
Wykład 4. Metoda Monte Carlo

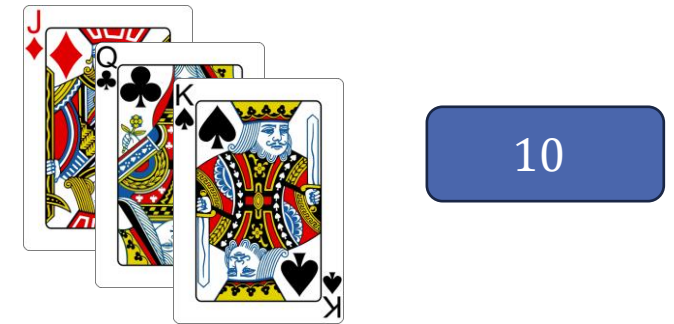
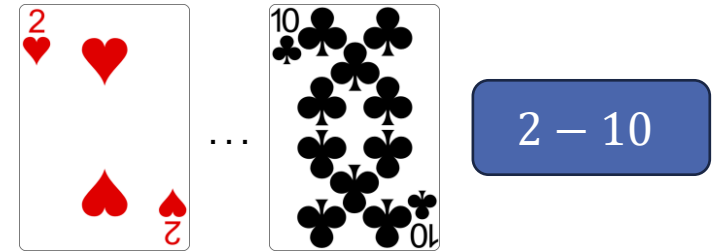
od bomb atomowych po Black Jacka

Piotr Morawiecki
24 marca 2025



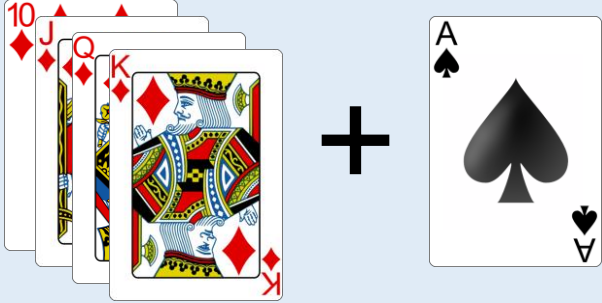
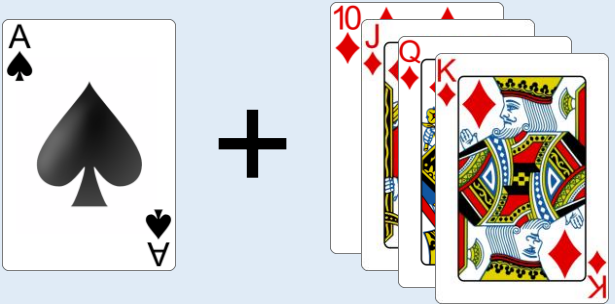
Motywacja: black jack (1/2)

- W grze *black jack* gracz dopiera kolejne karty z talii.
- Karty są warte następującą liczbę punktów:
 - Karty od dwójki do dziesiątki mają wartość równą numerowi karty.
 - Walet, dama i król mają wartość równą 10 punktów.
 - As ma wartość równą 1 lub 11, w zależności co jest lepsze dla gracza.
- Gracz w dowolnym momencie może skończyć dobierać karty.
- Celem gry jest uzyskanie jak najwyższej liczby punktów, jednak jeśli gracz przekroczy 21 punktów, to automatycznie przegrywa.

