

# Sprawozdanie z Laboratorium nr. 9

- Filip Pasternak
- Grupa lab. 7 - piątek 18:30

## ZAD 1

Wczytać sygnały, oraz wyznaczyć ich średnią.

```
x = [0 1 3];  
mean(x)
```

```
ans = 1.3333
```

```
data_w = data;  
mean(data_w)
```

```
ans = -2.4346e-05
```

```
mean(data_p)
```

```
ans = -4.2175e-04
```

## ZAD 2

Wyznaczyć RMS sygnałów.

```
rms(x)
```

```
ans = 1.8257
```

```
rms(data_w)
```

```
ans = 0.1855
```

```
rms(data_p)
```

```
ans = 0.1415
```

## ZAD 3

Wyznaczyć współczynnik szczytów sygnałów.

```
peak2rms(x)
```

```
ans = 1.6432
```

```
peak2rms(data_w)
```

```
ans = 4.2778
```

```
peak2rms(data_p)
```

```
ans = 4.7018
```

## ZAD 4

Wyznaczyć współczynnik peak2peak sygnałów.

```
peak2peak(x)
```

```
ans = 3
```

```
peak2peak(data_w)
```

```
ans = 1.5757
```

```
peak2peak(data_p)
```

```
ans = 1.2662
```

## ZAD 5

Wyznaczyć wartość maksymalną i minimalną sygnałów.

```
% Min i max x  
min(x)
```

```
ans = 0
```

```
max(x)
```

```
ans = 3
```

```
% Min i max waitrak.wav  
min(data_w)
```

```
ans = -0.7821
```

```
max(data_w)
```

```
ans = 0.7936
```

```
% Min i max przekladnia.wav  
min(data_p)
```

```
ans = -0.6008
```

```
max(data_p)
```

```
ans = 0.6654
```

## ZAD 6

Obliczyć wariancję sygnałów.

```
var(x)
```

```
ans = 2.3333
```

```
var(data_w)
```

```
ans = 0.0344
```

```
var(data_p)
```

```
ans = 0.0200
```

## ZAD 7

Obliczyć odchylenie standardowe sygnałów.

```
std(x)
```

```
ans = 1.5275
```

```
std(data_w)
```

```
ans = 0.1855
```

```
std(data_p)
```

```
ans = 0.1415
```

## ZAD 8

Obliczyć energię sygnałów.

```
% energia
```

```
energia_x = sum(abs(x).^2)
```

```
ans = 10
```

```
energia_w = sum(abs(data_w).^2)
```

```
ans = 1.5179e+03
```

```
energia_p = sum(abs(data_p).^2)
```

```
ans = 883.1704
```

## ZAD 9

Obliczyć średnią moc sygnałów.

```
power_x = bandpower(x)
```

```
power_x = 3.3333
```

```
power_w = bandpower(data_w)
```

```
power_w = 0.0344
```

```
power_p = bandpower(data_p)
```

```
power_p = 0.0200
```

## ZAD 10

Obliczyć współczynnik RMS sygnałów.

```
rss_x = rssq(x)
```

```
rss_x = 3.1623
```

```
rss_w = rssq(data_w)
```

```
rss_w = 38.9602
```

```
rss_p = rssq(data_p)
```

```
rss_p = 29.7182
```

## ZAD 11

Użyć funkcji *seqperiod()* dla podanych macierzy.

Funkcja ta służy do znajdowania okresu z jakim w kolumnach tworzą się pewne powtarzające się kombinacje np. 121212.

```
x_zad11_a = [4 0 1 6;  
             2 0 2 7;  
             4 0 1 5;  
             2 0 5 6];
```

```
x_zad11_b = [4 0 1 6;  
             2 0 2 7;  
             4 0 1 5;  
             2 0 5 6;  
             1 0 1 7];
```

```
seqperiod(x_zad11_a)
```

```
ans = 1x4
```

```
2     1     4     3
```

