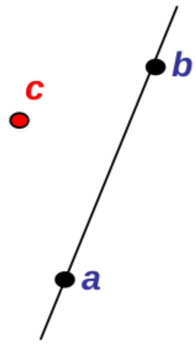
Szymon Budziak
Informatyka II rok, grupa poniedziałek 14:40 - 16:10

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.

3. Plan i sposób wykonania ćwiczenia

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

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

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

a)

	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)

	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)

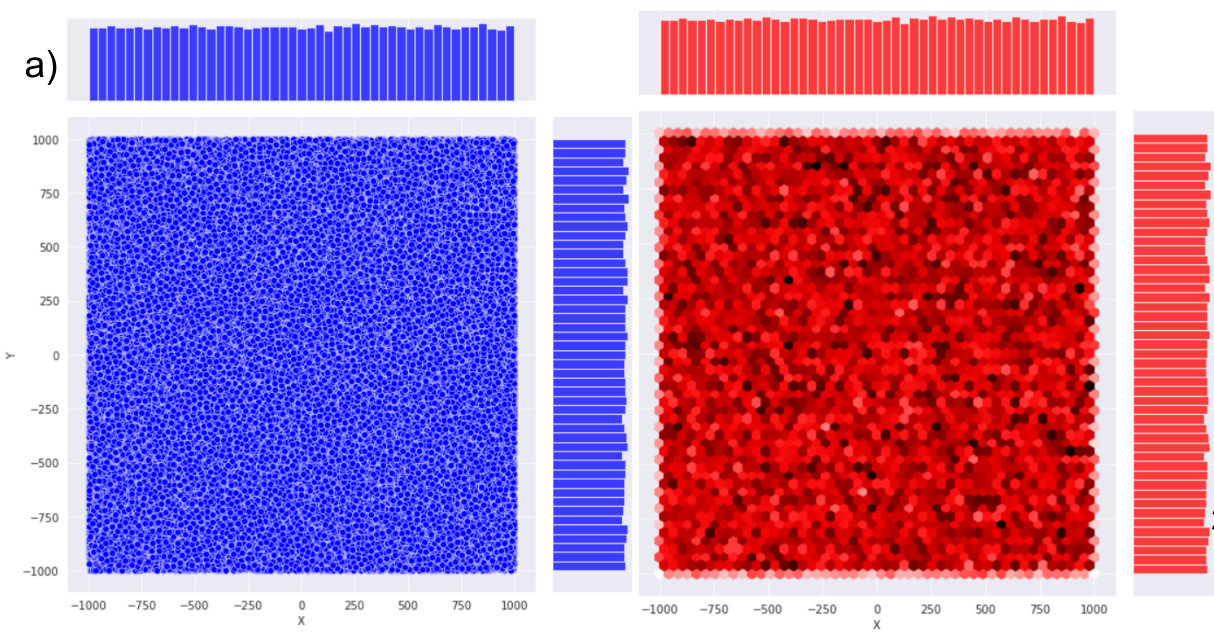
	X	Y
0	2.807673	-9.597759
1	-8.922250	4.515911
2	-5.893096	-8.079073
3	8.748726	-4.843531
4	-0.242502	9.997059

d)

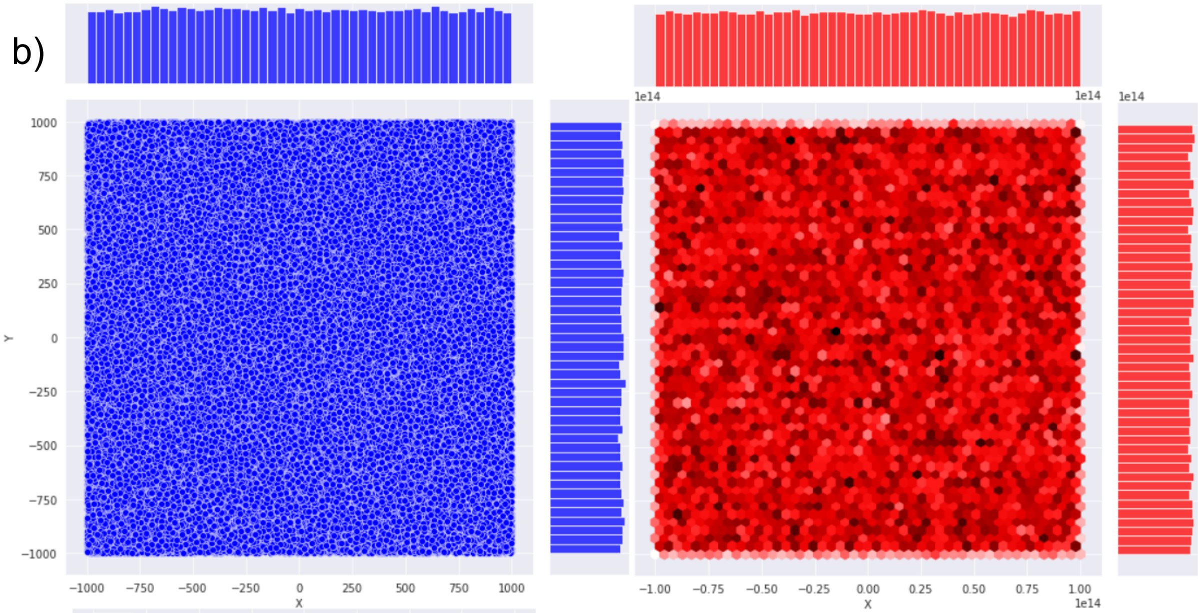
	X	Y
0	143.874008	7.243700
1	-360.162443	-17.958122
2	-324.810031	-16.190502
3	768.833667	38.491683
4	701.729060	35.136453

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.

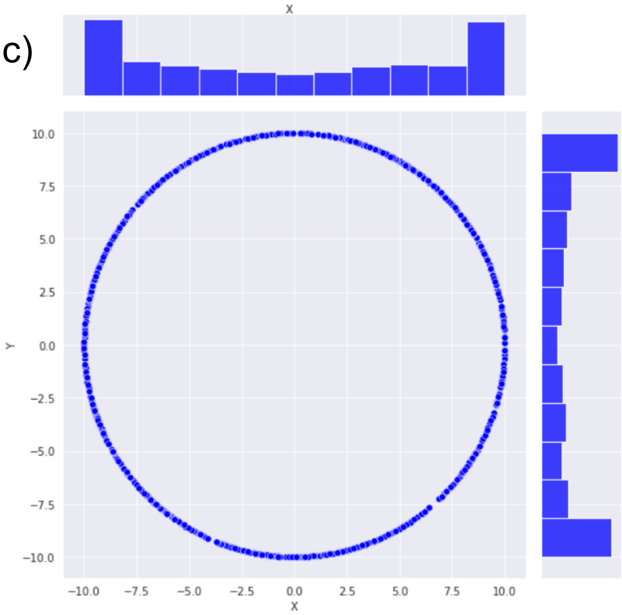
a)



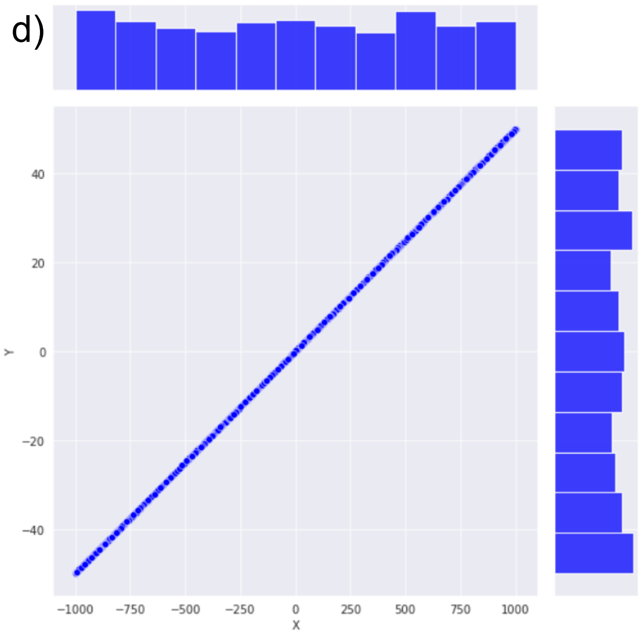
b)



c)



d)



