

Lista de Exercícios - 7 parte 4

May 11, 2021

Universidade Federal de Mato Grosso do Sul

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

7ª LISTA DE EXERCÍCIOS - parte 3

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

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

Para esse exercício, irei criar uma função para calcular a distribuição normal.

0.1 1 - Questao Supõe-se que uma moeda seja viciada de tal forma que a probabilidade em um lançamento der resultado “cara” seja 4 vezes maior que a probabilidade de sair “coroa”. Essa moeda é jogada 8 vezes. Calcule a probabilidade de ocorrer:

0.1.1 a) De ocorrer 7 caras.

```
[24]: calculo = binom.pmf(7, 8, 0.5)
calculo = calculo*4
print("{:3%}".format(calculo))
```

12.500000%

0.1.2 b) De dar pelo menos 1 caras.

```
[25]: calculo = binom.pmf(1, 8, 0.5)
calculo = calculo*4
print("{:3%}".format(calculo))
```

12.500000%

0.1.3 c) De não dar nenhuma coroa.

```
[26]: calculo = binom.pmf(0, 8, 0.5)
      calculo = calculo*4
      print("{:3%}".format(calculo))
```

1.562500%

0.2 2ª Questão – Um aluno de estatística foi fazer uma prova, porém ele não estudou sobre o assunto. Ao receber a prova, ele verificou que todas as questões são de múltipla escolha com 5 alternativas possíveis. Essa prova tinha 6 questões. E para não deixar a prova em branco, este aluno (que não estudou para a prova), marcou as opções de forma totalmente aleatória. Sabendo disso, qual a probabilidade do aluno ter:

0.2.1 a) Acertado todas as questões.

```
[27]: calculo = binom.pmf(6, 6, 0.2)
      print("{:3%}".format(calculo))
```

0.006400%

0.2.2 b) Errado todas as questões

```
[28]: calculo = binom.pmf(0, 6, 0.2)
      print("{:3%}".format(calculo))
```

26.214400%

0.2.3 c) Acertado mais da metade das questões

```
[29]: calculo = binom.pmf(4, 6, 0.2)
      print("{:3%}".format(calculo))
```

1.536000%

0.2.4 d) Acertado pelo menos 1 questão.

```
[30]: calculo = binom.pmf(1, 6, 0.2)
      print("{:3%}".format(calculo))
```

39.321600%

0.3 3ª Questão – Supondo que as consultas num banco de dados ocorrem com uma taxa média de 3 consultas por minuto. Qual a probabilidade de nos próximos 2 minutos, ocorram:

0.3.1 a) Menos que 3 consultas.

```
[31]: # 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
```

```
[32]: calculo = prob(0, 0, 3) + prob(1, 1, 3) + prob(2, 2, 3)
calculo
```

[32]: 0.4231900811268436

0.3.2 b) Mais que 5 consultas.

```
[33]: calculo = 1 - (prob(1, 1, 6) + prob(2, 2, 6) + prob(3, 3, 6) + prob(4, 4, 6) +
    ↪ prob(5, 5, 6))
calculo
```

[33]: 0.5567991108120549

0.4 4ª Questão – O número de falhas em certo tipo de placa plástica segue a distribuição Poisson, com uma taxa média de 0,05 defeito por metro quadrado. Na construção de um barco, é necessário cobrir uma superfície de 3 m x 2 m com essa placa. Calcule a probabilidade de:

0.4.1 a) Não haja falha nessa superfície.

```
[34]: calculo = prob(0, 6, 0.05)
calculo
```

[34]: 0.951229424500714

0.4.2 b) Mais que 1 falha nessa superfície.

```
[35]: calculo = prob(1, 6, 0.05)
calculo
```

[35]: 0.047561471225035706

0.5 5ª Questão – A idade de alunos da turma de estatística em uma sala de aula segue a distribuição normal de probabilidade. Sabendo que a média de idade dessa turma é de 20 anos e com um desvio padrão de 4 anos, calcule a probabilidade de um aluno ter:

0.5.1 a) Entre 18 e 23 anos.

```
[36]: def desvNormPadrao(lim1, lim2, desv):  
      z = (lim1-lim2)/desv  
      return z  
  
      # Calcular entre limites  
      z1 = desvNormPadrao(18, 20, 4) #0,1915  
      z2 = desvNormPadrao(23, 20, 4) #0,2734  
  
      z3 = 0.1915 + 0.2734  
      z3
```

[36]: 0.4649

0.5.2 b) Mais que 19 anos

```
[37]: def desvNormPadrao(lim1, lim2, desv):  
      z = (lim1-lim2)/desv  
      return z  
  
      # Calcular entre limites  
      z2 = desvNormPadrao(19, 20, 4) #0,0987  
      z3 = 0.0987 + 0.5  
      print(z3)
```

0.5987

0.5.3 c) Menos de 17 anos

```
[38]: def desvNormPadrao(lim1, lim2, desv):  
      z = (lim1-lim2)/desv  
      return z  
  
      # Calcular entre limites  
      z1 = desvNormPadrao(17, 20, 4) #0,2734  
  
      z3 = 1 - (0.2734 + 0.5)  
      print(z3)
```

0.22660000000000002

0.5.4 d) Entre 21 e 25 anos

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

```
# Calcular entre limites
z1 = desvNormPadrao(21, 20, 4) #0,0987
z2 = desvNormPadrao(25, 20, 4) #0,3944

z3 = (0.3944 - 0.0987)
print(z3)
```

0.29569999999999996

0.5.5 e) Entre 17 e 19 anos

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

# Calcular entre limites
z1 = desvNormPadrao(17, 20, 4) #0,2734
z2 = desvNormPadrao(19, 20, 4) #0,0987

print(0.2734-0.0987)
```

0.17469999999999997

0.6 6ª Questão – Suponha que o tempo de resposta na execução de um algoritmo segue uma distribuição normal de probabilidade com tempo médio de 23 segundos e desvio padrão de 4 segundos. Calcule a probabilidade ter um tempo de resposta de:

0.6.1 a) Menor que 25 segundos

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

# Calcular entre limites
z1 = desvNormPadrao(25, 23, 4) #0.1915
z3 = (0.1915 + 0.5)
print(z3)
```

0.6915

0.6.2 b) Entre 20 e 30 segundos

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

# Calcular entre limites
z1 = desvNormPadrao(20, 23, 4) #0.2734
```

```
z2 = desvNormPadrao(30, 23, 4) #0.4599
z3 = (0.2734 + 0.4599)
print(z3)
```

0.7333

```
[43]: ### c) Maior que 24 segundos
```

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

      # Calcular entre limites
      z1 = desvNormPadrao(24, 23, 4) #0.0987
      z3 = (0.5 - 0.0987)
      print(z3)
```

0.4013

```
[ ]:
```