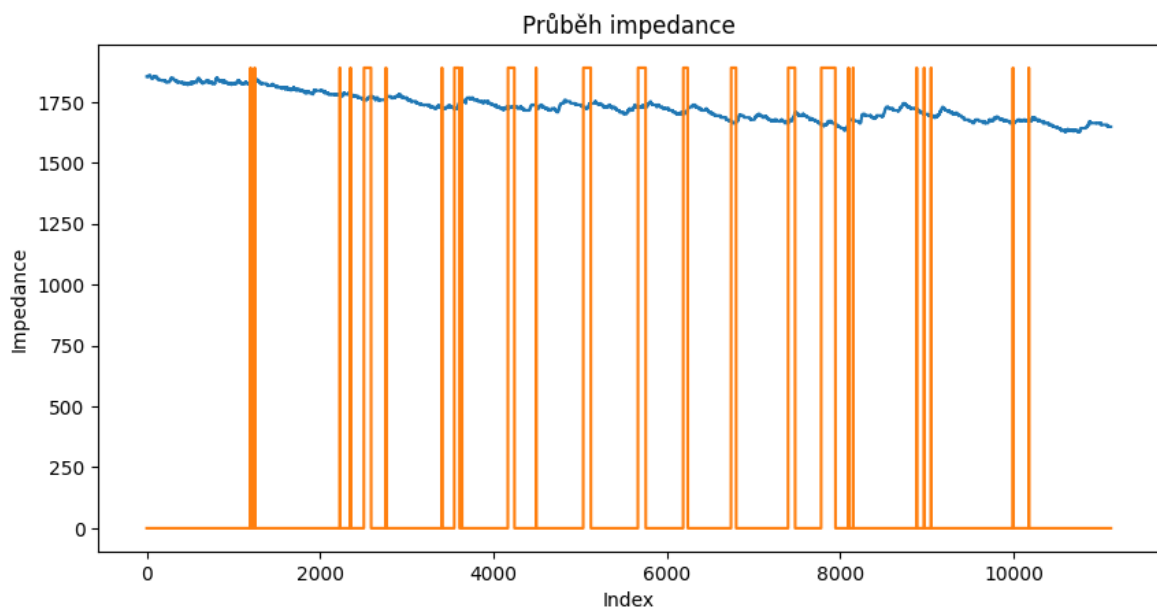


```
In [ ]: import numpy as np
import matplotlib.pyplot as plt
import pandas
import scipy as sc
pandas.set_option("display.precision", 3)
```

## Načtení dat

```
In [ ]: fileName = 'GSR_hrac1.xlsx'
dataHra2 = pandas.read_excel(fileName, sheet_name="Hra_2")
dataCopy = dataHra2
```

```
In [ ]: plt.plot(range(len(dataCopy)), dataCopy["Data"])
plt.plot(range(len(dataCopy)), dataCopy["Peak"])
figure = plt.gcf()
figure.set_figwidth(10)
plt.xlabel("Index")
plt.ylabel("Impedance")
plt.title("Průběh impedance")
plt.show()
```



## Úprava dat

```
In [ ]: # přepis stresové hodnoty 2 na 1
dataCopy.loc[(dataCopy.Stres == 2), "Stres"] = 1
impedance = list(dataCopy["Data"].copy().to_numpy())
stres = list(dataCopy["Stres"].copy().to_numpy())
```

## Pomocné funkce

```

In [ ]: def rozdELITStres(impData, stresData):
    """
    Funkce rozdělí sloupec impedancí do skupin podle stresového stavu.
    Výstupem jsou listy listů impedančních hodnot.
    """

    impedanceSkupiny = []
    aktualniSkupina = []

    for index in range(len(impData)):
        # ukončení cyklu, pokud je index na posledním řádku
        if index == len(impData) - 1:
            aktualniSkupina.append(impData[index])
            impedanceSkupiny.append(aktualniSkupina)
            break

        # zápis hodnot impedance v úseku se stejným stresovým stavem
        if stresData[index] == stresData[index + 1]:
            aktualniSkupina.append(impData[index])
        # zápis skupiny hodnot při změně stresového stavu
        elif stresData[index] != stresData[index + 1]:
            aktualniSkupina.append(impData[index])
            impedanceSkupiny.append(aktualniSkupina)
            aktualniSkupina = []

    # klidovaSkupina = impedanceSkupiny[0::2]
    # stresovaSkupina = impedanceSkupiny[1::2]
    return impedanceSkupiny[0::2], impedanceSkupiny[1::2]

def statistikaSkupin(listSkupin):

    prumerSkupin = [np.mean(skupina) for skupina in listSkupin]
    stdSkupin = [np.std(skupina) for skupina in listSkupin]
    plochaSkupin = [np.trapz(skupina, np.arange(len(skupina))) for skupina in listSkupin]
    momentSkupin = [sc.stats.moment(skupina, 2) for skupina in listSkupin]
    return {"prumer": prumerSkupin, "std": stdSkupin,
            "plocha": plochaSkupin, "moment": momentSkupin}

def entropieSkupin(listSkupiny):

    listEntropii = []
    for skupina in listSkupiny:
        histSkupina, _ = np.histogram(skupina)
        histSkupina = histSkupina[histSkupina != 0]
        histSkupina = histSkupina / len(histSkupina)
        listEntropii.append(-np.sum(histSkupina * np.log(histSkupina)))
    return listEntropii

def detekcePeakuSkupiny(listSkupin):
    """
    Funkce detekuje indexy a amplitudy peaků vstupního signálu.
    Dále spočítá jejich průměr a sumu.
    """

    listPocetPeaku = []
    listPrumerAmp = []
    listSumaAmp = []

    # převod skupin na 1D vektor hodnot
    for skupina in listSkupin:
        skupina = np.array(skupina)
        filtSkupina = sc.signal.medfilt(skupina, 5)
        # detekce peaku
        if len(skupina) > 2:
            indexySkupina, _ = sc.signal.find_peaks(skupina)
            amplitudySkupina = skupina[indexySkupina]