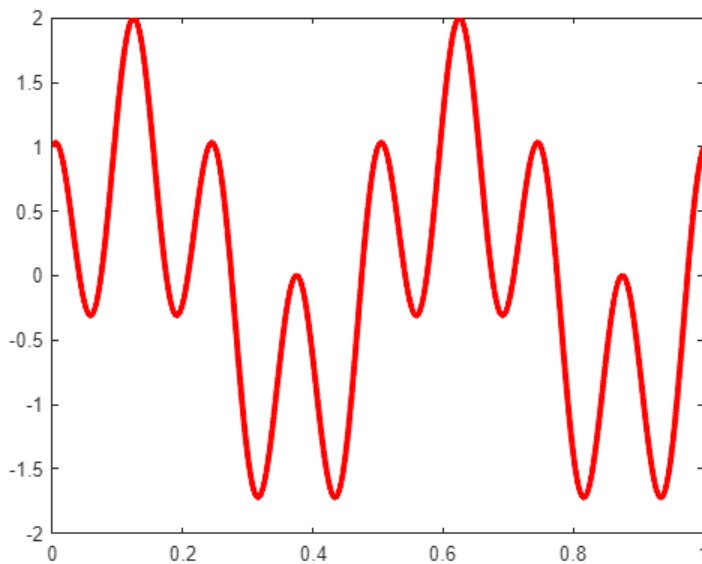
```
seqperiod(x_zad11_b)
```

```
ans = 1×4  
      5      1      4      3
```

## ZAD 12

Zapoznać się z funkcją *findpeaks()*

```
fs=5000;  
t=0:(1/fs):1;  
X=sin(2*pi*2*t);  
Y=cos(2*pi*8*t);  
s=X+Y;  
plot(t, s, 'r', 'LineWidth', 3);
```



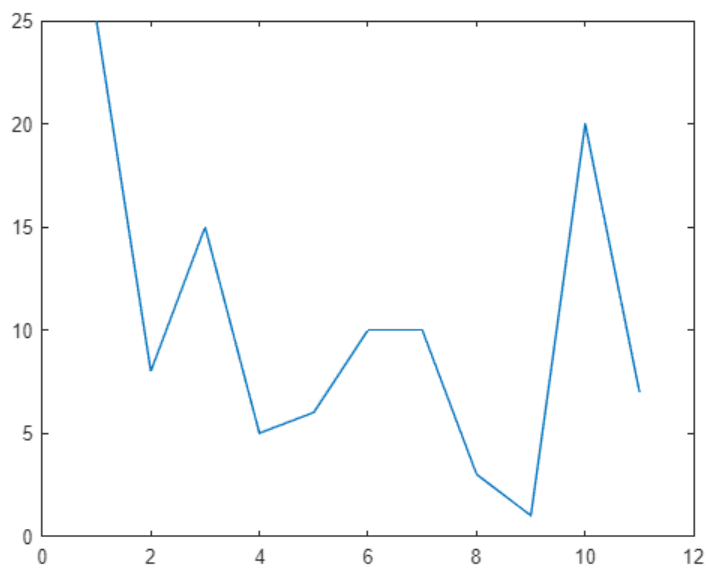
```
peaks = findpeaks(s)
```

```
peaks = 1×8  
      1.0314      2.0000      1.0314           0      1.0314      2.0000      1.0314 ...
```

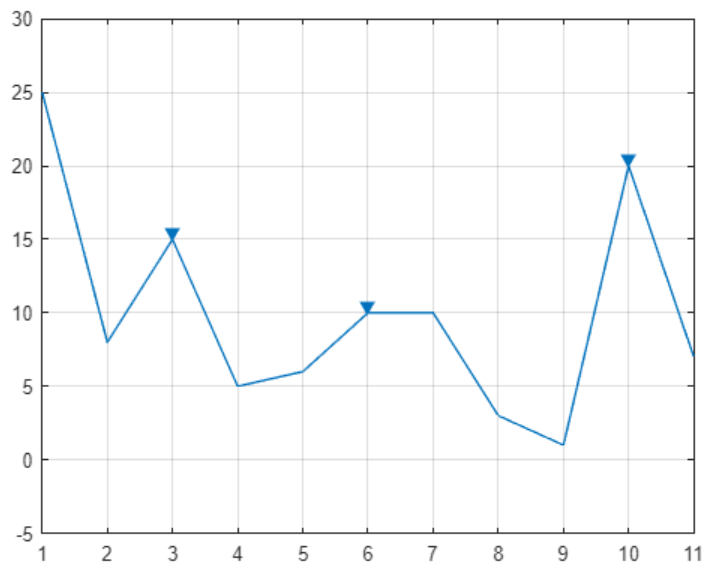
## ZAD 13

Użyć *findpeaks()* do danego zestawu danych.

```
s = [25 8 15 5 6 10 10 3 1 20 7];  
plot(s)
```



```
findpeaks(s)
```



```
peaks = findpeaks(s)
```

```
peaks = 1x3
      15      10      20
```

## ZAD 14

Zapoznać się z działaniem parametru SortStr funkcji *findpeaks()*.