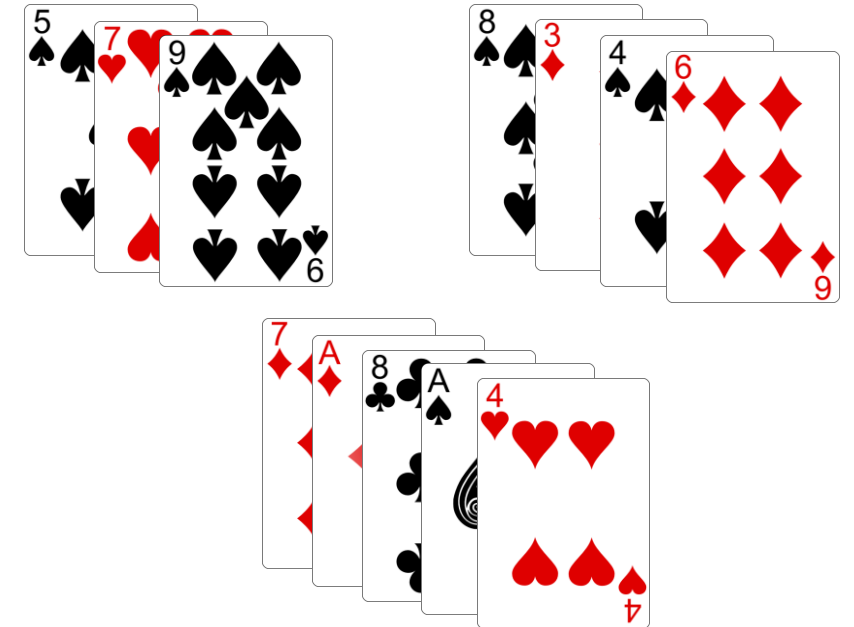


Motywacja: black jack (2/2)

- Wyzwanie: *Jaka jest szansa na uzyskanie dokładnie 21 punktów?*


$$P = \frac{16}{52} \cdot \frac{4}{51} = \frac{4}{221} \approx 0.024$$

$$P = \frac{4}{52} \cdot \frac{16}{51} = \frac{4}{221} \approx 0.024$$

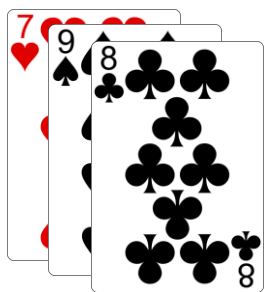
Mamy jednak wiele innych konfiguracji:



Problem staje się stanowczo zbyt złożony...

Metoda Monte Carlo

- Jeśli proces (np. gra w Black Jacka) jest zbyt złożony, aby można było przewidzieć jego wynik za pomocą podejścia analitycznego, może nam pomóc tzw. *metoda Monte Carlo*.
- Metoda polega na wielokrotnym losowym odgrywaniu danego procesu, a następnie na podstawie jego (losowych) wyników oszacowywanie szukanej wartości.



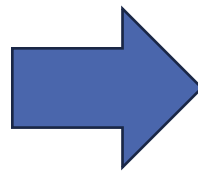
24



27



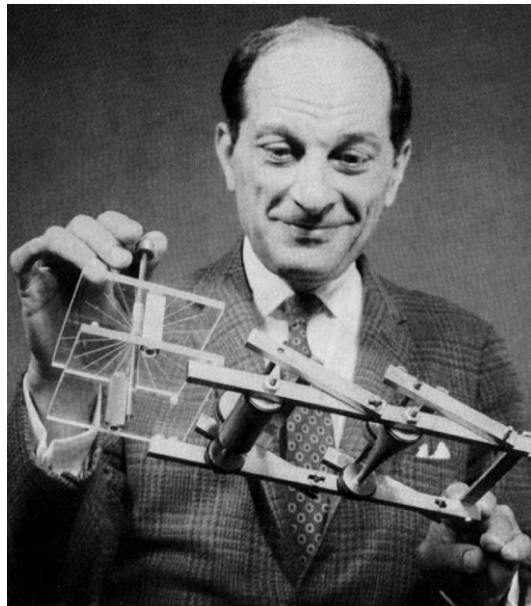
21



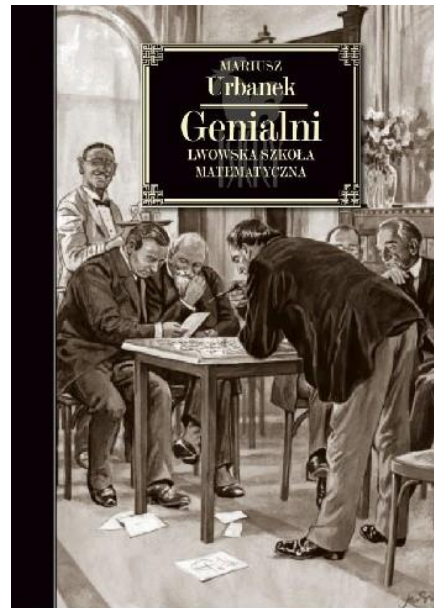
$$P = \frac{\text{ile razy uzyskaliśmy 21}}{\text{liczba powtórzeń}} = \frac{46}{1000} = 4.6\%$$

