

Prova __ P1

May 12, 2021

Universidade Federal de Mato Grosso do Sul

Campus de Campo Grande Estatística – Prof. Cássio Pinho dos Reis

Prova P1

Turma: Engenharia de Software RGA: 2021.1906.069-7 Aluno: Maycon Felipe da Silva Mota

```
[31]: import itertools as it
import pandas as pd
import numpy as np
import math
import statistics as stats
from scipy.stats import poisson
from scipy.stats import norm
from scipy.stats import binom
```

0.1 1ª Questão – Em uma escola de diversos idiomas de uma grande cidade que Fulaninho estuda, tem mil alunos matriculados regularmente. Desses, 600 são homens. Como política dessa escola, o aluno pode se matricular em somente um idioma, ou seja, ele não pode se matricular em 2 ou mais idiomas. Sabe-se ainda que 300 optaram por inglês e outros 500 por espanhol. Daqueles que optaram por inglês, 100 são homens e 200 mulheres optaram por espanhol. Um aluno é escolhido ao acaso para ganhar um prêmio. Calcule a probabilidade do vencedor desse prêmio:

```
[32]: def calcularProb(evento, espacoAmostrai):
    calculo = evento/espacoAmostrai
    return calculo
```

0.1.1 a) Ser homem e ter optado por inglês. (0,5 ponto)

```
[33]: # Valores
# Homem = 100 (Inglês), 300 (Espanhol), 200 (Outros), Total = 600
# Mulheres = 200 (Inglês), 100 (Espanhol), 0 (Outros), Total = 400
# Valor Total = 1000

homem_ingles = 100
```

```

valor_total = 1000
prob = calcularProb(100, 1000)
print("A probabilidade é de {:.3%}".format(prob))

```

A probabilidade é de 10.000%

0.1.2 b) Ser mulher ou ter optado por espanhol. (0,5 ponto)

```

[34]: valor_mulher = 400
      valor_espanhol = 500
      valor_mulher_espanhol = 200
      valor_espanhol = valor_espanhol - valor_mulher_espanhol
      valor_evento = valor_mulher + valor_espanhol # 700/1000
      valor_total = 1000
      prob = calcularProb(valor_evento, 1000)
      print("A probabilidade é de {:.3%}".format(prob))

```

A probabilidade é de 70.000%

0.1.3 c) Ser homem sabendo que o aluno optou por espanhol. (0,5 ponto)

```

[35]: homem_espanhol = 300
      valor_total = 1000
      prob = calcularProb(300, 1000)
      print("A probabilidade é de {:.3%}".format(prob))

```

A probabilidade é de 30.000%

0.2 2ª Questão – Luizinho possui 26 anos e faz faculdade de Estatística em uma Universidade. Nessa universidade, a média de idade dos alunos do curso de estatística é de 24 anos, possuindo também um desvio padrão de 5 anos. Calcule a probabilidade de um aluno ser escolhido ao acaso e ele ter:

```

[36]: def desvNormPadrao(lim1, lim2, desv):
      z = (lim1-lim2)/desv
      return z

```

0.2.1 a) Entre 25 e 28 anos. (1,0 ponto)

```

[37]: z1 = desvNormPadrao(28, 24, 5) # 0.2881
      z2 = desvNormPadrao(25, 24, 5) # 0.0793
      z3 = 0.2881-0.0793
      print(z1, z2, z3)
      print("A probabilidade é de {:.3%}".format(z3))

```

0.8 0.2 0.20880000000000004

A probabilidade é de 20.880%

0.2.2 b) Menos de 22 anos. (1,0 ponto)

```
[38]: z1 = desvNormPadrao(22, 24, 5) # 0.1554
      z3 = 0.5 - 0.1554

      print(z1, z2, z3)
      print("A probabilidade é de {:.3%}".format(z3))
```

-0.4 0.2 0.3446

A probabilidade é de 34.460%

0.3 3ª Questão – Joãozinho é o “rei” do par ou ímpar. Ele desenvolveu secretamente uma tática que o faz ganhar numa disputa de par ou ímpar, o dobro de vezes em que ele perde. Pedrinho indignado com essa situação tenta sempre descobrir que tática é essa, mas nunca tem sucesso. Numa última tentativa desesperadora, Pedrinho joga 7 vezes par ou ímpar com Joãozinho. Calcule a probabilidade de Pedrinho:

Iremos utilizar a função `binom.pmf`, que precisa dos seguintes parametros

Número de sucesso desejados (X):

Número de realizações de experimentos (P):

Probabilidade de sucesso:

0.3.1 a) Perder 5 vezes. (1,0 ponto)

```
[53]: # X = perder três vezes
      # P = qtd de jogadas
      # Y = taxa de acertos

      calculo = binom.pmf(5, 7, 0.66)
      print("A probabilidade é de: {:.3%}".format(calculo))
```

A probabilidade é de: 30.402%

[]:

0.3.2 b) Perder mais que 2 vezes. (1,0 ponto)

```
[55]: # X = perder três vezes
      # P = qtd de jogadas
      # Y = taxa de acertos

      p1 = binom.pmf(0, 7, 0.66) + binom.pmf(1, 7, 0.66) + binom.pmf(2, 7, 0.66)
      p2 = 1 - p1
```

```
print("A probabilidade é de: {:.3%}".format(p2))
```

A probabilidade é de: 95.078%

0.4 4ª Questão – O uso pacífico da radioatividade tem sido cada vez mais abrangente nas diversas áreas do conhecimento humano. Os elementos radioativos, quando bem manipulados, podem ser úteis ao seres humanos. As emissões de partículas radioativas têm sido modeladas por uma equipe de pesquisadores por meio de distribuições de probabilidade, com os seus valores dependendo da fonte utilizada. Suponha que o número de partículas alfa, emitidas por minuto, seja de 5 ocorrências por minuto de acordo com esses pesquisadores. Calcule a probabilidade de haver mais de 3 emissões por minuto. (1,5 ponto)

```
[39]: # Função para cálculo da probabilidade
```

```
def prob(x, t, y):  
    e = np.exp(1)  
    valor_1 = (e**(-y))*(y**x)  
    valor_2 = valor_1/math.factorial(x)  
    return valor_2
```

```
[46]: p0 = prob(0, 1, 5) #0.006737946999085469  
p1 = prob(1, 1, 5) #0.033689734995427344  
p2 = prob(2, 1, 5) #0.08422433748856836  
p3 = prob(3, 1, 5) #0.14037389581428059  
vprob = 1-0.26502591529736175  
print("A probabilidade é de {:.3%}".format(vprob))
```

A probabilidade é de 73.497%

0.5 5ª Questão – Para adequar os produtos às preferências dos clientes, um projetista de páginas de internet pretende conhecer o perfil dos indivíduos que acessam um de seus sites. Pensando nisso, ele fez uma pesquisa e levantou algumas características dos visitantes de seu site, tais como, sexo, idade, nível de instrução e o provedor utilizado. Geralmente, esse tipo de pesquisa envolve centenas ou milhares de respondentes. O nível de instrução é classificado se a pessoa possui completado o ensino fundamental, médio ou superior. Para essa pesquisa, o projetista selecionou de maneira totalmente ao acaso 400 homens e 600 mulheres, de acordo com a proporção da cidade em que mora. Sabendo de tudo isso, responda:

0.5.1 a) Qual tipo de amostragem foi realizado? (0,5 ponto)

Estratificada

0.5.2 b) Suponha, que o projetista utilize a população da cidade (de 20 mil habitantes) como parâmetro para estimar a quantidade amostral para esta pesquisa. Se ele quer uma amostra com um erro amostral de 6%, qual seria a quantidade amostral necessária? (1,0 ponto)

```
[41]: def calcularErroAmostral(porcentagem):  
        calculo = 1/(porcentagem ** 2)  
        return calculo  
def calcularPopulacao(eAmostral, populacao):  
        calculo = (eAmostral * populacao)/(eAmostral + populacao)  
        return calculo  
  
eAmostral = calcularErroAmostral(0.06)  
valorN = calcularPopulacao(eAmostral, 20000)  
  
print("Seria necessário o valor de {:.3f}".format(valorN))
```

Seria necessário o valor de 273.973

0.6 6ª Questão – Duas equipes de medição foram analisadas por 10 dias, e as quantidades de medições diárias foram anotadas:

```
[42]: equipe_a = [24, 16, 26, 10, 15, 12, 13, 24, 18, 12]  
equipe_b = [19, 16, 16, 18, 18, 19, 20, 17, 17, 15]
```

0.6.1 a) Qual equipe é mais produtiva em média? (0,5 ponto)

```
[43]: media_equipe_a = stats.mean(equipe_a)  
media_equipe_b = stats.mean(equipe_b)  
  
print(media_equipe_a, media_equipe_b)  
if media_equipe_a > media_equipe_b:  
    print('Equipe A é mais produtiva')  
if media_equipe_b > media_equipe_a:  
    print('Equipe B é a mais produtiva')  
# Equipe B
```

17 17.5

Equipe B é a mais produtiva

0.6.2 b) Qual equipe é mais consistente ao longo dos dias, ou seja, aquela equipe que ao longo dos dias, teve uma variabilidade menor de medições? (1,0 ponto)

```
[44]: def calcularCV(desv, media):  
        calculo = desv/media  
        return calculo  
  
desvP_equipe_a = stats.stdev(equipe_a)
```

```
desvP_equipe_b = stats.stdev(equipe_b)
cv_equipe_a = calcularCV(desvP_equipe_a, media_equipe_a)
cv_equipe_b = calcularCV(desvP_equipe_b, media_equipe_b)

print(desvP_equipe_a, desvP_equipe_b, cv_equipe_a, cv_equipe_b)

if cv_equipe_a < cv_equipe_b:
    print('Equipe A é mais consistente')
if cv_equipe_b < cv_equipe_a:
    print('Equipe B é mais consistente')
# Equipe B
```

```
5.773502691896258 1.5811388300841898 0.3396178054056622 0.09035079029052513
Equipe B é mais consistente
```

```
[45]: # Equipe B
```