

Python data one

Daten einlesen, anschauen, ...

engelbert gruber

10.12.2020

Rev. 0.3

INTRODUCTION

Ich wollte EKG-Daten auswerten ohne vorher ein EKG-Messgerät bauen zu müssen und ohne eine Messung machen zu müssen.

Im google “ecg signal sample data” und dann bekomme ich:

[ECG signals \(1000 fragments\)](https://physionet.org/about/database/), <https://physionet.org/about/database/>, [PTB-XL, a large publicly available electrocardiography dataset](#), [TELE ECG Database: 250 telehealth ECG records](#), ...

MATERIALS

1. Ein Dataset von TELE ECG
2. PC mit python3
3. Interesse

DIE DATEN

Ich habe die erste Datei von TELE ECG heruntergeladen “100_120.dat”

Das ist eine Textdatei mit fast 20-tausend Zeilen, die so anfängt

```
3.635870,0,1,1
3.709130,0,1,1
4.007597,0,1,1
4.498711,0,1,1
5.236739,0,1,1
5.510786,0,1,1
```

Das scheint eine CSV-Datei zu sein. CSV ist die Abkürzung für Comma Separated Values.

Csv Modul Versuch eins

Python hat ein csv-Modul ... ich probiere:

```
>>> import csv
>>> dir(csv) # zeigt den Inhalt des csv-Moduls
['Dialect', 'DictReader', 'DictWriter', 'Error', 'QUOTE_ALL', ...
```

Zu viel, aber da steht “reader” und ich will ja eine Datei lesen, also probiere ich weiter

```
>>> c = csv.reader('100_120.dat')
>>> for ln in c: print(ln)
...
['1']
['0']
['0']
['_']
['1']
['2']
['0']
['.']
['d']
['a']
['t']
```

“for ln in c: print(ln)” holt das nächste Ding von “c” bis es nichts mehr gibt.

Ich denke “c” sollte mir jeweils eine Zeile geben, deshalb gebe ich den Namen “ln”

(Abkürzung für “line”) und gebe das dan mit “`print(ln)`” aus.

Leider bekomme ich nur den Dateinamen und nicht den Inhalt der Datei.

```
>>> for ln in "100_120.dat": print(ln)
...
1
0
0
-
1
2
0
.
d
a
t
```

Das schaut aber anders aus. Wenn man das csv-Modul verwendet bekommt man eine Liste (array) in der der Character ist “['1']” .

Hilfe

Dann lesen wir eben etwas Hilfetext: google “python csv”

Die erste Funktion die in der Dokumentation steht ist :

```
csv.reader(csvfile, dialect='excel', **fmtparams)
```

Parameter 1: csvfile

Parameter 2: dialect='excel' hat den defaultwert 'excel', wenn wir nichts angeben wird das angenommen

Parameter 3: “**fmtparams” schaut kompliziert aus, das schauen wir uns an wenn es sein muss.

Csv reader Versuch zwei

Parameter 1: csvfile ist offensichtlich nicht der Name der Datei.

“file” ist leider nicht eindeutig. Das meint eine Datei auf der Platte oder im Anhang einer Email oder im Programmierkontext eine Variable die den Zugriff auf den Inhalt einer

Datei ermöglicht. Zum Beispiel in python:

```
>>> f = open("100_120.dat")
>>> f.readline()
'3.635870,0,1,1\n'
>>> f.readline()
'3.709130,0,1,1\n'
```

“f” ist dann mein file-Objekt von dem ich mit “readline” jeweils die nächste Zeile lese.

Ich probiere das mit csv.reader:

```
>>> f = open("100_120.dat")
>>> c = csv.reader(f)
>>> f = open("100_120.dat")
...
['-1.174875', '0', '1', '1']
['-1.172161', '0', '1', '1']
['-1.174875', '0', '1', '1']
['-1.180301', '0', '1', '1']
['-1.185728', '0', '1', '1']
```

Da werden jetzt die 20-tausend Zeilen ausgegeben. Die Zeilen sind schon zerlegt in die vier Werte, jede Zeile ist eine Liste ... von Strings.