Do podpunktów a) oraz b) zostały użyte dwa rodzaje wykresów a dla podpunktów c) oraz d) 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: $\varepsilon 1 = 10^{-14}$, $\varepsilon 2 = 10^{-10}$, $\varepsilon 3 = 10^{-8}$. Została również stworzono funkcja assign_to_dataframe, która przypisuje do odpowiedniego epsilona 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)

	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)

	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)

	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)

	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 dataframemom zauważyć, że w przykładzie a) oraz c) mamy takie same wartości dla różnych epsilonów oraz wyznaczników. W przykładzie b) oraz d) możemy zauważyć, że są one różne w zależności od użytego epsilona 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 MultiIndexing aby móc zagnieździć dla konkretnych epsilonów wszystkie wyznaczniki.

a)

		-1	0	1
e1	**1**	49846	0	50154
	2	49846	0	50154
	3	49846	0	50154
	4	49846	0	50154
e2	**1**	49846	0	50154
	2	49846	0	50154
	3	49846	0	50154
	4	49846	0	50154
e3	**1**	49846	0	50154
	2	49846	0	50154
	3	49846	0	50154
	4	49846	0	50154

b)

		-1	0	1
e1	**1**	50302	0	49698
	2	50303	3	49694
	3	50302	0	49698
	4	50299	7	49694
e2	**1**	50302	0	49698
	2	50303	3	49694
	3	50302	0	49698
	4	50299	7	49694
e3	**1**	50302	0	49698
	2	50303	3	49694
	3	50302	0	49698
	4	50299	7	49694

c)

		-1	0	1
e1	**1**	508	0	492
	2	508	0	492
	3	508	0	492
	4	508	0	492
e2	**1**	508	0	492
	2	508	0	492
	3	508	0	492
	4	508	0	492
e3	**1**	508	0	492
	2	508	0	492
	3	508	0	492
	4	508	0	492

d)

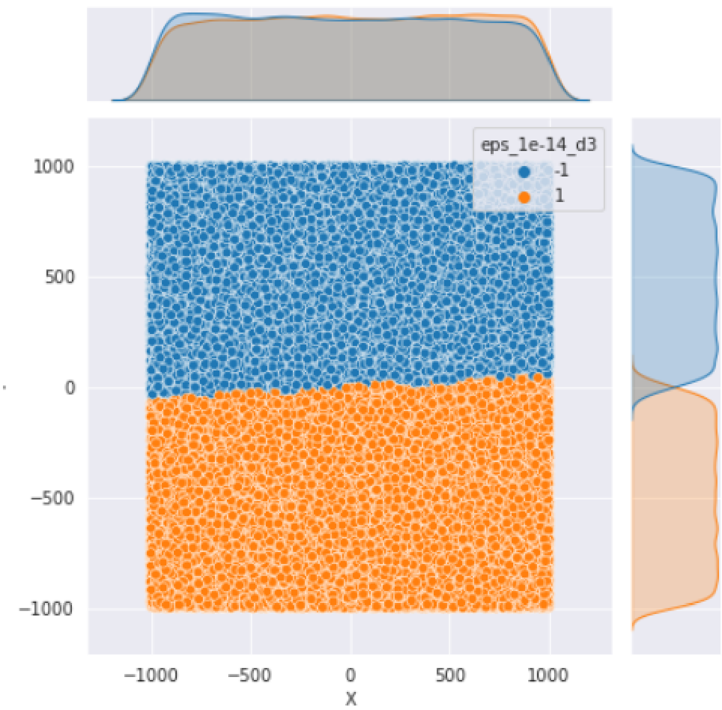
		-1	0	1
e1	**1**	0	1000	0
	2	149	702	149
	3	14	882	104
	4	165	680	155
e2	**1**	0	1000	0
	2	0	1000	0
	3	0	1000	0
	4	0	1000	0
e3	**1**	0	1000	0
	2	0	1000	0
	3	0	1000	0
	4	0	1000	0

e1, e2, e3 odpowiadają odpowiednio $\varepsilon 1 = 10^{-14}$, $\varepsilon 2 = 10^{-10}$, $\varepsilon 3 = 10^{-8}$ oraz 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

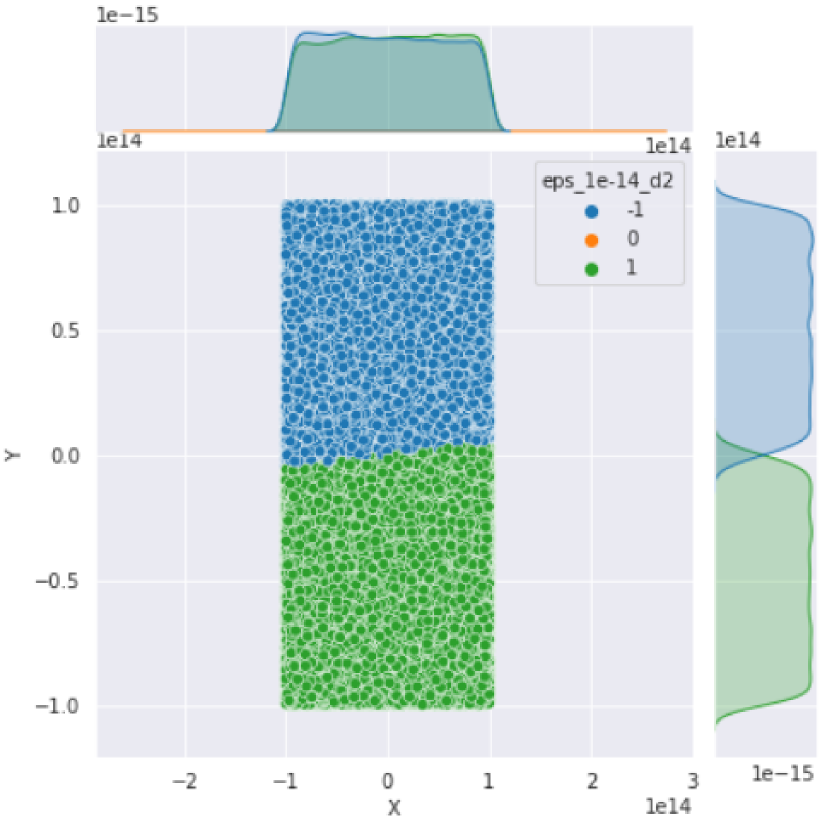
Kolory na wykresach są opisane w legendach które znajdują się obok wykresów albo pod nimi.

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

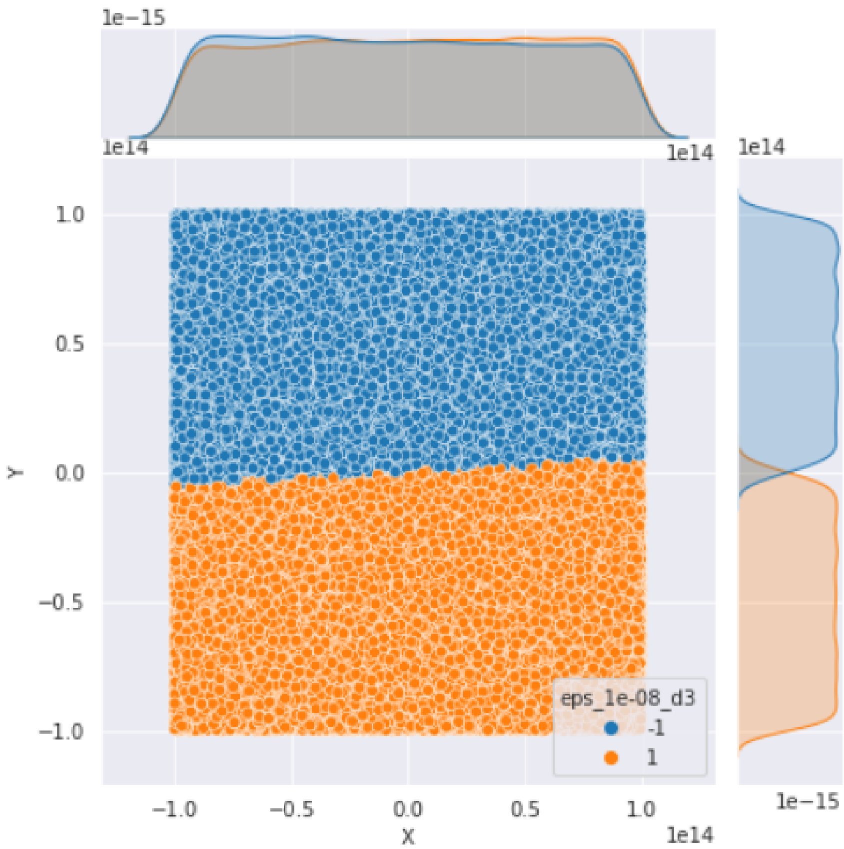


- klasyfikacja punktów z podpunktu b)

