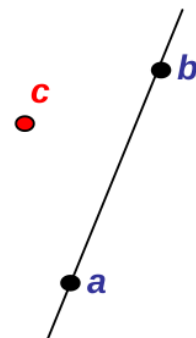


## Algorytmy geometryczne, laboratorium 1 - sprawozdanie

### 1. Opis ćwiczenia

Zadaniem które należy wykonać na laboratorium 1 jest określenie po której stronie prostej znajduje się punkt. Można to wykonać przy pomocy algorytmu obliczającego wartość jednego z dwóch wyznaczników:

1) Wyznacznik macierzy 3x3:  $\det(a, b, c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix}$



2) Wyznacznik macierzy 2x2:  $\det(a, b, c) = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix}$

Gdzie  $a_x$  i  $a_y$  to współrzędne odpowiednio x i y punktu a (tak samo z punktami b i c). Powyższe wyznaczniki pozwalają określić położenie punktu c względem prostej która jest wyznaczona przez punkty a i b. Jeżeli wyznacznik jest większy od 0 (albo od innej tolerancji dla zera - epsilon) to punkt znajduje się z lewej strony prostej, jeżeli jest mniejszy od 0 (albo od innej tolerancji dla zera którą zmieniamy na ujemną) to punkt znajduje się po prawej stronie prostej a jeżeli wartość wyznacznika jest równa 0 (lub równa epsilon) to punkt leży na prostej. Pomimo, że powyższe wyznaczniki są sobie równoważne to na skutek niedoskonałości reprezentacji liczb rzeczywistych w komputerze wyniki mogą się różnić w zależności od użytego wyznacznika.

### 2. Środowisko, biblioteki oraz użyte narzędzia

Ćwiczenie zostało wykonane w Jupyter Notebook i napisane w języku Python. Dodatkowo zostały użyte biblioteki takie jak pandas oraz numpy aby w przejrzysty sposób w dataframe przedstawiać wyniki z poszczególnych obliczeń. Do rysowania wykresów zostały użyte biblioteki matplotlib oraz seaborn, które w świetny sposób obrazują wykresy z danych które są w dataframe. Wszystko było wykonywane na

systemie operacyjnym Linux Ubuntu 20.04 oraz na procesorze Intel Core i5-7300HQ 2.50GHz.

### 3. Plan i sposób wykonania ćwiczenia

Na początku należało wygenerować zbiory punktów o współrzędnych rzeczywistych typu double:

- $10^5$  losowych punktów o współrzędnych z przedziału  $[-1000, 1000]$ ,
- $10^5$  losowych punktów o współrzędnych z przedziału  $[-10^{14}, 10^{14}]$ ,
- 1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu  $R=100$ ,
- 1000 losowych punktów o współrzędnych z przedziału  $[-1000, 1000]$  leżących na prostej wyznaczonej przez wektor (a, b), przyjmij:  
 $\mathbf{a} = [-1.0, 0.0]$ ,  $\mathbf{b} = [1.0, 0.1]$ .

5 pierwszych wierszy z dataframe kolejno z podpunktu a), b), c), d)

a) Tabela\_1

	X	Y
0	-396.973875	246.974268
1	789.282849	-905.516544
2	-424.323526	895.513647
3	-42.681272	-184.769776
4	-45.665334	438.109382

b) Tabela\_2

	X	Y
0	7.009587e+13	5.463838e+13
1	6.511091e+13	-1.524365e+13
2	4.218792e+13	-4.343438e+13
3	2.691518e+13	-2.781127e+13
4	-4.210157e+13	7.884060e+13

c) Tabela\_3

	X	Y
0	2.807673	-9.597759
1	-8.922250	4.515911
2	-5.893096	-8.079073
3	8.748726	-4.843531
4	-0.242502	9.997059

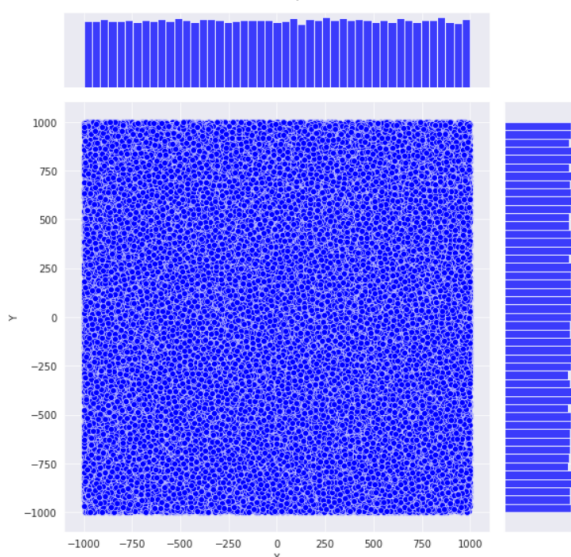
d) Tabela\_4

	X	Y
0	143.874008	7.243700
1	-360.162443	-17.958122
2	-324.810031	-16.190502
3	768.833667	38.491683
4	701.729060	35.136453

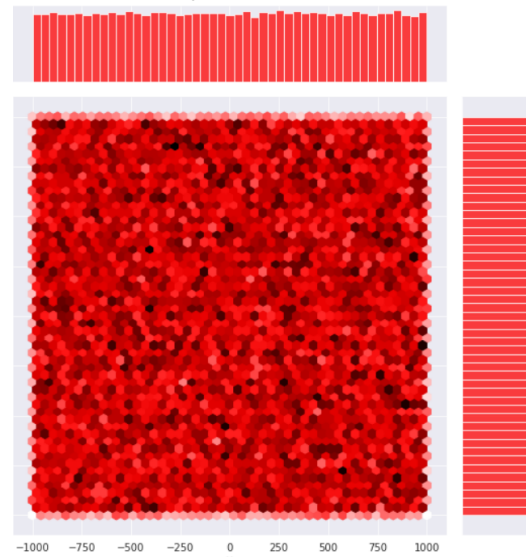
Do wygenerowania punktów z powyższych podpunktów użyto funkcji `uniform` z biblioteki `numpy` która znajduje się w module `random` (`np.random.uniform`). Użycie funkcji `random.uniform` z biblioteki `numpy` niż `uniform` ze standardowej biblioteki `Python` jest efektywniejsze, ponieważ działa szybciej oraz zapewnia znacznie większą liczbę rozkładów prawdopodobieństwa do wyboru.

Wraz z tworzeniem dataframe do każdego podpunktu wygenerowane zostały również wykresy rozkładu punktów za pomocą biblioteki `seaborn`.

