prova_de_estatística_1

August 21, 2021

1 Problema

1- Escolha aleatoriamente 50 números e utilizando a fórmula [k=1+(3,32*logN)+ln N] para o número de classes construa uma distribuição de frequência. OBS. COMO OS NÚMEROS SÃO ALEATÓRIOS NÃO PODERÁ TER PROVAS IGUAIS, CASO ISSO OCORRA AS DUAS SERÃO ZERADAS (2,0 pontos)

2- Calcule a média aritmética da distribuição que você criou na questão 1; (2,0 pontos)

3- Calcule a Mediana da distribuição criada acima (2,0 pontos)

4- Calcule o 1º e 3º quartil da distribuição acima (2,0 pontos)

5- Com os números que você escolheu construa um Boxplot. (2,0 pontos)

2 Import

[178]:
```
import scipy.stats                #importando a biblioteca scipy
import math                       #importando a biblioteca math
import random                     #importando a biblioteca random
import numpy  as np               #importanto a biblioteca numpy
import pandas as pd               #importando a biblioteca pandas
import matplotlib.pyplot as plt #importando a biblioteca matplotlib
```

2.1 1.1 - Escolhendo 50 números aleatórios entre 1 e 500

[179]:
```
amostra = np.random.randint(1,500,(1,50))
```

[180]:
```
amostra = amostra[0]
```

[181]:
```
amostra = amostra.tolist()
```

[228]:
```
amostra = sorted(amostra) # Ordernando valores sorteados
```

[229]:
```
amostra
```

[229]:
```
[27,
 32,
 73,
```

```
89,
95,
110,
121,
132,
145,
147,
149,
151,
153,
155,
160,
176,
178,
185,
190,
218,
219,
219,
223,
243,
257,
260,
261,
264,
284,
288,
288,
289,
290,
292,
326,
332,
336,
337,
346,
352,
370,
373,
381,
388,
418,
423,
423,
429,
477,
477]
```

2.2 1.2 - Distribuição de frenquência

[183]:
```
df = pd.DataFrame(np.array(amostra).reshape(50,1), columns = list("A")) #␣
↪Transformando lista em pandas dataframe
```

[184]:
```
df.head()                    # visualizando os 5 primeiros dados da tabela
```

[184]:
```
    A
0  27
1  32
2  73
3  89
4  95
```

[185]:
```
df = df.sort_values(by = "A", ascending=True) # organizando os elementos da␣
↪coluna única "A"
```

[186]:
```
df['fi'] = '-'
```

2.3 1.3 - Número de classes

[187]:
```
# fórmula k = 1+(3,32*logN)+ln N


k = 1+(3.32*math.log10(50)) + np.log(50)
```

[188]:
```
k
```

[188]:
```
10.552603419823727
```

[189]:
```
k = round(k) #arredondando
```

[190]:
```
k
```

[190]:
```
11
```

2.4 1.4 - Distribuição de frequências

[191]:
```
# Amplitude dos dados = Valor maior dos registros - menor valor
at = df['A'].max() - df['A'].min()
```

[192]:
```
at
```

[192]:
```
450
```

[193]:
```
# Amplitude dos intervalos

ai = at/k
```

[194]:
```
ai = math.ceil(ai) # Arredondamento para cima
ai
```

[194]:
```
41
```

[195]:
```
amplitudeintervalostxt  = []
# Menor valor da série
menor = round(df['A'].min(),1)

# Menor valor somado a amplitude
menor_amp = round(menor+ai,1)

valor = menor

while valor < df['A'].max():
    amplitudeintervalostxt.append('{} - {}'.
 ↪format(round(valor,1),round(valor+ai,1)))
    valor = valor + ai
```

[196]:
```
amplitudeintervalostxt
```

[196]:
```
['27 - 68',
 '68 - 109',
 '109 - 150',
 '150 - 191',
 '191 - 232',
 '232 - 273',
 '273 - 314',
 '314 - 355',
 '355 - 396',
 '396 - 437',
 '437 - 478']
```

[197]:
```
amplitudeintervaloslist = []
intervalo       = []

primeirotermo = round(df['A'].min(),1)
segundotermo  = primeirotermo + ai
intervalo.append(primeirotermo)
intervalo.append(segundotermo)
amplitudeintervaloslist.append(intervalo)

for i in range(len(amplitudeintervalostxt) - 1):
    amplitudeintervaloslist.append([])
    amplitudeintervaloslist[i+1].append(amplitudeintervaloslist[i][-1])
    amplitudeintervaloslist[i+1].append(amplitudeintervaloslist[i][-1]+ai)
```