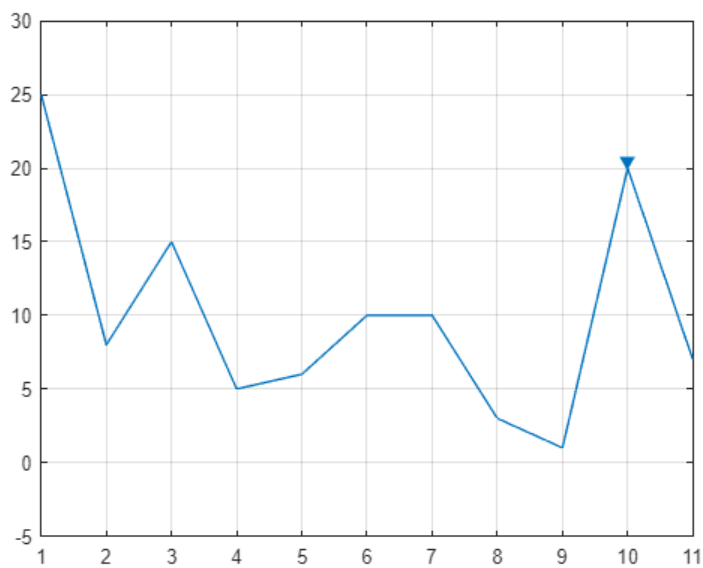
```
peaks = findpeaks(s, "SortStr", "descend")
```

```
peaks = 1×3  
      20      15      10
```

