

# “Simple Data Visualization”

## Practical Assignment 3

Informatik 1 für Biomedical Engineering

*Technische Universität Graz*

---

---

### 1. Tasks

#### 1.1. Funktionen (1pt)

Verwenden Sie eine saubere main Funktion für Ihr Programm. Verwenden Sie auch sonst überall Funktionen, wo dies für Sie Sinn ergibt.

#### 1.2. Kommandozeilenparameter (2pt)

Ihr Programm soll mit einer Reihe von Kommandozeilenparametern umgehen müssen. Verwenden Sie erneut das `argparse`<sup>1</sup> package.

Folgende Parameter sind zu implementieren:

`-i / --in` Pfad zur Input Datei. Default: `ass_2.p`

`-o / --out` Pfad zu ihren Outputs. Default: `/`

Selbstverständlich ist zu überprüfen, ob die angegebene Input Datei vorhanden ist. Wenn nicht, dann soll Ihr Programm eine entsprechende Ausgabe machen und sich sauber (mit `exit()`) beenden.

Selbiges gilt für den `--out` Parameter. Dies muss ein Ordner sein. Überprüfen Sie, ob dies der Fall ist. Wenn nicht, beenden sie auch hier das Programm sauber (mit `exit()`).

Die Default-Werte sind zu verwenden, sollte der Parameter nicht im Aufruf vorhanden sein. Es ist nicht notwendig, zu überprüfen, ob Sie für die Pfade Lese- und Schreibrechte besitzen

#### 1.3. Kombination getrennter Signale (8pt)

Öffnen Sie das Pickle aus Assignment 2. Suchen Sie nach getrennten Records. Diese sind erkennbar durch Buchstaben am Ende des Namens (z.B.: `drive17a` und `drive17b`). Kombinieren Sie alle zusammengehörenden Teil-Records zu einem zusammenfassenden Record. Überprüfen Sie, ob die anderen Parameter zusammenpassen (samples per frame, usw). Sollten die Records nicht kompatibel sein, kombinieren Sie die Teile nicht. Bedenken Sie, dass Sie manche Werte im Record und den Signalen anpassen müssen, da sie nun mehr Einträge haben. Fehlende Daten (also eine leere Liste im `data` Feld) soll ebenfalls dazu führen, dass die Records nicht kombiniert werden.

---

<sup>1</sup><https://docs.python.org/3.6/library/argparse.html>

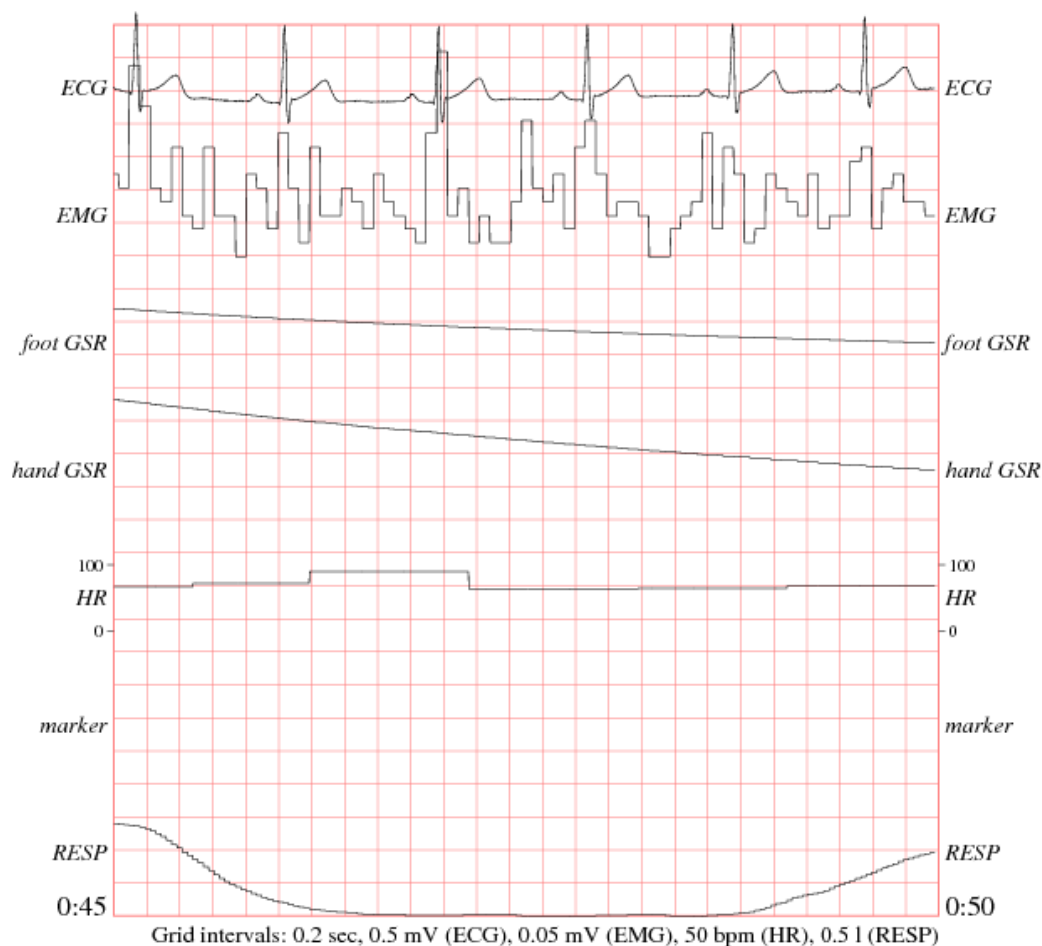
Kombinieren Sie Records nur, wenn ein a-Record (also z.B.: `drive17a`) vorhanden ist und es keine Lücken bis zum letzten Record gibt. `a,b,c,d` kann kombiniert werden, `a,c,d` aber nicht, da der Record `b` fehlt.

