{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"name": "Copy of CS480 Assignment 1.ipynb",

"provenance": [],

"collapsed\_sections": [],

"include\_colab\_link": true

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

}

},

"cells": [

{

"cell\_type": "markdown",

"metadata": {

"id": "view-in-github",

"colab\_type": "text"

},

"source": [

"<a href=\"https://colab.research.google.com/github/evalsamis/cs480student/blob/main/Copy\_of\_CS480\_Assignment\_1.ipynb\" target=\"\_parent\"><img src=\"https://colab.research.google.com/assets/colab-badge.svg\" alt=\"Open In Colab\"/></a>"

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "asaW4NA0jK20"

},

"source": [

"\n",

"# Assignment 1\n"

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Xv1Wc4ECfZwV",

"outputId": "d367d735-5b7f-4362-bdc7-8594147539ab"

},

"source": [

"%pylab inline"

],

"execution\_count": 10,

"outputs": [

{

"output\_type": "stream",

"text": [

"Populating the interactive namespace from numpy and matplotlib\n"

],

"name": "stdout"

},

{

"output\_type": "stream",

"text": [

"/usr/local/lib/python3.6/dist-packages/IPython/core/magics/pylab.py:161: UserWarning: pylab import has clobbered these variables: ['plot']\n",

"`%matplotlib` prevents importing \* from pylab and numpy\n",

" \"\\n`%matplotlib` prevents importing \* from pylab and numpy\"\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "LD\_A0Idy5NY0"

},

"source": [

"from scipy.misc import electrocardiogram # scipy is a python scientific computing library\n",

"ecg = electrocardiogram() # let's load the scipy ECG dataset"

],

"execution\_count": 20,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "NFw0i1\_4-ixY"

},

"source": [

"SAMPLING\_RATE = 360 # this data was sampled with 360 Hz"

],

"execution\_count": 21,

"outputs": []

},

{

"cell\_type": "markdown",

"metadata": {

"id": "MXCidk9L\_VG8"

},

"source": [

"\*\*Task 1:\*\* Plot the data in the range from 9 to 11 seconds! [30 Points]"

]

},

{

"cell\_type": "code",

"metadata": {

"id": "s2n86tUP\_pMZ"

},

"source": [

"# TODO your plotting code goes here\n",

"# please take the sampling rate into account to only plot the range\n",

"# between 9 and 11 seconds"

],

"execution\_count": 22,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "J2E2b5Z\_mGSh",

"outputId": "5a82947d-0614-481f-843e-8bb7cd0930bb"

},

"source": [

"ecg[0:10] # first ten values in ecg (indexing)"

],

"execution\_count": 23,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([-0.245, -0.215, -0.185, -0.175, -0.17 , -0.17 , -0.185, -0.17 ,\n",

" -0.16 , -0.15 ])"

]

},

"metadata": {

"tags": []

},

"execution\_count": 23

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "6qM8YTMXnoBR"

},

"source": [

"from9to11seconds = ecg[8\*SAMPLING\_RATE:10\*SAMPLING\_RATE]"

],

"execution\_count": 25,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "V8r7nM6Ao9yx"

},

"source": [

"from10to12seconds = ecg[9\*SAMPLING\_RATE:11\*SAMPLING\_RATE]"

],

"execution\_count": 27,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 282

},

"id": "qErxYrwfohIE",

"outputId": "954c164d-7a8e-4c89-8994-79045a7c06b2"

},

"source": [

"plt.plot(from9to11seconds)"

],

"execution\_count": 26,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"[<matplotlib.lines.Line2D at 0x7f16ac76e599>]"

]

},

"metadata": {

"tags": []

},

"execution\_count": 27

},

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 282

},

"id": "tLJ6PwFSpFOl",

"outputId": "f5bd5f5e-9633-4e8e-8aa3-bfdae90c1dbf"

},

"source": [

"plt.plot(from10to12seconds)"

],

"execution\_count": 26,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"[<matplotlib.lines.Line2D at 0x7f16ac753470>]"

]

},

"metadata": {

"tags": []

},

"execution\_count": 27

},

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "M7Y0hThdAGte"

},

"source": [

"\*\*Task 2:\*\* Use Neurokit to detect the R spikes! [30 Points]"

]

},

{

"cell\_type": "code",

"metadata": {

"id": "uUlv03HQ6zmn"