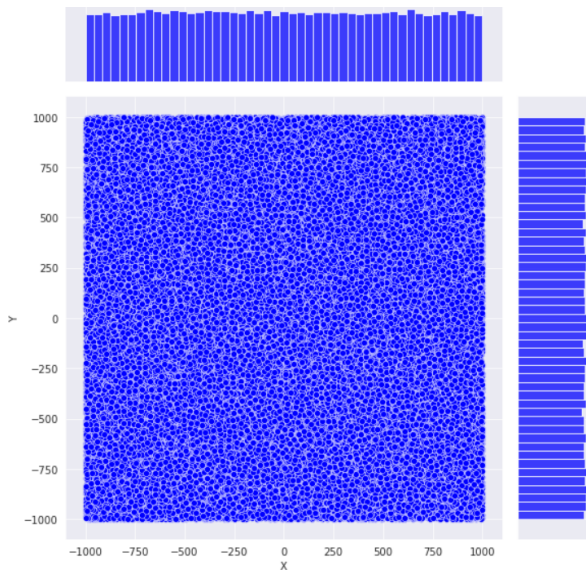
a) Wykres\_1



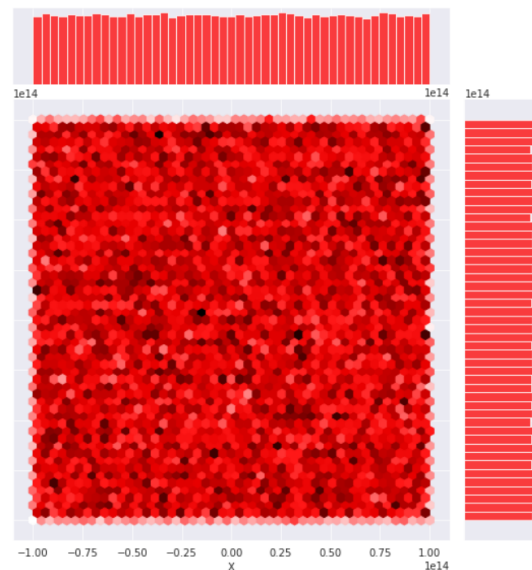
Wykres\_2



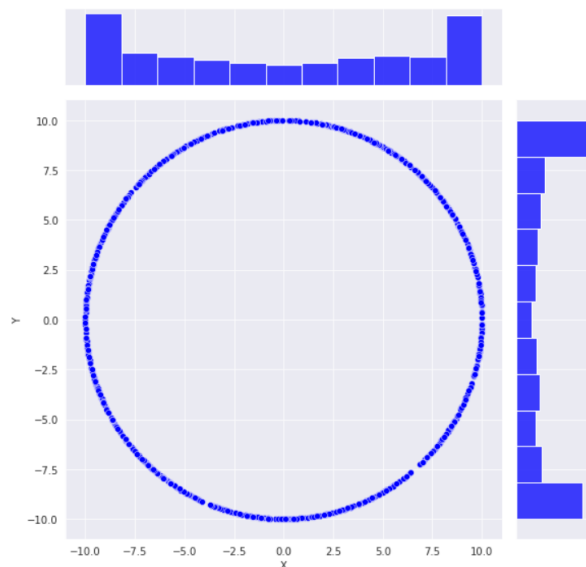
b) Wykres\_3



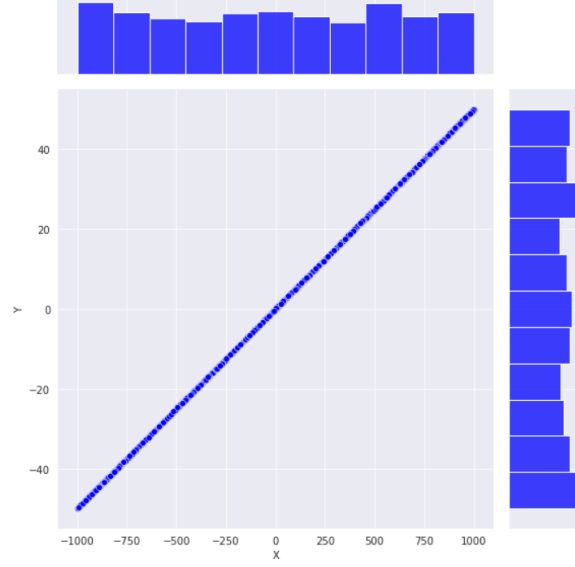
Wykres\_4



c) Wykres\_5



d) Wykres\_6



Dla Wykres\_1, Wykres\_2, Wykres\_3, Wykres\_4 zostały użyte dwa rodzaje wykresów a dla podpunktów Wykres\_5 i Wykres\_6 jeden rodzaj. Pierwszy z nich to wykres punktowy (scatter plot), został on użyty również w podpunktach c) i d). Drugi z wykresów to wykres rodzaju hexagon (hexagon plot) na którym możemy zobaczyć w których miejscach jest większe zagęszczenie punktów a w których mniejsze (ciemniejszy kolor oznacza większą liczbę punktów w danym obszarze, jaśniejszy oznacza mniejszą liczbę). Wykresy słupkowe u góry wykresów oraz z boku pokazują dystrybucję w danych obszarach odpowiednio x-ów u góry oraz y-ów z prawego boku. Również osie są podpisane u dołu 'X' oraz z lewego boku 'Y'.

Następnie zaimplementowano cztery funkcje obliczające wyznaczniki, które pozwolą nam zdeterminować po której stronie prostej wyznaczonej przez dwa punkty leży trzeci punkt:

- `det_1` - funkcja obliczająca wyznacznik 3x3, zaimplementowana samodzielnie
- `det_2` - funkcja obliczająca wyznacznik 2x2, zaimplementowana samodzielnie
- `det_3` - funkcja obliczająca wyznacznik 3x3, zaimplementowana przy pomocy biblioteki `numpy`
- `det_4` - funkcja obliczająca wyznacznik 2x2, zaimplementowana przy pomocy biblioteki `numpy`

Kolejnym krokiem było sklasyfikowanie punktów z poszczególnych podpunktów ze względu na położenie w stosunku do prostej wyznaczonej przez wektor ( $a = [-1.0, 0.0]$ ,  $b = [1.0, 0.1]$ ) w celu porównania wyników dla różnych wyznaczników. Do zrealizowania tego została zaimplementowana dodatkowa funkcja `classify_points` która klasyfikuje punkty. Zwraca ona wyniki -1 (punkt leży po lewej stronie prostej), 0 (punkt leży na prostej) i 1 (punkt leży po prawej stronie prostej). Klasyfikację punktów powtórzono dla różnych wartości tolerancji błędów wynoszących:  $\epsilon_1 = 10^{-14}$ ,  $\epsilon_2 = 10^{-12}$ ,  $\epsilon_3 = 10^{-10}$ ,  $\epsilon_4 = 10^{-8}$ . Została również stworzona funkcja `assign_to_dataframe`, która przypisuje do odpowiedniego epsilon oraz odpowiedniego wyznacznika liczbę punktów które zostały sklasyfikowane do -1, 0 lub 1. Na końcu również sprawdzono wyniki dla różnej precyzji obliczeń - `float32`.

#### 4. Klasyfikacja poszczególnych podziałów punktów

Do wszystkich istniejących dataframe które odpowiadają każdemu podpunktowi dołączane są nowe kolumny w których znajdują się wartości (-1, 0, 1) dla różnych epsilonów i różnych wyznaczników.

a) Tabela\_5

	X	Y	eps_1e-14_d1	eps_1e-14_d2	eps_1e-14_d3	eps_1e-14_d4	eps_1e-10_d1
0	-396.973875	246.974268	-1	-1	-1	-1	-1
1	789.282849	-905.516544	1	1	1	1	1
2	-424.323526	895.513647	-1	-1	-1	-1	-1
3	-42.681272	-184.769776	1	1	1	1	1
4	-45.665334	438.109382	-1	-1	-1	-1	-1

b) Tabela\_6

	X		Y	eps_1e-14_d1	eps_1e-14_d2	eps_1e-14_d3	eps_1e-14_d4	eps_1e-10_d1
0	7.009587e+13	5.463838e+13		-1	-1	-1	-1	-1
1	6.511091e+13	-1.524365e+13		1	1	1	1	1
2	4.218792e+13	-4.343438e+13		1	1	1	1	1
3	2.691518e+13	-2.781127e+13		1	1	1	1	1
4	-4.210157e+13	7.884060e+13		-1	-1	-1	-1	-1

c) Tabela\_7

	X		Y	eps_1e-14_d1	eps_1e-14_d2	eps_1e-14_d3	eps_1e-14_d4	eps_1e-10_d1
0	2.807673	-9.597759		1	1	1	1	1
1	-8.922250	4.515911		-1	-1	-1	-1	-1
2	-5.893096	-8.079073		1	1	1	1	1
3	8.748726	-4.843531		1	1	1	1	1
4	-0.242502	9.997059		-1	-1	-1	-1	-1

d) Tabela\_8

	X		Y	eps_1e-14_d1	eps_1e-14_d2	eps_1e-14_d3	eps_1e-14_d4	eps_1e-10_d1
0	143.874008	7.243700		0	-1	0	0	0
1	-360.162443	-17.958122		0	0	0	-1	0
2	-324.810031	-16.190502		0	0	0	-1	0
3	768.833667	38.491683		0	0	0	0	0
4	701.729060	35.136453		0	0	1	0	0