Wenn ein Record erfolgreich kombiniert wurde, entfernen Sie die Teil-Records aus der Datenstruktur.

Speichern Sie am Ende Ihres Programmes die Datenstruktur ein letztes mal unter dem Namen `ass_3.p` in jenem Ordner, der durch `--out` definiert wurde.

#### 1.4. Visualisierung (14pt)

Visualisieren Sie alle Records mithilfe von Matplotlib. Versuchen Sie dabei, einen Plot ähnlich dem folgenden Stil zu erstellen:



Die sensorspezifischen Informationen (Grid intervals: 0.2sec...) am unteren Ende des Beispiels sind zu ignorieren.

Erstellen Sie für jeden Record zwei Plots. Einmal mit den gesamten Sensordaten und einmal mit dem Ausschnitt von Sekunde 45 bis Sekunde 50. Zusätzlich speichern Sie jeden Plot einmal als `.png` und einmal als `.pdf`. (Also insgesamt 4 Plots pro Record). Die Namen der Plots soll folgendes Schema haben: der Name des Records + `_short` für den Ausschnitt

und `+_full` für die Visualisierung der gesamten Daten. Speichern Sie diese im Order, der mittels `--out` definiert wurde.

Beispiel: `drive01` soll folgende Plots erzeugen:

- `drive01_full.png`
- `drive01_short.png`
- `drive01_full.pdf`
- `drive01_short.pdf`

Plotten Sie nur Records, die in zumindest einem Sensor Daten geladen haben. Plotten Sie nur Sensoren, welche auch Daten haben. Das Verwenden von **seaborn** ist erlaubt!

Die Stufenstruktur bei Linienplots erhalten Sie mit `"pyplot.step()"`. Sie können `time.strftime()` verwenden, um die Indizes zu Timestamps umzuwandeln. Beachten Sie hierbei die Signalfrequenz!

**Hinweis:** Passen Sie bei den `_short` Plots die `y_limits` so an, dass die Daten des Ausschnitts den verfügbaren Platz ausnutzen.

### 1.5. BONUS: Visualisierung mit Bokeh(2pt\*)

Erzeugen Sie ähnliche Plots wie im letzten Task. Verwenden Sie jedoch das **bokeh** package. Die gespeicherten Dateien sollen im `.html` Format sein. Es gibt hier also pro Record 2 Dateien (einmal `_full` und einmal `_short`. Speichern Sie die Plots erneut im definierten `--out` Ordner.

### 1.6. BONUS: Bokeh Hovertool(3pt\*)

Verwenden Sie das bokeh Hovertool<sup>2</sup> um ein Overlay zu erzeugen. Dort sollen die Werte aller Sensoren und der genaue Timestamp angezeigt werden. Benutzen Sie hierfür auch die `datasource` Objekte.

## 2. Packages

Folgende Packages sind für dieses Assignment zugelassen und/oder notwendig:

- `argparse`, `bokeh`, `collections`, `math`, `matplotlib`, `numpy`, `os`, `pickle`, `scipy`, `seaborn`, `struct`, `sys`, `time`

## 3. Beschränkungen

- Fügen Sie keine nutzlosen Komponenten Ihrer Abgabe hinzu
- Importieren Sie keine extra zusätzlichen packages
- Verwenden Sie keine nicht angeführten Kommandozeilenparameter

---

<sup>2</sup><https://bokeh.pydata.org/en/latest/docs/reference/models/tools.html>

## 4. Datei Header

All Ihre Quelldateien in Ihrer Abgabe müssen gleich zu Beginn einen Kommentar mit folgenden Informationen enthalten:

- Author: – Ihr Name
- MatNr: – Ihre Matrikelnummer
- Description: – Generelle Beschreibung der Datei
- Comments: – Kommentare, Erklärungen, usw

Bitte einfach den folgenden Code in Ihre Dateien kopieren und den Inhalt anpassen. Je nach PDF-Reader müssen Sie eventuell die Leerzeichen/Einrückungen per Hand anpassen.