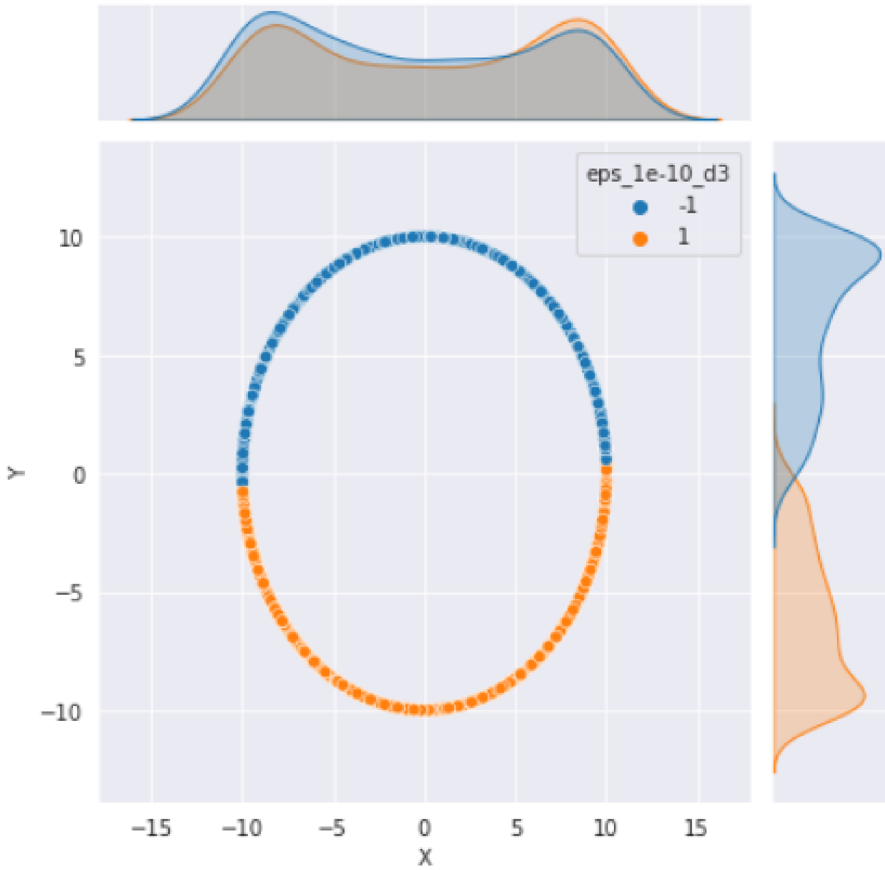
dla ε = 10^{-14} i det_2 (wyznacznik 2x2 samodzielnie)



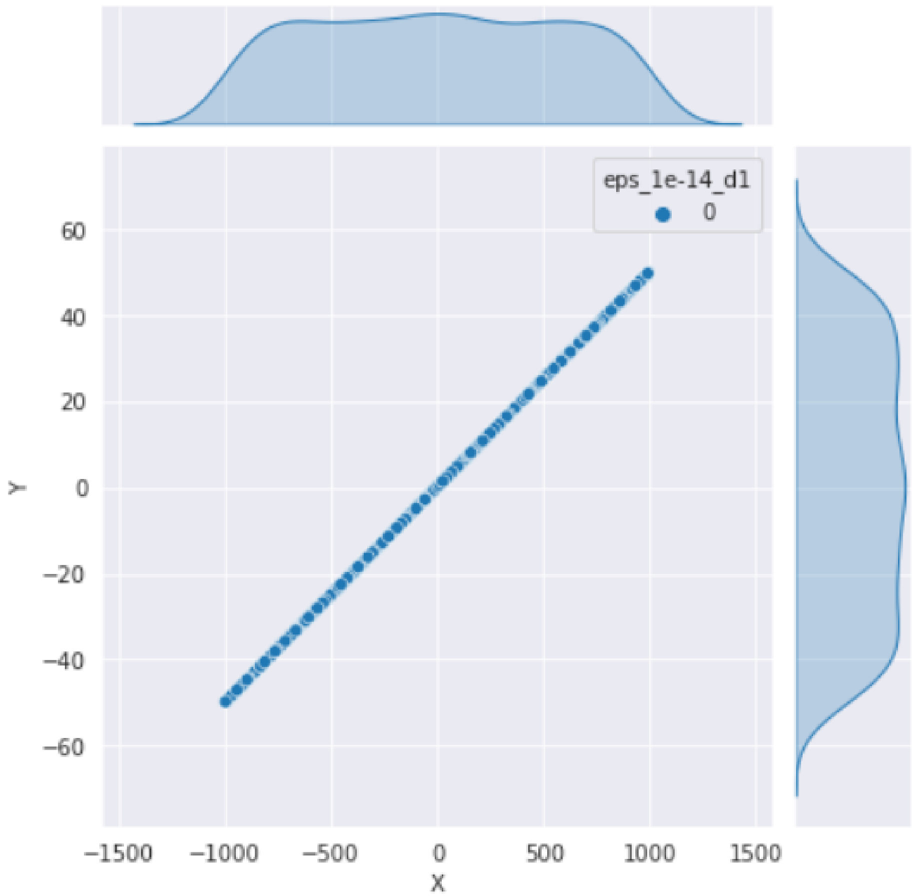
dla ε = 10^{-14} i det_2 (wyznacznik 2x2 samodzielnie)



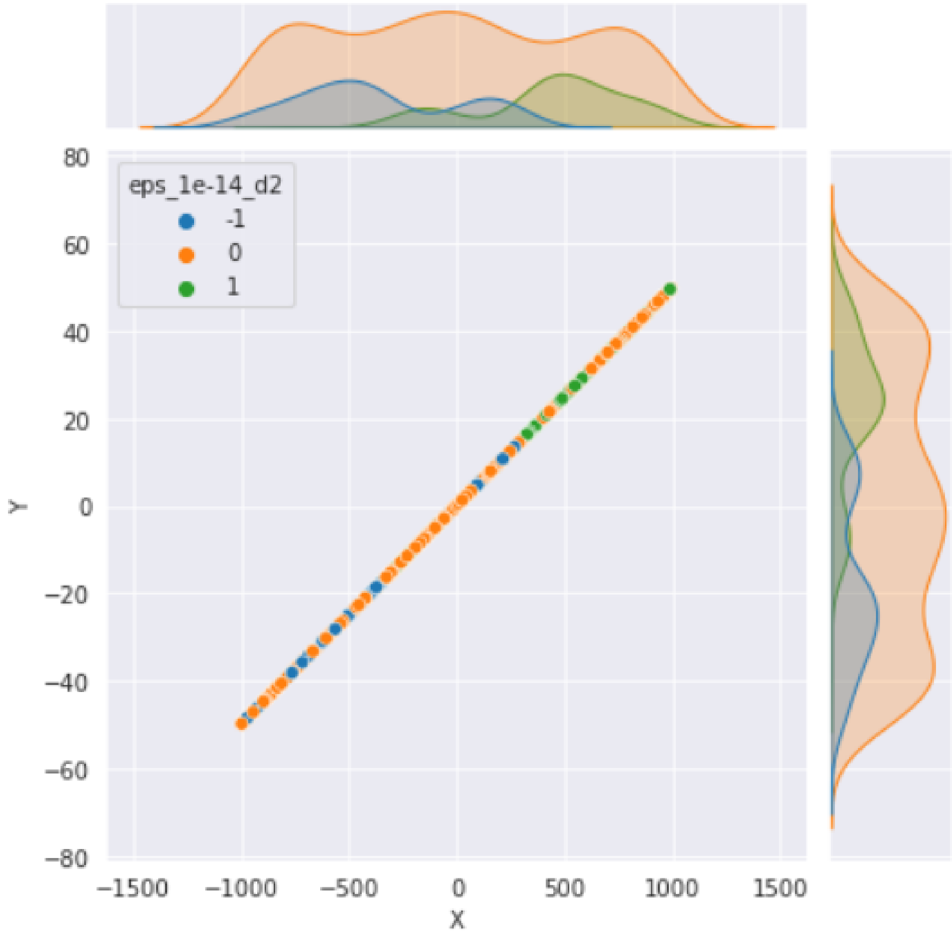
- klasyfikacja punktów z podpunktu c) (wszystkie wykresy są identyczne dla wszystkich ε i wyznaczników)