Powyższe dataframe to tylko fragment całości. Całe dataframe ze wszystkimi obliczonymi wartościami dla różnych epsilonów oraz różnych wyznaczników znajduje się w Jupyter Notebooku. Możemy dzięki tym dataframe zauważyć, że w przykładzie Tabela\_5 i Tabela\_7 mamy takie same wartości dla różnych epsilonów oraz wyznaczników. W Tabela\_6 i Tabela\_8 możemy zauważyć, że są one różne w zależności od użytego epsilon oraz wyznacznika. Również zostały stworzone dataframe, które zawierają dane ile punktów zostało przydzielonych do odpowiednich wartości (-1, 0, 1) dla różnych wartości epsilon oraz różnych wyznaczników dla odpowiednich podpunktów. W tych dataframe zostało użyte MultilIndexing aby móc zagnieździć dla konkretnych epsilonów wszystkie wyznaczniki.



a) Tabela_9					b) Tabela_10					c) Tabela_11					d) Tabela_12				
-1 0 1					-1 0 1					-1 0 1					-1 0 1				
10**-08	1	49910	0	50090	10**-08	1	49884	0	50116	10**-08	1	499	0	501	10**-08	1	0	1000	0
	2	49910	0	50090		2	49878	10	50112		2	499	0	501		2	0	1000	0
	3	49910	0	50090		3	49884	0	50116		3	499	0	501		3	0	1000	0
	4	49910	0	50090		4	49881	10	50109		4	499	0	501		4	0	1000	0
10**-10	1	49910	0	50090	10**-10	1	49884	0	50116	10**-10	1	499	0	501	10**-10	1	0	1000	0
	2	49910	0	50090		2	49878	10	50112		2	499	0	501		2	0	1000	0
	3	49910	0	50090		3	49884	0	50116		3	499	0	501		3	0	1000	0
	4	49910	0	50090		4	49881	10	50109		4	499	0	501		4	0	1000	0
10**-12	1	49910	0	50090	10**-12	1	49884	0	50116	10**-12	1	499	0	501	10**-12	1	0	1000	0
	2	49910	0	50090		2	49878	10	50112		2	499	0	501		2	83	818	99
	3	49910	0	50090		3	49884	0	50116		3	499	0	501		3	0	1000	0
	4	49910	0	50090		4	49881	10	50109		4	499	0	501		4	99	794	107
10**-14	1	49910	0	50090	10**-14	1	49884	0	50116	10**-14	1	499	0	501	10**-14	1	0	1000	0
	2	49910	0	50090		2	49878	10	50112		2	499	0	501		2	137	708	155
	3	49910	0	50090		3	49884	0	50116		3	499	0	501		3	16	879	105
	4	49910	0	50090		4	49881	10	50109		4	499	0	501		4	140	716	144

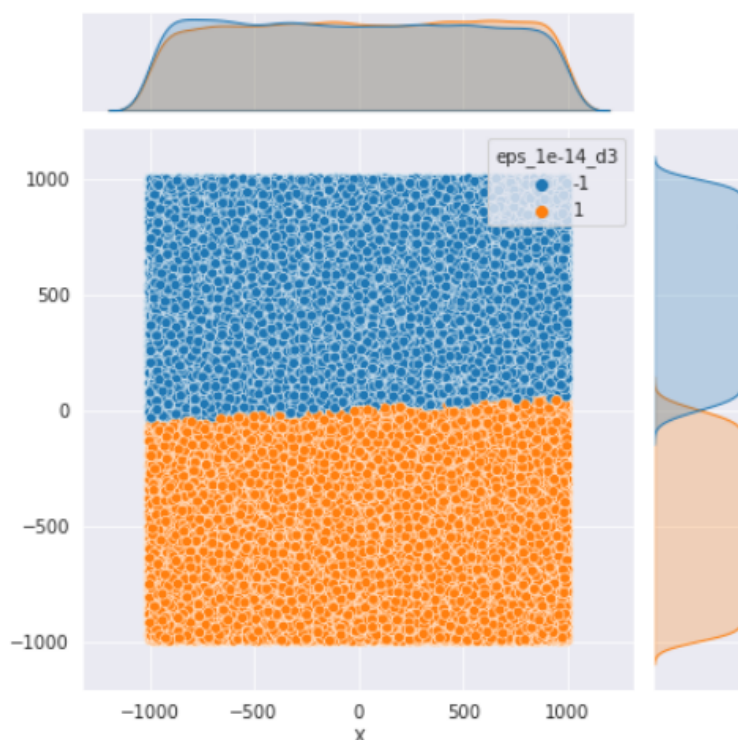
1, 2, 3, 4 odpowiada odpowiednio det\_1, det\_2, det\_3, det\_4.

## 5. Wybrane wykresy klasyfikacji punktów dla różnych podpunktów, epsilonów oraz wyznaczników

Kolory na wykresach są opisane w legendach które znajdują się obok wykresów albo pod nimi.

- klasyfikacja punktów z podpunktu a) (wszystkie wykresy są identyczne dla wszystkich  $\epsilon$  i wyznaczników)

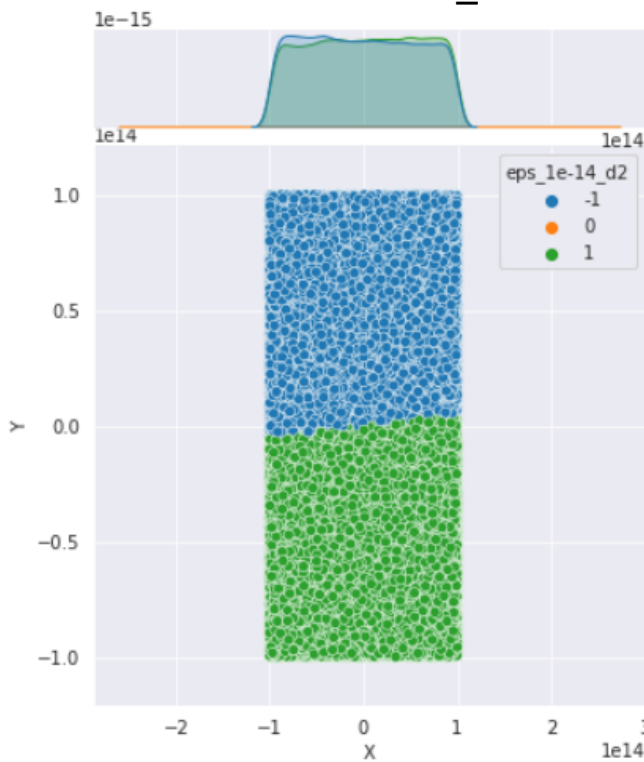
Wykres\_7



- klasyfikacja punktów z podpunktu b)

