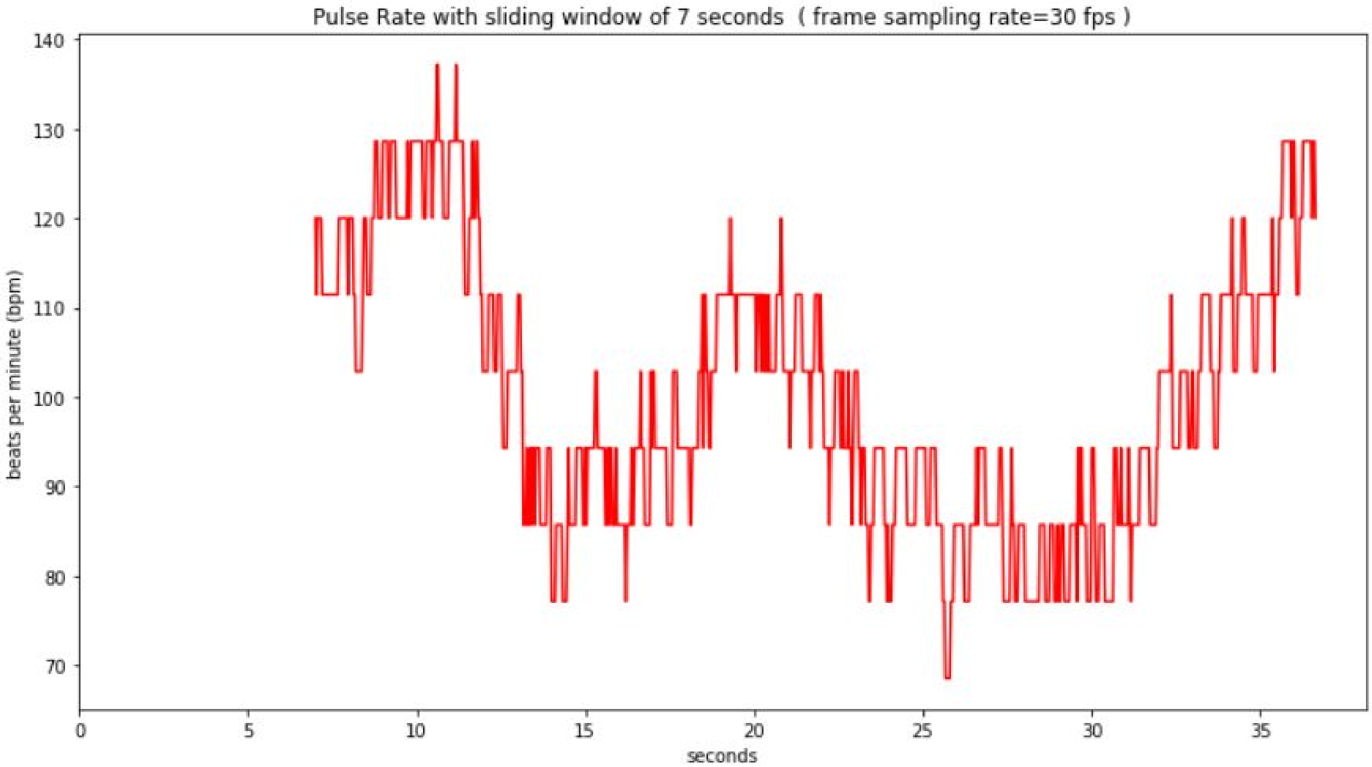
- Average pulse rate = 77.84 bpm




```
The average pulse rate (in BPM) by time series analysis  77.84590690208657
```

1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

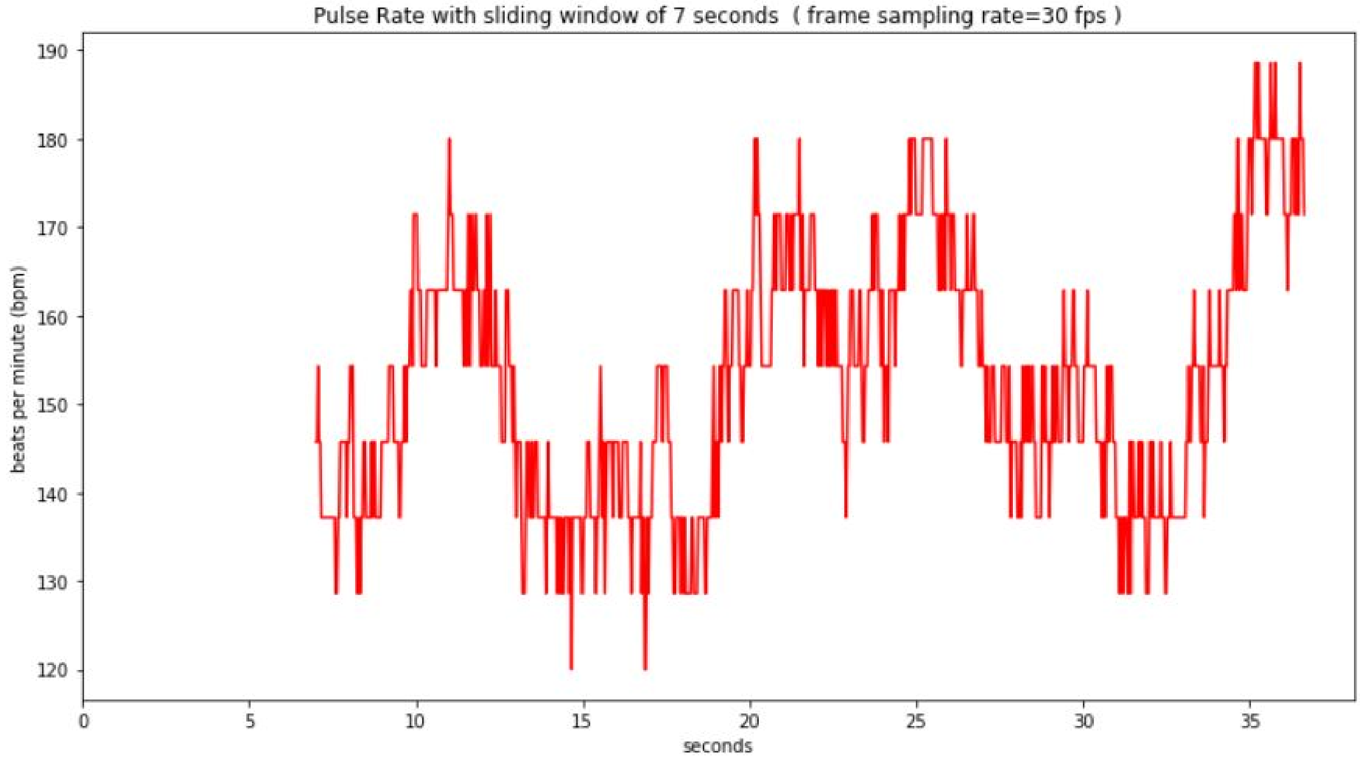
- Average pulse rate = 100.09 bpm




```
The average pulse rate (in BPM) by time series analysis  100.09309791332264
```

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

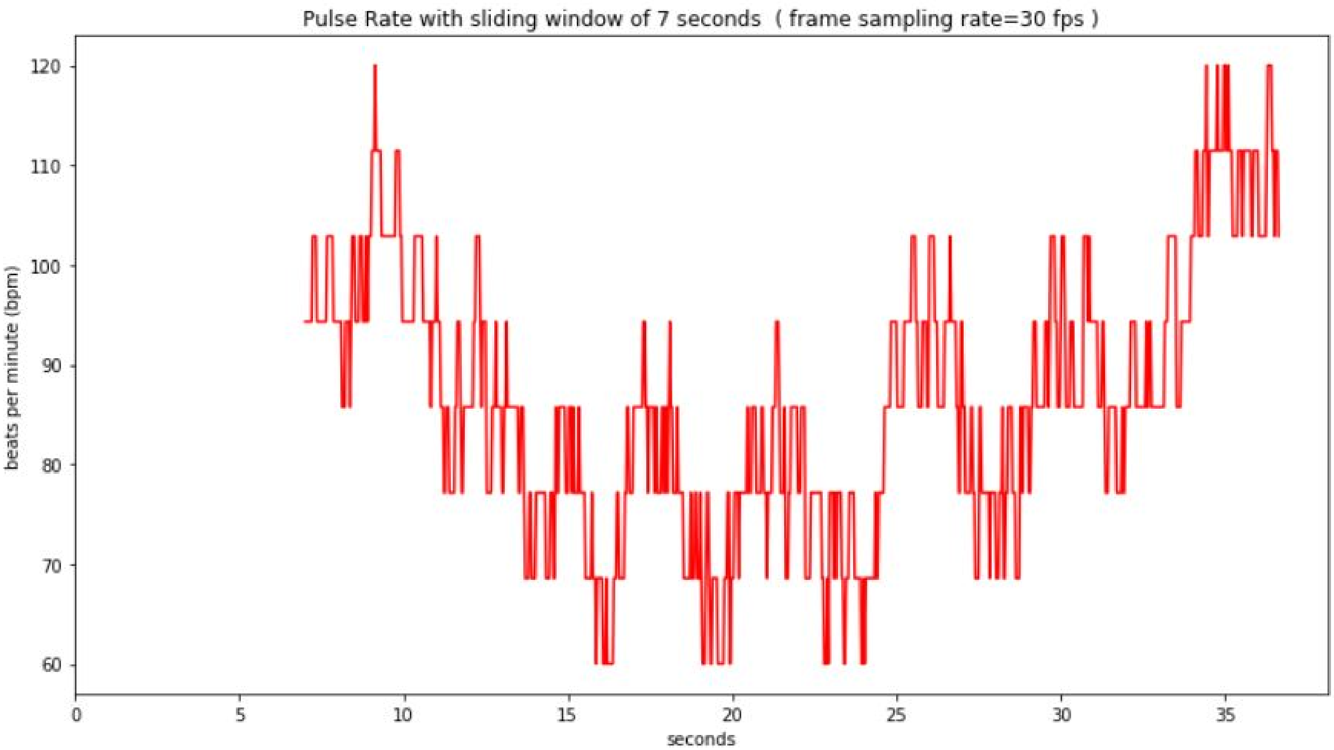
• Average pulse rate = 152 bpm




```
The average pulse rate (in BPM) by time series analysis  152.56179775280867
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

• Average pulse rate = 86 bpm



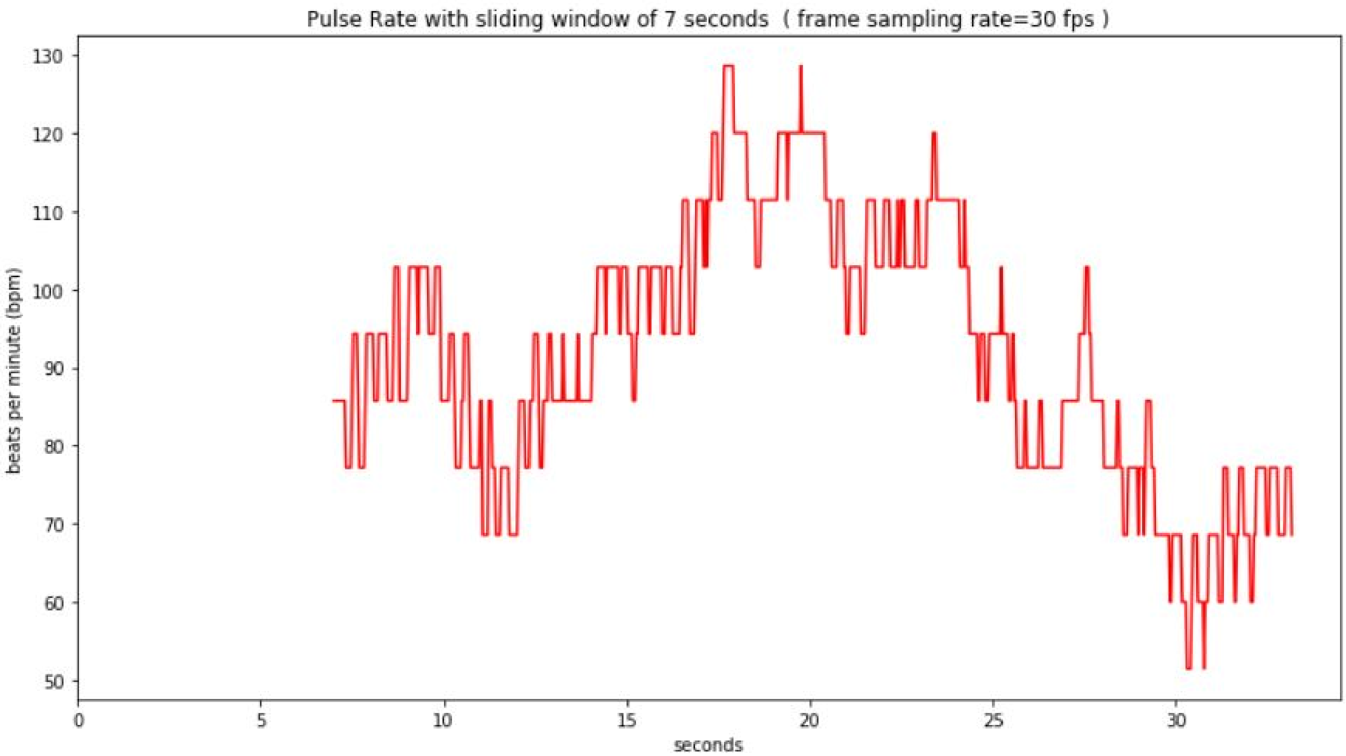