**Beispielheader:**

```
#####  
# Author:      Patrick Kasper  
# MatNr:       0730294  
# Description:  The main file. Assignment only has 1 file...  
# Comments:    This is the example comment. I just made it a bit  
#              longer so it spans across multiple lines.  
#####
```



## 5. Coding Standard

Für diese Lehrveranstaltung orientieren Sie sich am offiziellen PEP 8 Standard<sup>3</sup>. Dieser Beschreibt grundsätzliche Formalitäten im Bezug auf Ihren Code. Folgendes ist besonders zu Beachten:

**Sprache.** Code schreibt man in Englisch. Im internationalen Zeitalter ist es notwendig, dass auch jemand am anderen Ende der Welt verstehen kann, was Sie programmiert haben. Ihr gesamter Quellcode muss daher auf Englisch geschrieben sein. Dies betrifft sowohl die Kommentare also auch Variablennamen und Ähnliches.

**Leerzeichen statt Tabulatoren.** Python basiert auf Einrückungen, anstatt auf geschwungenen Klammern. Theoretisch gibt es die Möglichkeit, Leerzeichen (spaces) oder Tabulatoren (tabs) zu verwenden. PEP8 schreibt aber, 4 Leerzeichen als Einrückungen vor. Die meisten Python Programmierumgebungen werden automatisch 4 Leerzeichen einfügen, wenn Sie auf die Tabulator-Taste drücken.

**Sprechende Namen.** Verwenden Sie kurze, aber sprechende Namen für Ihre Variablen, Funktionen, usw. Es muss eindeutig aus dem Namen hervorgehen, was die Aufgabe des Elements ist. Für simple Iterationen kann ein einfaches `i` ausreichend sein, aber dies kann schnell zu Chaos führen. Sollten Sie Variablen haben, die keine Aufgabe haben und nicht verwendet werden, schreibt PEP 8 vor, einen einfachen Unterstrich (`_`) zu verwenden.

---

<sup>3</sup><https://www.python.org/dev/peps/pep-0008/>

**72 Zeichen Zeilenlänge.** In Ihrem Code darf keine Zeile mehr als 72 Zeichen haben. Dies ist Voraussetzung, um den Code ausdrucken, oder auf allen Geräten darstellen zu können. PEP 8 limitiert in manchen Fällen die Länge auf 79 Zeichen und in anderen auf 72. Bitte halten Sie sich in Ihrem Code an die untere Schranke (72). **Hinweis:** Dies inkludiert die Einrückungen mittels Leerzeichen!

## 6. Automatisierte Tests

Ihr Programm wird mit dem folgenden Befehl ausgeführt:

```
>python assignment_3.py <command line params>
```

Vergewissern Sie sich, dass Ihre Abgabe allen Beschränkungen, die in dieser Angabe erwähnt sind, konform ist.

## 7. Abgabe

### 7.1. Deadline

12. Jänner 2018 um 23:59:59

Eine Spätabgabe ist nicht vorgesehen. Ausnahme bilden hier Notfälle. Sollte das Abgabesystem nicht online sein, verlängert sich die Deadline automatisch um 24 Stunden. Sollten Sie, aus diversen Gründen, nicht in der Lage sein, Ihre Abgabe hochzuladen, kontaktieren Sie Ihren Tutor *VOR* der Deadline. (**Hinweis:** Urlaub oder Ähnliches wird nicht als Grund akzeptiert!)

### 7.2. Hochladen der Abgaben

Assignments werden stets als Archive abgegeben. Erlaubt sind hier die Formate *.zip*, und *.tar.gz*.

Zusätzlich zu ihren Quelldateien, soll Ihre Abgabe auf eine Datei namens `readme.txt` beinhalten. Das Vorhandensein der Datei ist Pflicht, ihr Inhalt aber optional. Sie soll folgenden Inhalt haben: (i) Die Zeit, die Sie benötigt haben, um die Aufgabenstellung zu absolvieren. (ii) Feedback, wo Sie Probleme hatten. Ihr Feedback ermöglicht es, verbreitete Probleme zu erkennen und die Vorlesung und Tutoriumseinheiten entsprechend anzupassen. Abgaben erfolgen auf der Palme Website. Bitte prüfen Sie vor der Abgabe diese Kriterien:

- Datei- und Ordnerstruktur (siehe unten)
- Kommentarheader in jeder Quelldatei
- Coding Standard

### 7.3. Struktur der Abgabe

```
├─ assignment_3.zip (or assignment_3.tar.gz)
│   └─ assignment_3.py
│       └─ readme.txt
```

Bitte fügen Sie den Datensatz und das input pickle **NICHT** Ihrer Abgabe hinzu. Diese werden automatisch bei den Tests in die entsprechenden Ordner kopiert.