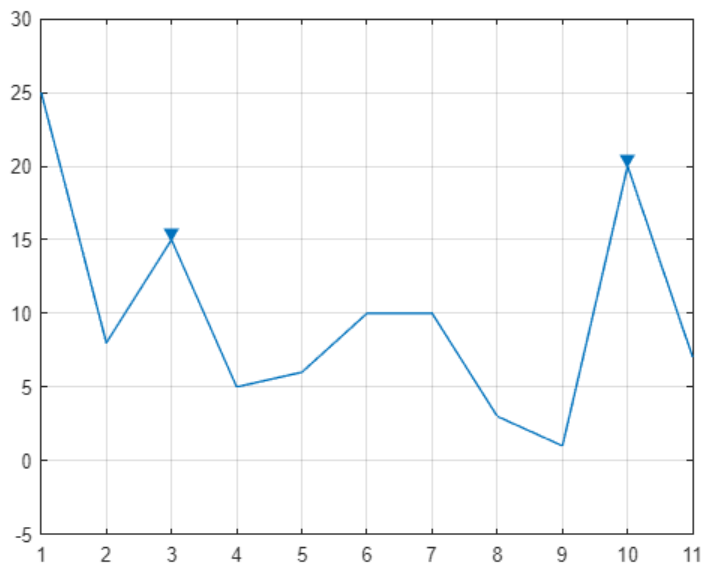
## ZAD 15

Zapoznać się z parametrem Threshold funkcji *findpeaks()*

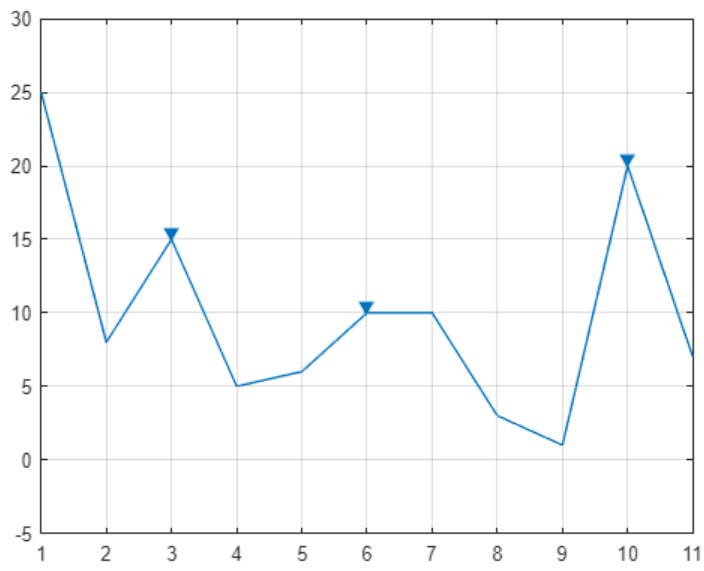
```
findpeaks(s, "Threshold", 10)
```



```
findpeaks(s, "Threshold", 5)
```



```
findpeaks(s, "Threshold", 0)
```