```
The average pulse rate (in BPM) by time series analysis  86.35955056179708
```

Activity : Exercise- push up (In the dark)

- Video name : ”pushup_dark.mp4”
- Video duration : ”00:00:43” (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)
- Sliding window used for bpm = 7 sec (last 7 seconds)

1. Every pixel is taken into consideration

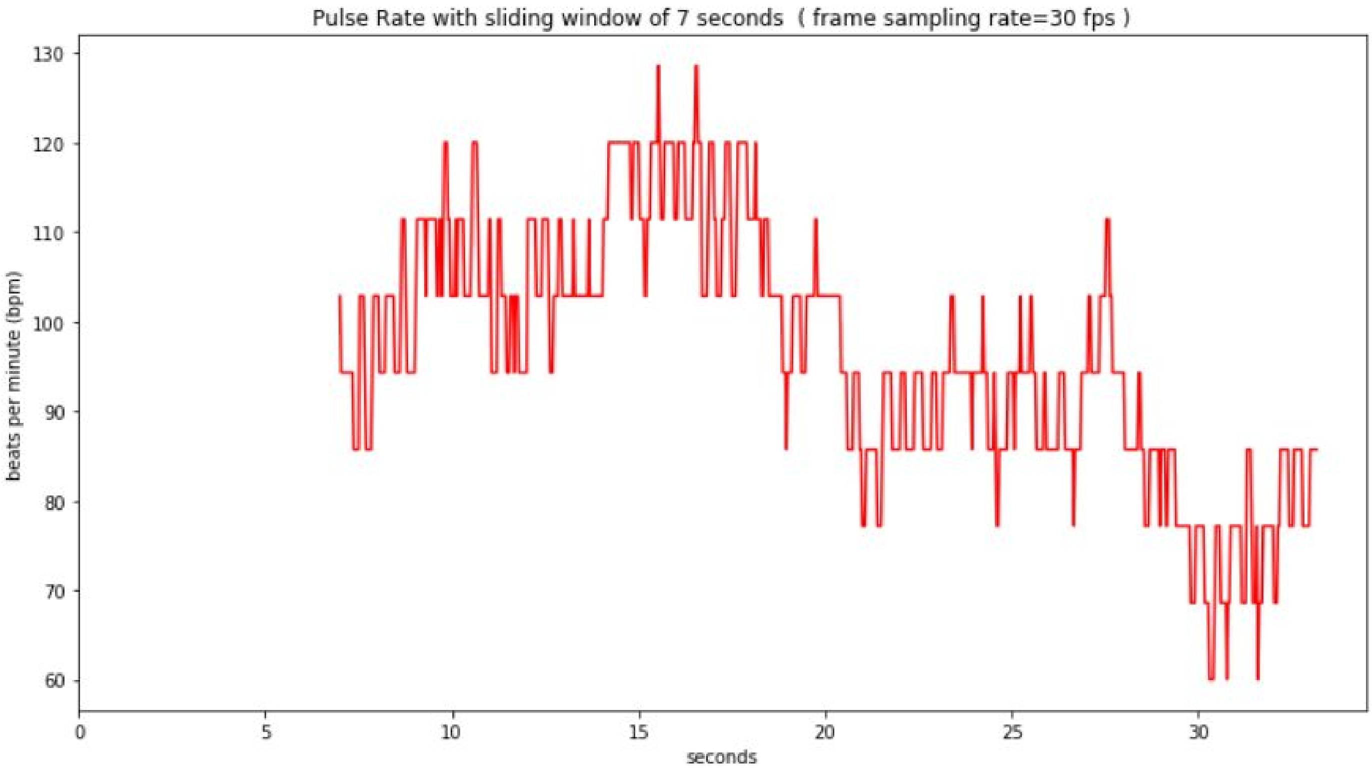
- Average pulse rate = 92 bpm




```
The average pulse rate (in BPM) by time series analysis  91.89749182115615
```

1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

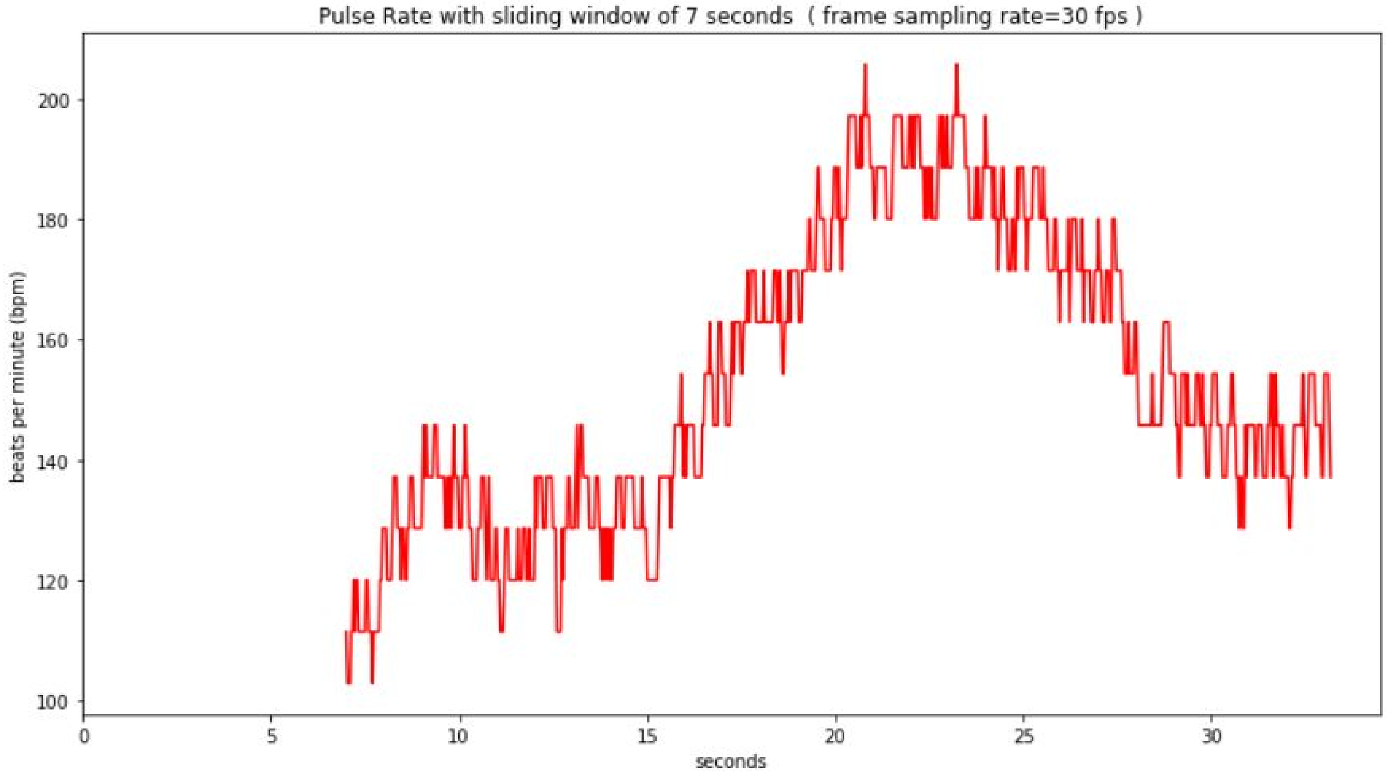
- Average pulse rate = 96 bpm




```
The average pulse rate (in BPM) by time series analysis  96.51035986913898
```

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

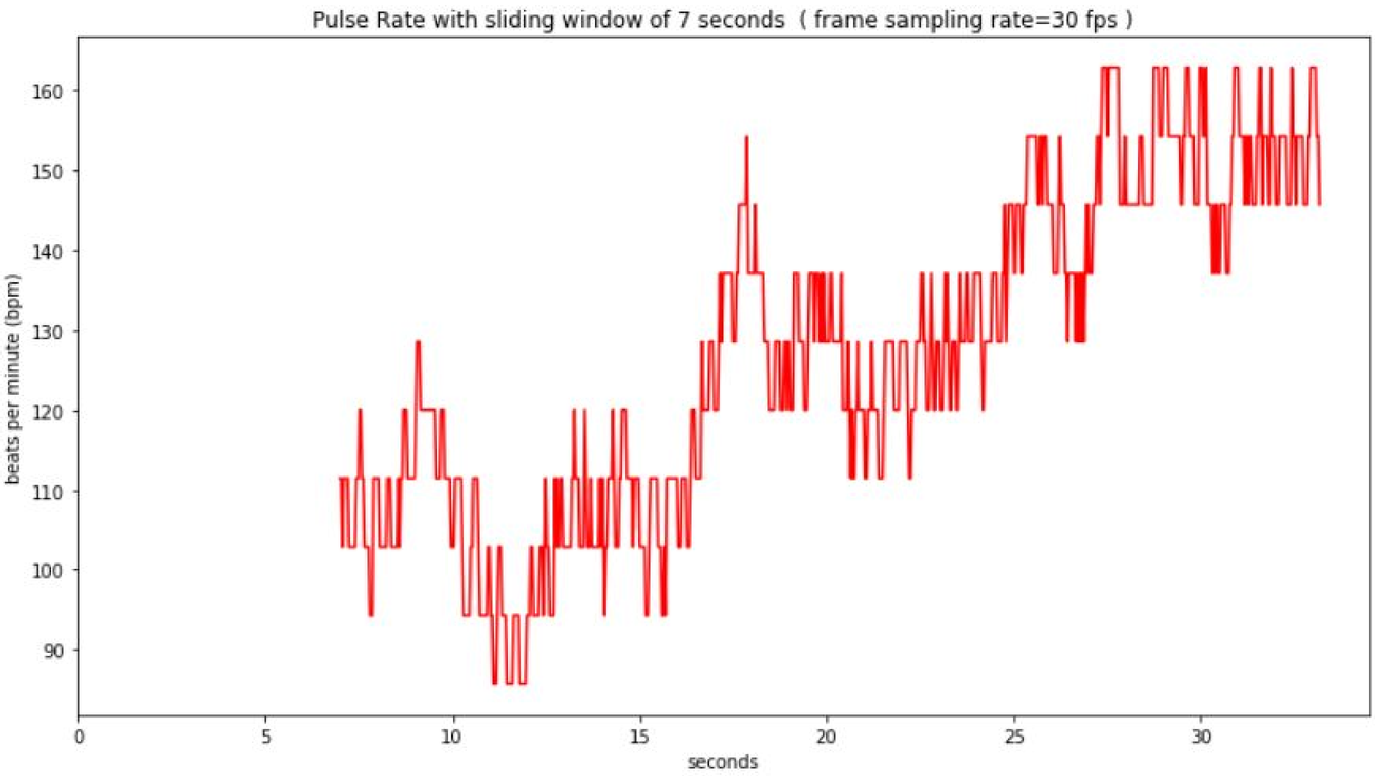
- Average pulse rate = 153 bpm




```
The average pulse rate (in BPM) by time series analysis  153.40239912759012
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

- Average pulse rate = 126 bpm




```
The average pulse rate (in BPM) by time series analysis  126.7829880043618
```

Activity : Exercise- crunches (In the dark)

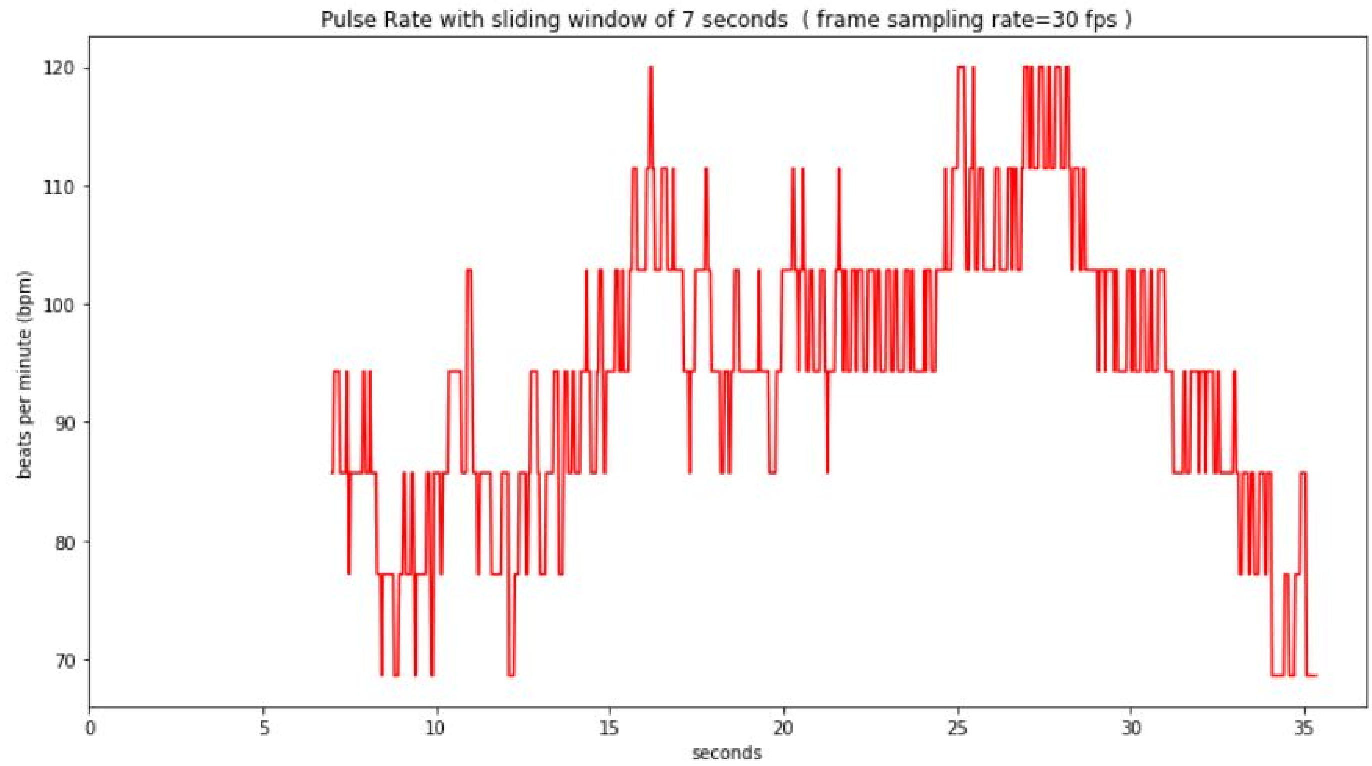
• Video name : "crunches_dark.mp4"