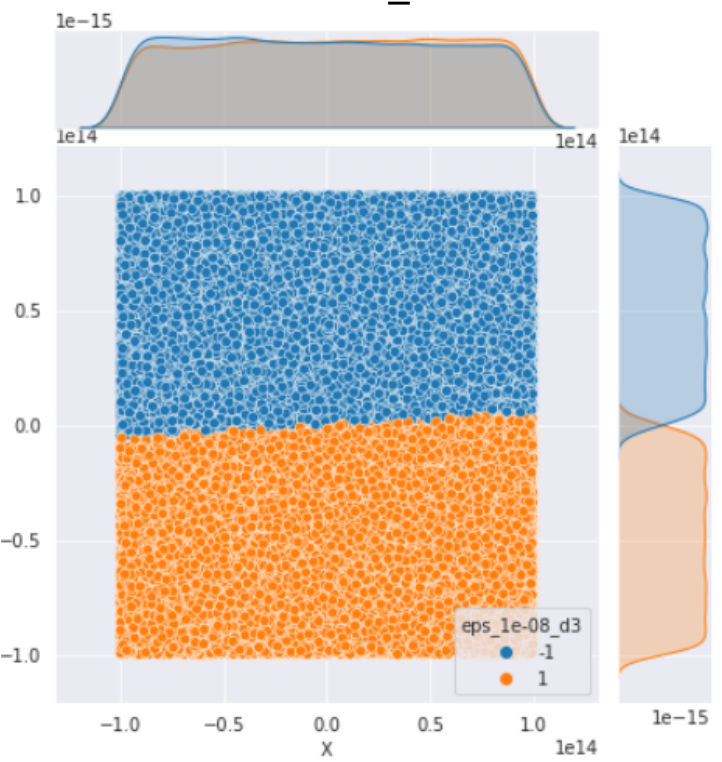
Wykres\_8

$\varepsilon = 10^{-14}$  det\_2



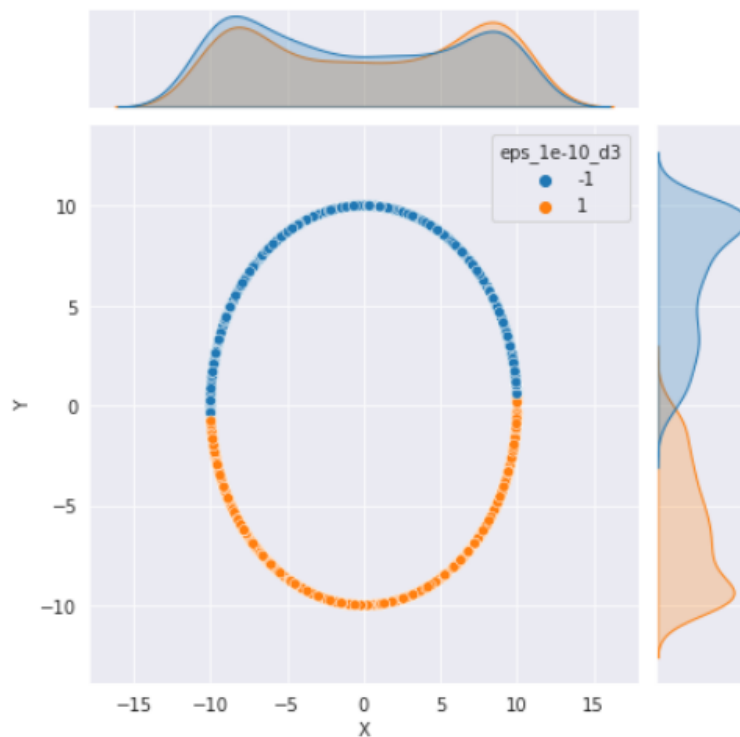
Wykres\_9

$\varepsilon = 10^{-8}$  det\_3



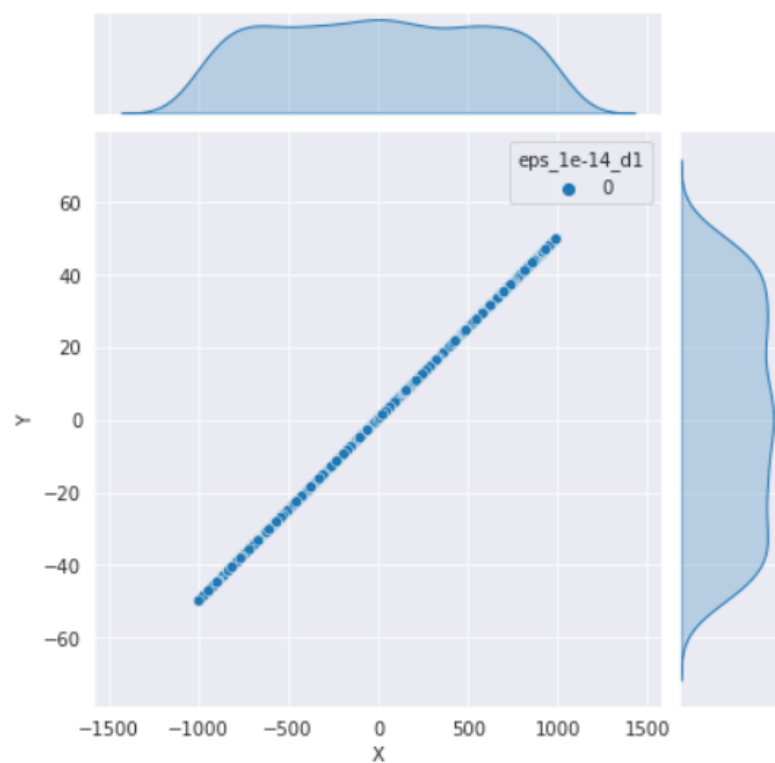
- klasyfikacja punktów z podpunktu c) (wszystkie wykresy są identyczne dla wszystkich  $\varepsilon$  i wyznaczników)

Wykres\_10

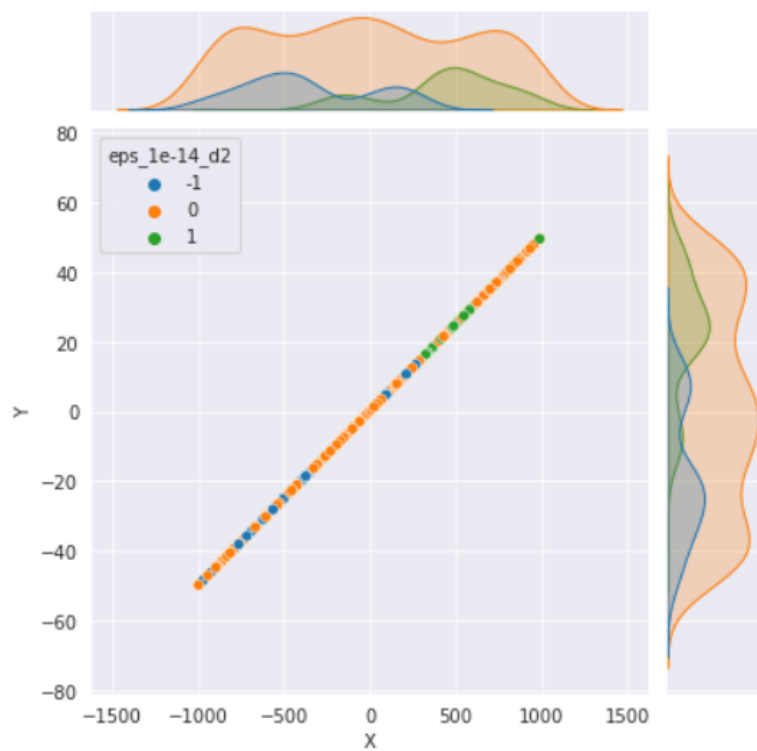


- klasyfikacja punktów z podpunktu c)

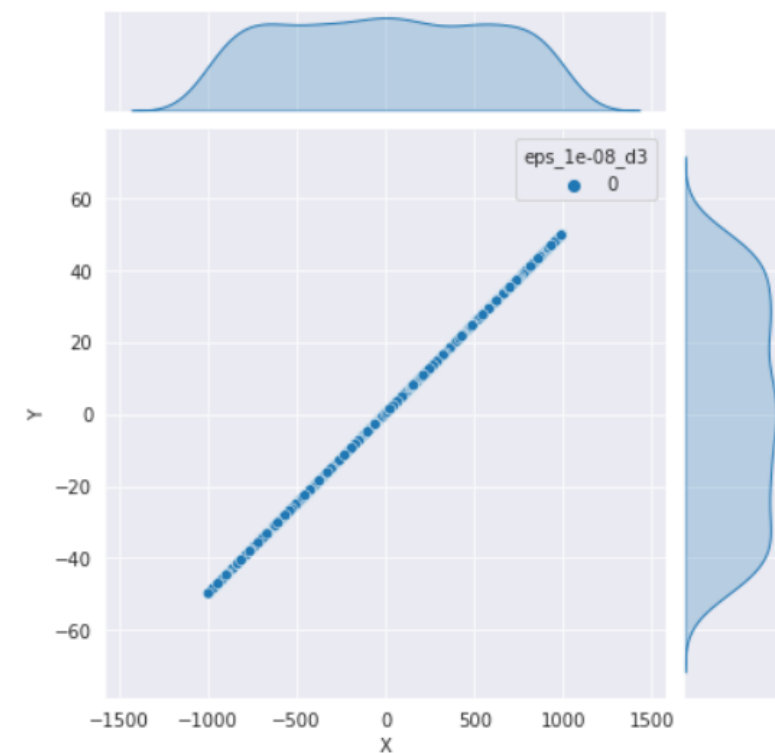
Wykres\_11  
 $\varepsilon = 10^{-14} \text{ det\_1}$