Historia metody Monte Carlo

- Metoda Monte Carlo została zapoczątkowana przez polskiego matematyka Stanisława Ulama.



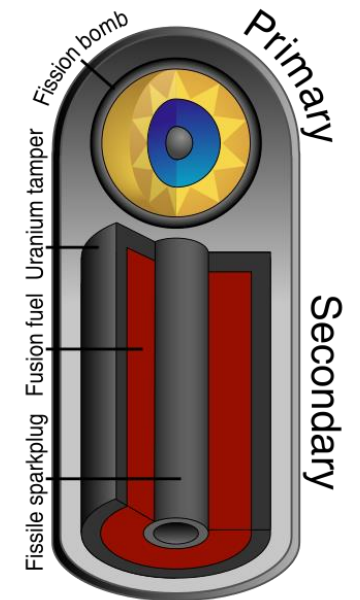
Stanisław Ulam
(1909 - 1984)



Lwowska Szkoła
Matematyki



Projekt Manhattan

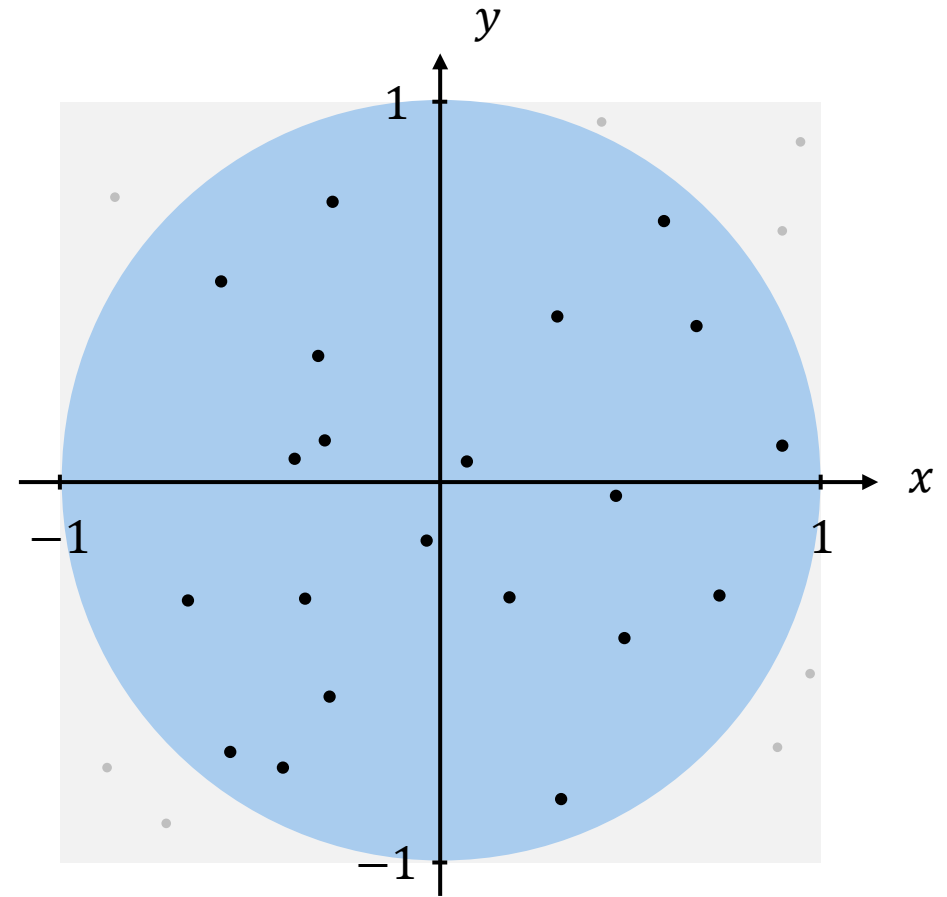


Konfiguracja
Teller-Ulama

Szacowanie powierzchni koła (1/5)

