**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №1**

**по дисциплине: Статистические методы обработки**

**экспериментальных данных**

**Тема: «Формирование и первичная обработка выборки. Ранжированный и интервальный ряды»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 7381 |  | Павлов А.П. |
| Преподаватель |  | Середа В.И. |

Санкт-Петербург

2021

**Цель работы.**

Ознакомление с основными правилами формирования выборки и подготовки выборочных данных к статистическому анализу.

**Постановка задачи.**

Осуществить формирование репрезентативной выборки заданного объема из имеющейся генеральной совокупности экспериментальных данных. Осуществить последовательное преобразование полученной выборки в ранжированный, вариационный и интервальный ряды. Применительно к интервальному ряду построить и отобразить графически полигон, гистограмму и эмпирическую функцию распределения для абсолютных и относительных частот. Полученные результаты содержательно проинтерпретировать.

**Выполнение работы.**

Для выполнения лабораторной работы выбран язык программирования Python. Код программы представлен в приложении А.

Произведена выборка из представленной генеральной совокупности экспериментальных данных размером 107 элементов. Результат представлен на рис. 1. Выборка сформирована из генеральной совокупности путем случайного выбора необходимого количества элементов.

Text, letter

Description automatically generated

Рисунок 1 ­– Сформированная выборка

Ранжированный ряд представлен на рис.2.

Text, letter

Description automatically generated

Рис. 2. Ранжированный ряд

Вариационный ряд представлен на рис.3.

Text

Description automatically generated with medium confidence

Рис. 3. Вариационный ряд

Для построения интервальный ряда требуется рассчитать количество различных групп. Требуемые расчёты представлены на рис. 4. Построенный интервальный ряд представлен на рис. 5.

A screenshot of a computer

Description automatically generated with low confidence

Рис. 4. Расчёты для интервального ряда

Table

Description automatically generated

Рис. 5. Интервальный ряд с частотами

На рис. 6 представлены построенные полигон и гистограмма абсолютных частот.

Chart, histogram

Description automatically generated

Рис. 6. Гистограмма и полигон абсолютных частот

На рис. 7 представлены построенные полигон и гистограмма относительных частот.

Chart, line chart, histogram

Description automatically generated

Рис. 7. Гистограмма и полигон относительных частот

Для интервального ряда абсолютных частот построена эмпирическая функция распределения. Построенная эмпирическая функция распределения представлена на рис. 8.

Chart, box and whisker chart

Description automatically generated

Рис. 8. Эмпирическая функция распределения для интервального ряда абсолютных частот

Для интервального ряда относительных частот построена эмпирическая функция распределения. Построенная эмпирическая функция распределения представлена на рис. 9.

Chart, box and whisker chart

Description automatically generated

Рис. 9. Эмпирическая функция распределения для интервального ряда относительных частот

**Выводы.**

В ходе выполнения лабораторной работы был повторён и применён на практике материал с правилами формирования случайной выборки и подготовкой выборочных данных к статистическому анализу. В практической части были сделаны следующие этапы работы: формирование выборки, преобразование в ранжированный, вариационный и интервальный ряды, построение по интервальному ряду гистограмм, полигонов частот и эмпирической функции распределения.

Приложение А

исходный код разработанной прораммы

import csv  
import random  
import math  
from collections import Counter  
import matplotlib.pyplot as plt  
import numpy as np  
from statsmodels.distributions.empirical\_distribution import ECDF  
  
random\_seed = 79  
data\_elem\_size = 107  
  
def getDataFromFile(attribureNum):  
 with open(**'machine.data'**, newline=**''**) as csvfile:  
 spamreader = csv.reader(csvfile, delimiter=**','**)  
 data = []  
 for row in spamreader:  
 data.append(int(row[attribureNum]))  
 return data  
  
def generateSample(data):  
 random.seed(random\_seed)  
 data = random.sample(data, data\_elem\_size)  
 return data  
  
def printData(data):  
 count = 0  
 for i in data:  
 count = count + 1  
 if(count > 20):  
 count = 0  
 print(i)  
 else:  
 print(i, end =**" "**),  
 print()  
  
data = getDataFromFile(8)  
data = generateSample(data)  
print(**"Сформированная выборка:"**)  
printData(data)  
print(**"Ранжированный ряд:"**)  
printData(sorted(data))  
  
countedData = Counter(sorted(data))  
count = 0  
print(**'Варианционный ряд:'**)  
for i in sorted(set(data)):  
 count = count + 1  
 if (count > 10):  
 count = 0  
 print(**f'**{i:**3**} **==>** {countedData[i]:**1**}**'**)  
 else:  
 print(**f'**{i:**3**} **==>** {countedData[i]:**1**}**'**, end=**" "**)  
print()  
  
intervalNum = int(1 + 3.322 \* math.log10(data\_elem\_size))  
print(**'Количество интервалов определенное по формуле Стерджесса = '**,intervalNum)  
print(**'Минимальное значение в выборке = '**, min(data))  
print(**'Максимальное значение в выборке = '**, max(data))  
dataRange = max(data) - min(data)  
intervalSize = int(dataRange / intervalNum)  
print(**'Размах выборки = '**, max(data) - min(data))  
  
isInBucket = lambda x: min(int((abs(x) - min(data)) / dataRange \* intervalNum), intervalNum-1)  
borders = [(min(data) + dataRange/intervalNum\*i, min(data) + dataRange/intervalNum\*(i+1)) for i in range(intervalNum)]  
buckets = [[] for i in range(intervalNum)]  
for value in data:  
 buckets[isInBucket(value)].append(value)  
  
print(**'Интервальный ряд'**)  
print(**' Интервал абс. част. относ. част'**)  
for i in range(0, intervalNum):  
 print(**f'**{(min(data) + intervalSize\*i):**3**} **-** {min(data) + intervalSize\*(i+1):**3**}**'  
 f'**{len(buckets[i]):**10**} {len(buckets[i])/len(data):**25**}**'**)  
  
fig, ax = plt.subplots()  
ax.hist(data, bins=intervalNum, density=False, edgecolor=**'black'**, facecolor=**'white'**)  
center\_of\_borders = [(border[0] + border[1])/2 for border in borders]  
y = [len(bucket) for bucket in buckets]  
ax.plot(center\_of\_borders, y, **'--k'**)  
ax.set\_xlabel(**'Значение'**)  
ax.set\_ylabel(**'Абсолютная частота'**)  
ax.set\_title(**'Гистограмма и полигон абсолютных частот'**)  
fig.tight\_layout()  
plt.show()  
  
fig, ax = plt.subplots()  
ax.hist(data, intervalNum, weights=np.ones(len(data)) / len(data), density=False, edgecolor=**'black'**, facecolor=**'white'**)  
center\_of\_borders = [(border[0] + border[1])/2 for border in borders]  
y = [len(bucket)/data\_elem\_size for bucket in buckets]  
ax.plot(center\_of\_borders, y, **'--k'**)  
ax.set\_xlabel(**'Значение'**)  
ax.set\_ylabel(**'Относительная частота'**)  
ax.set\_title(**'Гистограмма и полигон относительных частот'**)  
fig.tight\_layout()  
plt.show()  
  
maxBucketLen = 0  
for bucket in buckets:  
 if(maxBucketLen < len(bucket)/len(data)):  
 maxBucketLen = len(bucket)/len(data)  
  
absValues = []  
sum = 0  
print(sum)  
for bucket in buckets:  
 sum = sum + len(bucket)/len(data)  
 absValues.append(sum)  
  
ecdf = ECDF(absValues)  
fig, ax = plt.subplots()  
ax.set\_xlabel(**'x'**)  
ax.set\_ylabel(**'F(x)'**)  
ax.set\_title(**'Эмпирическая функция распределения относительных частот'**)  
ax.axis(xmin=maxBucketLen, xmax=max(ecdf.x))  
ax.axis(ymin=-0.05, ymax=1.05)  
  
for i in range(len(ecdf.x)-1):  
 xs = [0, ecdf.x[i]]  
 ys = [ecdf.y[i]] \* 2  
 ax.plot(xs, ys, **'r:'**, alpha=0.2)  
 ax.plot(ecdf.x[i], ecdf.y[i], **"k."**) *# точки* xs = [ecdf.x[i], ecdf.x[i+1]]  
 ys = [ecdf.y[i]] \* 2  
 ax.plot(xs, ys, **'k-'**)  
ax.plot(ecdf.x[-1], ecdf.y[-1], **"k."**)  
plt.show()