DATEN ANSCHAUEN

Zur Visualisierung von Daten mit python ist das erste in der Liste immer “matplotlib”.

“Import matplotlib” findet bei mir nichts, also installiere ich das Modul.

```
python3 -m pip install matplotlib
```

Danach kann ich das Modul importieren.

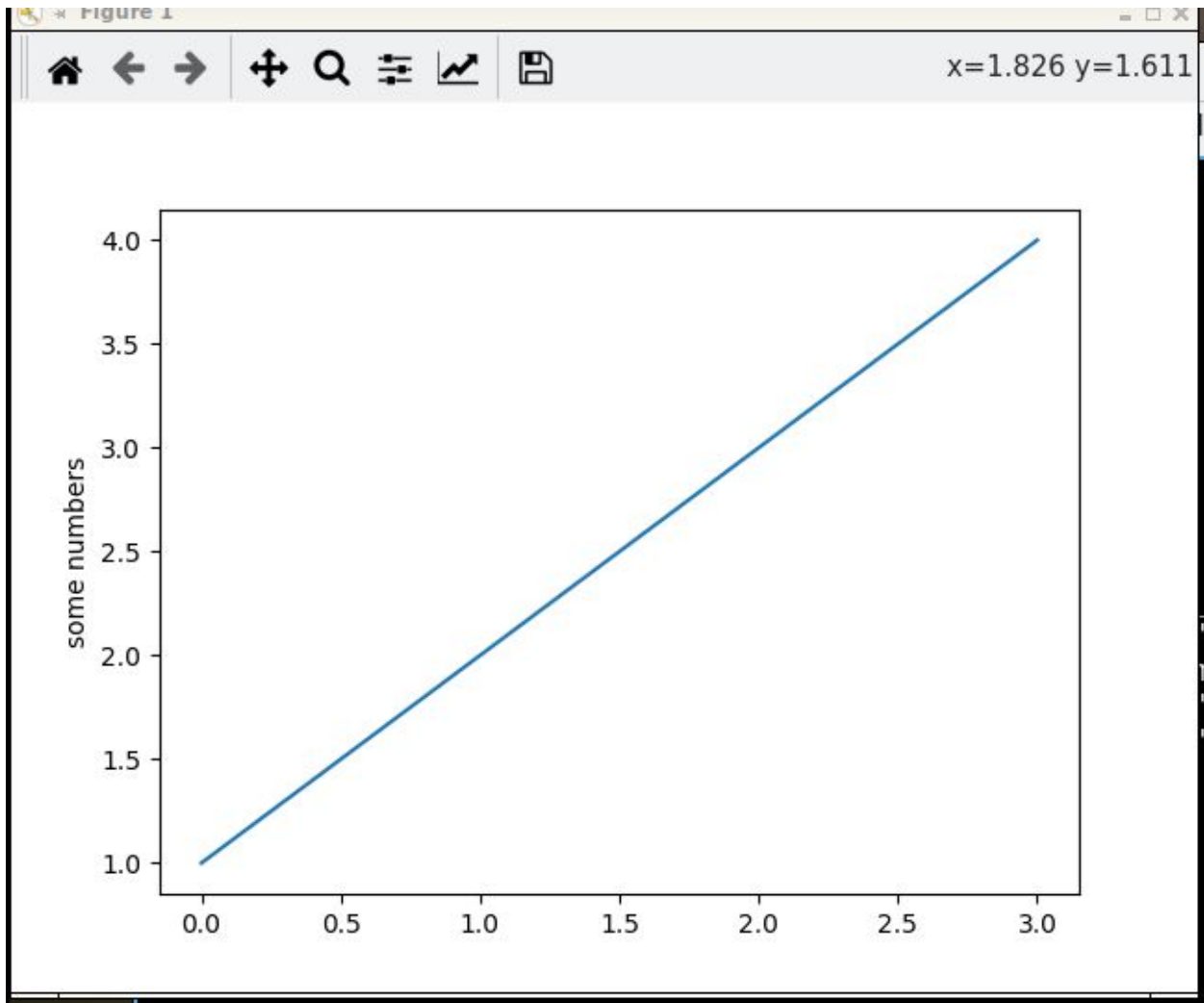
Nach dem fehlschlag beim csv-Modul, schaue ich dieses mal sofort in die Dokumentation.