- Za pomocą metody Monte Carlo spróbujemy oszacować pole koła o promieniu $r = 1$.
- Powierzchnia koła to $P = \pi r^2 = \pi$.
- Oszacujemy ją losowo umieszczając punkty w kwadracie 2×2 .
- Powierzchnię koła znajdujemy z:

$$\frac{\text{liczba punktów w kole}}{\text{liczba punktów w kwadracie}} \approx \frac{\text{powierzchnia koła}}{\text{powierzchnia kwadratu}}$$



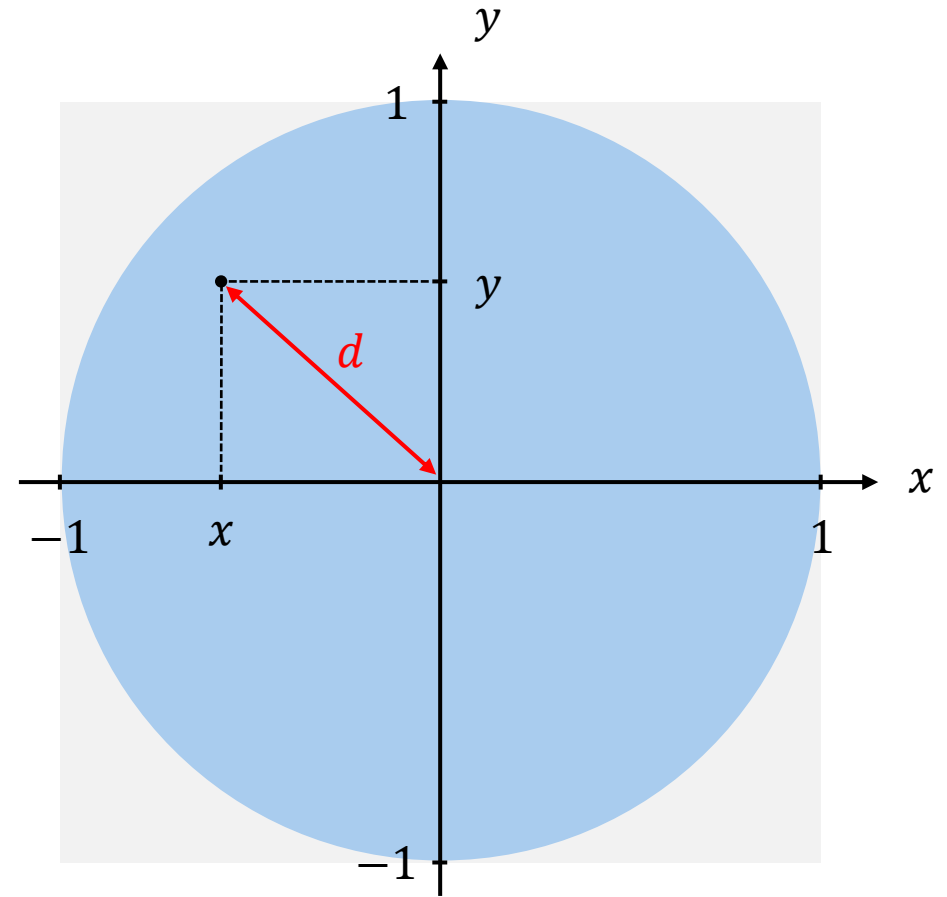
Szacowanie powierzchni koła (2/5)