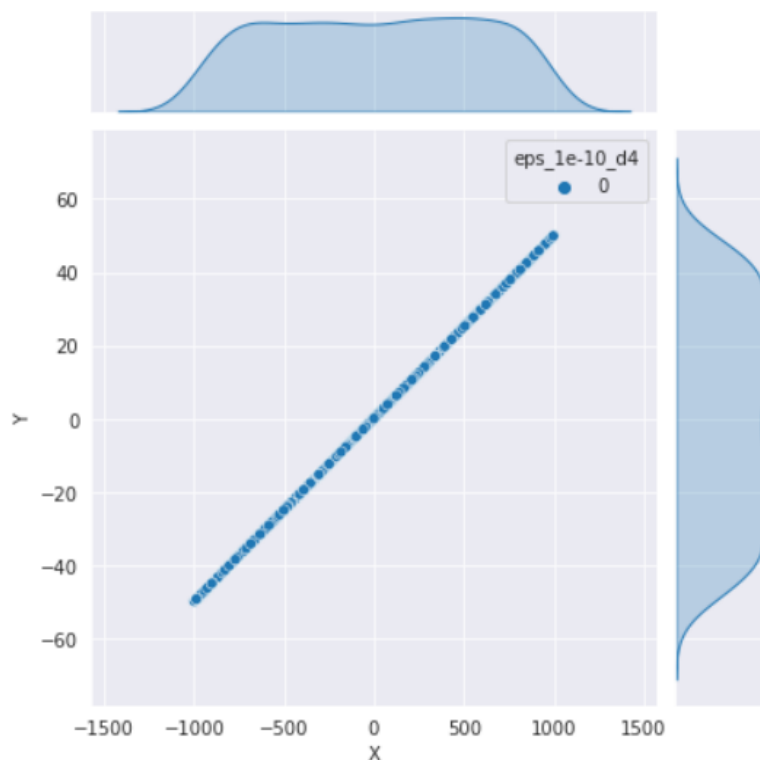
Wykres\_12  
 $\varepsilon = 10^{-14} \text{ det\_2}$



Wykres\_12  
 $\varepsilon = 10^{-8} \text{ det\_3}$



Wykres\_13  
 $\varepsilon = 10^{-10} \text{ det\_4}$

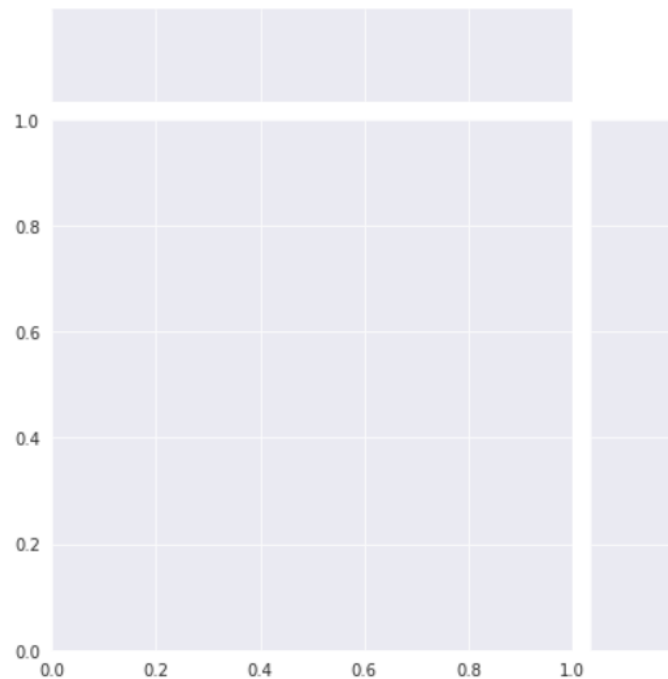




Wykresy dla punktów, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilon

- W podpunkcie a) wyniki dla wszystkich 4 wyznaczników oraz dla różnych epsilon są takie same, nie różnią się między sobą więc wykresy są puste. Wszystkie punkty zostały sklasyfikowane tak samo.

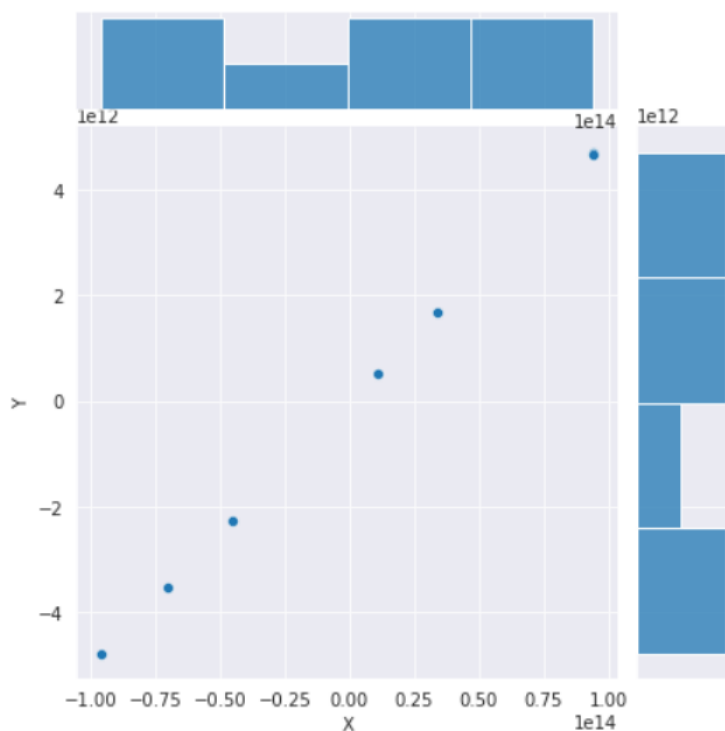
Wykres\_14



- W podpunkcie b) występują punkty które zostały inaczej zaklasyfikowane przez różne wyznaczniki.