<https://matplotlib.org/tutorials/index.html#introductory>

...probier ... blätter probier falscher Einstieg.

Besser eins weiter : [Pyplot tutorial](#) Beispiel eins:

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```



Versuch mit den EKG-Daten

Die Idee

```
#!/usr/bin/python3
import csv
import matplotlib.pyplot as plt
c = csv.reader( open("100_120.dat") )
# jede Zeile hat vier durch "," getrennte Werte
```

```
# z.B. -1.188441,0,1,1
# TODO data = die Werte der ersten Spalte aus der Datei
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

Werte einlesen

Wir brauchen ein Array (eine Liste) die nur die Werte der ersten Spalte enthält.

```
import csv
data = []
for ln in csv.reader(open('100_120.dat')):
    data.append( float(ln[0]) )
```

Wir brauchen Zahlenwerte “float” und nicht Zeichenketten “strings”.

“Data.append” hängt den Wert an die Liste “data” an.

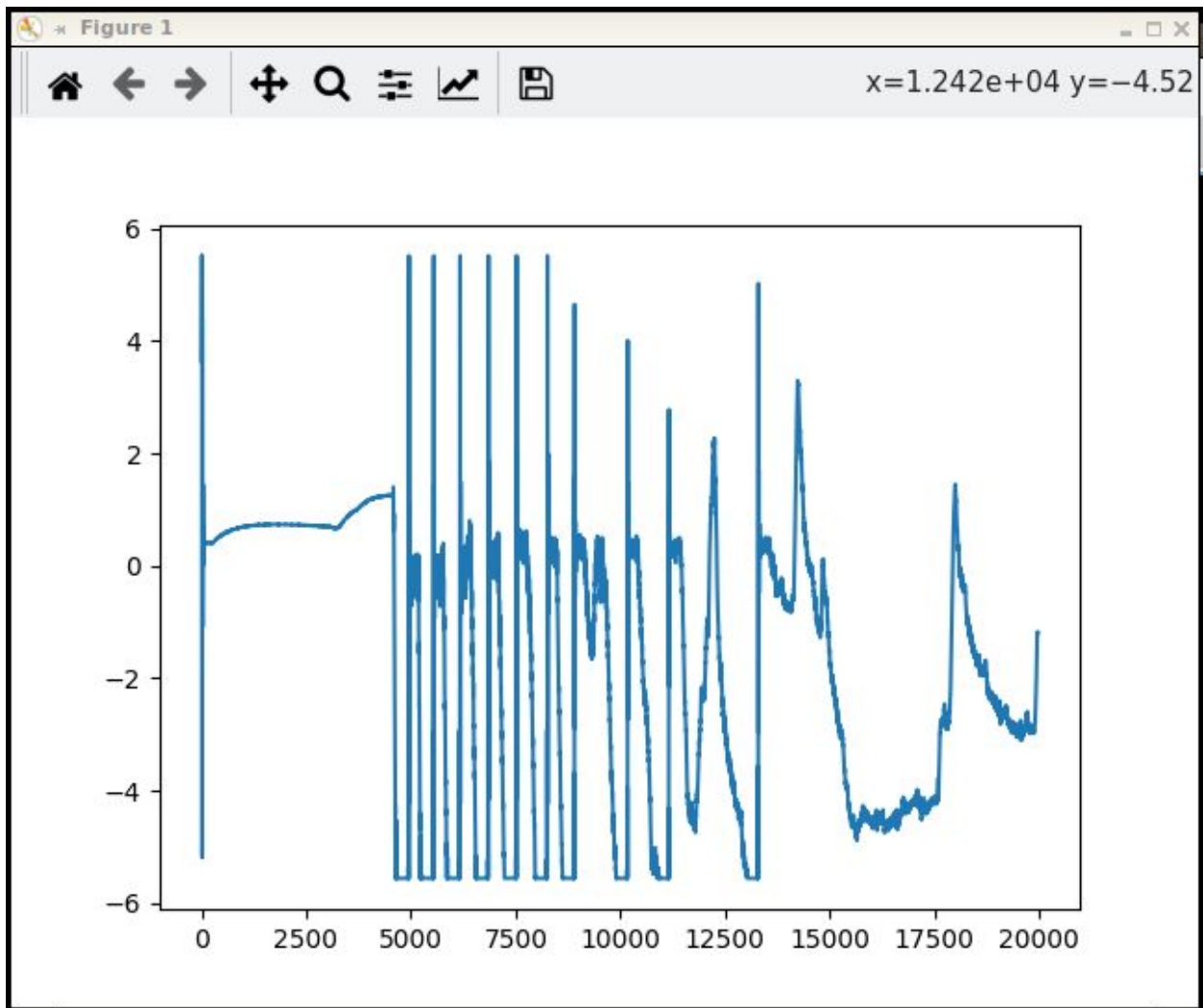
Danach enthält “data” :