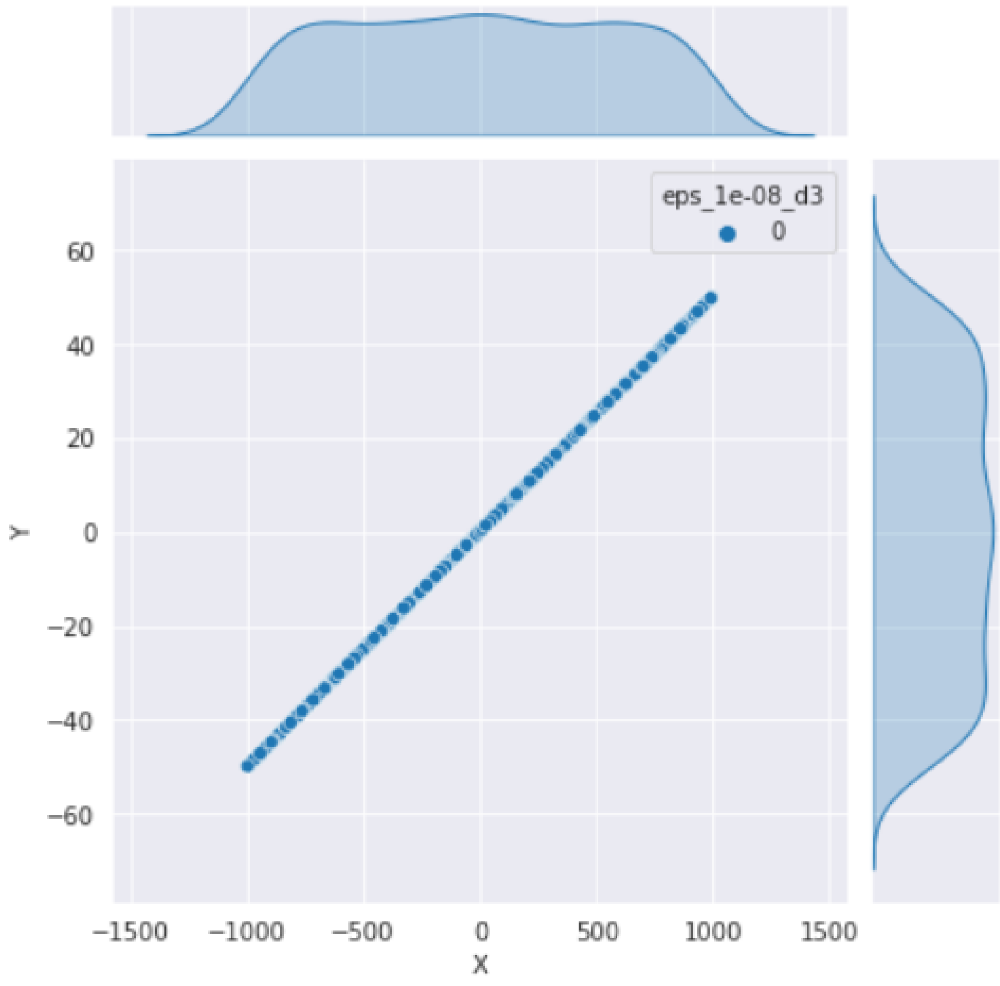
- klasyfikacja punktów z podpunktu c)
  dla $\varepsilon = 10^{-14}$ i det_1 (wyznacznik 3x3 samodzielnie)



dla ε = 10^{-14} i det_2 (wyznacznik 2x2 samodzielnie)

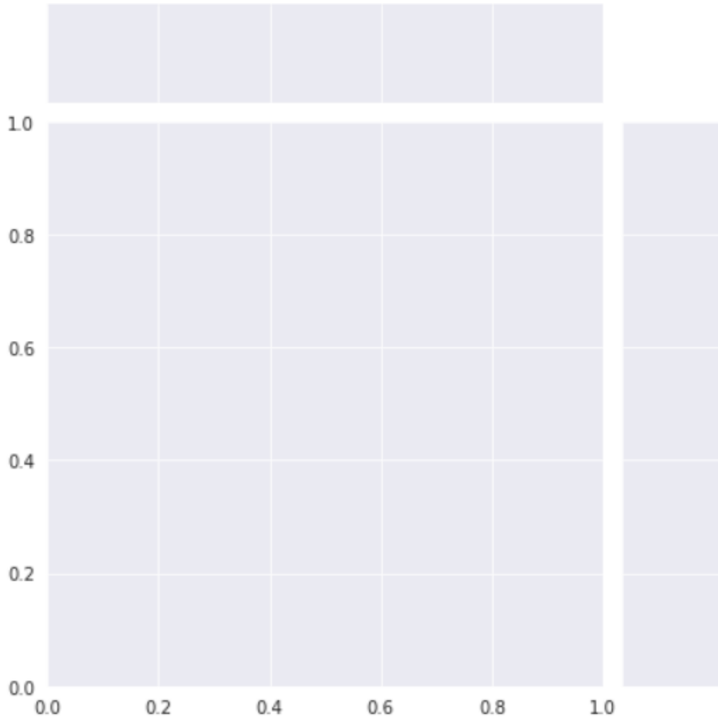


dla ε = 10^{-8} i det_3 (wyznacznik 3x3 przy pomocy numpy)



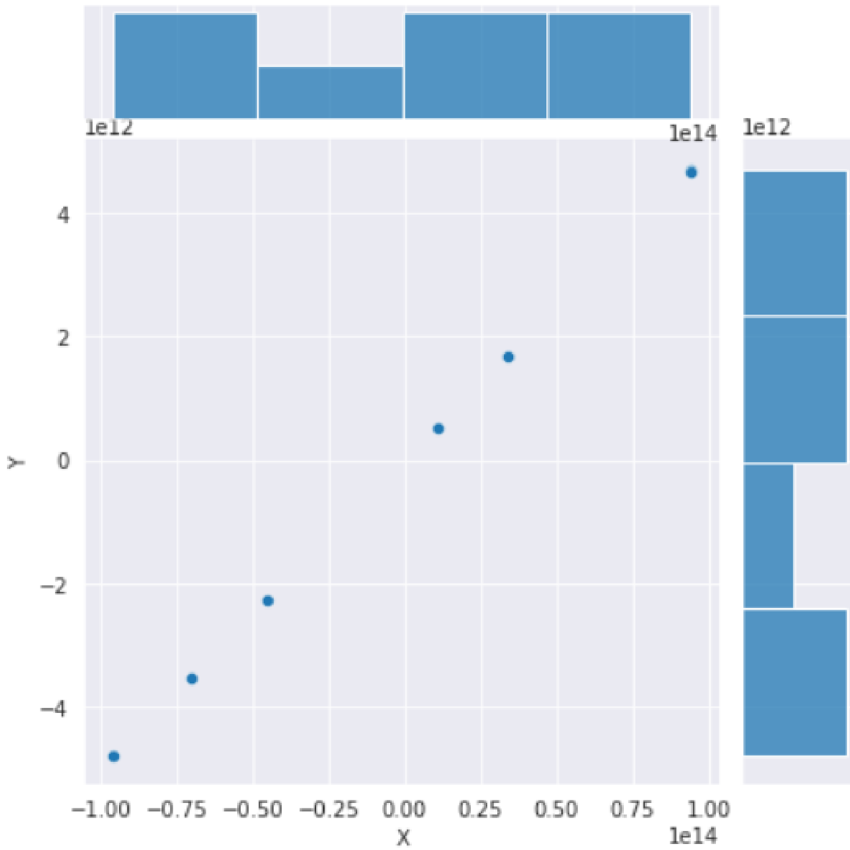
Wykresy dla punktów, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilona

- W podpunkcie a) 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.