• Video duration : "00:00:46" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)

• Sliding window used for bpm = 7 sec (last 7 seconds)

1. Every pixel is taken into consideration

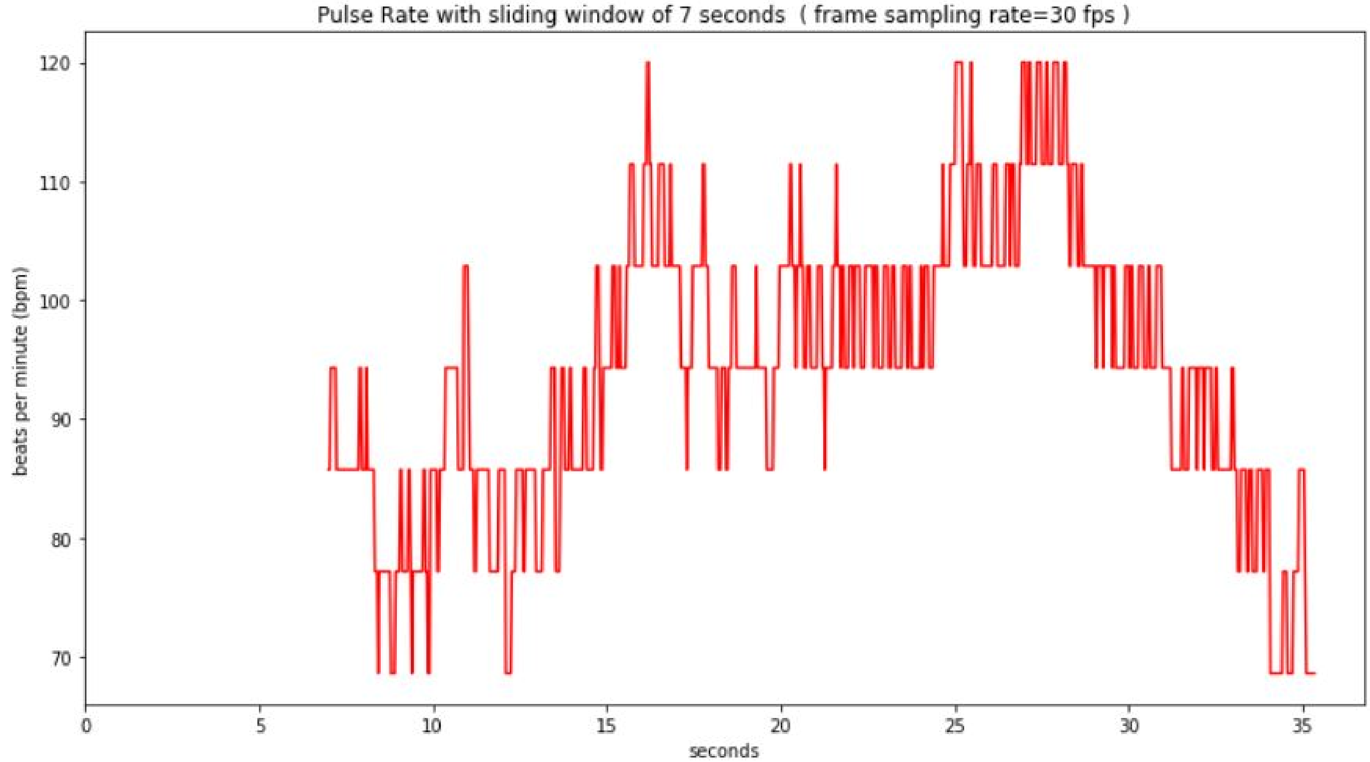
• Average pulse rate =94 bpm



The average pulse rate (in BPM) by time series analysis 94.31593083767076

1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

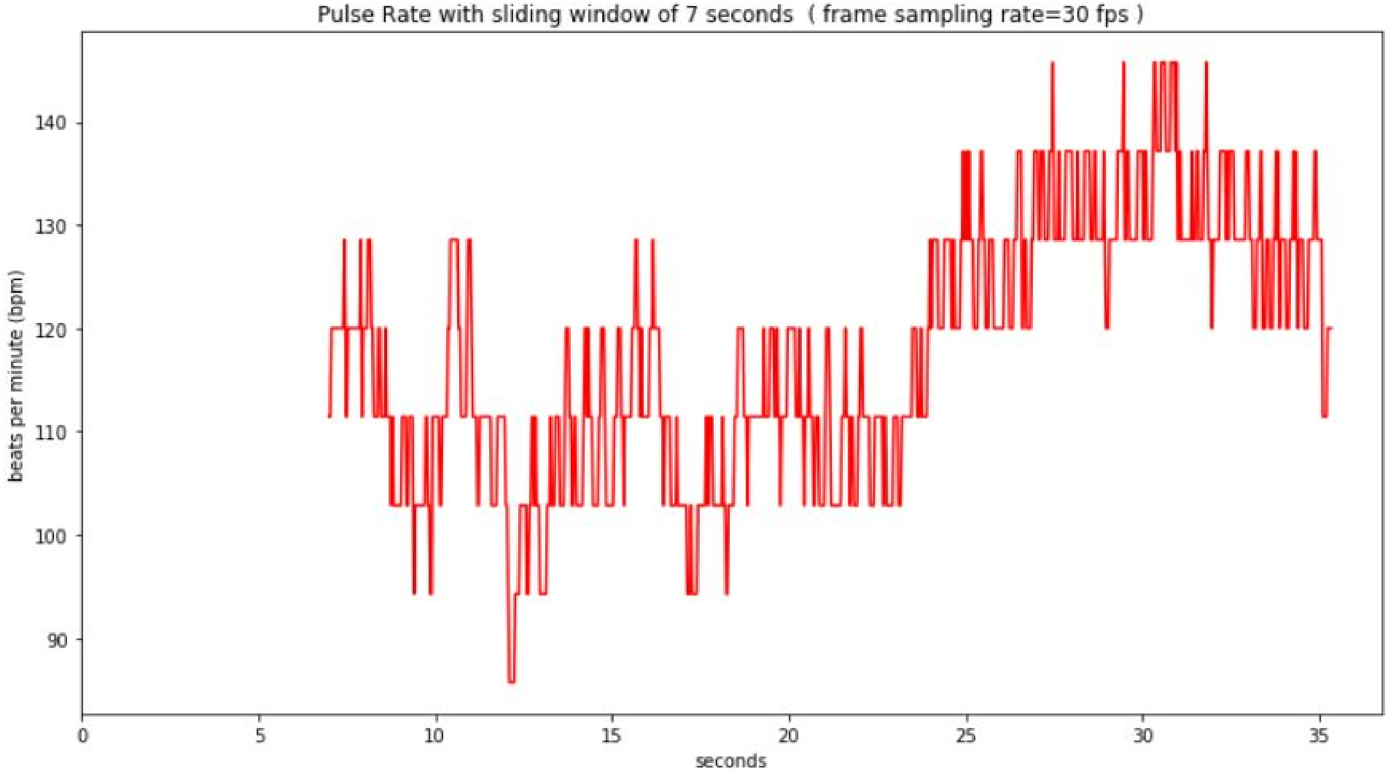
- Average pulse rate = 94 bpm



The average pulse rate (in BPM) by time series analysis 94.19506462984802

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

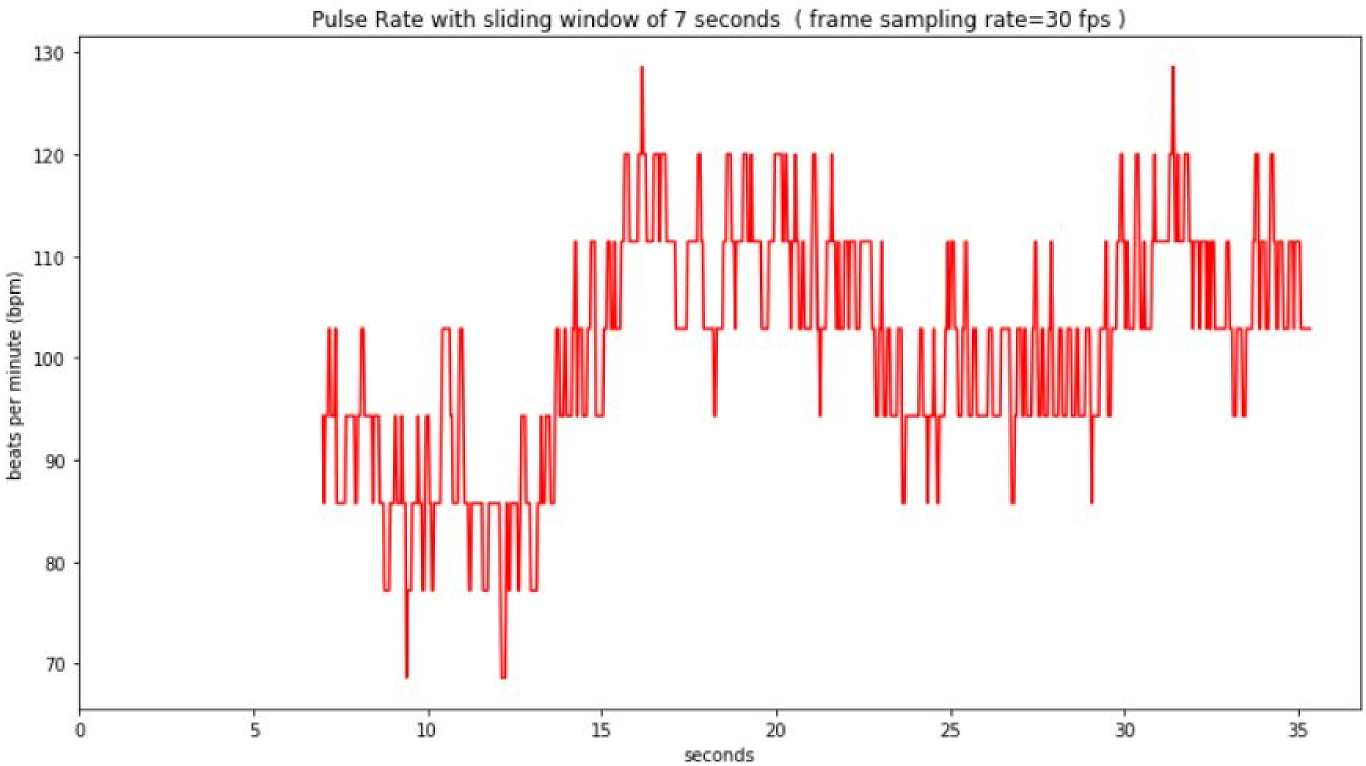
- Average pulse rate = 117 bpm




```
The average pulse rate (in BPM) by time series analysis  117.86469699513285
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

- Average pulse rate = 101 bpm




