Atividade 03\_31/08/23 – Estatística Aplicada Profa: Malu 3ºADS Noturno

Aluno: Matheus da Silva Gomes

Ex.01, 02

Texto, Carta

Descrição gerada automaticamente

Ex.03

Texto

Descrição gerada automaticamente

import statistics  
# import numpy as np  
# import sys  
# import os  
import exercicio\_06 as ex  
  
# sys.stdout.reconfigure(encoding='utf-8')  
  
  
def verif\_quartil(lista, n\_quartil):  
 if n\_quartil == 1:  
 position = int(len(lista) \* 0.25)  
 return lista[position]  
 elif n\_quartil == 2:  
 position = int(len(lista) \* 0.5)  
 return lista[position]  
 elif n\_quartil == 3:  
 position = int(len(lista) \* 0.75)  
 return lista[position]  
 elif n\_quartil == 4:  
 return lista[-1]  
 else:  
 print("Valor quartil inválido!")  
 return  
  
  
def percentil(lista, n\_percent):  
 position = int(len(lista) \* n\_percent) - 1  
 return lista[position]  
  
  
numeros = "120 250 250 251 251 785 458 124 245 125 \  
145 254 654 563 562 456 125 145 258 145 \  
145 895 145 785 458 800 900 400 500 365 \  
201 365 654 568 251 365 145 896 145 256 \  
254 365 154 215 458 254 258 145 369 547 \  
589 698 698 789 544 456 356 548 569 598 \  
896 785 456 256 123 154 452 258 145 695 \  
"  
lista\_tratada = list(map(int, numeros.split()))  
media = statistics.mean(lista\_tratada)  
print(f"Média aritmética: {media:.2f}")  
  
moda = statistics.mode(lista\_tratada)  
print(f"Moda: {moda}")  
  
lista\_ordenada = sorted(lista\_tratada)  
# print(lista\_ordenada)  
mediana = statistics.median(lista\_ordenada)  
  
print(f"Mediana: {int(mediana)}\t\tPosição 35 de 70: {lista\_ordenada[35]} \t\tPosição 36 de 70: {lista\_ordenada[36]}")  
  
print(f"Quartil 0{3}: {verif\_quartil(lista\_ordenada, n\_quartil=3)}")  
  
print(f"Percentil ..P{23}: {percentil(lista\_ordenada, n\_percent=0.23)}")  
  
at = ex.amplitude(lista\_ordenada)  
k = ex.qtde\_linhas(len(lista\_ordenada))  
intervalo = ex.intervalo(at, k)  
cont00 = min(lista\_ordenada)  
print("\nFrequência")  
print(f'n\_linhas = ou {k[0]} ou {k[1]} ou {k[2]}')  
print(f'Intervalo = de {int(intervalo)} em {int(intervalo)}\n')  
ex.cont\_ocorrencias(lista\_tratada, intervalo, cont00)

Ex.04

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

import math  
import statistics as stats  
import sys  
  
# sys.stdout.reconfigure(encoding='utf-8')  
  
  
def cond\_lista(lista):  
 return list(sorted(map(int, lista.split())))  
  
  
def calc\_moda(lista):  
 lista = cond\_lista(lista)  
 moda = stats.mode(lista)  
 print(f"Moda: {moda}")  
  
  
def calc\_media(lista):  
 lista = cond\_lista(lista)  
 media = stats.mean(lista)  
 print(f"Média aritmética: {media:.2f}")  
  
  
def calc\_mediana(lista):  
 lista = cond\_lista(lista)  
 print(f"Mediana: {stats.median(lista)}")  
  
  
def amplitude(lista): # At  
 xmax = max([x for x in lista])  
 xmin = min([x for x in lista])  
 return xmax - xmin  
  
  
def qtde\_linhas(n\_elementos): # k  
 opt0 = int(math.sqrt(n\_elementos))  
 opt1 = opt0 + 1  
 opt2 = opt0 - 1  
 return [opt2, opt0, opt1] # ex.: opt0 = 7 então 6, 7, 8  
  
  
def intervalo(amplitude, qtde\_linhas):  
 n\_real = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
 for i in range(len(n\_real)):  
 for j in range(len(qtde\_linhas)):  
 if (amplitude + n\_real[i]) % qtde\_linhas[j] == 0:  
 ic = (amplitude + n\_real[i]) / qtde\_linhas[j]  
 return ic  
 else:  
 continue  
 return "erro"  
  
  
def hit\_count(lista, intervalo, cont):  
 while cont < max(lista):  
 n\_ocorrencia = 0  
 for i in range(len(lista)):  
 if lista[i] >= cont and lista[i] < cont + intervalo:  
 n\_ocorrencia += 1  
 print("De ", int(cont), " até", int(cont + intervalo), " = ", n\_ocorrencia,  
 '\t ', int(round((n\_ocorrencia/len(lista)) \* 100)), "%")  
 cont += intervalo  
  
  
def verif\_quartil(lista, n\_quartil):  
 lista = cond\_lista(lista)  
 if n\_quartil == 1:  
 position = int(len(lista) \* 0.25)  
 print(f"Quartil {n\_quartil}: {lista[position]}")  
 elif n\_quartil == 2:  
 position = int(len(lista) \* 0.5)  
 print(f"Quartil {n\_quartil}: {lista[position]}")  
 elif n\_quartil == 3:  
 position = int(len(lista) \* 0.75)  
 print(f"Quartil {n\_quartil}: {lista[position]}")  
 elif n\_quartil == 4:  
 print(f"Quartil {n\_quartil}: {lista[-1]}")  
 else:  
 print("Valor quartil inválido!")  
 return  
  
  
def quintil(lista, n\_quintil):  
 lista = cond\_lista(lista)  
 if n\_quintil == 1:  
 position = int(len(lista) \* 0.2)  
 print(f"Quintil {n\_quintil}: {lista[position]}")  
 elif n\_quintil == 2:  
 position = int(len(lista) \* 0.4)  
 print(f"Quintil {n\_quintil}: {lista[position]}")  
 elif n\_quintil == 3:  
 position = int(len(lista) \* 0.6)  
 print(f"Quintil {n\_quintil}: {lista[position]}")  
 elif n\_quintil == 4:  
 position = int(len(lista) \* 0.8)  
 print(f"Quintil {n\_quintil}: {lista[position]}")  
 elif n\_quintil == 5:  
 print(f"Quartil {n\_quintil}: {lista[-1]}")  
 else:  
 print("Valor quartil inválido!")  
 return  
  
  
def decil(lista, n\_decil):  
 lista = cond\_lista(lista)  
 position = (int(len(lista) \* (n\_decil \* 0.1)) - 1)  
 print(f"Decil: {lista[position]}")  
  
  
def percentil(lista, n\_percent):  
 lista = cond\_lista(lista)  
 position = int(len(lista) \* n\_percent) - 1  
 print(lista[position])  
  
  
def calc\_frequency(lista):  
 lista\_tratada = cond\_lista(lista)  
 at = amplitude(lista\_tratada)  
 k = qtde\_linhas(len(lista\_tratada))  
 interval = intervalo(at, k)  
 cont00 = min(lista\_tratada)  
 print(f'Frequência:\nn\_linhas = ou {k[0]} ou {k[1]} ou {k[2]}')  
 print(f'Intervalo = de {int(interval)} em {int(interval)}')  
 hit\_count(lista\_tratada, interval, cont00)  
  
  
# if \_\_name\_\_ == "\_\_main\_\_": ///oq que essa bagaça faz mesmo??