### Отчет по дисциплине “Математическая статистика”

##### Описательная статистика

Выполнил: Султанов Нурсиль

Ильнарович 09-012

Задание 2. Раздел 2.

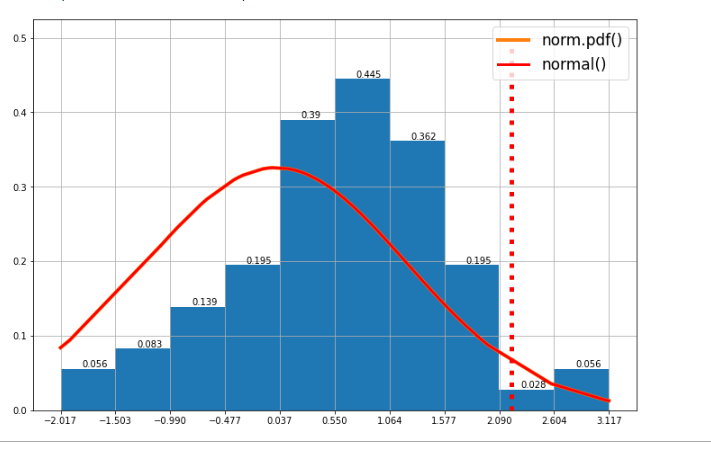
Цель работы:В этом задании требуется произвести проверку гипотезы для предоставленной выборки о типе её распределения. У задания есть несколько вариантов: проверка гипотезы согласия критерием хиквадрат и критерием Колмогорова. К каждому из заданий прилагаются дополнительные параметры (гипотеза H0 и уровень значимости α).

Проверка согласия хи-квадрат.

1.1. Характеристики выборки

|  |  |
| --- | --- |
| **Характеристика** | **Значение** |
| Объём выборки | 70 |
| Минимум | -2.016 |
| Максимум | 3.11 |
| Размах | 5.134 |
| Альфа | 0.1 |
| Стандартное отклонение | 1.22 |
| P-значение | 0.0646 |
| С (критическое значение) | 2.204 |
| Тестовая статистика | 1.821 |

1.2 Посторение гистограммы



Код программы

import csv

with open("r2z2.csv", encoding='utf-8') as r\_file:

file\_reader = csv.reader(r\_file, delimiter = ",")

x = []

n = 0

for row in file\_reader:

if n > 0:

x.append(float(row[0]))

n+=1

n-=1

x.sort()

print("Объем выборки: ", n)

print("Минимум: ", x[0])

print("Максимум: ", x[n-1])

print("Размах: ", round(x[n-1] - x[0],3))

import math

def normal(x):

M=0

sigma = 1.5

return (1./math.sqrt(sigma\*2\*math.pi))\*math.exp(-(x\*x)/(2\*sigma))

from scipy import integrate

def normal\_cdf(a,b):

v, err = integrate.quad(normal, a, b)

return v

import numpy as np

import matplotlib.pylab as plt

import statsmodels as sm

from scipy.stats import norm

fig=plt.figure(figsize=(12,8))

ax=fig.add\_subplot()

ax.xaxis.set\_major\_locator(plt.IndexLocator(base=0.7334, offset=0))

n, bins,pitches = ax.hist(x,bins=int(np.round(len(x)/10)), density=True)

for i in pitches:

height = round(i.get\_height(), 3)

ax.annotate(height, (i.get\_x() + 0.2, height+0.002))

ax.grid()

y = []

for i in range(len(x)):

y.append(normal(x[i]))

plt.plot(x, norm.pdf(x,loc = 0,scale = math.sqrt(1.5)), linewidth= 4, label=r'norm.pdf()')

plt.plot(x, y, linewidth =3, color = "r", label=r'normal()')

plt.legend(loc='best', fontsize=17)

plt.show()

phi = []

p = []

# Тестовые данные

for i in n:

tmp = float(i)

phi.append(tmp)

# Предполагаемые данные

for i in range(7):

tmp\_p = normal\_cdf(bins[i],bins[i+1])

p.append(tmp\_p)

# Статистика

test\_stat = 0

for i in range(7):

test\_stat = test\_stat + ((phi[i] - len(x)\*p[i])\*\*2)/(len(x)\*p[i])

print(test\_stat)