```
The average pulse rate (in BPM) by time series analysis  101.07436629175783
```

Activity : Resting in sunlight

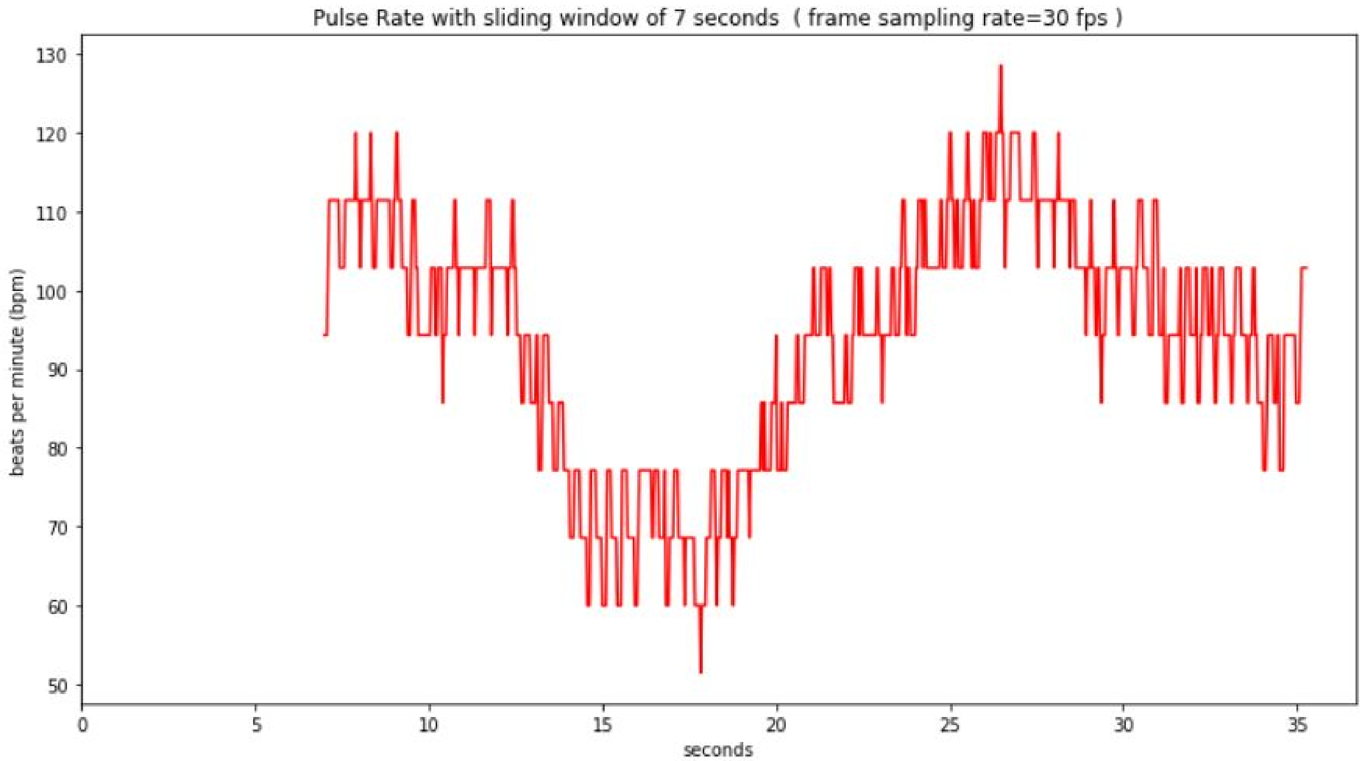
• Video name : "resting_sunlight.mp4"

• Video duration : "00:00:45" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)

• Sliding window used for bpm = 7 sec (last 7 seconds)

1. Every pixel is taken into consideration

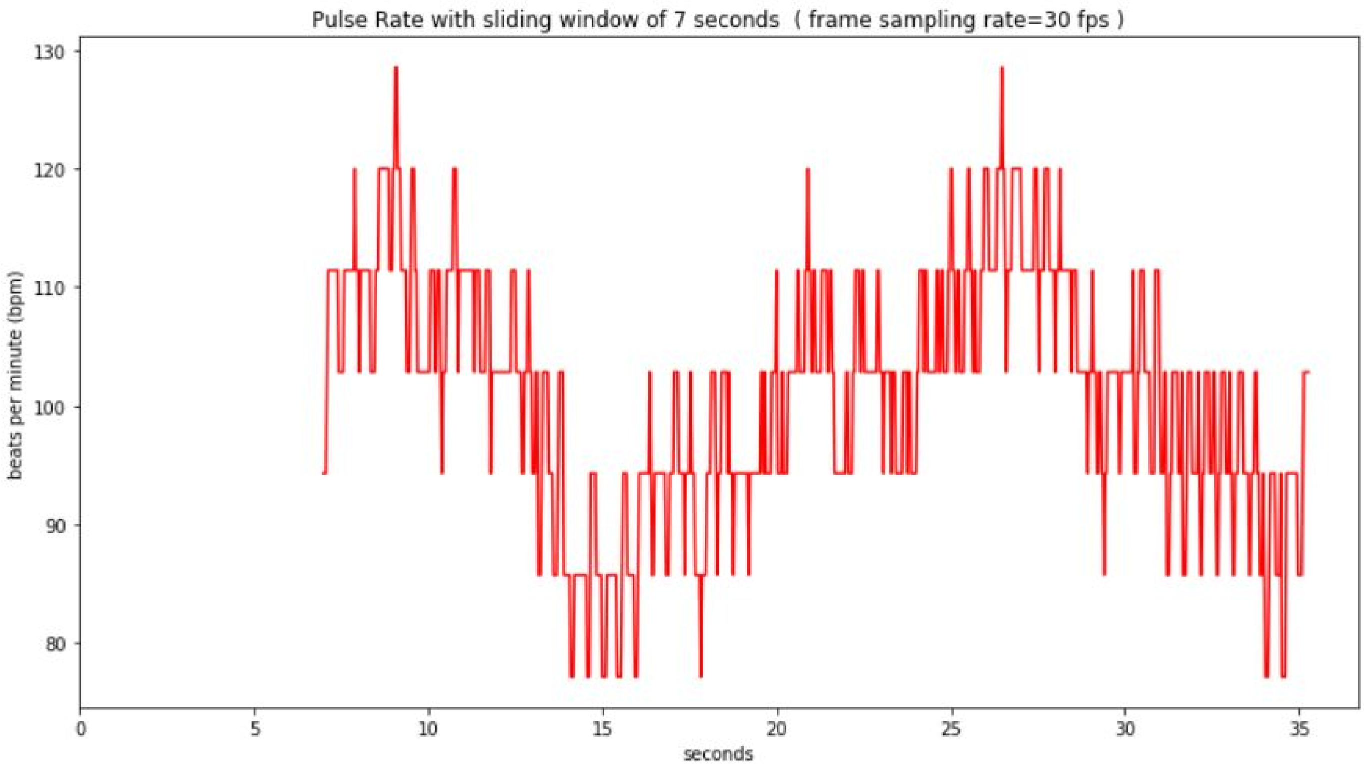
Average pulse rate = 94 bpm




```
The average pulse rate (in BPM) by time series analysis  94.20494699646669
```

1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

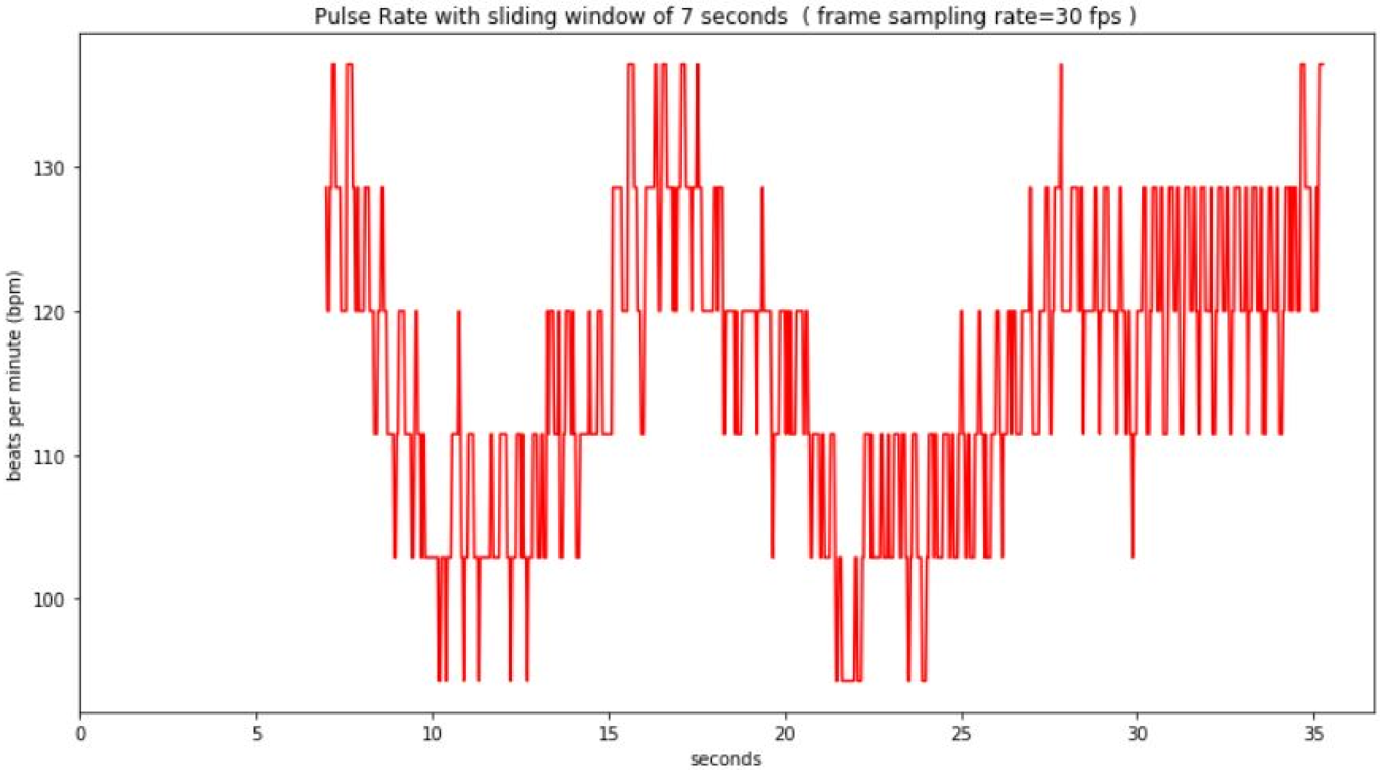
Average pulse rate = 101 bpm




```
The average pulse rate (in BPM) by time series analysis  101.19131751640624
```

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

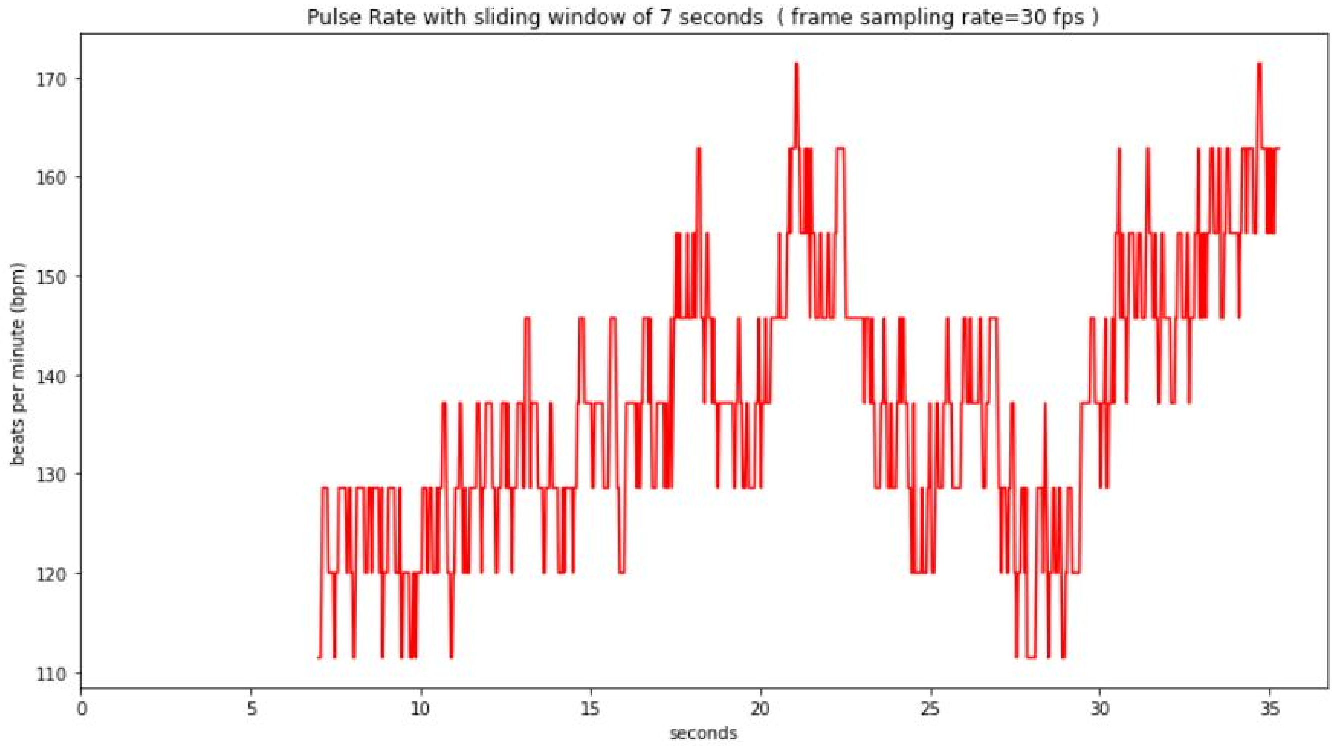
- Average pulse rate = 116 bpm




```
The average pulse rate (in BPM) by time series analysis  116.02221100454373
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

- Average pulse rate = 137 bpm




```
The average pulse rate (in BPM) by time series analysis  137.04189803129734
```

Conclusion:
According to the results of the above experiments with decreasing the resolution, is visible that the pulse rate increases due to the decrease in resolution. It maybe possible due to the fact that the SNR decreases due to decrease in the resolution of a frame, and hence there might be more peak occuring due to the noise component.

Comment on Low resolution: The low resolution was giving unexpected flutuations in the heartbeat per minute value. It may be possible due to low signal to noise ratio .

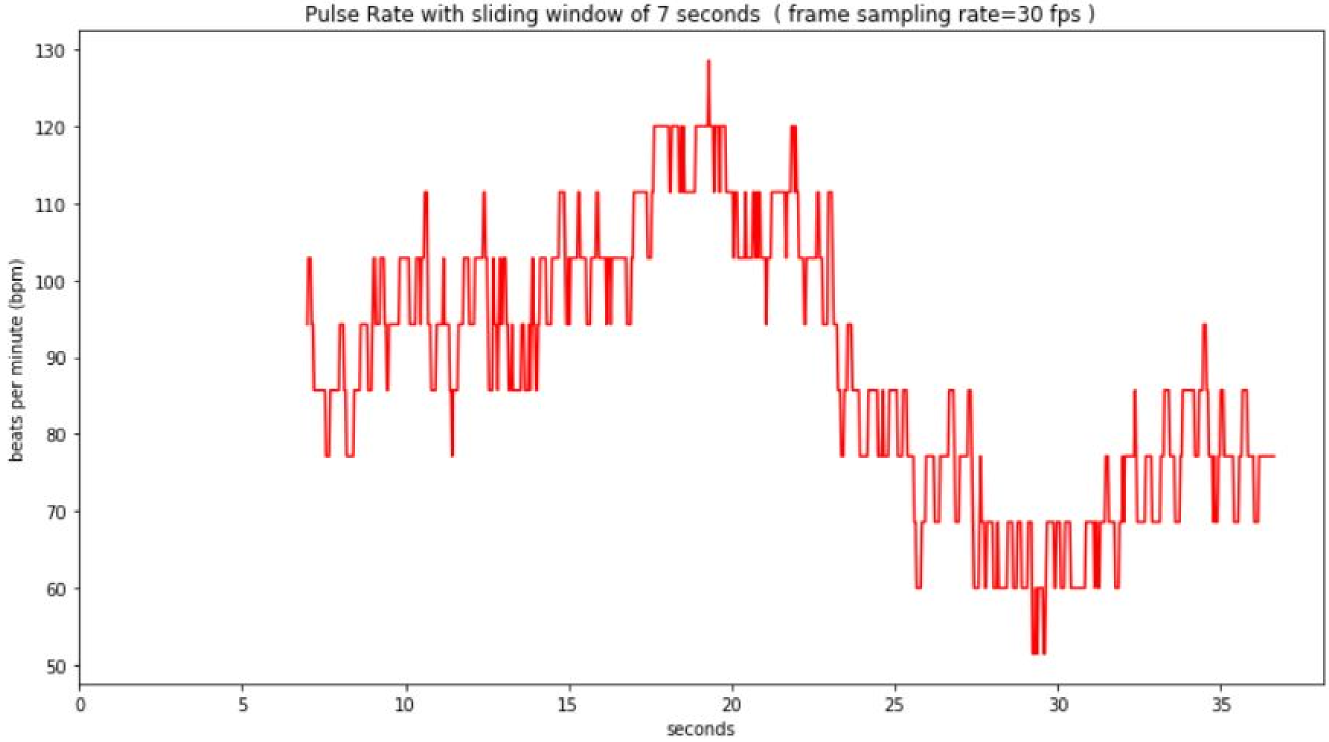
Comment on how to increase the SNR : We could increase the SNR by cropping the corner, boundary pixels. In the below analysis, I have cropped the side regions or can say that the center part is taken keeping 20 pixels gap from each side . That will help in boosting the SNR .
As the corner , boundary pixels are more susceptible to the noise from the environment, therefore my first priority is to remove these pixels from participating in the calculations and effect the result. Therefore , I used cropping technique to remove these pixels.
The cropping code is written in the program. The analysis is as follows :