```
>>> len(data)
19971
```

19971 Werte.

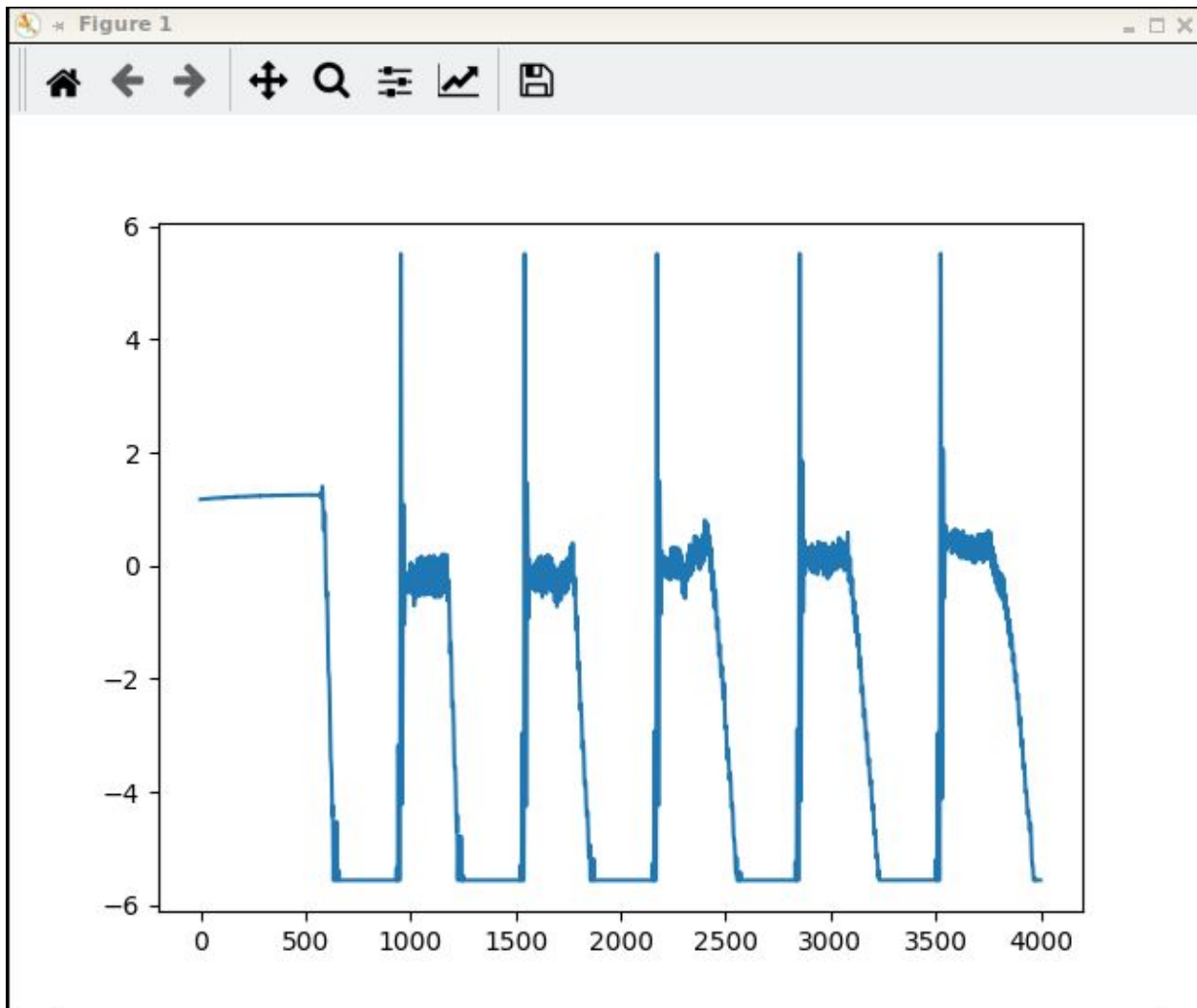
Werte darstellen