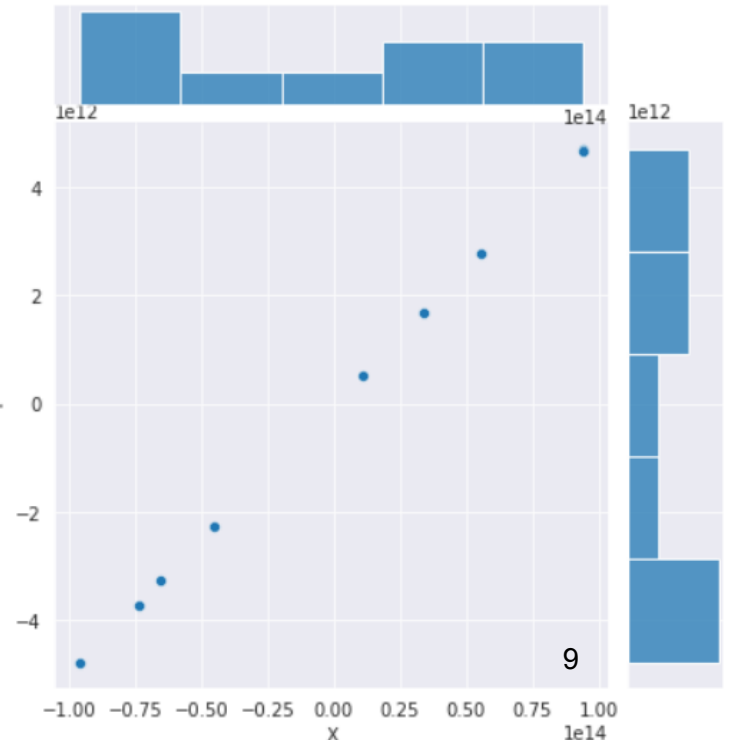
Wykres\_15

$\epsilon = 10^{-14}$  det\_1 z  $\epsilon = 10^{-10}$  det\_3



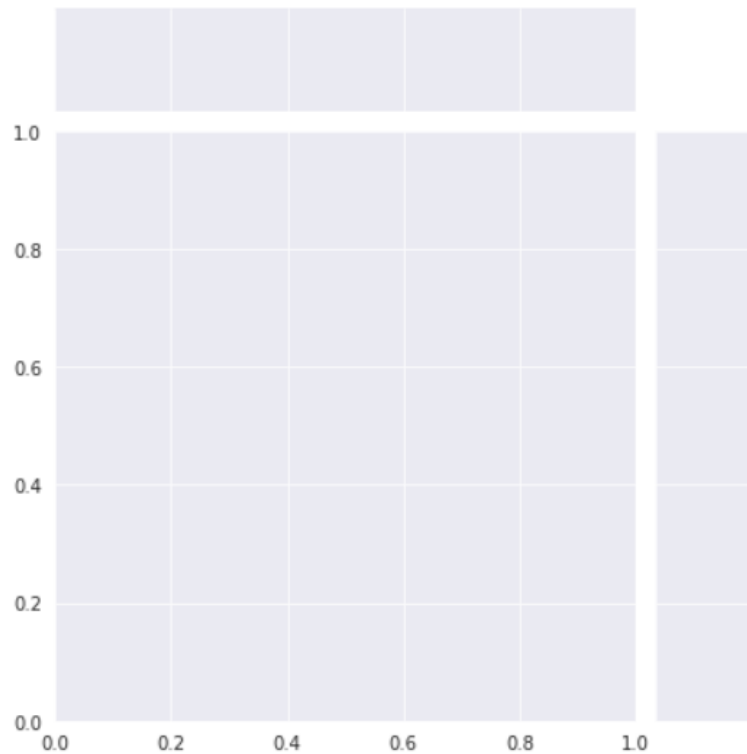
Wykres\_16

$\epsilon = 10^{-14}$  det\_3 z  $\epsilon = 10^{-12}$  det\_1



- W podpunkcie c) wyniki dla wszystkich 4 wyznaczników oraz dla różnych epsilonów są takie same, nie różnią się między sobą więc wykresy są puste. Wszystkie punkty zostały sklasyfikowane tak samo.

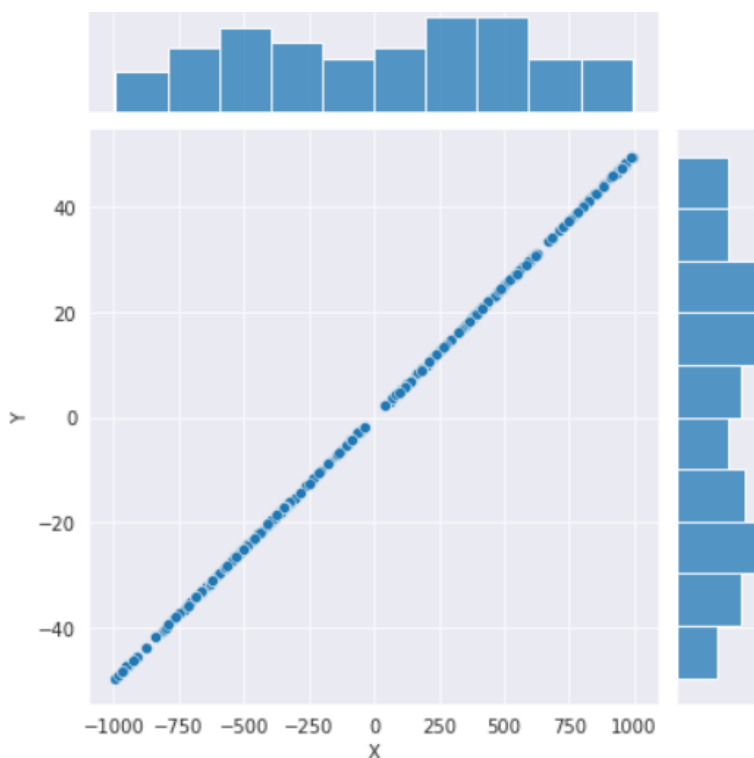
Wykres\_17



- W podpunkcie d) występują punkty które zostały inaczej zaklasyfikowane przez różne wyznaczniki.

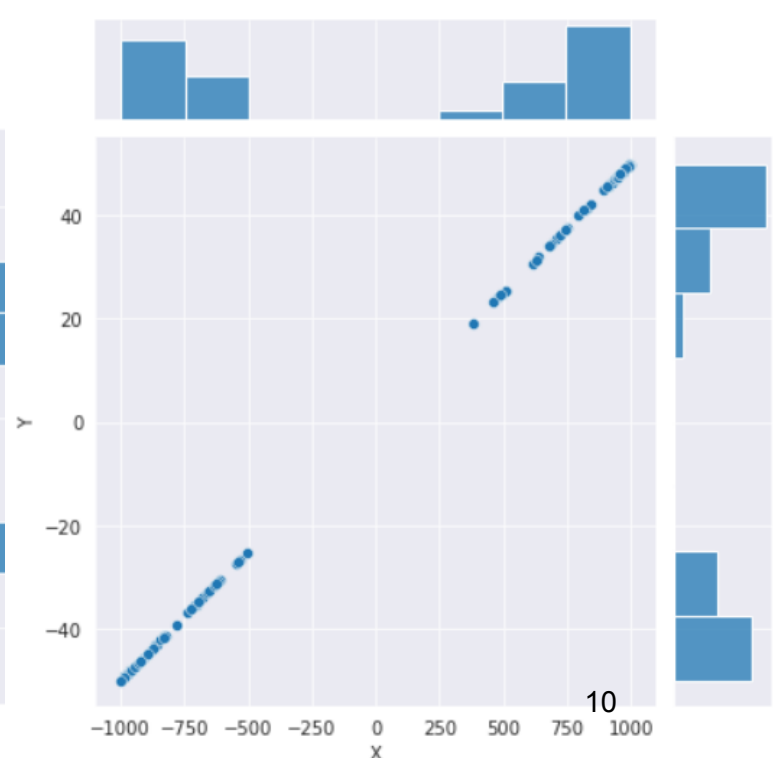
Wykres\_18

dla  $\varepsilon = 10^{-14}$  det\_1 z  $\varepsilon = 10^{-14}$  det\_2



Wykres\_19

$\varepsilon = 10^{-14}$  det\_3 z  $\varepsilon = 10^{-12}$  det\_4



## 6. Użycie innej precyzji (float32)

Wykresy dla podpunktów b) i d) po użyciu innej precyzji. Przedstawione są wykresy tylko dla podpunktów b) i d) ponieważ były to jedyne podpunkty w których występowała rozbieżność w wynikach dla różnych wyznaczników i epsilonów. Zostały stworzone nowe dataframe dla wartości po użyciu innej precyzji.

5 pierwszych wierszy z dataframe kolejno z podpunktu b) i d) po zmianie precyzji na float32

b) Tabela\_13

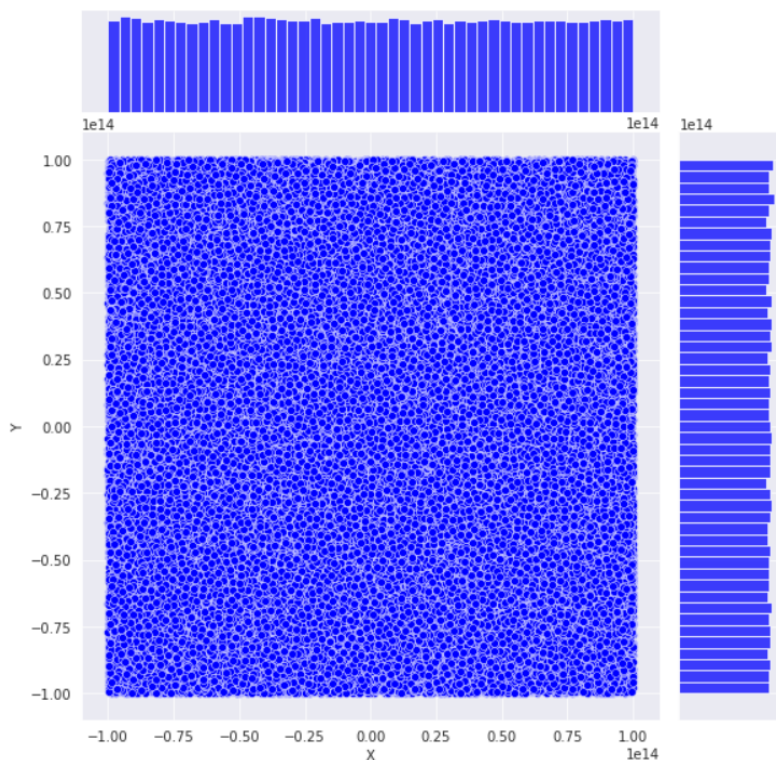
	X	Y
0	-2.407648e+13	-6.095850e+12
1	8.156353e+13	5.364797e+13
2	-1.927577e+13	5.362377e+13
3	-2.333296e+13	4.585456e+13
4	8.642588e+13	8.367844e+13