```

```

else:
    indexySkupina = 0
    amplitudySkupina = 0

if indexySkupina.size > 0:
    pocetPeaku = len(indexySkupina)
    prumerAmplitud = np.mean(amplitudySkupina)
    sumaAmplitud = np.sum(amplitudySkupina)
else:
    pocetPeaku = prumerAmplitud = sumaAmplitud = 0

listPocetPeaku.append(pocetPeaku)
listPrumerAmp.append(prumerAmplitud)
listSumaAmp.append(sumaAmplitud)

return {"pocetIndexu": listPocetPeaku, "prumer": listPrumerAmp, "suma": listSumaAmp}

```

```

In [ ]: klidSkupiny, stresSkupiny = rozdelitStres(impedance, stres)
print(f"Počet skupin, klid: {len(klidSkupiny)}, stres: {len(stresSkupiny)}")

```

Počet skupin, klid: 25, stres: 24

```

In [ ]: klidStatistikaSkupin = statistikaSkupin(klidSkupiny)
klidEntropie = entropieSkupin(klidSkupiny)
klidPeakySkupin = detekcePeakuSkupiny(klidSkupiny)

klidHodnoty = zip(klidStatistikaSkupin['prumer'], klidStatistikaSkupin['std'],
                  klidStatistikaSkupin['plocha'], klidEntropie,
                  klidStatistikaSkupin['moment'], klidPeakySkupin['pocetIndexu'],
                  klidPeakySkupin['prumer'], klidPeakySkupin['suma'])
sloupce = ["Průměr", "Směr. odchylka", "Plocha", "Entropie", "Moment",
           "Počet indexů", "Průměr amplitud", "Suma amplitud"]

klidTabulka = pandas.DataFrame(klidHodnoty, columns=sloupce)
klidTabulka.set_index(pandas.Index(range(1, len(klidSkupiny) + 1), name="Skupina"))

```

Out[ ]:

	Průměr	Směr. odchylka	Plocha	Entropie	Moment	Počet indexů	Průměr amplitud	Suma amplitud
<b>Skupina</b>								
1	1834.246	8.895	2.196e+06	-350.810	79.115	110	1835.055	201856
2	1831.821	1.483	6.961e+04	-14.038	2.199	6	1833.333	11000
3	1802.149	15.693	1.755e+06	-235.488	246.268	121	1803.273	218196
4	1780.456	4.132	2.012e+05	-20.918	17.073	8	1783.500	14268
5	1769.897	5.539	2.566e+05	-16.772	30.681	14	1774.071	24837
6	1762.058	4.933	3.013e+05	-32.781	24.334	21	1765.000	37065
7	1751.138	17.194	1.115e+06	-132.701	295.620	51	1754.510	89480
8	1727.919	3.397	2.315e+05	-24.823	11.542	17	1730.647	29421
9	1743.714	0.700	4.708e+04	-31.205	0.490	0	0.000	0
10	1745.512	14.352	9.112e+05	-94.570	205.990	63	1745.381	109959
11	1725.704	8.021	4.297e+05	-36.316	64.344	19	1727.895	32830
12	1733.508	12.868	9.205e+05	-99.872	165.596	32	1732.375	55436
13	1726.447	13.532	9.375e+05	-104.046	183.126	40	1728.525	69141
14	1722.811	14.778	7.443e+05	-70.689	218.394	35	1722.029	60271
15	1706.692	19.470	8.517e+05	-86.224	379.081	36	1710.583	61581
16	1681.077	9.738	1.007e+06	-130.835	94.831	35	1682.429	58885
17	1684.625	13.191	5.054e+05	-39.671	174.002	24	1684.375	40425
18	1645.822	7.425	2.485e+05	-12.069	55.133	7	1649.143	11544
19	1676.822	0.569	7.378e+04	-63.305	0.324	2	1677.000	3354
20	1706.732	21.287	1.232e+06	-162.950	453.140	41	1708.073	70031
21	1708.936	5.287	1.316e+05	-2.104	27.957	6	1708.167	10249
22	1697.472	2.867	1.205e+05	-35.262	8.221	2	1698.000	3396
23	1684.830	17.084	1.577e+06	-228.408	291.868	64	1688.703	108077
24	1672.283	3.730	2.876e+05	-36.575	13.914	8	1673.750	13390
25	1652.760	15.341	1.554e+06	-229.389	235.356	78	1656.821	129232

```

In [ ]: stresStatistikaSkupin = statistikaSkupin(stresSkupiny)
stresEntropie = entropieSkupin(stresSkupiny)
stresPeakySkupin = detekcePeakuSkupiny(stresSkupiny)

stresHodnoty = zip(stresStatistikaSkupin['prumer'], stresStatistikaSkupin['std'],
                  stresStatistikaSkupin['plocha'], stresEntropie,
                  stresStatistikaSkupin['moment'], stresPeakySkupin['pocetIndexu'],
                  stresPeakySkupin['prumer'], stresPeakySkupin['suma'])
sloupce = ["Průměr", "Směr. odchylka", "Plocha", "Entropie", "Moment",
           "Počet indexů", "Průměr amplitud", "Suma amplitud"]

stresTabulka = pandas.DataFrame(stresHodnoty, columns=sloupce)
stresTabulka.set_index(pandas.Index(range(1, len(stresSkupiny) + 1), name="Skupina"))