```
#!/usr/bin/python3
import csv
import matplotlib.pyplot as plt
data = []
for ln in csv.reader(open('100_120.dat')):
    data.append( float(ln[0]) )
plt.plot(data)
plt.show()
```



Besser nur zwischen 4000 und 8000:

```
plt.plot(data[4000:8000])  
plt.show()
```



Time waits for ...

Anstatt der Messpunkt/Samplenummer brauchen wir die Zeit um die Pulsfrequenz ausrechnen zu können. Schauen wir auf [TELE ECG Database: 250 telehealth ECG records](#) nach ob eine Samplerate angegeben ist.

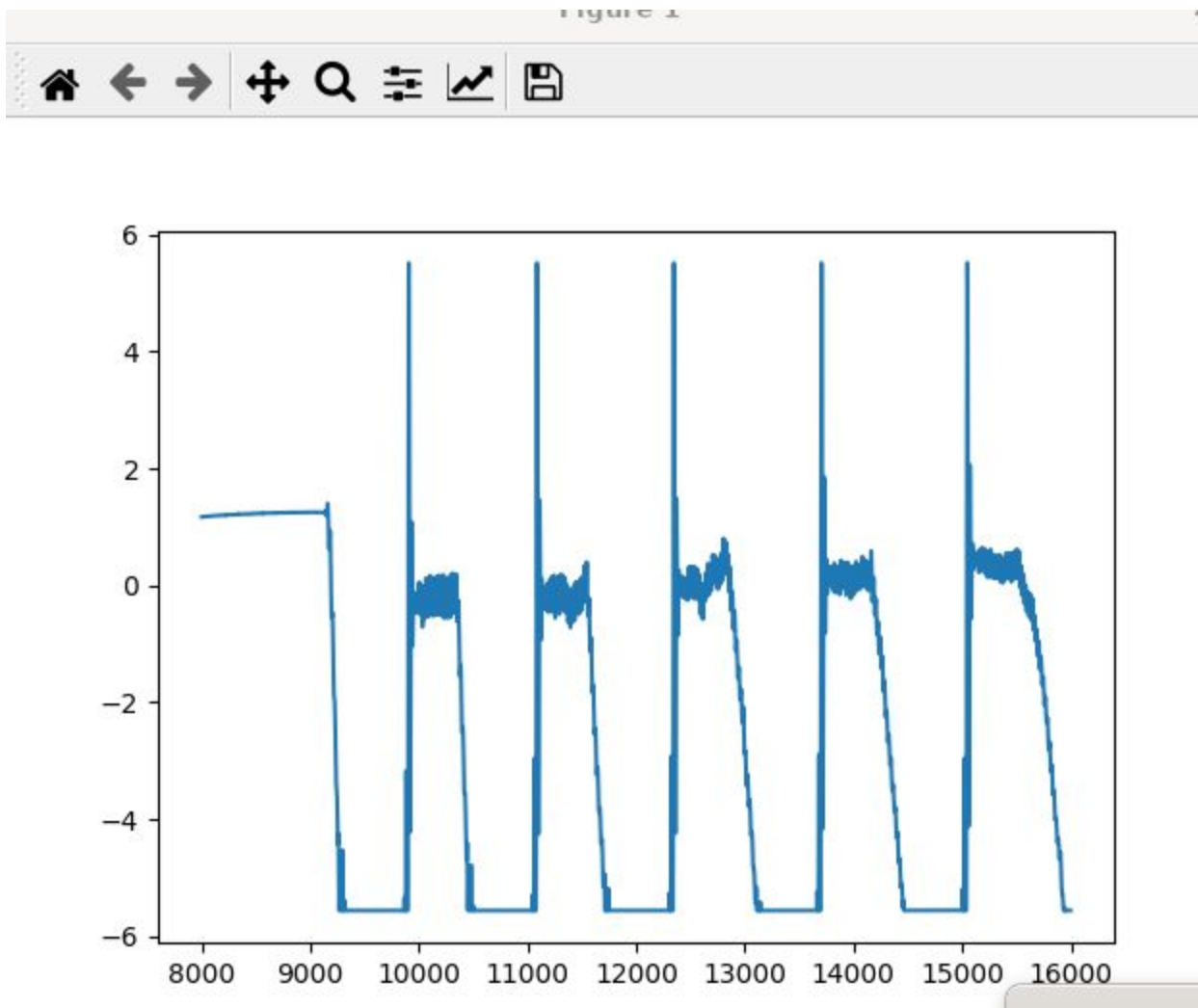
This ECG is sampled at a rate of 500 Hz using

Bei 1000 ist eine steigende Flanke, bei 1500 die nächste. 500 Hz bedeutet 500 Samples pro Sekunde, ungefähr 1Hz Pulsfrequenz ... kann stimmen (1kHz wäre ... Tinnitus?) .

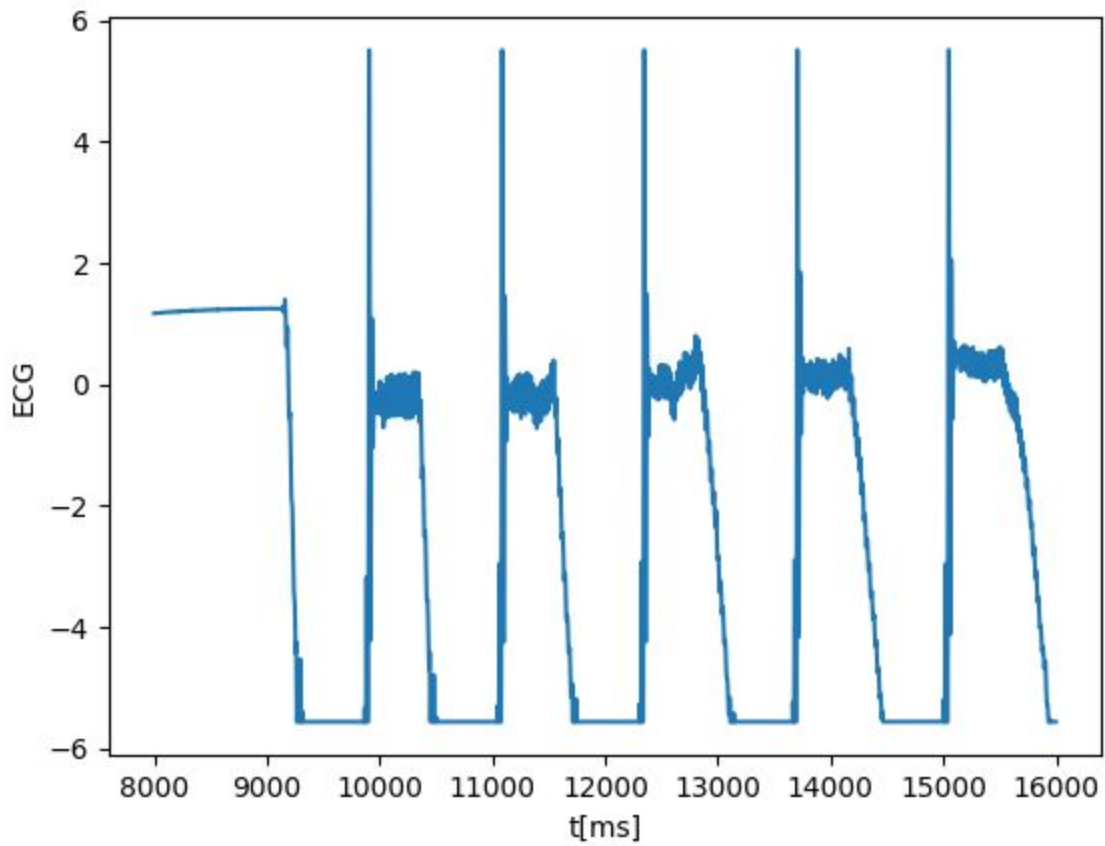
Wir benötigen ein Array in dem die Zeitwerte stehen.

Das offensichtlichste ist in der ""-Schleife das Array mit den Zeitwerten füllen.