Activity : Resting mode(In the dark)

- Video name : "resting_dark.mp4"
- Video duration : "00:00:45" (I have trimmed starting 5 and ending 5 seconds because they may contain

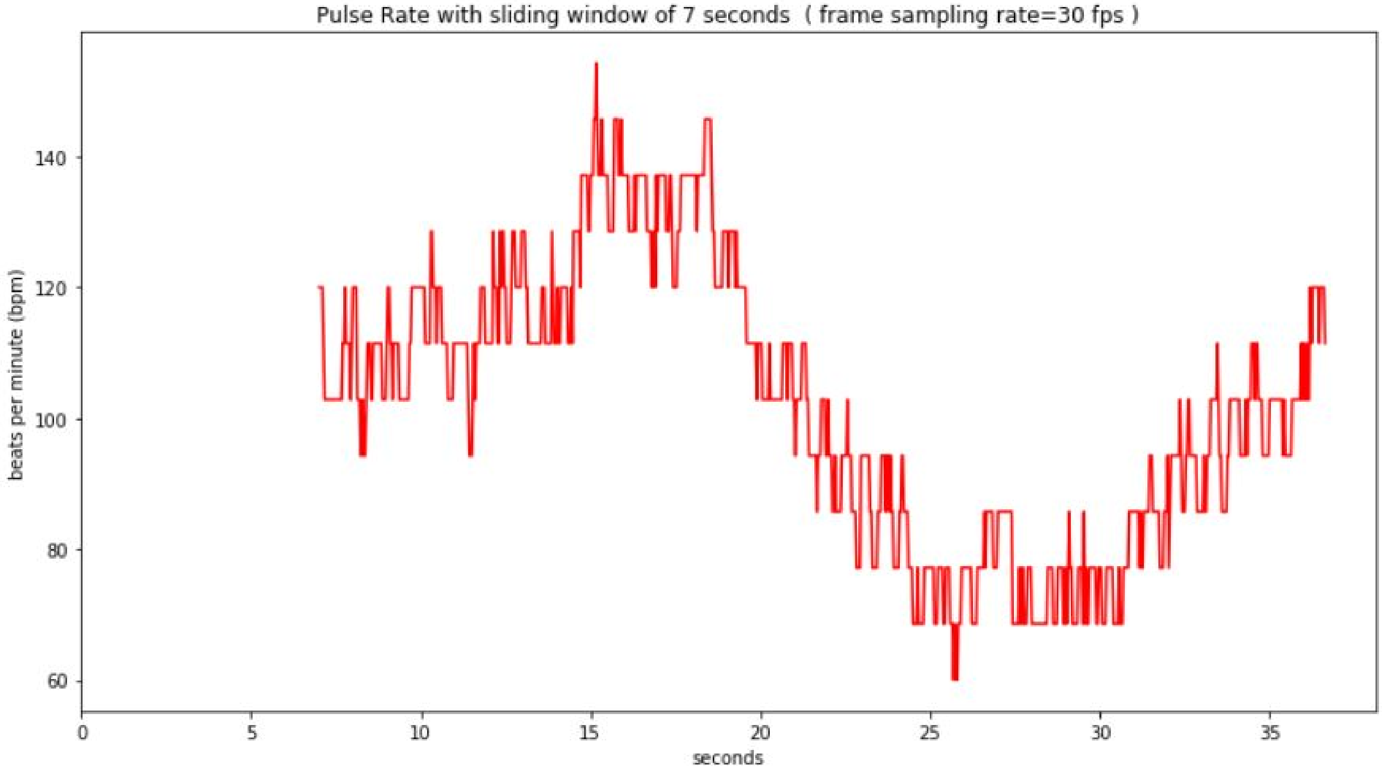
more noise . This is done in the code.)
• Sliding window used for bpm = 7 sec (last 7 seconds)
• The center pixels are taken keeping 20 pixels gap from the boundary.

1. Every pixel is taken into consideration.
• Average pulse rate = 88 bpm




```
The average pulse rate (in BPM) by time series analysis  88.63242375601942
```

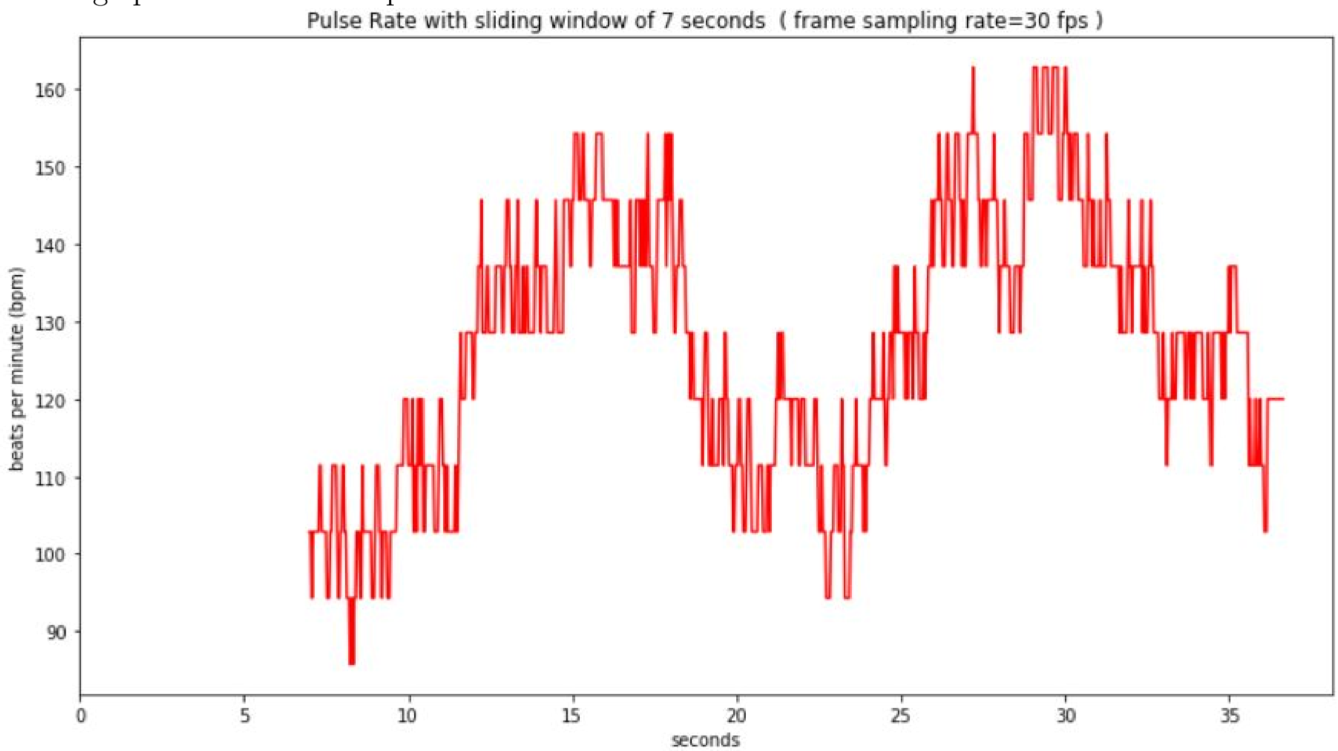
1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis
• Average pulse rate = 101 bpm




```
The average pulse rate (in BPM) by time series analysis  101.9325842696631
```

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

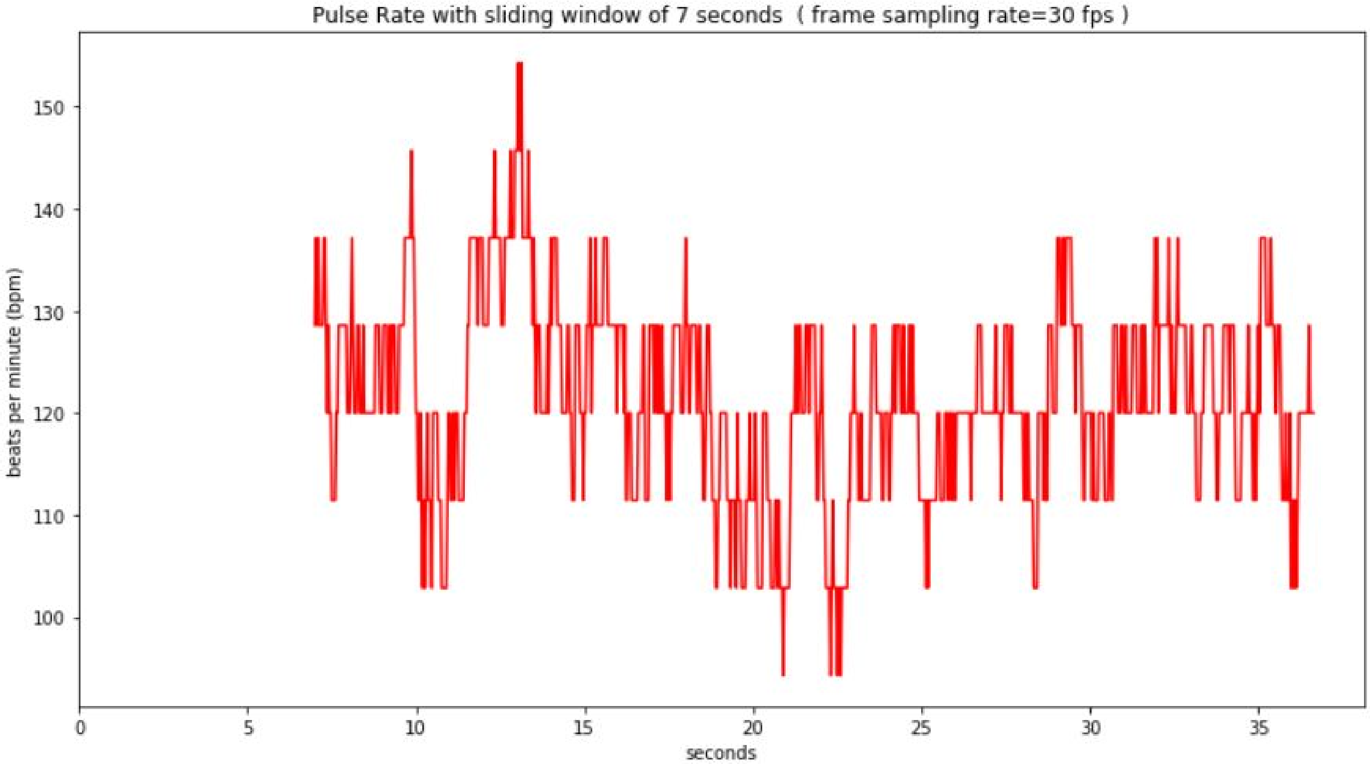
- Average pulse rate = 126 bpm




```
The average pulse rate (in BPM) by time series analysis  126.88603531300204
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

- Average pulse rate = 121 bpm



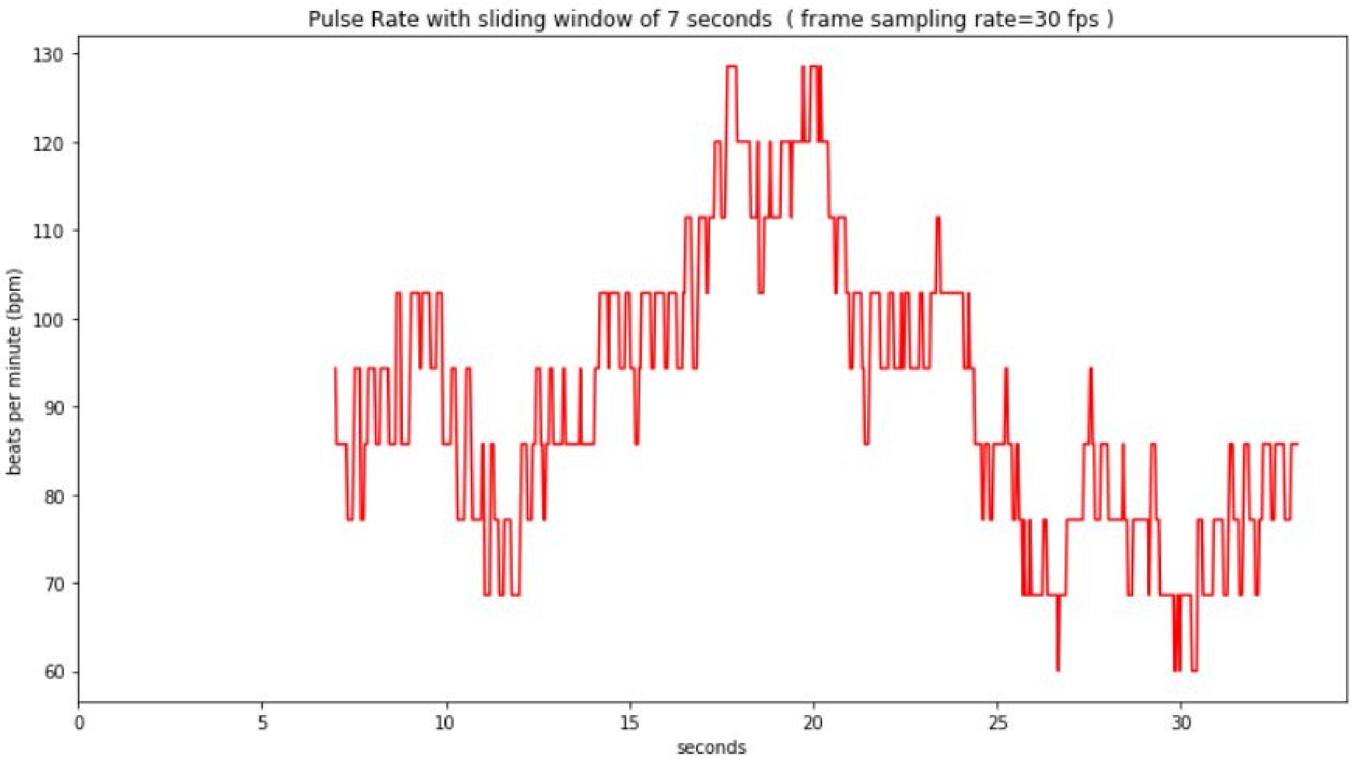