},

"source": [

"# now can you see the P Q R S T waves? it's actually a pretty good signal despite\n",

"# the shift in the baseline"

],

"execution\_count": 28,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "CkvNQr36oFor",

"outputId": "2337c94b-998d-4d21-de17-18ed4e23b2da"

},

"source": [

"# we now need to install the neurokit2 library\n",

"!pip install neurokit2\n",

"import neurokit2 as nk"

],

"execution\_count": 29,

"outputs": [

{

"output\_type": "stream",

"text": [

"Requirement already satisfied: neurokit2 in /usr/local/lib/python3.6/dist-packages (0.1.0)\n",

"Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages (from neurokit2) (1.4.1)\n",

"Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages (from neurokit2) (1.1.5)\n",

"Requirement already satisfied: scikit-learn in /usr/local/lib/python3.6/dist-packages (from neurokit2) (0.22.2.post1)\n",

"Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from neurokit2) (1.19.5)\n",

"Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-packages (from neurokit2) (3.2.2)\n",

"Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.6/dist-packages (from pandas->neurokit2) (2.8.1)\n",

"Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-packages (from pandas->neurokit2) (2018.9)\n",

"Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-packages (from scikit-learn->neurokit2) (1.0.0)\n",

"Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-packages (from matplotlib->neurokit2) (0.10.0)\n",

"Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->neurokit2) (2.4.7)\n",

"Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/dist-packages (from matplotlib->neurokit2) (1.3.1)\n",

"Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (from python-dateutil>=2.7.3->pandas->neurokit2) (1.15.0)\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "2SyzFo7ZE4gM"

},

"source": [

"# we can use neurokit's function to detect the R spikes\n",

"\_, rpeaks = nk.ecg\_peaks(ecg, sampling\_rate=SAMPLING\_RATE)"

],

"execution\_count": 30,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "7eee64Di-Jvx",

"outputId": "a19321dd-0520-40ed-de1c-0c905ab7a536"

},

"source": [

"# look at rpeaks - the numbers indicate the samples that are R spikes\n",

"print(rpeaks['ECG\_R\_Peaks'][0:10]) # here only the first ten"

],

"execution\_count": 31,

"outputs": [

{

"output\_type": "stream",

"text": [

"[ 125 343 552 748 944 1130 1317 1501 1691 1880]\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 265

},

"id": "Cok\_u4AI-JyW",

"outputId": "f5b589eb-6973-40ad-ae41-adecaf81f367"

},

"source": [

"# you can also plot the first two detected R spikes\n",

"plot = nk.events\_plot(rpeaks['ECG\_R\_Peaks'][0:2], ecg[0:SAMPLING\_RATE])"

],

"execution\_count": 32,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "C4QphQFI-J1O"

},

"source": [

"# TODO Can you write code to plot the R spikes between 9 and 11 seconds?\n",

"# Hint: one solution could be to just detect peaks in the range you specified for Task 1\n",

"# Hint 2: you might need to extend the range by +- 1 second to see the five peaks from Task 1"

],

"execution\_count": 33,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "diNT6yCYtwo7"

},

"source": [

"# extending the range by 1 second on each side seemed to large, so I extended the range by 1/3 of the sampling rate on each side\n",

"new\_range = ecg[9\*SAMPLING\_RATE-(SAMPLING\_RATE//3):11\*SAMPLING\_RATE+(SAMPLING\_RATE//3)]"

],

"execution\_count": 34,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "pI-d1B5csCYS"

},

"source": [

"\_, rpeaks\_9to11 = nk.ecg\_peaks(new\_range, sampling\_rate=SAMPLING\_RATE)"

],

"execution\_count": 35,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "9pDovL\_LtYgD",

"outputId": "a3562429-147e-49f7-d3ac-3d49d4072db7"

},

"source": [

"plot = nk.events\_plot(rpeaks\_9to11['ECG\_R\_Peaks'], new\_range)\n",

"plt.xlabel(\"time\")"

],

"execution\_count": 36,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Text(0.5, 0, 'time')"

]

},

"metadata": {

"tags": []

},

"execution\_count": 36

},

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Nvq66z4KGQqt"

},

"source": [

"\*\*Task 3:\*\* Detect Heartrate. [15 Points]"

]

},

{

"cell\_type": "code",

"metadata": {

"id": "Lysq6bR5oFmD"

},

"source": [

"# TODO Does the patient have a regular heart rate? If yes, which one? (estimate is fine)\n",

"# Hint: you can use the rpeaks for that "

],

"execution\_count": 37,

"outputs": []

},

{

"cell\_type": "markdown",

"metadata": {

"id": "31SpxZDeNRlE"

},

"source": [

"We can calculate the heart rate manually with the rpeaks of the ecg by calculating their distance but the neurokit2 package provides a helpful function called `ecg\_rate()` to return the heart rate in a numpy array."