d) Tabela\_14

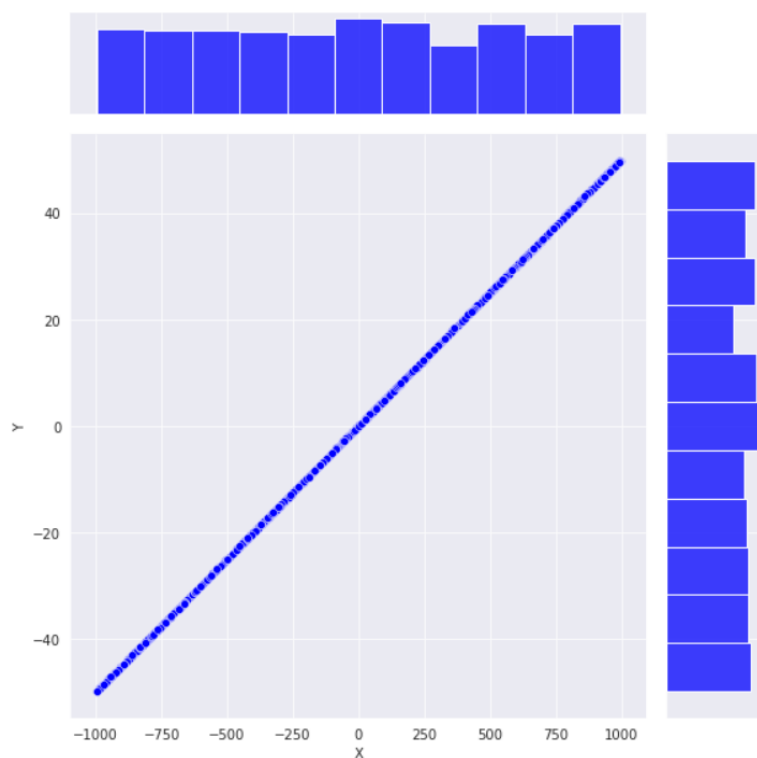
	X	Y
0	-694.067871	-34.653393
1	-569.224426	-28.411221
2	-405.260193	-20.213009
3	-321.059631	-16.002981
4	718.095032	35.954754

Wygenerowane wykresy rozkładu punktów dla podpunktów b) i d) po zmianie precyzji.

b) Wykres\_20



d) Wykres\_21



Do nowych dataframe, które odpowiadają podpunktom b) i d) po zmianie ich wartości przez użycie float32 dołączane są nowe kolumny w których znajdują się wartości (-1, 0, 1) dla różnych epsilon'ów i różnych wyznaczników. Użyte epsilon'y to  $\epsilon_1 = 10^{-16}$  i  $\epsilon_2 = 10^{-10}$ .

b) Tabela\_15

	X		Y	eps_1e-16_d1	eps_1e-16_d2	eps_1e-16_d3	eps_1e-16_d4	eps_1e-10_d1
0	-2.407648e+13	-6.095850e+12		1	1	1	1	1
1	8.156353e+13	5.364797e+13		-1	-1	-1	-1	-1
2	-1.927577e+13	5.362377e+13		-1	-1	-1	-1	-1
3	-2.333296e+13	4.585456e+13		-1	-1	-1	-1	-1
4	8.642588e+13	8.367844e+13		-1	-1	-1	-1	-1

d) Tabela\_16

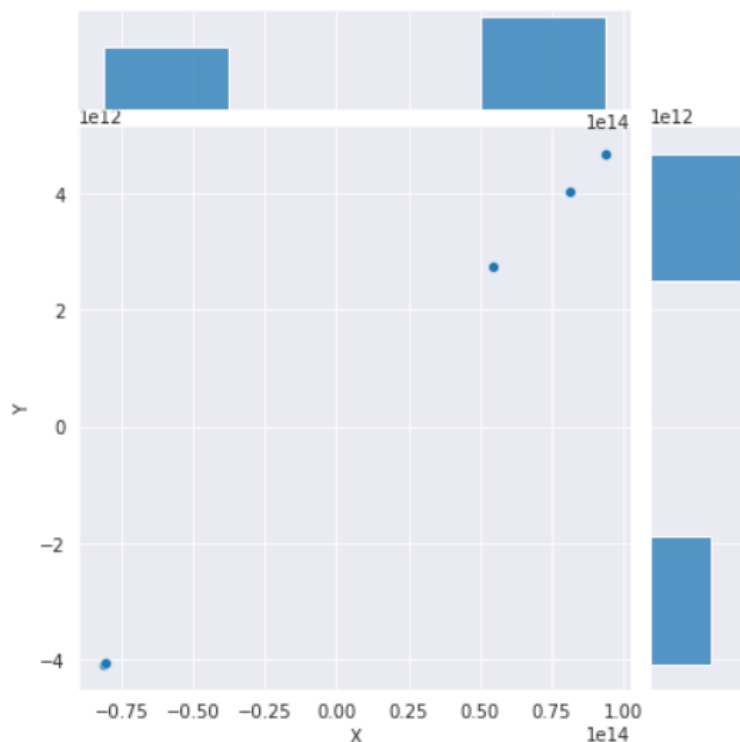
	X		Y	eps_1e-16_d1	eps_1e-16_d2	eps_1e-16_d3	eps_1e-16_d4	eps_1e-10_d1
0	-694.067871	-34.653393		-1	-1	-1	-1	-1
1	-569.224426	-28.411221		-1	-1	-1	-1	-1
2	-405.260193	-20.213009		-1	-1	-1	-1	-1
3	-321.059631	-16.002981		-1	-1	-1	-1	-1
4	718.095032	35.954754		-1	-1	-1	-1	-1

Wybrane wykresy dla punktów o zmienionej precyzji, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilon.

- dla podpunktu b)