```
The average pulse rate (in BPM) by time series analysis  121.40609951845933
```

Activity : Exercise- push up (In the dark)

- Video name : "pushup_dark.mp4"
- Video duration : "00:00:43" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)
- Sliding window used for bpm = 7 sec (last 7 seconds)

1. Every pixel is taken into consideration

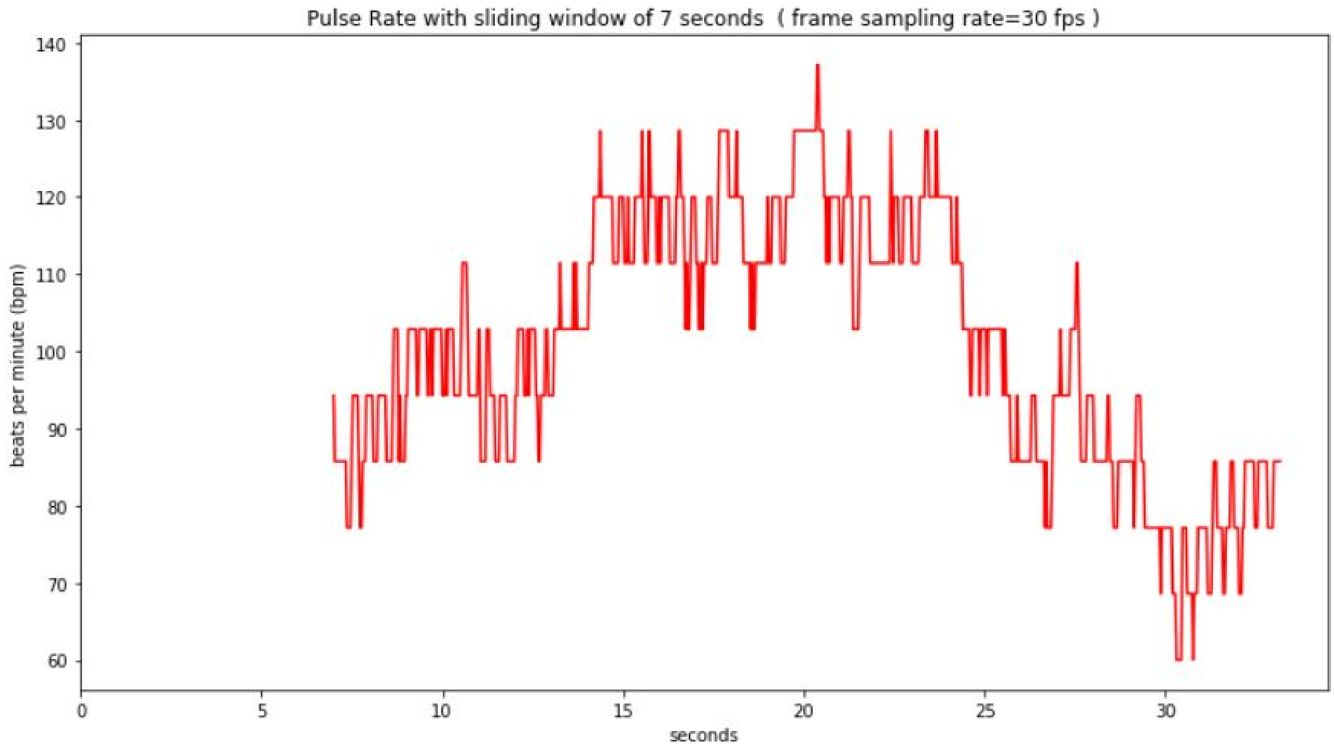
- Average pulse rate = 90 bpm




```
The average pulse rate (in BPM) by time series analysis  90.88331515812438
```

1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

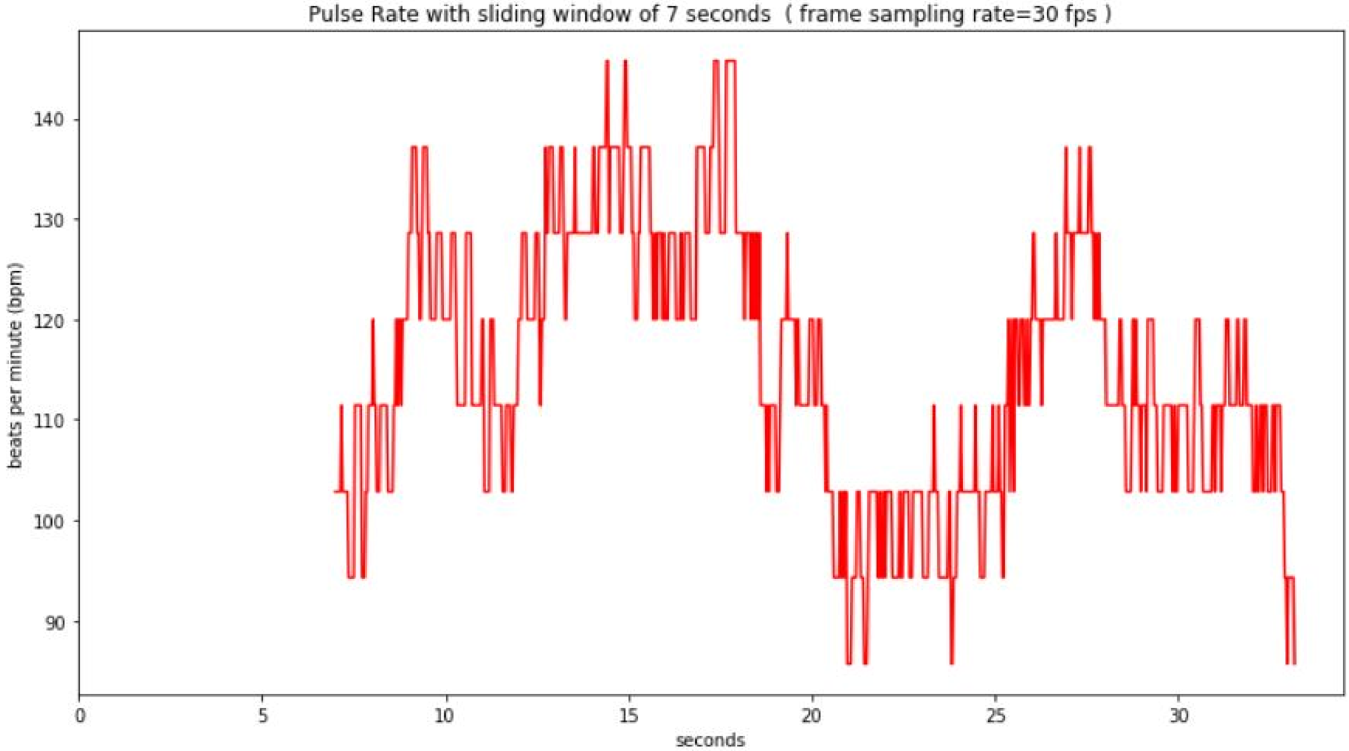
- Average pulse rate = 100 bpm




```
The average pulse rate (in BPM) by time series analysis  100.89422028353363
```

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

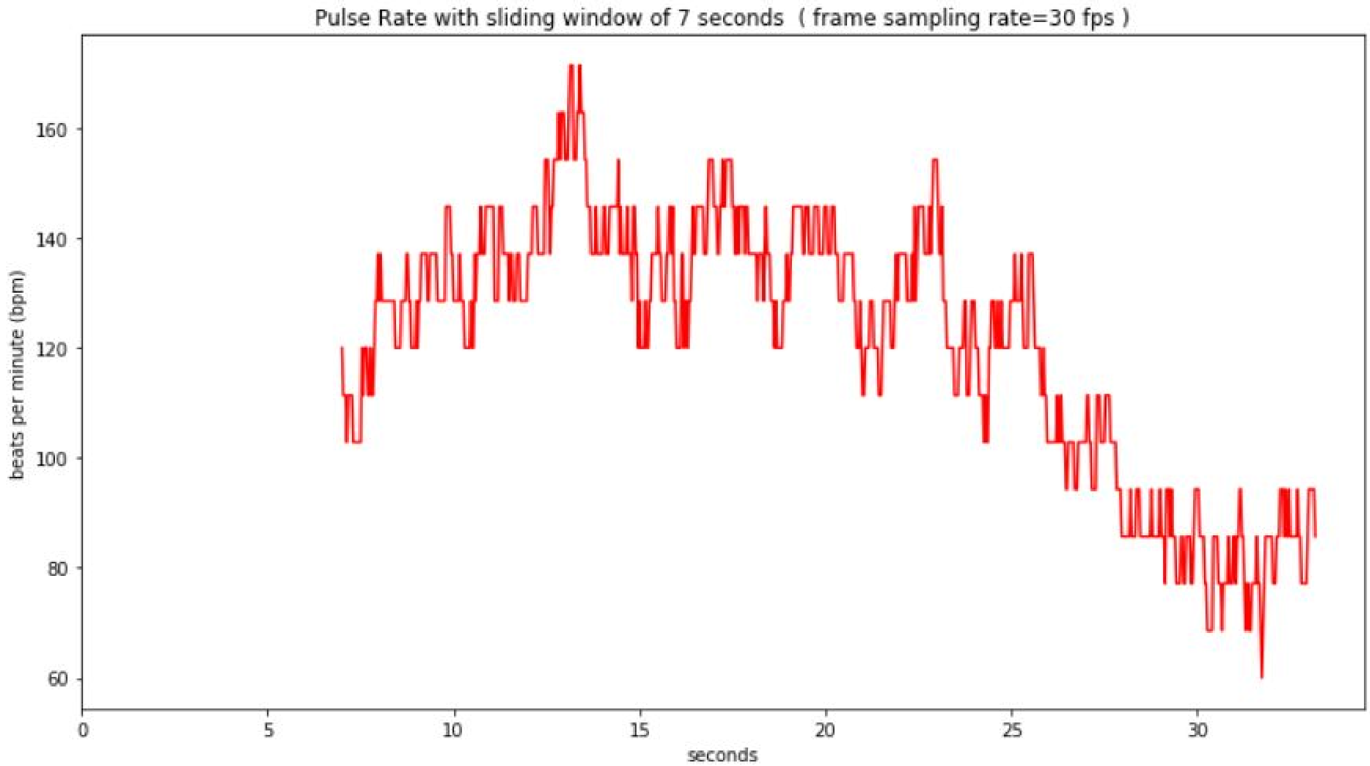
- Average pulse rate = 115 bpm




```
The average pulse rate (in BPM) by time series analysis  115.17993456924748
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