- Jak sprawdzić czy punkt (x, y) znajduje się wewnątrz koła?
- Odległość od środka to:

$$d = \sqrt{x^2 + y^2}$$

- Punkt znajduje się w kole kiedy $d \leq r$.
Zatem:

$$\begin{aligned}d &\leq 1 \\ \sqrt{x^2 + y^2} &\leq 1 \\ x^2 + y^2 &\leq 1\end{aligned}$$

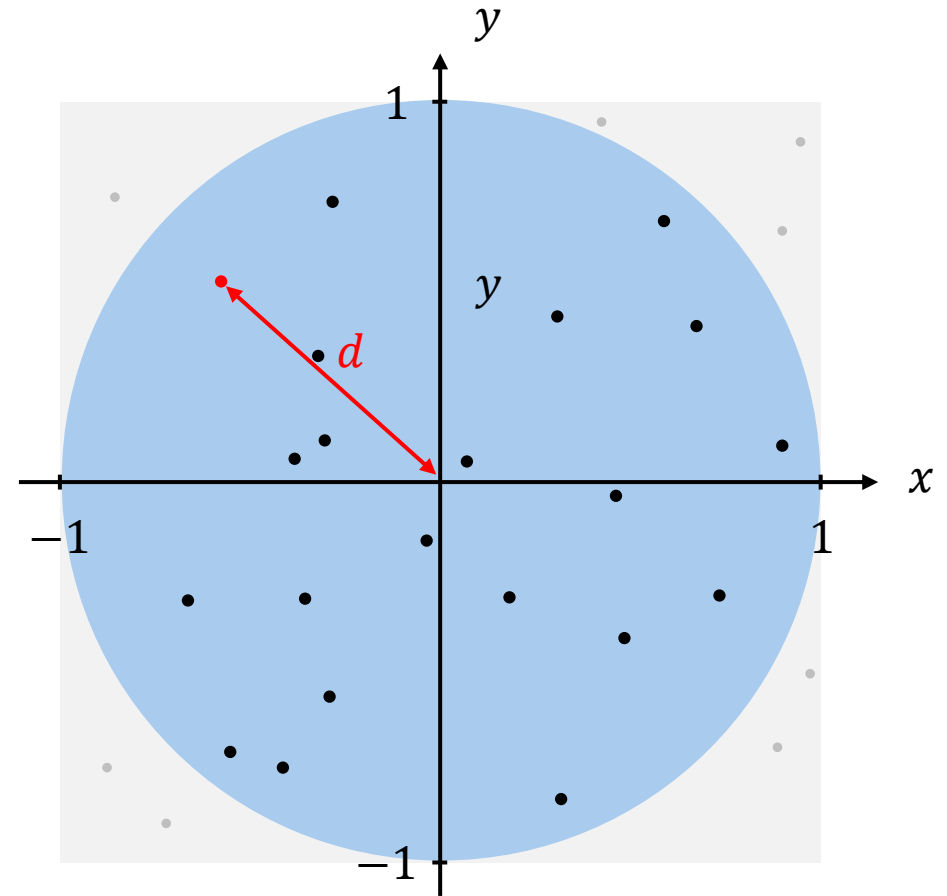


Szacowanie powierzchni koła (3/5)

```
import random

def estimate_pi(num_samples):
    num_inside = 0
    for i in range(num_samples):
        x = random.uniform(-1, 1)
        y = random.uniform(-1, 1)
        if x**2 + y**2 <= 1:
            num_inside += 1
    return 4 * num_inside / num_samples

print(estimate_pi(1000))
```



Szacowanie powierzchni koła (4/5)

```
for i in range(10):  
    print(estimate_pi(100))
```

2.80	3.08	3.24	3.08	2.92
2.92	3.20	3.12	3.12	2.92

```
for i in range(10):  
    print(estimate_pi(10000))
```

