##projekt PYTHON

-- UWAGA: przeczytać pliki ze slacka nt. generatorów

Sprawdza: wiedza statystyczna + wiedza algorytmiczna

Timeline: 3-4 zajęć projektowych

\*Bez przerostu formy nad treścią. Prezentacja to pokazanie wyników działania kodu. Nie trzeba

\* robić ogromnej otoczki typu kolor wykresu itd.

Zadanie w projekcie:

1. implementacja metody: generatora ciagu liczb

2. ciag spelnia testy statystyczne

3. zilustrowanie tych dane - wykresy

Komputer działa deterministycznie, nie potrafi wygenerować losowych danych.

Generatory ustalają zmienne na podstawie wyznaczników, które wydają nam się losowe.

Tematy:

1. LCG (liniowy generator kongruentny) - RUNDOOM - metoda deterministyczna

2. LFG (generator Fibonacciego) - VENUS - metoda deterministyczna

3. generator liczb z rozkładu liniowego/normalnego (2 metody) - APACZE

4. generator metodą eliminacji - trzeba zdefiniować metodę za pomocą funkcji - kilka rozkładów (3-4) do uzgodnienia z grupą - ESPERANTO

Krzysiek:

są dwie metody ale tylko metoda Boxa-Muellera jest do zaimplementowania oraz metoda z centralnego twierdzenia granicznego

w drugim dokumencie jest dokladnie opisane jak je zrobic.

FUnkcje sinus cosinus sa wbudowane w scipy albo w numpy.

Najpierw wybieramy liczbe losowa (w numpy). Rozklad jednostajny jest warunkiem poczatkowym.Str. 18 drugiego pliku

Krzysiek przesle strone gdzie jest to opisane algorytmicznie.

Wyjsciowa funkcja (standardowo jest 0,1) - zadana średnia i zadane odhylenie standardowe - trzeba napisac funkcje.

ALGORYTM - plik w ktorym to jest opisane - str. 16.

Nastepnie analiza tych n + statystyka czy spelnia test czy nie

Losujemy liczby, warto zobaczyc to ile razy + robimy test statystyczny

potem drugi raz i znowu test (oczywiscie w petli tzw. metoda monte carlo - dla n=7 losowalismy to 1000 razy i 50% nie przeszlo testu statystycznego; losujemy 30 liczb losowych z kazdego algorytmu i sprawdzamy czy przechodza test statystyczny)

1. Napisac 2 funkcje do losowania z rozkladow + porownanie testow statystycznych

2. wykresy.

3. Do funkcji warto zrobic scenariusze np. ze wprowadzamy parametr, ale gdy wprowadzimy zly to pojawi sie jakis warning.

4. Porownanie tego z normalna funkcja numpy.

Dla rozkladu normalnego sa dostepne w numpy

test kołmogorowa smirnoffa???

test Shafiro (Scipy) jest info w necie o tym jak interpretowac wyniki tego testu

TEST METODA MONTECARLO (19.01.2019)

Losujemy 100 liczb i dla nich robimy te testy.

jesli powtorzymy to 10000 razy to wiemy ze dla ciagu tych liczb mamy w 80% pozytywny wynik testu, a dla 1000 tylko w 90%.

To sie robi dla metody o centralnym twierdzeniu granicznym

dla metody boxa muellera

najlepiej funkcja (argument, liczba iteracji oraz funkcja losujaca tzn. albo numpy, albo boxmueller, albo ctg)

Rezultat - ladna tabelka ktora porownuje rezultalty tych tabel

CENTRALNE TWIERDZENIE GRANICZNE – 20.01.2019

Statystyka prawdopodobieństwa.

*Pisalismy kod do CTG.*

*Ostatecznie Krzysiek go napisał.*

1. Zmienna losowa X ma rozklad normalny N(mi, sigma) [mi = srednia w populacji generalnej; sigma = odchylenie standardowe] N oznacza r.normalny
2. Należy obliczyć prawdopodobieństwo, ze zmienna x przyjmie wartości w jakims przedziale (miedzy np. -1 do 1)
3. Aby to obliczyć należy przeprowadzić standardyzacje zmiennej N(mi,sigma) na taki inny układ Z (0,1) <czyli taki gdzie

import numpy as np

import matplotlib.pyplot as plt

x\_n = 30 ## N (duże N) ze slajdu

n = 10000 ## n - liczba liczb do wygenerowania

Y = []

for k in range(n):

X = np.random.uniform(-1,1 , size=x\_n)

y = np.mean(X-np.mean(X) )/(np.std(X)/ np.sqrt(x\_n))

Y.append(y)

plt.hist(Y)

plt.show()