[198]:
```
amplitudeintervaloslist
```

[198]:
```
[[27, 68],
 [68, 109],
 [109, 150],
 [150, 191],
 [191, 232],
 [232, 273],
 [273, 314],
 [314, 355],
 [355, 396],
 [396, 437],
 [437, 478]]
```

[199]:
```
amplitudeintervalostxt = np.array(amplitudeintervalostxt)
```

[200]:
```
df2 = pd.DataFrame(amplitudeintervalostxt, columns =['amplitude_intervalos'])
```

[201]:
```
df2
```

[201]:

	amplitude_intervalos
0	27 - 68
1	68 - 109
2	109 - 150
3	150 - 191
4	191 - 232
5	232 - 273
6	273 - 314
7	314 - 355
8	355 - 396
9	396 - 437
10	437 - 478

[202]:
```
provasnointervalolist = [[]]
copiaamostra = amostra[:]

end = False

for j in range(len(amplitudeintervaloslist)):
    while copiaamostra[0] < amplitudeintervaloslist[j][-1] and end == False:
        provasnointervalolist[j].append(copiaamostra[0])
        if len(copiaamostra) > 1:
            del copiaamostra[0]
        elif len(copiaamostra) == 1:
            end = True
    else:
        provasnointervalolist.append([])
```

```
        if end == True:
            del provasnointervalolist[-1]
```

[203]:
```
provasnointervalolist
```

[203]:
```
[[27, 32],
 [73, 89, 95],
 [110, 121, 132, 145, 147, 149],
 [151, 153, 155, 160, 176, 178, 185, 190],
 [218, 219, 219, 223],
 [243, 257, 260, 261, 264],
 [284, 288, 288, 289, 290, 292],
 [326, 332, 336, 337, 346, 352],
 [370, 373, 381, 388],
 [418, 423, 423, 429],
 [477, 477]]
```

[204]:
```
df2['provas_intervalo'] = provasnointervalolist
```

[205]:
```
df2
```

[205]:

	amplitude_intervalos	provas_intervalo
0	27 - 68	[27, 32]
1	68 - 109	[73, 89, 95]
2	109 - 150	[110, 121, 132, 145, 147, 149]
3	150 - 191	[151, 153, 155, 160, 176, 178, 185, 190]
4	191 - 232	[218, 219, 219, 223]
5	232 - 273	[243, 257, 260, 261, 264]
6	273 - 314	[284, 288, 288, 289, 290, 292]
7	314 - 355	[326, 332, 336, 337, 346, 352]
8	355 - 396	[370, 373, 381, 388]
9	396 - 437	[418, 423, 423, 429]
10	437 - 478	[477, 477]

[206]:
```
df2['frequencia_absoluta'] = df2['provas_intervalo'].map(lambda x : len(x))
```

[207]:
```
df2
```

[207]:

	amplitude_intervalos	provas_intervalo
0	27 - 68	[27, 32]
1	68 - 109	[73, 89, 95]
2	109 - 150	[110, 121, 132, 145, 147, 149]
3	150 - 191	[151, 153, 155, 160, 176, 178, 185, 190]
4	191 - 232	[218, 219, 219, 223]
5	232 - 273	[243, 257, 260, 261, 264]
6	273 - 314	[284, 288, 288, 289, 290, 292]
7	314 - 355	[326, 332, 336, 337, 346, 352]
8	355 - 396	[370, 373, 381, 388]

```
9             396 - 437                       [418, 423, 423, 429]
10            437 - 478                                 [477, 477]

    frequencia_absoluta
0                     2
1                     3
2                     6
3                     8
4                     4
5                     5
6                     6
7                     6
8                     4
9                     4
10                    2
```

[208]:
```
df2 = df2.assign(frequencia_acumulada=df2.frequencia_absoluta.cumsum())
```

[209]:
```
df2['frequencia_relativa'] = df2['frequencia_absoluta'].map(lambda x: (x/
↪len(amostra))*100)
```

[210]:
```
df2
```

[210]:
```
   amplitude_intervalos                          provas_intervalo  \
0                27 - 68                                  [27, 32]
1                68 - 109                               [73, 89, 95]
2               109 - 150             [110, 121, 132, 145, 147, 149]
3               150 - 191  [151, 153, 155, 160, 176, 178, 185, 190]
4               191 - 232                       [218, 219, 219, 223]
5               232 - 273                  [243, 257, 260, 261, 264]
6               273 - 314             [284, 288, 288, 289, 290, 292]
7               314 - 355             [326, 332, 336, 337, 346, 352]
8               355 - 396                       [370, 373, 381, 388]
9               396 - 437                       [418, 423, 423, 429]
10              437 - 478                                 [477, 477]

    frequencia_absoluta  frequencia_acumulada  frequencia_relativa
0                     2                     2                 4.0
1                     3                     5                 6.0
2                     6                    11                12.0
3                     8                    19                16.0
4                     4                    23                 8.0
5                     5                    28                10.0
6                     6                    34                12.0
7                     6                    40                12.0
8                     4                    44                 8.0
9                     4                    48                 8.0
```

```
10                   2                    50                 4.0
```