```
#!/usr/bin/python3
import csv
import matplotlib.pyplot as plt
data = []
time_in_ms = []
for ln in csv.reader(open('100_120.dat')):
    time_in_ms.append( len(time_in_ms) * 2 )
    data.append( float(ln[0]) )
plt.plot(time_in_ms[4000:8000],data[4000:8000])
plt.show()
```



Und Beschriften



```
#!/usr/bin/python3
import csv
import matplotlib.pyplot as plt
data = []
time_in_ms = []
for ln in csv.reader(open('100_120.dat')):
    time_in_ms.append( len(time_in_ms) * 2 )
    data.append( float(ln[0]) )
plt.plot(time_in_ms[4000:8000],data[4000:8000])
plt.ylabel('EKG')
plt.xlabel('t[ms]')
plt.show()
```

UND WEITER

Verschiedene Möglichkeiten weiter zu machen, falls man noch nicht probiert hat :

- for-Schleifen
- `data.append(map(float, ln))`
- `time_in_ms = [x * 2 for x in range(len(data))]`
- Es gibt Datenanalysepakete in python, so wie matplotlib zum Anzeigen.
numpy kann besser mit großen Datenmengen umgehen.
scipy ist für wissenschaftliche Berechnungen.
- oder man versucht ein EKG-Gerät zu simulieren. Die Daten aus der Datei einzeln lesen und versuchen den Puls zu detektieren.

REFERENCES

- Matplotlib: <https://matplotlib.org/tutorials/introductory/pyplot.html>
- python csv: <https://docs.python.org/3/library/csv.html>
- "Time and tide waits for no man" (Quote from Shakespeare, Julius Caesar)
- [TELE ECG Database: 250 telehealth ECG records](#)

REVISION HISTORY

- 0.3 10.12.2020 - Und weiter
- 0.2 10.12.2020 - time
- 0.1 09.12.2020 - EKG-Kurve