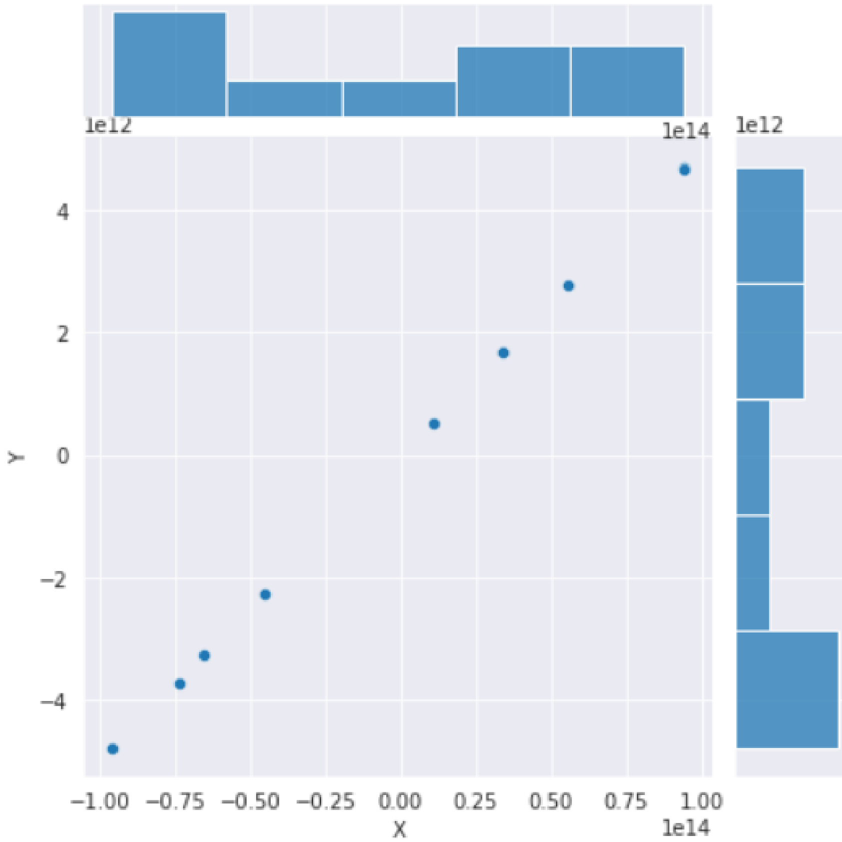


- W podpunkcie b) występują punkty które zostały inaczej zaklasyfikowane przez różne wyznaczniki.

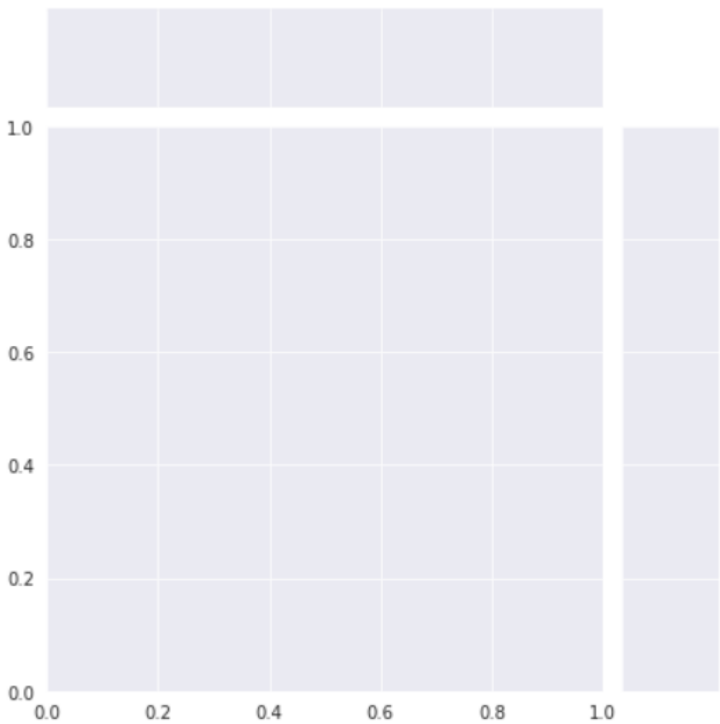
dla $\varepsilon = 10^{-14}$ i det_1 z $\varepsilon = 10^{-14}$ i det_2



dla ε = 10^{-14} i det_3 z ε = 10^{-8} i det_4

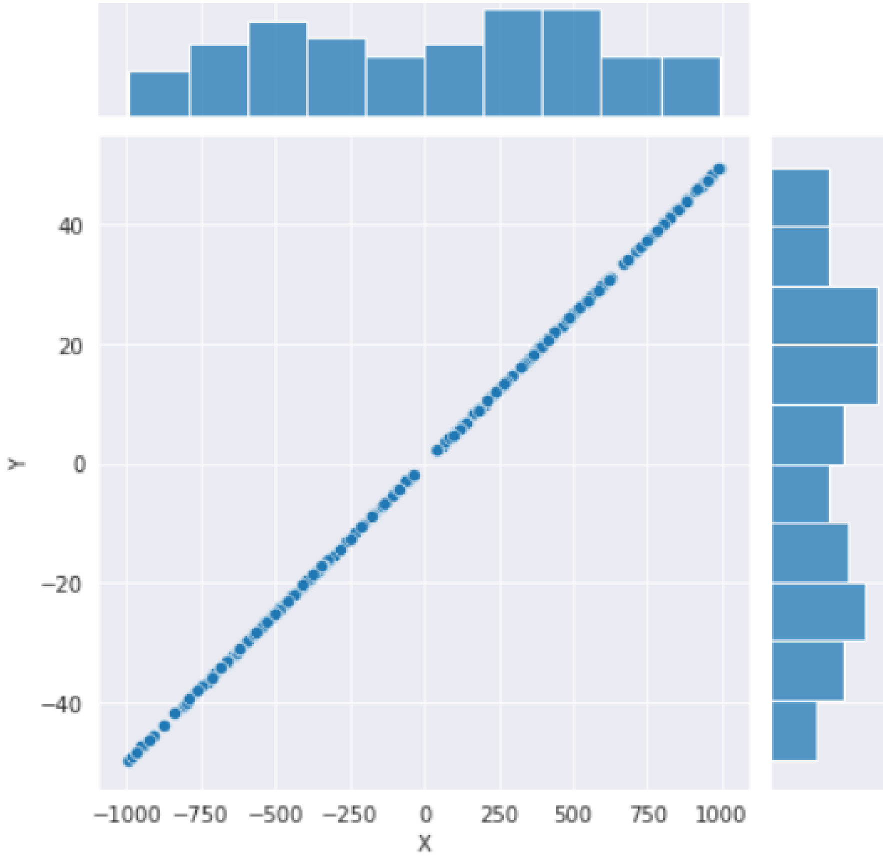


- W podpunkcie c) wyniki dla wszystkich 4 wyznaczników oraz dla różnych epsilonów są takie same, nie różnią się między sobą więc wykresy są puste. Wszystkie punkty zostały sklasyfikowane tak samo.

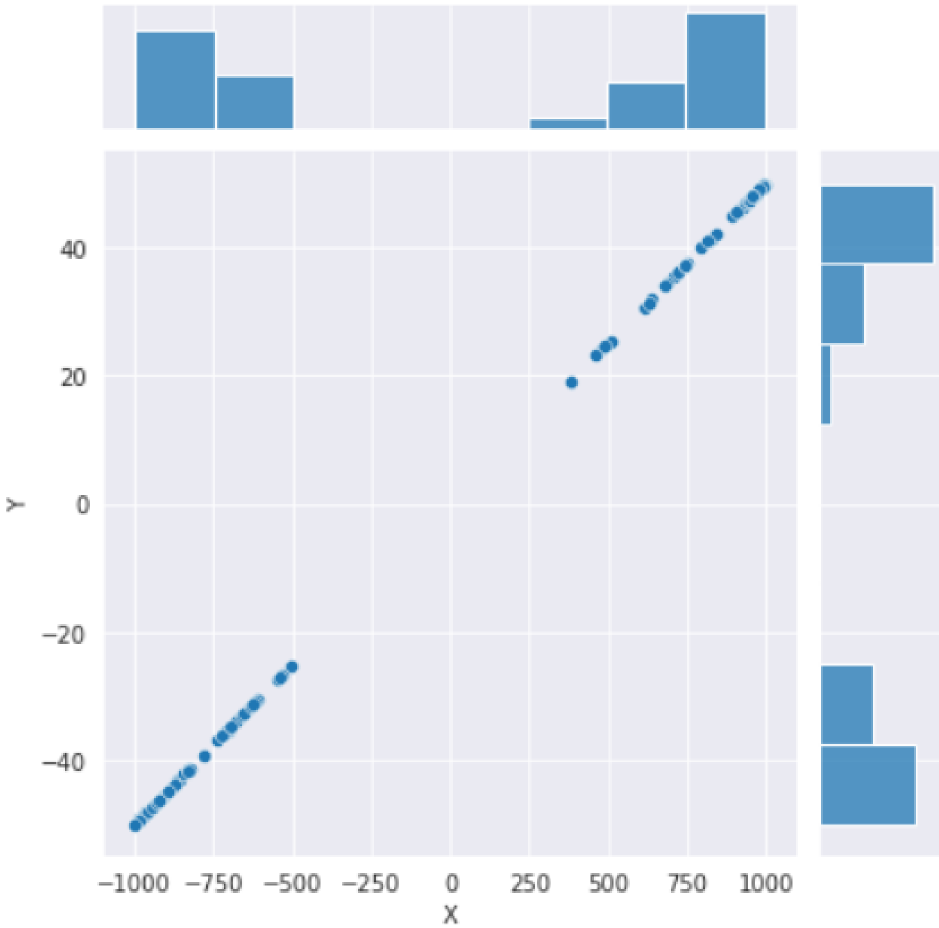


- W podpunkcie d) występują punkty które zostały inaczej zaklasyfikowane przez różne wyznaczniki.

dla $\varepsilon = 10^{-14}$ i det_2 z $\varepsilon = 10^{-10}$ i det_3



dla $\varepsilon = 10^{-14}$ i det_3 z $\varepsilon = 10^{-10}$ i det_3



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)

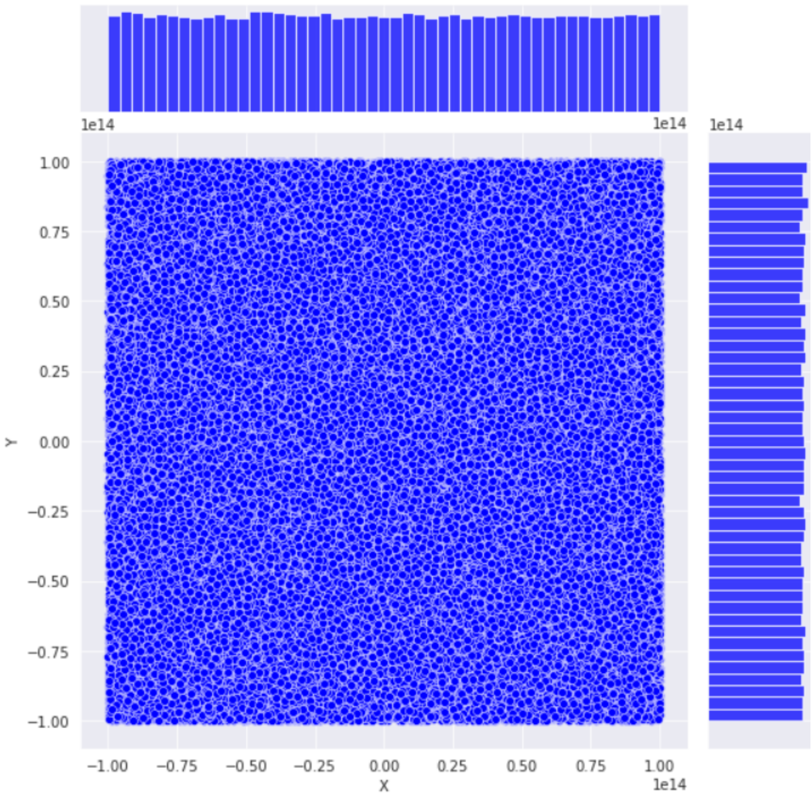
	X	Y
0	-2.407648e+13	-6.095850e+12
1	8.156353e+13	5.364797e+13
2	-1.927577e+13	5.362377e+13
3	-2.333296e+13	4.585456e+13
4	8.642588e+13	8.367844e+13

d)

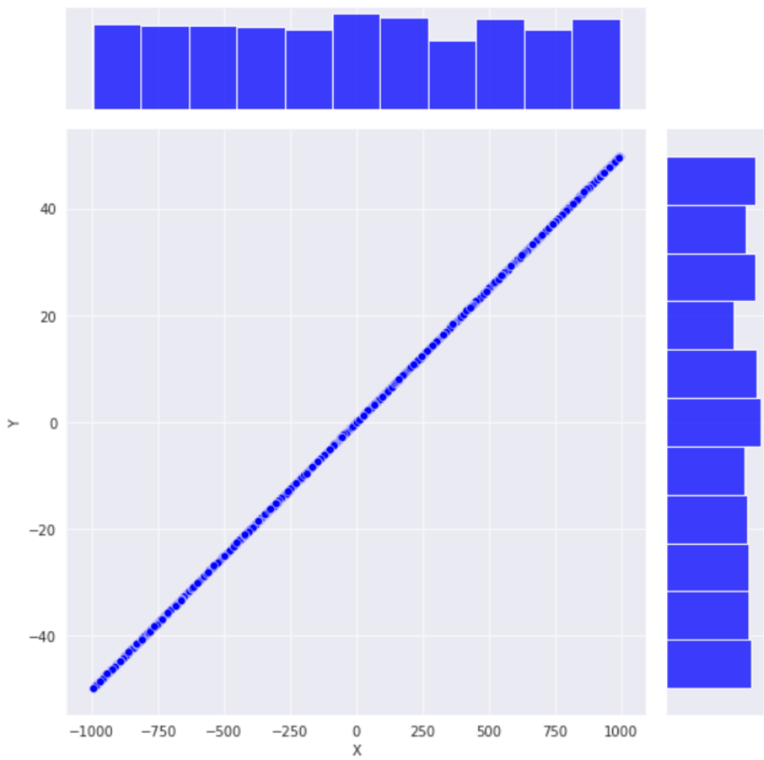
	X	Y
0	-694.067871	-34.653393
1	-569.224426	-28.411221
2	-405.260193	-20.213009
3	-321.059631	-16.002981
4	718.095032	35.954754

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

b)



d)



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 epsilonów i różnych wyznaczników. Użyte epsilony to $\varepsilon 1 = 10^{-16}$ i $\varepsilon 2 = 10^{-10}$.

b)