]

},

{

"cell\_type": "code",

"metadata": {

"id": "1hlUFeNBIQ2Y"

},

"source": [

"rpeaks, info = nk.ecg\_peaks(ecg, sampling\_rate=SAMPLING\_RATE)\n",

"ecg\_rate = nk.signal\_rate(rpeaks, sampling\_rate=SAMPLING\_RATE, desired\_length=len(rpeaks))"

],

"execution\_count": 38,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "bWrqXeuD00yc",

"outputId": "7068c5f8-1253-43b5-ee91-f3f8bc1d2b6f"

},

"source": [

"print(ecg\_rate)"

],

"execution\_count": 39,

"outputs": [

{

"output\_type": "stream",

"text": [

"[100.63668257 100.63668257 100.63668257 ... 81.81818182 81.81818182\n",

" 81.81818182]\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 296

},

"id": "Ocpxx5SEBAEs",

"outputId": "84e0868f-04bd-41cd-e221-bea9c786a281"

},

"source": [

"plt.plot(ecg\_rate) # plotted heart rate\n",

"plt.xlabel(\"time\")\n",

"plt.ylabel(\"heart rate\")"

],

"execution\_count": 40,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"Text(0, 0.5, 'heart rate')"

]

},

"metadata": {

"tags": []

},

"execution\_count": 40

},

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "vf0iUFBi1-iL",

"outputId": "11fe3520-a65a-4f18-cac1-d93e79aa646b"

},

"source": [

"# calculate average heart rate of the patient\n",

"numpy.mean(ecg\_rate)"

],

"execution\_count": 41,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"101.3766390964127"

]

},

"metadata": {

"tags": []

},

"execution\_count": 41

}

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "nfeYLl\_wL6lv"

},

"source": [

"The patient has heart rate 101 bpm which is higher than a resting heart rate of 60-70 bpm. We don't know the current state of the patient, we can only make assumptions and assume that the heart rate is elevated, causing tachycardia. From the plot we see a few outliers of the heart rate but eventhough it is beating around the 100 beats."

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "zvaJYO\_1Izkc"

},

"source": [

"\*\*Task 4:\*\* Detect P Q S T Peaks. [25 Points]"

]

},

{

"cell\_type": "code",

"metadata": {

"id": "4N5tbGEFIyyr"

},

"source": [

"# We know the R spikes now but what about P Q S T waves?\n",

"# TODO Can you use nk.ecg\_delineate to detect them in the range of 9 to 11 seconds?\n",

"# Hint: Look at the tutorial: https://neurokit2.readthedocs.io/en/latest/examples/ecg\_delineate.html#Locate-other-waves-(P,-Q,-S,-T)-and-their-onset-and-offset\n",

"# Hint 2: The plot will have blue, orange, green, and red circles :)"

],

"execution\_count": 42,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 265

},

"id": "UPc5ibvkBWfU",

"outputId": "2cf28c79-4cf3-4778-ed7f-f55925ac12f0"

},

"source": [

"\_, waves\_peak = nk.ecg\_delineate(new\_range, rpeaks\_9to11['ECG\_R\_Peaks'], sampling\_rate=SAMPLING\_RATE, show=True, show\_type='peaks')"

],

"execution\_count": 43,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "Yg\_kHJPrIy3p"

},

"source": [

"# TODO Does the detection work well?"