```

C:\Users\vojte\AppData\Local\Temp\ipykernel\_14284\1306398198.py:35: RuntimeWarning: Precision loss occurred in moment calculation due to catastrophic cancellation. This occurs when the data are nearly identical. Results may be unreliable.

```
momentSkupin = [sc.stats.moment(skupina, 2) for skupina in listSkupin]
```

Out[ ]:

	Průměr	Směr. odchylka	Plocha	Entropie	Moment	Počet indexů	Průměr amplitud	Suma amplitud
<b>Skupina</b>								
1	1827.500	2.784	12792.0	-0.017	7.750	0	0.000	0
2	1838.750	0.661	12872.0	-4.038	0.438	0	0.000	0
3	1777.250	1.199	12441.0	-0.215	1.438	1	1779.000	1779
4	1779.889	1.370	14238.0	-0.485	1.877	1	1781.000	1781
5	1764.159	5.773	142895.5	-7.542	33.329	3	1762.333	5287
6	1758.750	1.392	12310.0	-0.654	1.938	0	0.000	0
7	1728.250	0.968	12098.0	-2.899	0.938	0	0.000	0
8	1729.119	7.742	100284.5	-6.184	59.935	5	1726.200	8631
9	1748.500	3.122	12240.0	1.125	9.750	1	1750.000	1750
10	1728.787	1.369	127929.0	-28.685	1.874	5	1729.000	8645
11	1729.000	0.000	19019.0	-29.819	0.000	0	0.000	0
12	1730.908	7.524	148851.5	-4.812	56.612	4	1735.000	6940
13	1729.652	1.690	152210.0	-35.568	2.856	7	1732.714	12129
14	1707.731	4.311	87092.5	-1.162	18.581	5	1707.800	8539
15	1664.897	2.934	94897.0	-15.528	8.610	4	1669.000	6676
16	1678.912	11.462	132627.5	-0.448	131.380	5	1676.200	8381
17	1661.841	6.040	270884.5	-22.572	36.487	6	1669.000	10014
18	1672.750	2.107	11709.0	1.040	4.438	1	1677.000	1677
19	1675.400	2.154	15078.0	0.432	4.640	1	1677.000	1677
20	1718.500	2.784	12029.0	1.743	7.750	1	1717.000	1717
21	1700.000	0.000	11900.0	-16.636	0.000	0	0.000	0
22	1694.000	0.000	11858.0	-16.636	0.000	0	0.000	0
23	1662.778	0.629	13302.0	-5.199	0.395	0	0.000	0
24	1671.857	1.457	10030.0	-0.119	2.122	0	0.000	0

```
In [ ]: klidPole = np.array([klidStatistikaSkupin['prumer'], klidStatistikaSkupin['std'],
                             klidStatistikaSkupin['plocha'], klidEntropie,
                             klidStatistikaSkupin['moment'], klidPeakySkupin['pocetIndexu'],
                             klidPeakySkupin['prumer'], klidPeakySkupin['suma']])

stresPole = np.array([stresStatistikaSkupin['prumer'], stresStatistikaSkupin['std'],
                      stresStatistikaSkupin['plocha'], stresEntropie,
                      stresStatistikaSkupin['moment'], stresPeakySkupin['pocetIndexu'],
                      stresPeakySkupin['prumer'], stresPeakySkupin['suma']])

print(f"Pole hodnot: klid: {klidPole.shape}, stres: {stresPole.shape}")
```

Pole hodnot: klid: (8, 25), stres: (8, 24)