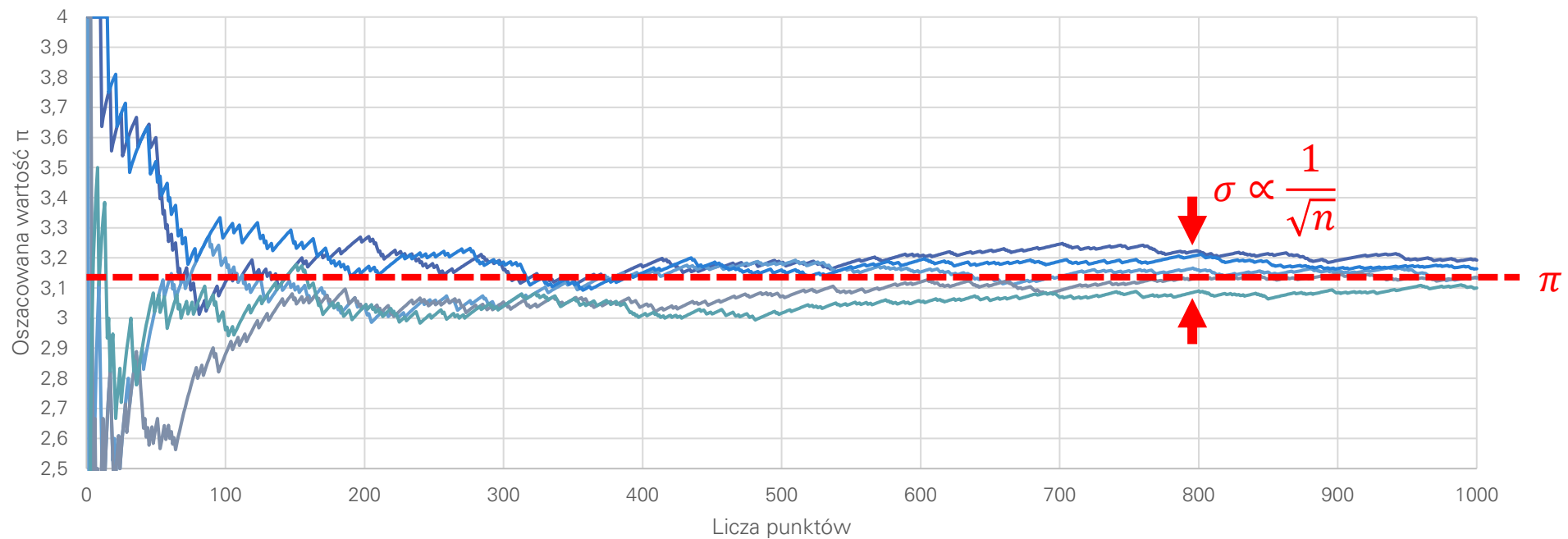
3.1412	3.1492	3.1520	3.1480	3.1064
3.1208	3.1720	3.1464	3.1384	3.1512

```
for i in range(10):  
    print(estimate_pi(1000000))
```