[211]:
```
df2 = df2.assign(frequencia_relativa_acumulada=df2.frequencia_relativa.cumsum())
```

[212]:
```
df2
```

[212]:
```
   amplitude_intervalos                          provas_intervalo  \
0               27 - 68                                  [27, 32]
1               68 - 109                              [73, 89, 95]
2              109 - 150              [110, 121, 132, 145, 147, 149]
3              150 - 191  [151, 153, 155, 160, 176, 178, 185, 190]
4              191 - 232                        [218, 219, 219, 223]
5              232 - 273                  [243, 257, 260, 261, 264]
6              273 - 314              [284, 288, 288, 289, 290, 292]
7              314 - 355              [326, 332, 336, 337, 346, 352]
8              355 - 396                        [370, 373, 381, 388]
9              396 - 437                        [418, 423, 423, 429]
10             437 - 478                                  [477, 477]

    frequencia_absoluta  frequencia_acumulada  frequencia_relativa  \
0                     2                     2                  4.0
1                     3                     5                  6.0
2                     6                    11                 12.0
3                     8                    19                 16.0
4                     4                    23                  8.0
5                     5                    28                 10.0
6                     6                    34                 12.0
7                     6                    40                 12.0
8                     4                    44                  8.0
9                     4                    48                  8.0
10                    2                    50                  4.0

    frequencia_relativa_acumulada
0                             4.0
1                            10.0
2                            22.0
3                            38.0
4                            46.0
5                            56.0
6                            68.0
7                            80.0
8                            88.0
9                            96.0
10                          100.0
```

3 2 - Calculando a média aritmética

[213]:
```
media = np.mean(amostra)
```

[214]:
```
media
```

[214]: 251.02

4 3 - Achando a mediana

[215]:
```
mediana = np.median(amostra)
```

[216]:
```
mediana
```

[216]: 258.5

5 4 - Calculando o primeiro e o terceiro quartil

[218]:
```
q1 = np.percentile(df.A, 25)
```

[219]:
```
q1 # primeiro quartil
```

[219]: 153.5

[220]:
```
median = np.percentile(df.A, 50)
```

[221]:
```
median
```

[221]: 258.5

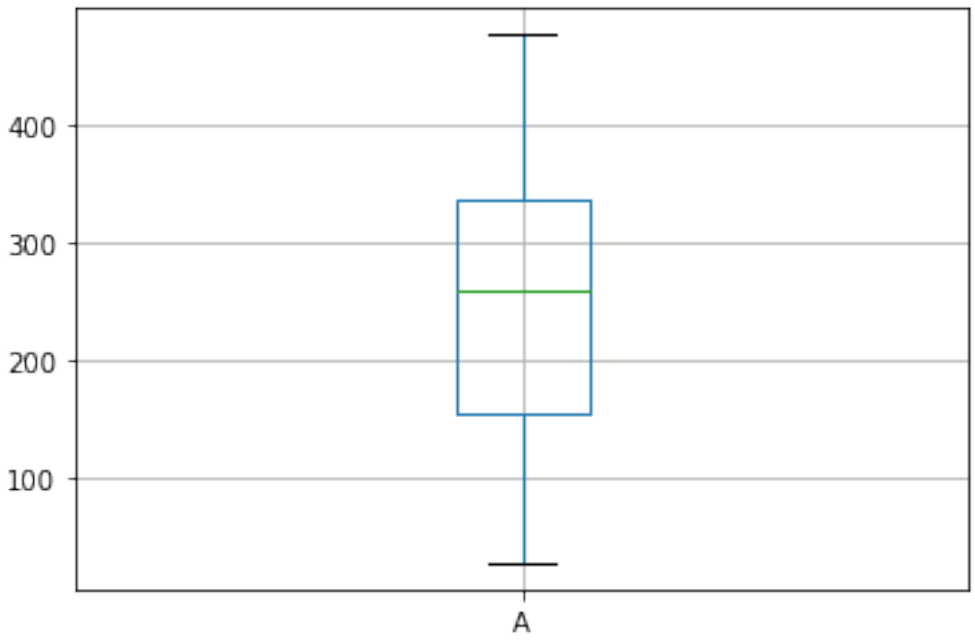
[222]:
```
q2 = np.percentile(df.A, 75)
```

[223]:
```
q2 # terceiro quartil
```

[223]: 336.75

6 5 - Construindo boxplot com valores

[227]:
```
boxplot = df.boxplot()
```


400
300
200
100
A

[ ]: