**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ **«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В. Г. ШУХОВА»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Системный анализ**

**Тема: Разработка датчика случайных чисел, распределенных по гауссовскому закону**

Выполнил: ст. группы ВТ-31

Подкопаев Антон Валерьевич

Проверил: проф. ПО и ВТАС

Полунин Александр Иванович

**Белгород 2020**

**Цель работы:** на примере программы показать нормальное распределение,

задав большое количество случайных величин из шести элементов в любом диапазоне, и построить его график.

**Задания для выполнения к работе**

Используя датчик равномерно распределенных случайных чисел получить гауссовскую случайную величину. Значение ее математического ожидания равно номеру N студента в журнале, величина дисперсии равна . Правильность программы проверить, как и в первой лабораторной, построением на одном графике гистограммы гауссовской величины и ее функции плотности вероятности.

**Результат работы программы**

**Изображение выглядит как снимок экрана, компьютер, монитор, рабочий стол

Автоматически созданное описание**

*Приложение*

import sys

import random

import numpy

import math

from scipy.stats import norm

import matplotlib.pyplot as pyplot

from matplotlib.backends.backend\_qt4agg import FigureCanvasQTAgg as FigureCanvas

from matplotlib.figure import Figure

from PyQt5.QtWidgets import \* #QApplication, QMainWindow, QMenu, QVBoxLayout, QHBoxLayout, QFormLayout, QGridLayout, QSizePolicy, QMessageBox, QWidget, QPushButton, QLineEdit, QLabel

from PyQt5.QtGui import QIcon

n = 6

mx = 13

sigma = math.sqrt(1 / mx)

sz = 10000

col = 25

def getRandomNumbers(sz):

result = []

for i in range (0, sz):

result.append(random.random())

return result

def getGaussian(sz):

result = []

for i in range(0, sz):

v = getRandomNumbers(n)

x = sigma \* math.sqrt(2) \* (numpy.sum(v) - n / 2) + mx

result.append(x)

return result

class PlotWidget(QWidget):

def \_\_init\_\_(self, parent=None):

super(PlotWidget, self).\_\_init\_\_(parent)

self.initUI()

def initUI(self):

self.mainLayout = QVBoxLayout(self)

self.figure = Figure()

self.canvas = FigureCanvas(self.figure)

self.mainLayout.addWidget(self.canvas)

def plot(self):

self.figure.clear()

result = getGaussian(sz)

min = numpy.min(result)

max = numpy.max(result)

step = (abs(min) + max)/ 10

a = numpy.arange(min, max, step)

data = numpy.full\_like(a, 0)

x\_axis = numpy.arange(min - 1, max + 1, 0.001)

ax = self.figure.add\_subplot(111)

ax.plot(x\_axis, norm.pdf(x\_axis, mx, sigma), color = "mediumblue")

ax.hist(result, col, density=True, color = "cyan")

self.canvas.draw()

class MainWindow(QMainWindow):

def \_\_init\_\_(self):

super(MainWindow, self).\_\_init\_\_()

self.initUI()

self.connectUI()

self.plotWidget.plot()

def initUI(self):

self.centralWidget = QWidget(self)

self.plotLayout = QVBoxLayout(self.centralWidget)

self.plotWidget = PlotWidget()

self.plotLayout.addWidget(self.plotWidget)

self.buttonsLayout = QVBoxLayout(self.centralWidget)

self.generateButton = QPushButton("New set of numbers")

self.buttonsLayout.addWidget(self.generateButton)

self.plotLayout.addLayout(self.buttonsLayout)

self.setCentralWidget(self.centralWidget)

self.dataLayout = QGridLayout(self.centralWidget)

#self.muEdit = QLineEdit()

#self.sigmaEdit = QLineEdit()

#self.muLabel = QLabel("mu")

#self.sigmaLabel = QLabel("sigma")

#self.dataLayout.addWidget(self.muLabel)

#self.dataLayout.addWidget(self.muEdit)

#self.dataLayout.addWidget(self.sigmaLabel)

#self.dataLayout.addWidget(self.sigmaEdit)

self.plotLayout.addLayout(self.dataLayout)

def connectUI(self):

self.generateButton.clicked.connect(self.plotWidget.plot)

#def setParams(newMu, newSigma):

# mu = newMu

# sigma = newSigma

if \_\_name\_\_ == "\_\_main\_\_":

app = QApplication([])

window = MainWindow()

window.resize(800, 600)

window.setWindowTitle("1")

window.show()

sys.exit(app.exec\_())