	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

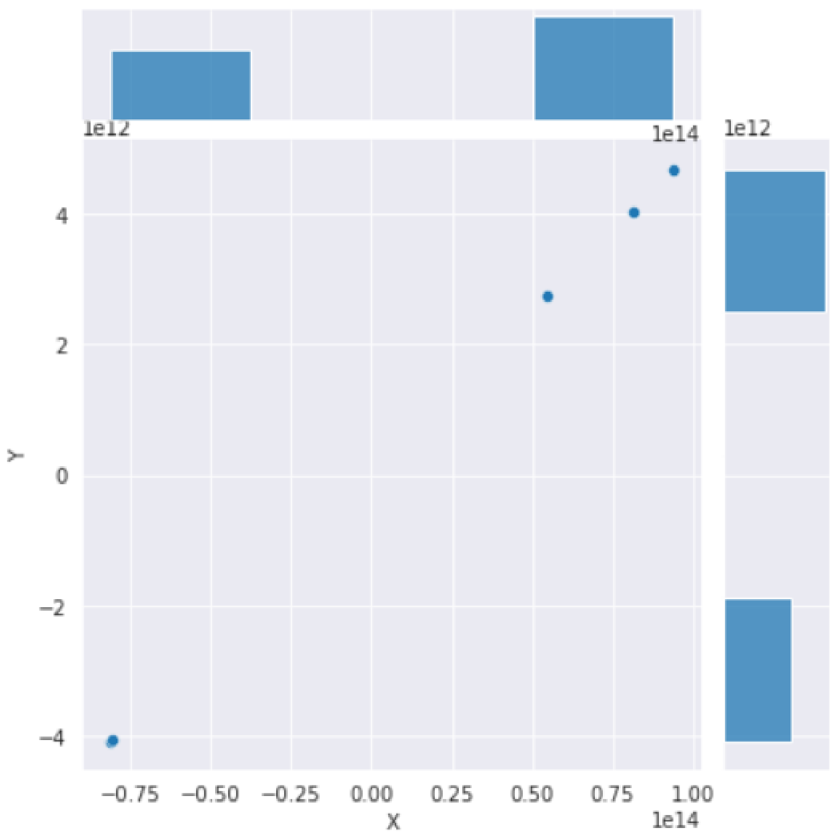
d)

	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

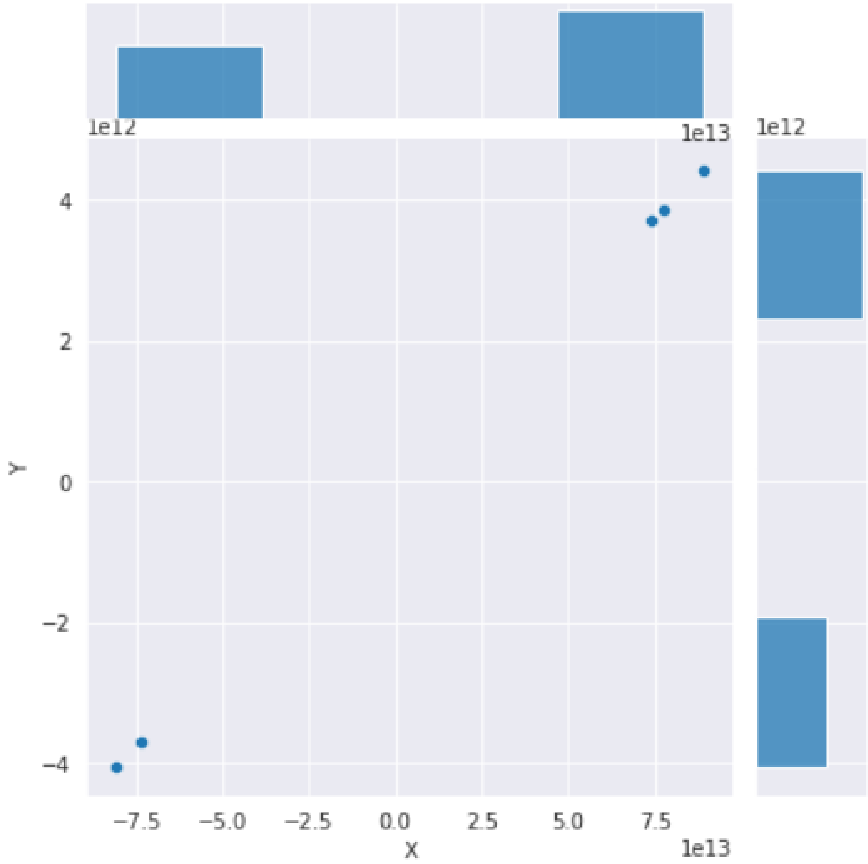
Wybrane wykresy dla punktów o zmienionej precyzji, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilona.

- dla podpunktu b)

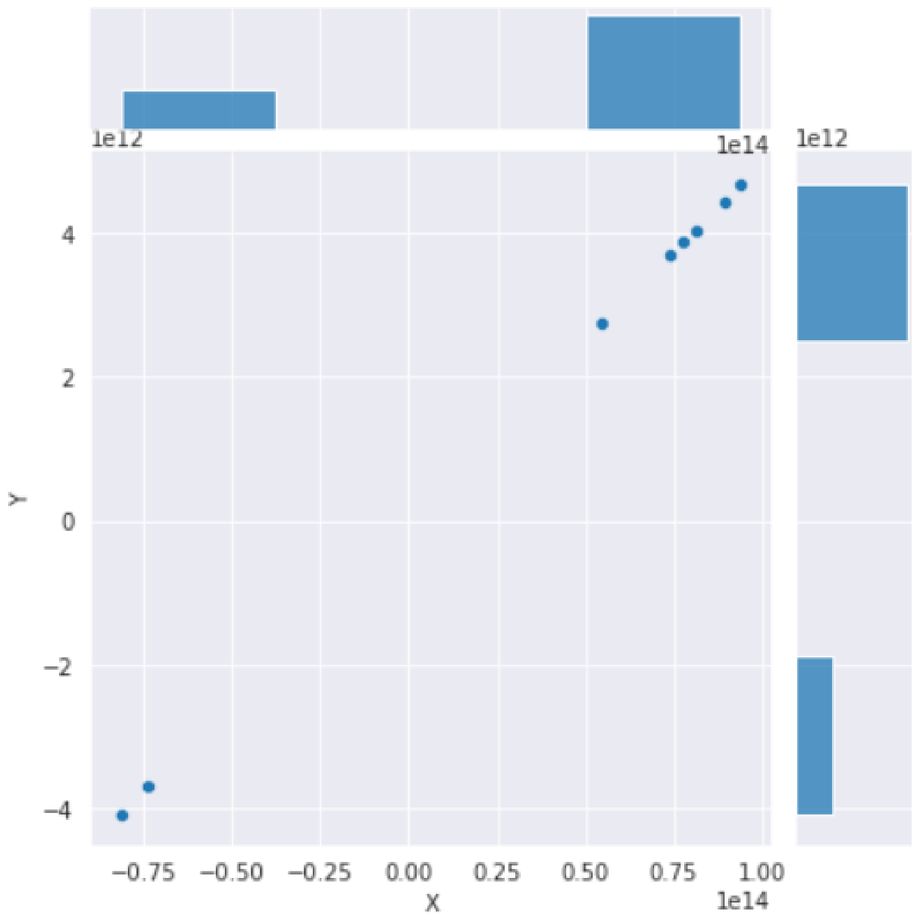
dla $\varepsilon = 10^{-16}$ i det_2 z $\varepsilon = 10^{-16}$ i det_3