- Average pulse rate = 121 bpm



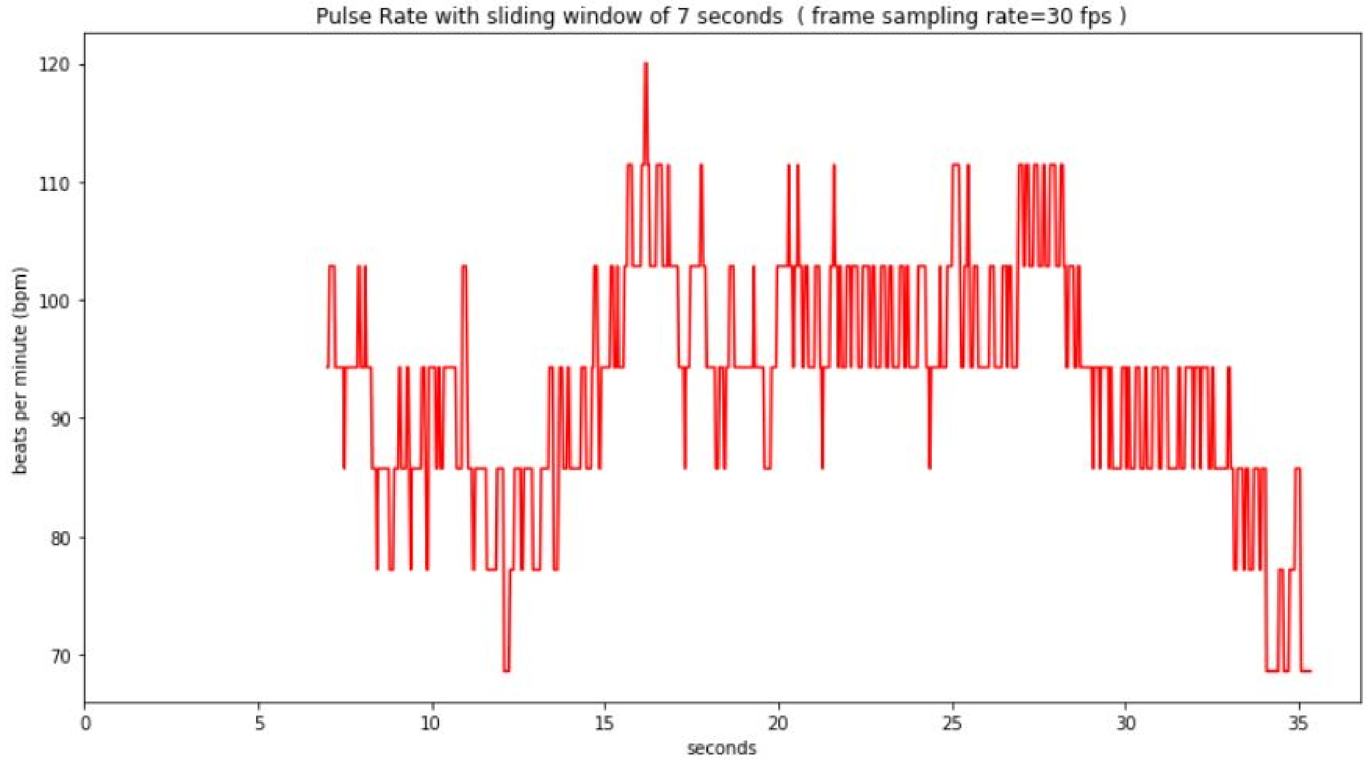

```
The average pulse rate (in BPM) by time series analysis  121.03598691384937
```

Activity : Exercise- crunches (In the dark)

• Video name : "crunches_dark.mp4"
• Video duration : "00:00:46" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)
• Sliding window used for bpm = 7 sec (last 7 seconds)

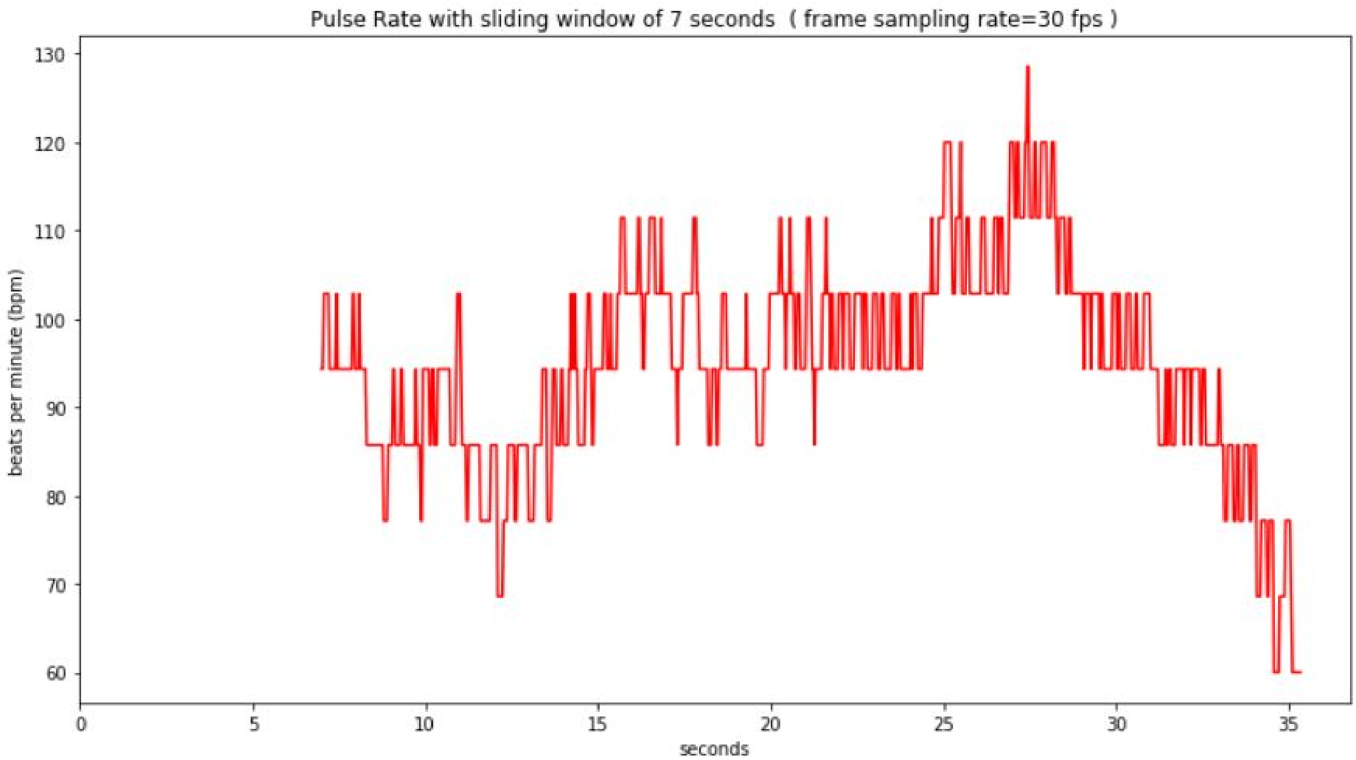
1. Every pixel is taken into consideration

• Average pulse rate = 93 bpm




```
The average pulse rate (in BPM) by time series analysis  93.1374853113994
```

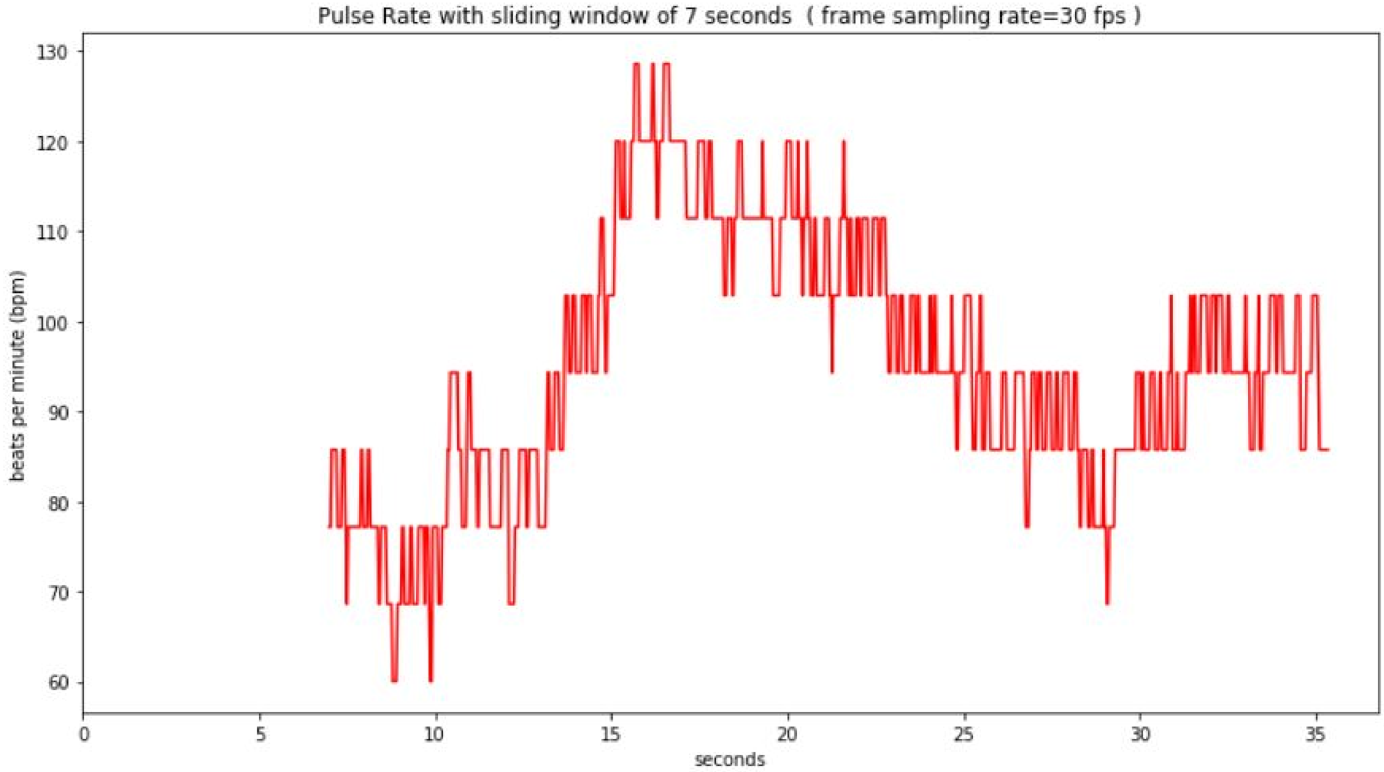
1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis




```
The average pulse rate (in BPM) by time series analysis  95.05120026859218
```

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

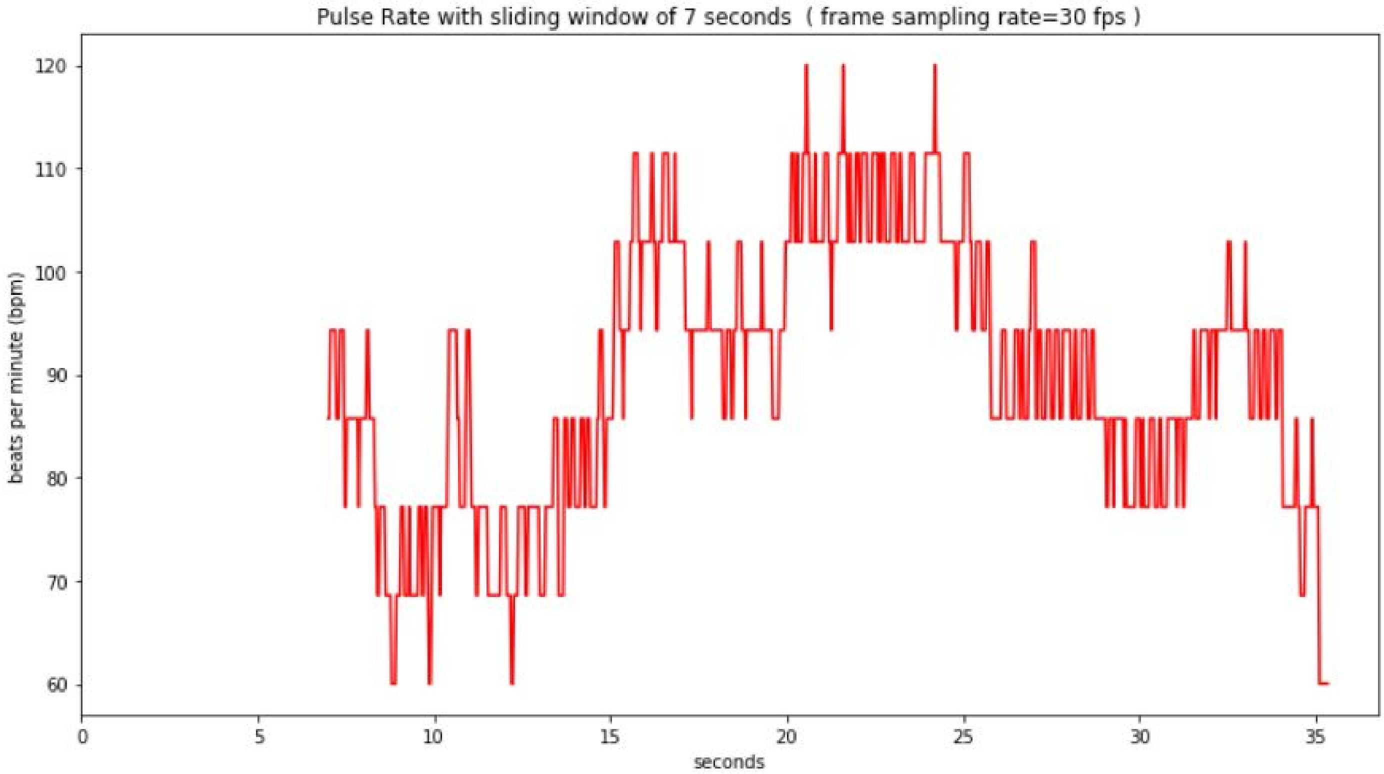
- Average pulse rate = 95 bpm




```
The average pulse rate (in BPM) by time series analysis  95.02098371663608
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

- Average pulse rate =89 bpm



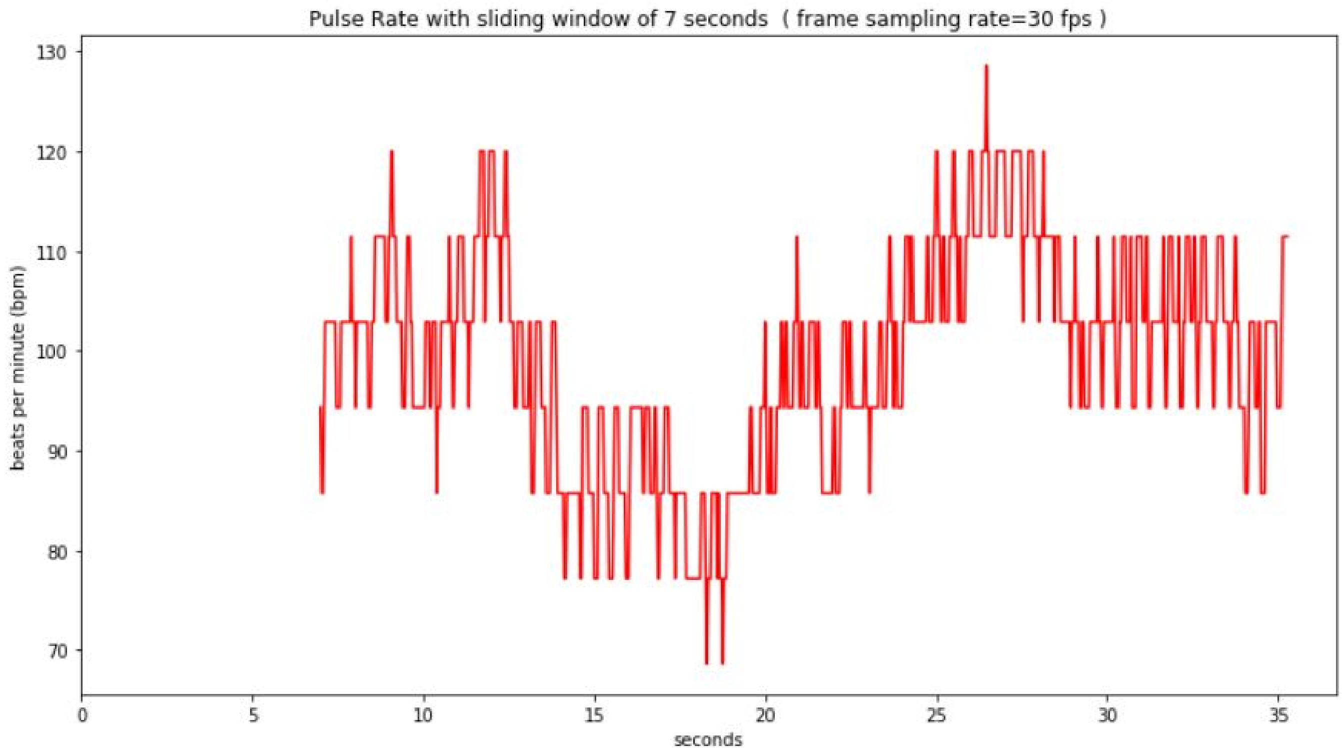

