

Matematyka

Moduł `math` w bibliotece standardowej

Biblioteka `math` implementuje podstawowe operacje matematyczne. Pełna lista funkcji, wraz z opisami, dostępna jest po wywołaniu komendy `help(math)`. Funkcje biblioteki `math` wykonują operacje na pojedynczych liczbach (nie na listach).

```
1 import math
2
3 math.sin()
4 math.cos()
5 math.tan()
6 math.pi
```

Moduł `statistics` w bibliotece standardowej

Moduł `statistics` pozwala na wykonywanie podstawowych operacji statystycznych, w tym obliczanie średnich, wariancji i odchylenia standardowego.

```
1 import statistics
2
3 statistics.avg()
4 statistics.mean()
5 statistics.stdev()
```

Moduł `random` w bibliotece standardowej

```
1 import random
2
3 random.sample()
4 random.random()
```

Moduł `matplotlib` (nie w bibliotece standardowej)

Moduł `matplotlib` pozwala na rysowanie wykresów i diagramów. Jest to bardzo rozbudowana biblioteka z setkami opcji konfiguracyjnych. Najczęściej używanym modułem biblioteki `matplotlib` jest moduł `pyplot`, który implementuje szereg funkcji umożliwiających rysowanie wykresów 2d.

Podstawowe użycie jest następujące.

```
1 from matplotlib import pyplot as plt
```

```
1 plt.plot(0, 0, 'o')
2 plt.show()
```

```
1 x1 = [x*0.01 for x in range(0,628)]
2 y1 = [math.sin(x*0.01)+random.gauss(0, 0.1) for x in range(0,628)]
3 plt.plot(x1, y1)
4
5 x2 = [x*0.5 for x in range(0,round(63/5))]
6 y2 = [math.cos(x*0.5) for x in range(0,round(63/5))]
7 plt.plot(x2, y2, 'o-')
8
9 plt.show()
```

Zadania kontrolne

Obliczanie odległości między dwoma punktami - Euclidean Distance

Dla dwóch (constant) punktów A i B o podanych koordynatach napisz program, który obliczy odległość między nimi wykorzystując algorytm Euclidesa.

Napisz tę funkcję tak, żeby przeszła doctest:

```
1 def euclidean_distance(A, B):
2     """
3     >>> euclidean_distance((0,0), (1,0))
4     1.0
5
6     >>> euclidean_distance((0,0), (1,1))
7     1.4142135623730951
8
9     >>> euclidean_distance((0,1), (1,1))
10    1.0
11
12    >>> euclidean_distance((0,10), (1,1))
13    9.055385138137417
14    """
15    pass
```

Zadanie z gwiazdką

: Przekształć algorytm tak, aby działał w N wymiarowej przestrzeni.

Wyliczanie odległości w celu oszacowania przynależności do zbioru. Zwróć uwagę, że bez względu na ilość wymiarów wzór się niewiele różni.

Zadanie z gwiazdką 2

:

Wygeneruj 100 losowych punktów (rozkład gaussa o średniej 0, dowolnym odchyleniu standardowym(np. 0.2)) wokół dwóch dowolnie wybranych punktów (np. A=[0, 1], B=[2, 4]).

Napisz do tego celu funkcję, która przejdzie doctest:

```
1 def random_point(center, std=0.2):
2     """
3     >>> random.seed(1); random_point((0,0), std=0.2)
4     (0.2576369506310926, 0.2898891217399542)
5
6     >>> random.seed(1); random_point((0,0))
7     (0.2576369506310926, 0.2898891217399542)
8
9     >>> random.seed(1); random_point((2,5), std=10)
10    (14.881847531554628, 19.494456086997708)
11
12    >>> random.seed(1); random_point((2,5), std=(0.1, 12))
13    (2.1288184753155464, 22.393347304397253)
14    """
15    pass
```

Wyrysuj te punkty na wykresie (możesz użyć funkcji `plt.axis('equal')` żeby osie wykresu były w tej samej skali). Punkt A i punkty wygenerowane na jego podstawie wyrysuj kolorem czerwonym (argument `color='red'` w funkcji `plt.plot`), a punkt B i punkty wygenerowane na jego podstawie wyrysuj kolorem niebieskim. Możesz do tego celu napisać funkcję `plot_point(point, color)`, która przyjmuje punkt (dwuelementowy tuple, lub listę, z czego pierwszy element to współrzędna x, a druga to y), i kolor i doda ten punkt do aktualnie aktywnego rysunku.

Korzystając z funkcji napisanej w ćwiczeniu powyżej oblicz odległość od każdego z punktów do punktów A i B oraz na podstawie tej odległości zaklasyfikuj te punkty (jeżeli punkt jest bliżej punktu A to należy do zbioru A, jeżeli jest bliżej do zbioru B to należy do zbioru B). Narysuj nowy wykres, na którym punkty ze zbioru A będą narysowane kolorem czerwonym, a punkty ze zbioru B kolorem niebieskim.

Czy dwa wykresy są takie same? Co się stanie jeżeli będziemy zwiększali odchylenie standardowe przy generacji punktów? Albo przybliżymy do siebie punkty A i B?

Przeliczenia trygonometryczne

Napisz program, który wczyta od użytkownika wielkość kąta w stopniach i wyświetli wartość czterech podstawowych funkcji trygonometrycznych (sin, cos, tg, ctg) o ile dla danego kąta jest to możliwe.

Zadanie z gwiazdką

: Jeżeli funkcja trygonometryczna nie istnieje dla danego kąta, zwróć wyjątek `ValueError('dla tego kąta wartość funkcji nie istnieje')`

Lotto

Napisz program, który wyświetli 6 losowych i nie powtarzających się liczb z zakresu od 1 do 49.

Podpowiedź

: - `random.randrange()` - `random.sample()`

Pytania

: - Czym są liczby pseudolosowe? - Czy da się stworzyć program czysto losowy? - Dlaczego?

Pole trójkąta

Napisz program, który obliczy pole trójkąta, pod warunkiem że użytkownik poda wysokość i długość podstawy tego trójkąta. Uwzględnij, że wysokość i długość podstawy mogą być liczbami niecałkowitymi. Wykorzystaj doctest do przetestowania funkcji.