dla ε = 10^{-16} i det_2 z ε = 10^{-16} i det_4

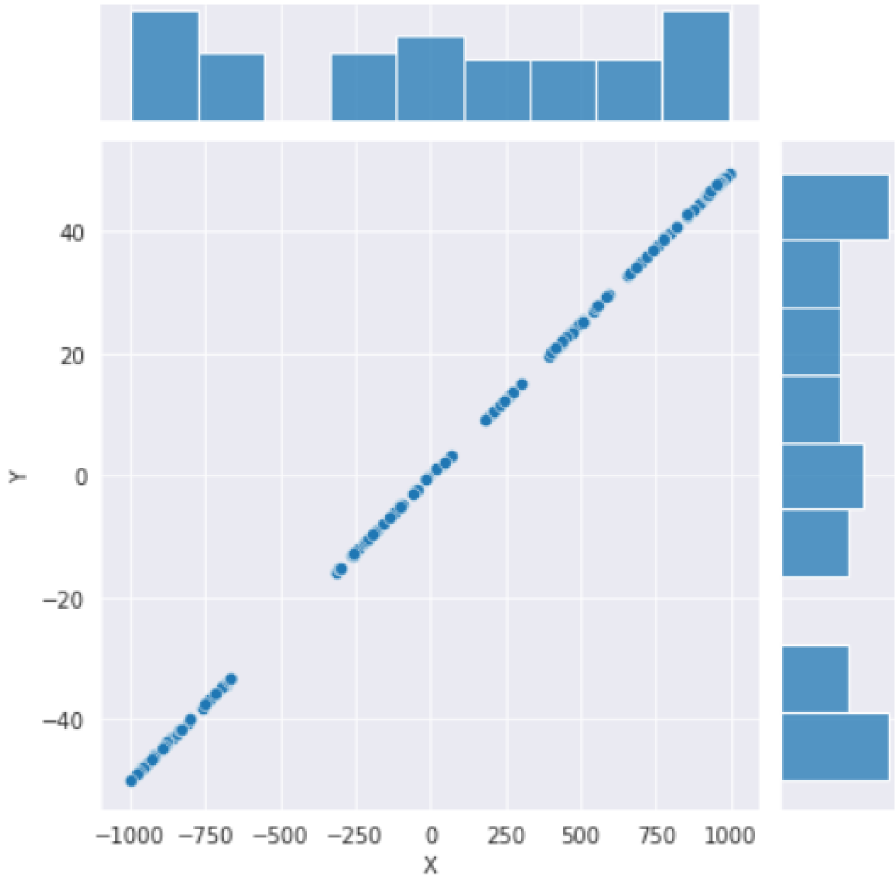


dla ε = 10^{-16} i det_3 z ε = 10^{-10} i det_4

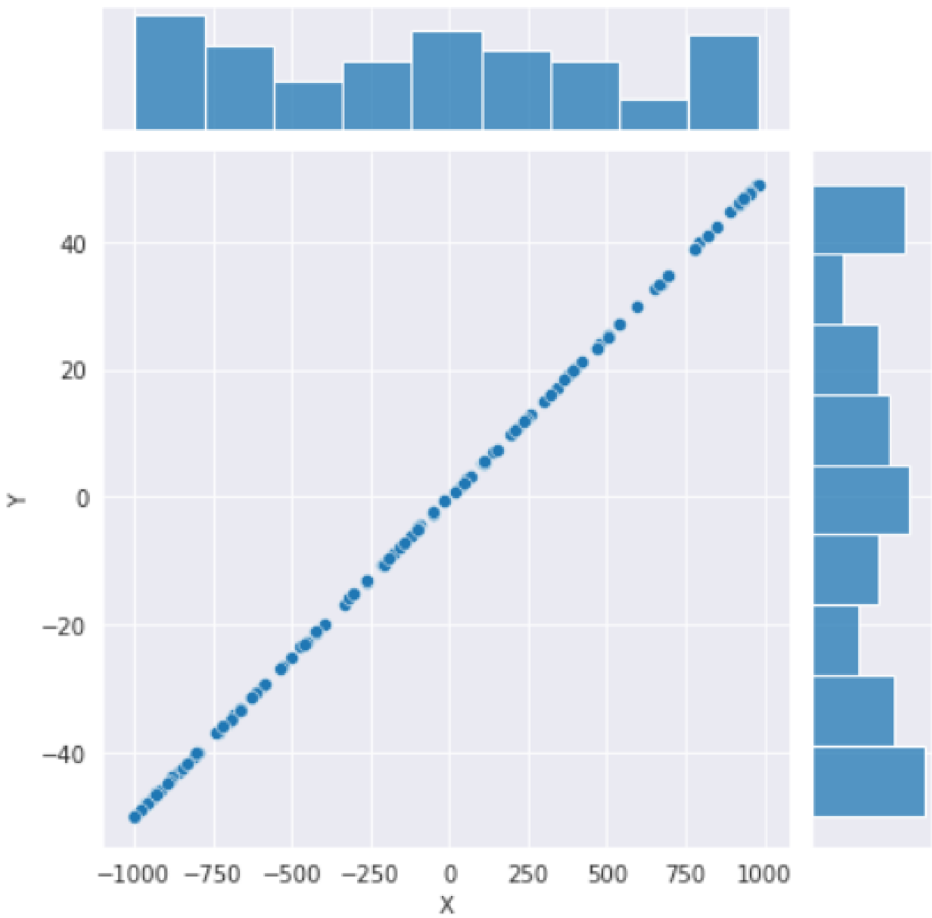


- dla podpunktu d)

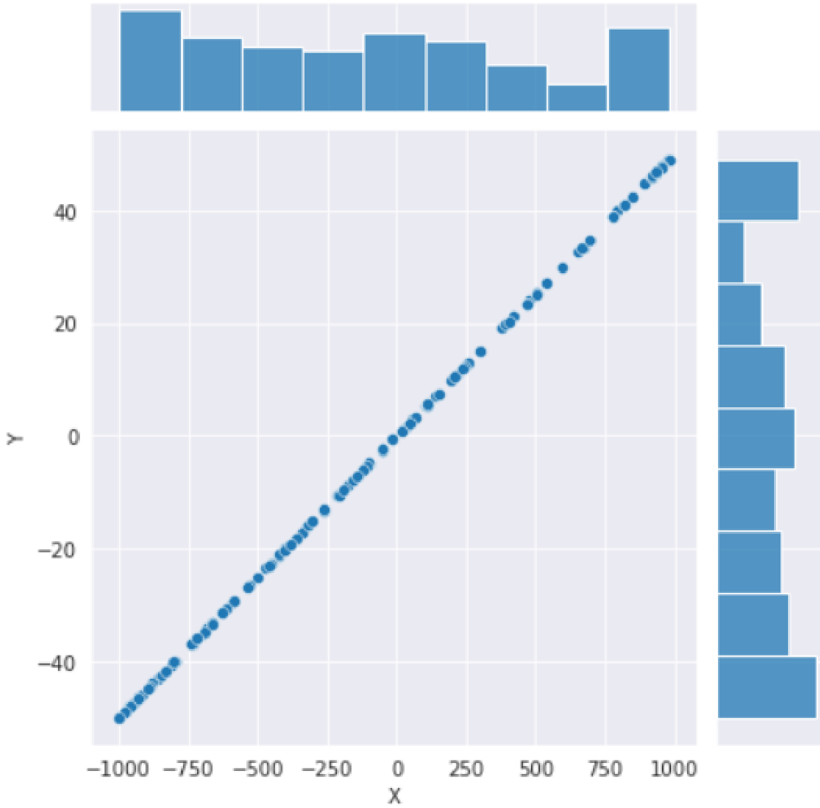
dla ε = 10^{-16} i det_1 z ε = 10^{-10} i det_1



dla ε = 10^{-16} i det_2 z ε = 10^{-16} i det_3



dla ε = 10^{-16} i det_3 z ε = 10^{-10} i det_4



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

b)

		-1	0	1
e1	**1**	49850	0	50150
	2	49848	5	50147
	3	49850	0	50150
	4	49848	3	50149
e2	**1**	49850	0	50150
	2	49848	5	50147
	3	49850	0	50150
	4	49848	3	50149

d)

		-1	0	1
e1	**1**	465	48	487
	2	419	164	417
	3	481	62	457
	4	437	128	435
e2	**1**	410	184	406
	2	410	184	406
	3	410	184	406
	4	410	184	406

7. Wnioski

Klasyfikacja punktów z podpunktów a) oraz c) dla większości przypadków dawała identyczne wyniki niezależnie od użytego wyznacznika czy epsilona (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 epsilona 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 $\varepsilon >= 10^{-14}$.

Na wykresach, które ilustrują różnice w klasyfikacji przy użyciu różnych wyznaczników oraz epsilonów 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 $\varepsilon >= 10^{-14}$ oraz na ogół najgorsze wyniki dawał det_4 (wyznacznik 2x2 przy pomocy numpy).