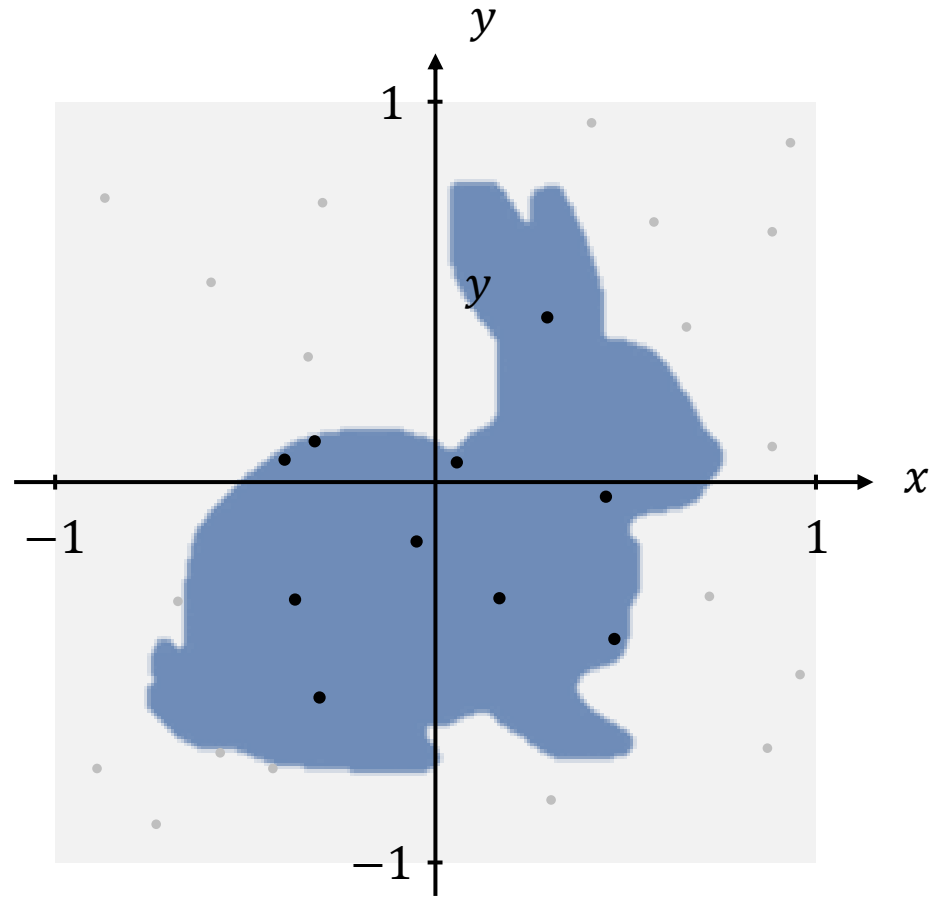
3.144908	3.142296	3.140420	3.143548	3.141392
3.141396	3.141936	3.143892	3.142024	3.142344

Szacowanie powierzchni koła (5/5)

- Zgodnie z *Prawem Wielkich Liczb*, gdy liczba prób losowego doświadczenia dąży do nieskończoności, średnia arytmetyczna wyników tych prób dąży do ich wartości oczekiwanej.

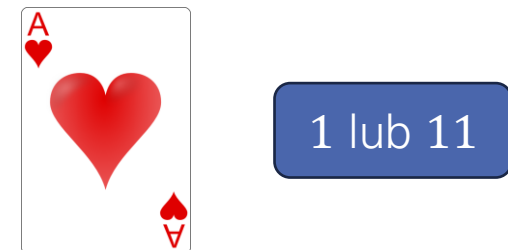
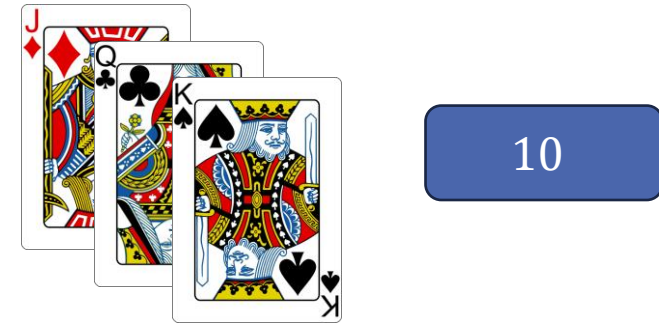
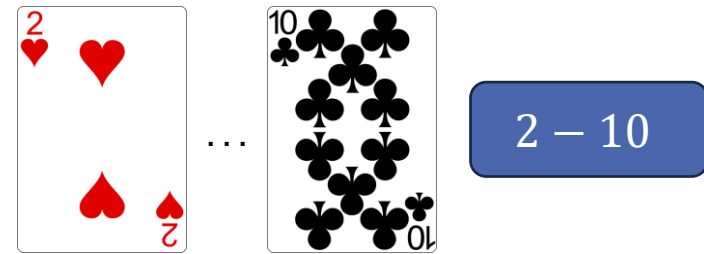


Szacowanie pola bardziej złożonych figur



Case study: black jack (1/3)

- **Wyzwanie:** *Jaka jest szansa na uzyskanie dokładnie 21 punktów?*
- Zasymulujemy n rund Black Jacka.
- Dobieramy karty tak długo aż nie uzyskamy 21 punktów lub więcej.
- Na podstawie stosunku liczby gier w których uzyskaliśmy 21 punktów do wszystkich gier oszacujemy poszukiwane prawdopodobieństwo.