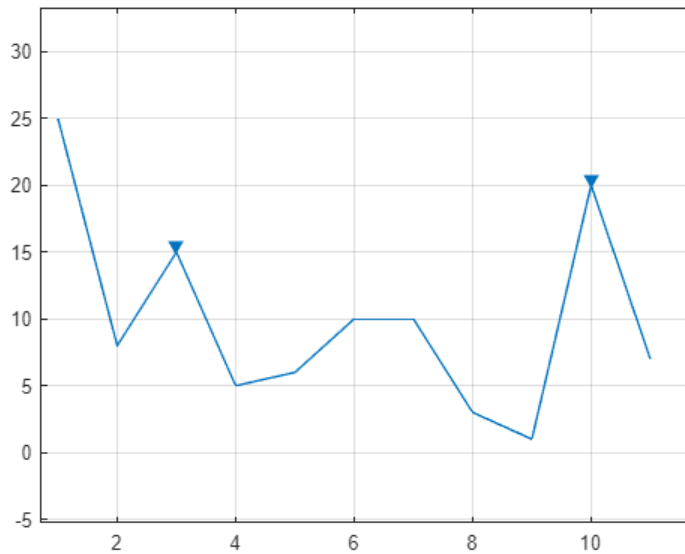




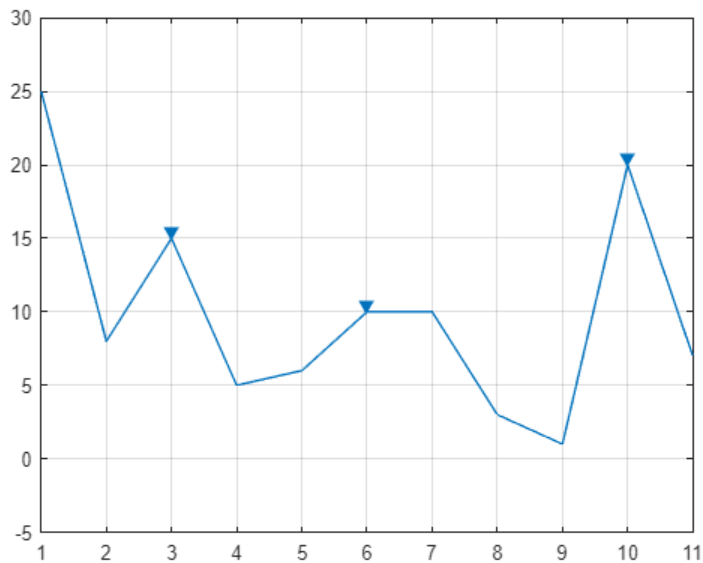
## ZAD 16

Zapoznać się z parametrem `MinPeakHeight` funkcji `findpeaks()`

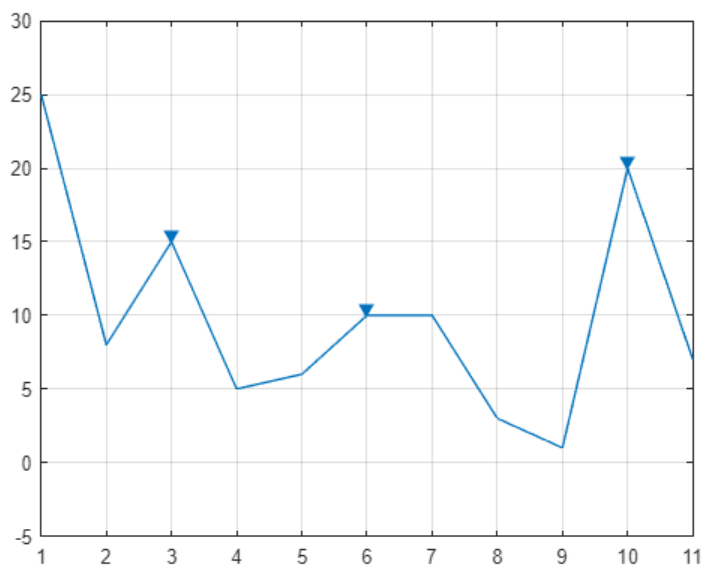
```
findpeaks(s, "MinPeakHeight", 12)
```



```
findpeaks(s, "MinPeakHeight", 8)
```



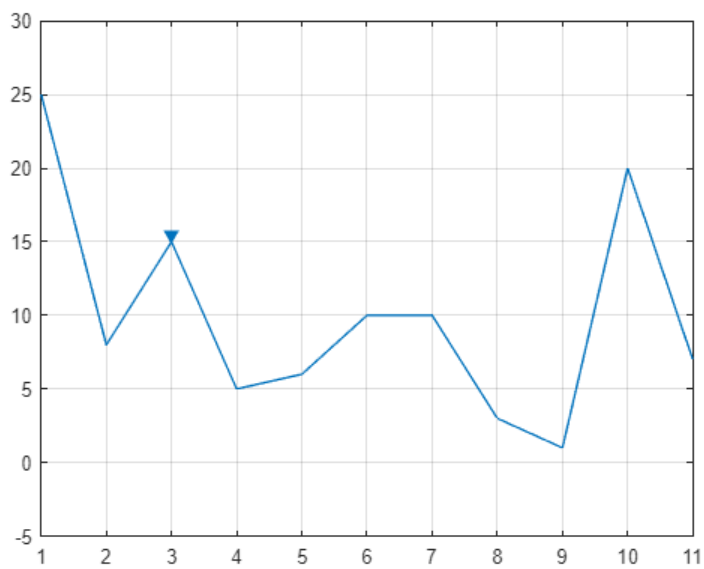
```
findpeaks(s, "MinPeakHeight", 3)
```



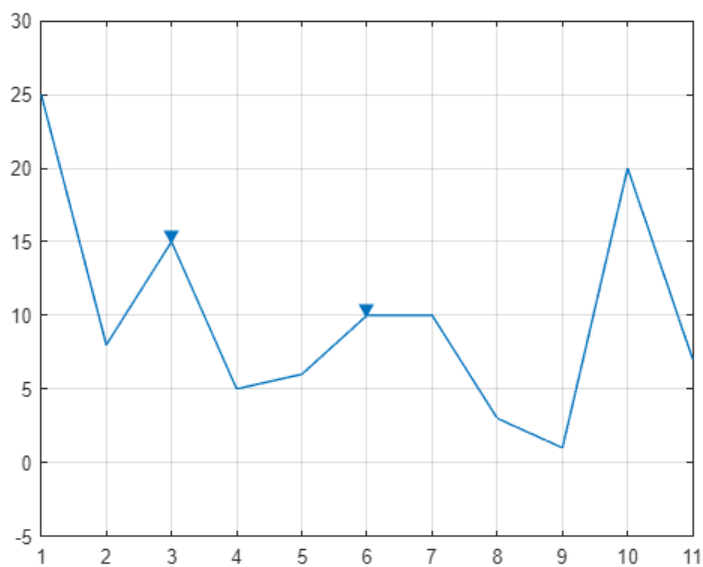
## ZAD 17

Zapoznać się z parametrem NPeaks funkcji *findpeaks()*

```
findpeaks(s, "NPeaks", 1)
```