],

"execution\_count": 44,

"outputs": []

},

{

"cell\_type": "markdown",

"metadata": {

"id": "kWbj\_hj1PQyx"

},

"source": [

"No, the peaks and the plot can not be read easy. \n",

"From the documentation for ecg\_delineate() we also can clean the ecg signal before:"

]

},

{

"cell\_type": "code",

"metadata": {

"id": "SYcBtw8oCOca"

},

"source": [

"ecg\_clean = nk.ecg\_clean(new\_range)"

],

"execution\_count": 45,

"outputs": []

},

{

"cell\_type": "markdown",

"metadata": {

"id": "iw6mYYqYPZDx"

},

"source": [

"and then plot it to get a better graph, not easy to read, though:"

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 265

},

"id": "gEJSxRV5EHkI",

"outputId": "6754fb1a-2609-496d-d814-fe6d712d9188"

},

"source": [

"\_, waves\_peak = nk.ecg\_delineate(ecg\_clean, rpeaks\_9to11['ECG\_R\_Peaks'], sampling\_rate=SAMPLING\_RATE, show=True, show\_type='peaks')"

],

"execution\_count": 46,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "markdown",

"metadata": {

"id": "Y5Yra7gTJzEj"

},

"source": [

"\*\*Bonus Task:\*\* Detect P Q S T Peaks for the full signal before and after filtering. [33 Points]"

]

},

{

"cell\_type": "code",

"metadata": {

"id": "BrbtS7rxKXGQ"

},

"source": [

"# TODO As above use the nk.ecg\_delineate function to detect peaks for the whole signal\n",

"# Hint: For plotting, 1000 samples / 5 rpeaks gives a nice overview."

],

"execution\_count": 47,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "Z5Hkw0sMI-LK"

},

"source": [

"\_, rpeaks\_all = nk.ecg\_peaks(ecg, sampling\_rate=SAMPLING\_RATE)"

],

"execution\_count": 48,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 265

},

"id": "rbBbZQoTJI7f",

"outputId": "9dcf7d98-651e-40c7-fc4e-bb61bb33d9b6"

},

"source": [

"# delineate function on the whole ecg with 5 rpeaks:\n",

"\_, waves\_peak = nk.ecg\_delineate(ecg, rpeaks\_all['ECG\_R\_Peaks'][0:5], sampling\_rate=SAMPLING\_RATE, show=True, show\_type='peaks')"

],

"execution\_count": 49,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "wHk4nLQ2KoEE"

},

"source": [

"# TODO Let's filter the signal and see if the detection improves.\n",

"# Hint: You can use nk.signal\_filter to define a bandpass. \n",

"# Hint 2: Which lowcut and highcut frequencies work well? Play with it :)\n",

"# Hint 3: You don't need to re-calculate the rpeaks since they are pretty accurate in general."

],

"execution\_count": 50,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "z4Z6EPMOFPzd"

},

"source": [

"filtered\_ecg = nk.signal\_filter(ecg, lowcut=10, highcut=30, method='butterworth')"

],

"execution\_count": 51,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "i5016wt1F075"

},

"source": [

"\_, rpeaks\_filtered = nk.ecg\_peaks(filtered\_ecg, sampling\_rate=SAMPLING\_RATE)"

],

"execution\_count": 52,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 265

},

"id": "sCXTPfQ8FY1H",

"outputId": "134c0e15-dc5e-47da-f04f-05b38085dd40"

},

"source": [

"\_, waves\_peak = nk.ecg\_delineate(filtered\_ecg, rpeaks\_filtered['ECG\_R\_Peaks'][0:5], sampling\_rate=SAMPLING\_RATE, show=True, show\_type='peaks')"

],

"execution\_count": 53,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"tags": [],

"needs\_background": "light"

}

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "s64ssUs4LmJq"

},

"source": [

"# TODO Does the filtering improve the result?"

],

"execution\_count": 54,

"outputs": []

},

{

"cell\_type": "markdown",

"metadata": {

"id": "NhowUt4sLj0q"

},

"source": [

"Yes, we can see that the P, Q, S and T peaks they are visible and aligned depending on the band we choose for lowcut and highcut. 10 to 40 of different frequencies."

]

},

{

"cell\_type": "code",

"metadata": {

"id": "qElvYaCgK\_r3"

},

"source": [

"#\n",

"# Great job!!\n",

"#\n",

"# , ; , .-'\"\"\"'-. , ; ,\n",

"# \\\\|/ .' '. \\|//\n",

"# \\-;-/ () () \\-;-/\n",

"# // ; ; \\\\\n",

"# //\_\_; :. .; ;\_\_\\\\\n",

"# `-----\\'.'-.....-'.'/-----'\n",

"# '.'.-.-,\_.'.'\n",

"#jgs '( (..-'\n",

"# '-'\n",

"#"

],

"execution\_count": 57,

"