Case study: black jack (2/3)

Metoda Monte Carlo

```
import random

blackjack = 0
n = 10000

for i in range(n):
    cards = [2, 3, 4, 5, 6, 7, 8, 9, 10, "J", "Q", "K", "A"]
    cards = cards * 4
    picked_cards = []

    while score(picked_cards) < 21:
        random_card = random.choice(cards)
        picked_cards.append(random_card)
        cards.remove(random_card)

    if score(picked_cards) == 21:
        blackjack += 1

print(blackjack / n)
```

Funkcja do sumowania punktów na kartach

```
def score(picked_cards):
    score = 0
    n_aces = 0

    for card in picked_cards:
        if card == "J" or card == "Q" or card == "K":
            score += 10
        elif card == "A":
            score += 1
            n_aces += 1
        else:
            score += card

    for i in range(n_aces):
        if (score <= 11):
            score += 10

    return score
```

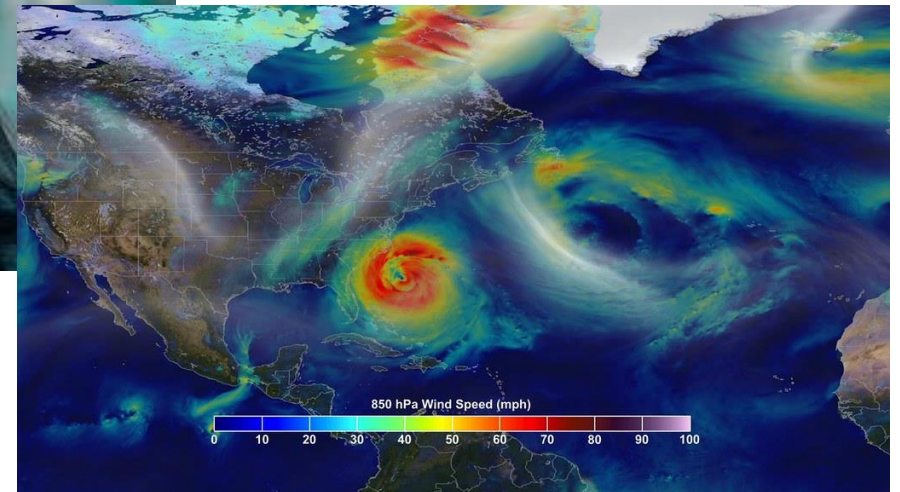
Case study: black jack (3/3)

- W wyniku uzyskałem następujące prawdopodobieństwa:

0.1869	0.1779	0.1725	0.1768	0.1772	0.182	0.1704	0.1838	0.181	0.1757
0.1825	0.1789	0.1784	0.1812	0.1808	0.1794	0.1793	0.1817	0.1845	0.1758

- Zatem prawdopodobieństwo uzyskania 21 punktów wynosi około 17 – 19%.
 - Inne ciekawe pytania:
 - Średnio po dobraniu ilu kartach uzyskujemy 21 punktów?
 - Jaka jest szansa, że będzie wśród nich As?
 - Jaka jest szansa na uzyskanie co najmniej 20, 19, 18, itd. punktów?
 - ...
-

Inne zastosowania metody Monte Carlo



Podsumowanie wykładu

- Jeśli proces jest zbyt złożony, aby można było przewidzieć jego wynik za pomocą podejścia analitycznego, może nam pomóc tzw. *metoda Monte Carlo*.
- W metodzie tej wielokrotnie losowo odgrywamy danego procesu, a następnie na podstawie jego wyników oszacowujemy szukaną wartości (np. szansę uzyskania 21 punktów w *Black Jacku*).
- Zgodnie z *Prawem Wielkich Liczb*, im więcej powtórzeń wykonamy, tym dokładniejsze będzie nasze oszacowanie.