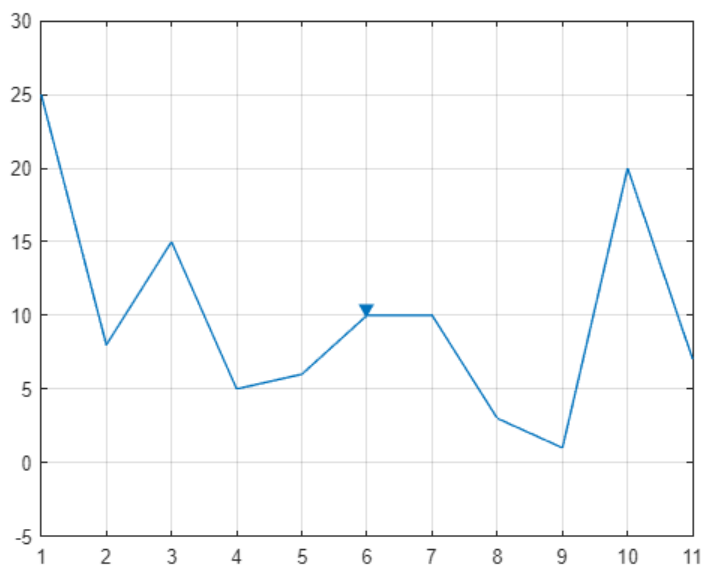
```
findpeaks(s, "NPeaks", 2)
```



## ZAD 18

Zapoznać się z parametrem `MinPeakWidth` funkcji `findpeaks()`

```
findpeaks(s, "MinPeakWidth", 1)
```



## ZAD 19

Użyć uzyskanej wiedzy do porównania ze sobą sygnałów podanych w ćwiczeniu z zastosowaniem K-NN.

```
data_p20 = data_p;
```

```
data_w20 = data_w;
```

```
% Porównywanie
```

```
dif_rms_23_20_w = sum(abs(rms(data_w23) - rms(data_w20)))
```

```
dif_rms_23_20_w = 0.0068
```

```
dif_rms_23_21_w = sum(abs(rms(data_w23) - rms(data_w21)))
```

```
dif_rms_23_21_w = 0.0036
```

```
dif_rms_24_20_w = sum(abs(rms(data_w24) - rms(data_w20)))
```

```
dif_rms_24_20_w = 8.8342e-04
```

```
dif_rms_24_21_w = sum(abs(rms(data_w24) - rms(data_w21)))
```

```
dif_rms_24_21_w = 0.0023
```

```
dif_rms_23_20_p = sum(abs(rms(data_p23) - rms(data_p20)))
```

```
dif_rms_23_20_p = 0.0054
```

```
dif_rms_23_21_p = sum(abs(rms(data_p23) - rms(data_p21)))
```

```
dif_rms_23_21_p = 0.0053
```

```
dif_rms_24_20_p = sum(abs(rms(data_p24) - rms(data_p20)))
```

```
dif_rms_24_20_p = 0.0034
```

```
dif_rms_24_21_p = sum(abs(rms(data_p24) - rms(data_p21)))
```

```
dif_rms_24_21_p = 0.0033
```

## Pytania:

1. Podstawowymi parametrami sygnałów są średnia kwadratowa (RMS), wartość średnia, wariancja, lub RSS (Root Sum Square)
2. Parametry sygnałów mogą być zastosowane do rozpoznawania obiektów wszelkiego typu, jak inteligentny monitoring lub wszelkie programy zamieniające mowę na tekst.
3. Wartość złożona z RMS, RSS i wartości maksymalnej sygnału nie będzie dla nas użyteczna z uwagi na silne skorelowanie tych parametrów.
4. *findpeaks()* pozwala na znalezienie ekstremów podanym sygnale oraz daje dużą elastyczność co do stawiania warunków jakie ekstrema mają zostać wzięte pod uwagę. Została przetestowana w zadaniach 11-18.

## **Wnioski:**

Matlab posiada szeroki wachlarz możliwości dotyczący interpretowania i opisywania sygnałów za pomocą przydatnych parametrów. Posiada wbudowane funkcje do ich wyznaczania co znacznie ułatwia działania na tych sygnałach i nas odciąża w kwestii uporczywych wzorów matematycznych. Parametry te można wykorzystywać do porównywania ze sobą różnych sygnałów co może zostać, przy odpowiedniej ilości danych, użyte do porównywania ze sobą różnych sygnałów i znajdowania między nimi podobieństw.