```
The average pulse rate (in BPM) by time series analysis  89.91438643612553
```

Activity : Resting in sunlight

- Video name : "resting_sunlight.mp4"
- Video duration : "00:00:45" (I have trimmed starting 5 and ending 5 seconds because they may contain more noise . This is done in the code.)
- Sliding window used for bpm = 7 sec (last 7 seconds)

1. Every pixel is taken into consideration

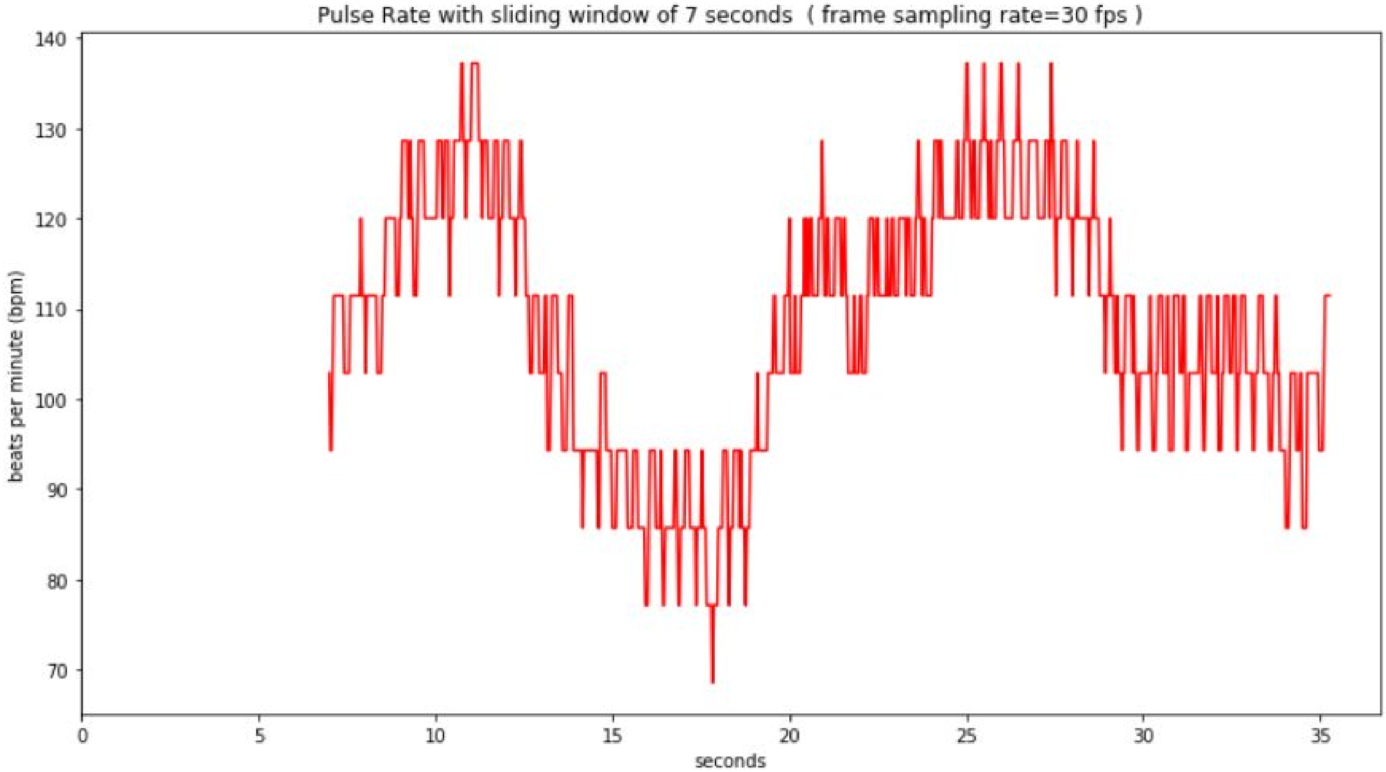
- Average pulse rate = 99 bpm




```
The average pulse rate (in BPM) by time series analysis  99.39424533064117
```

1. 1 out of every 10 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

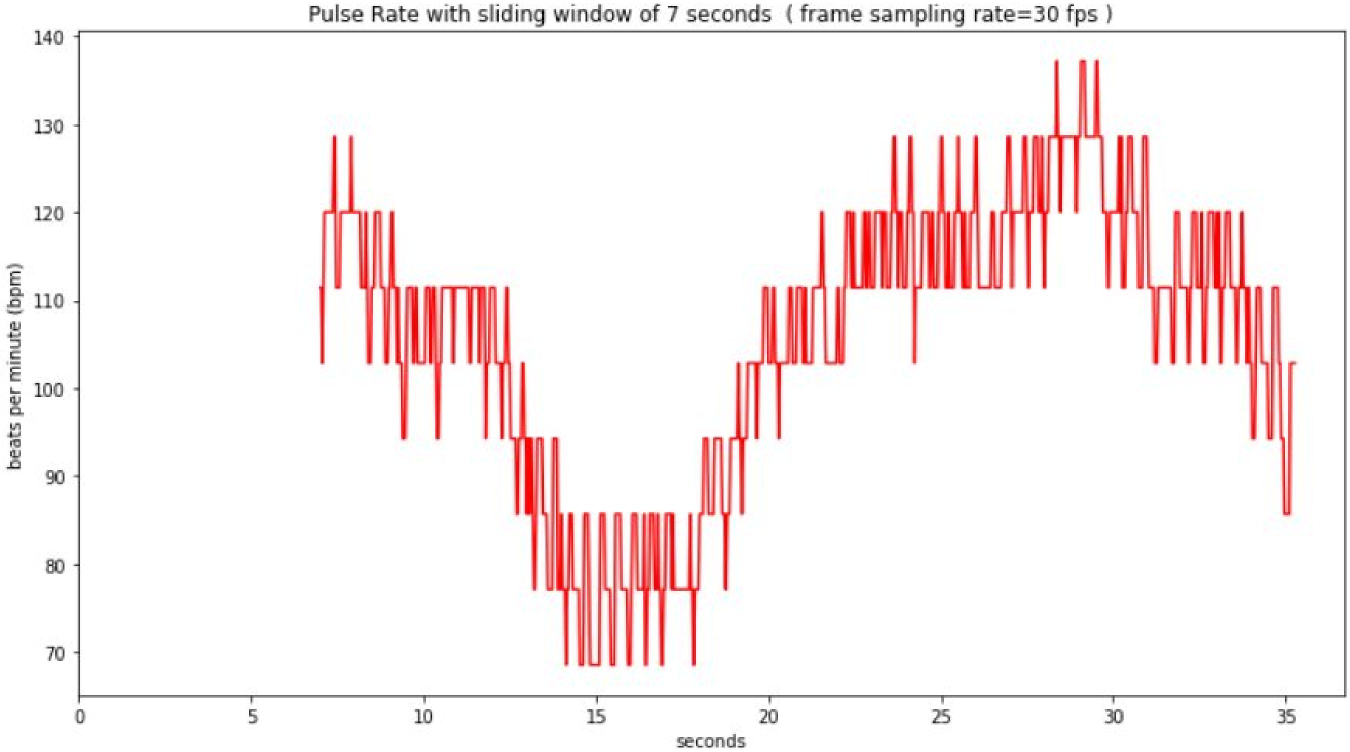
- Average pulse rate =108 bpm




```
The average pulse rate (in BPM) by time series analysis  108.79353861685995
```

2. 1 out of every 50 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

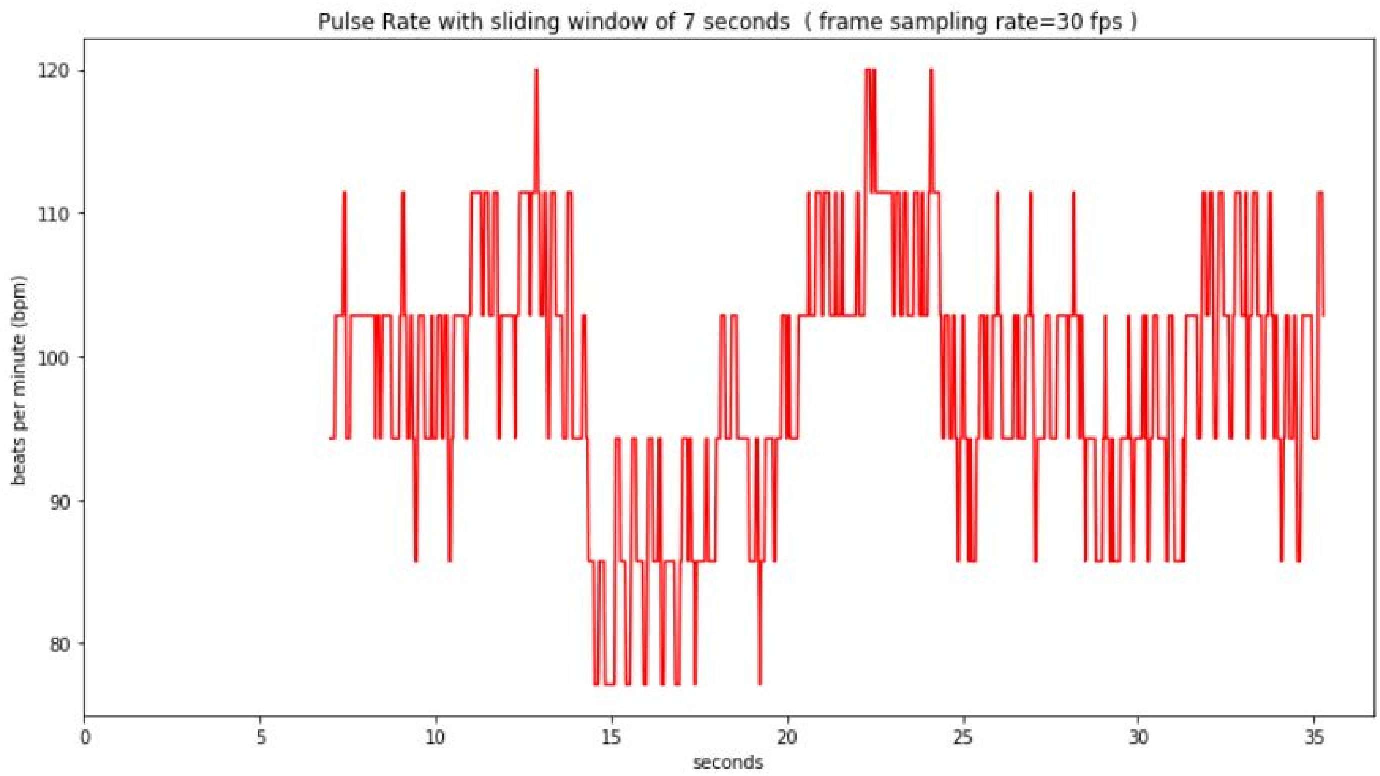
- Average pulse rate =105 bpm




```
The average pulse rate (in BPM) by time series analysis  105.91620393740556
```

3. 1 out of every 100 pixel is choosen along x-axis and 1 every 10 pixel is choosen along y-axis

- Average pulse rate =98 bpm




```
The average pulse rate (in BPM) by time series analysis  98.47551741544724
```

Conclusion of Bumping the SNR by removing the boundary pixels:
Almost in all the cases above, there is a improvement in the SNR after removing the boundary pixels. The heartbeat bpm seems to stabilise by removing the boundary pixels.

Solution: (c) **Trade-off between Low resolution vs Low FPS**
As per the results of the experiments completed in part(b) and part(c), their is not such relationship visible between low resolution and low FPS .
Tradeoff can be observed when we will have fixed amount of pixels , then tradeoff can be experienced as follows :
1. If we decrease the number of frames, then pixel per frame will increase. In that case the quality of our frames will be high and their can be better results as SNR can be high.
2. If we decrease the pixels per frame , then the number of frames will increase and the bpm can be measured with much higher precision. But the condition is our device should be able to capture at that fps rate.