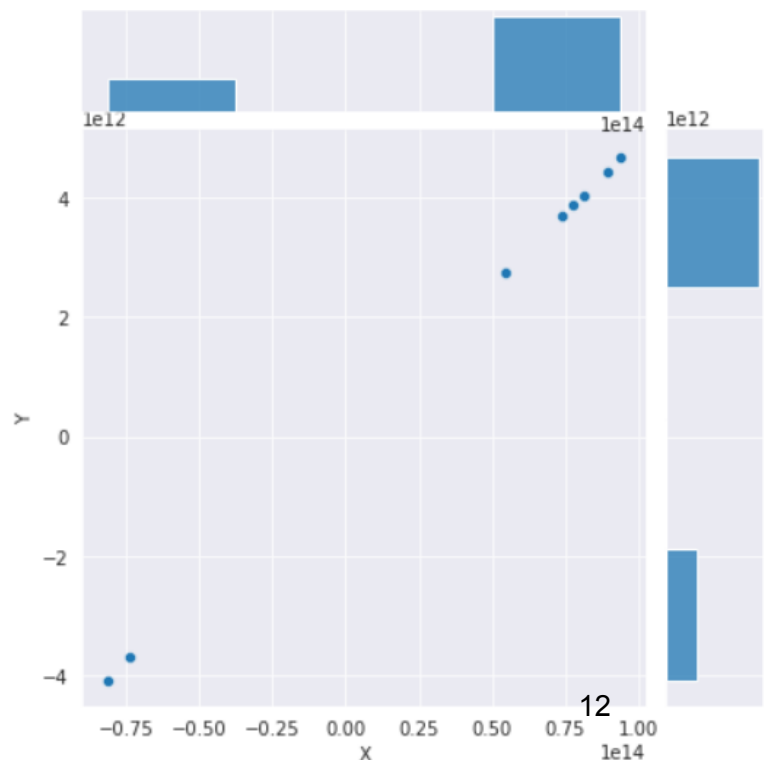
Wykres\_22

$\epsilon = 10^{-16}$  det\_3 z  $\epsilon = 10^{-12}$  det\_1



Wykres\_23

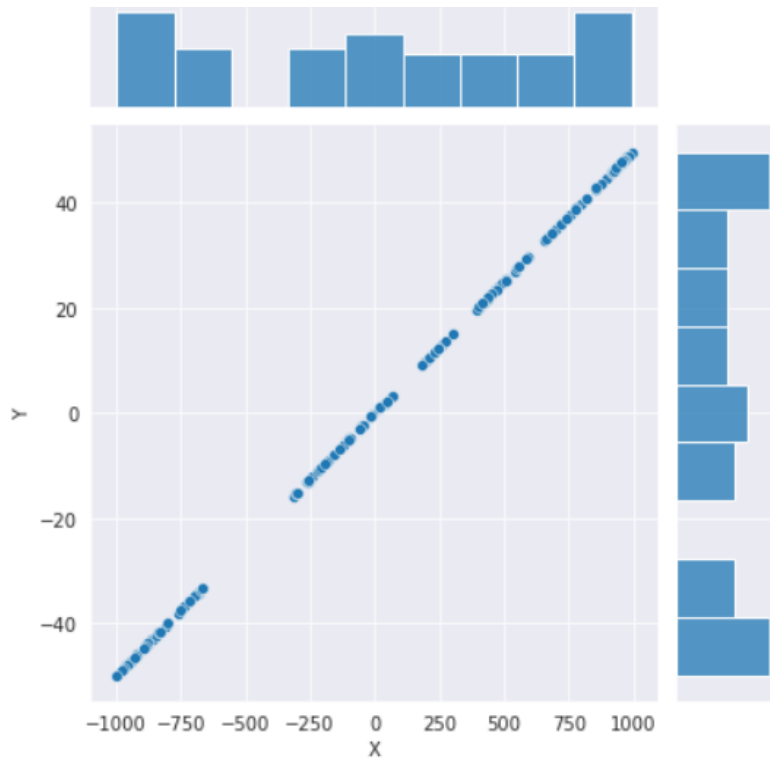
$\epsilon = 10^{-16}$  det\_1 z  $\epsilon = 10^{-8}$  det\_4



- dla podpunktu d)

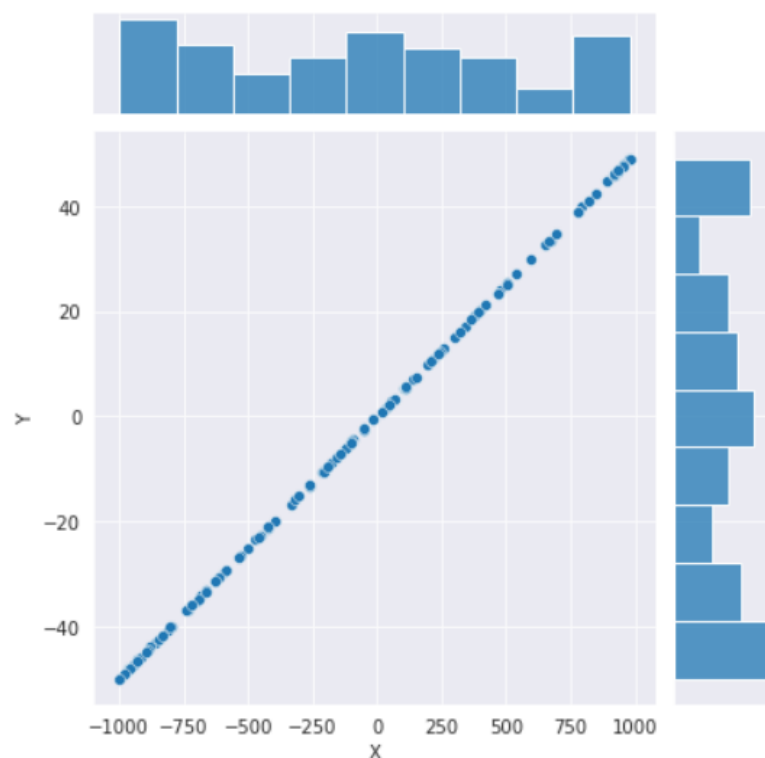
Wykres\_24

dla  $\varepsilon = 10^{-16}$  det\_1 z  $\varepsilon = 10^{-12}$  det\_1



Wykres\_25

$\varepsilon = 10^{-16}$  det\_2 z  $\varepsilon = 10^{-16}$  det\_3



Wykresy dla punktów, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilon oraz innej precyzji.

b) Tabela\_17

		-1	0	1
<b>10**-08</b>	<b>1</b>	49593	0	50407
	<b>2</b>	49591	5	50404
	<b>3</b>	49593	0	50407
	<b>4</b>	49592	5	50403
<b>10**-12</b>	<b>1</b>	49593	0	50407
	<b>2</b>	49591	5	50404
	<b>3</b>	49593	0	50407
	<b>4</b>	49592	5	50403
<b>10**-16</b>	<b>1</b>	49593	0	50407
	<b>2</b>	49591	5	50404
	<b>3</b>	49593	0	50407
	<b>4</b>	49592	5	50403

d) Tabela\_18

		-1	0	1
<b>10**-08</b>	<b>1</b>	405	180	415
	<b>2</b>	405	180	415
	<b>3</b>	405	180	415
	<b>4</b>	405	180	415
<b>10**-12</b>	<b>1</b>	405	179	416
	<b>2</b>	405	179	416
	<b>3</b>	405	179	416
	<b>4</b>	421	139	440
<b>10**-16</b>	<b>1</b>	456	53	491
	<b>2</b>	416	155	429
	<b>3</b>	470	67	463
	<b>4</b>	432	122	446



## 7. Wnioski

Klasyfikacja punktów z podpunktów a) oraz c) dla większości przypadków dawała identyczne wyniki niezależnie od użytego wyznacznika czy epsilon (tolerancji). Spowodowane jest to tym, że istnieje bardzo małe prawdopodobieństwo, iż punkt będzie znajdował się dokładnie na prostej, z dokładnością do użytej tolerancji. W podpunkcie b) wybór epsilon nie miał znaczenia jednak wybór wyznacznika już tak. Dla wszystkich przypadków dla `det_2` wynikiem ilości punktów leżących na prostej jest 3, natomiast dla `det_4` jest to 7. Jest to spowodowane najprawdopodobniej tym, że współrzędne tych punktów mają duże wartości bezwzględne. Operacja klasyfikacji punktów wymaga jednak dużej precyzji, która jest tracona z powodu dużych wartości które są generowane. Mantysa liczby zmiennoprzecinkowej jest skończona a dla dużej liczby będzie przechowywana informacja o pewnej liczbie cyfr, przez co pomijane są cyfry znajdujące się na odległych miejscach dziesiętnych, które są ważne w tym ćwiczeniu. W dataframeach możemy zauważyć iż najlepsze wyniki otrzymuje się dla wartości  $\epsilon \geq 10^{-14}$ .

Na wykresach, które ilustrują różnice w klasyfikacji przy użyciu różnych wyznaczników oraz epsilon można zaobserwować, że ilość różnic zwiększa się wraz ze zwiększeniem wartości bezwzględnej współrzędnych punktów. Możemy dojść do wniosku, że niezależnie od tolerancji wszystkie wyznaczniki działają najlepiej dla punktów o bezwzględnie mniejszych współrzędnych. Można to zauważyć na niektórych wykresach liniowych.

Wyniki otrzymane przy użyciu innej precyzji (zamianie liczb na `float32`) są zdecydowanie gorsze od tych uzyskanych dla danych o precyzji `float64`. Mimo tego można jednak zauważyć, iż wyznaczniki biblioteczne sprawdzały się gorzej od tych zaimplementowanych samodzielnie. Najlepsze wyniki otrzymywane były dla wyznacznika `det_1` dla  $\epsilon \geq 10^{-14}$  oraz na ogół najgorsze wyniki dawał `det_4` (wyznacznik 2x2 przy pomocy numpy).