Atividade 6
Análise de Dados
E Big Data

Aluno: Gabriel Moraes Preihsner de la Cerda

R.A.: 320137555

RESOLVA DETALHADAMENTE DISCUTINDO A SOLUÇÃO DOS PROBLEMAS.

CONSTRUIR UM SCRIPT PYTHON ONDE FOR NECESSÁRIO.

1) Os mancais são largamente utilizados nas indústrias de automóveis com a finalidade de “apoiar” as peças giratórias dos automóveis, como os eixos, por exemplo, reduzindo assim o atrito entre as peças. Existem diversos tipos de mancais, mas podemos grosseiramente divididos em duas categorias: os lisos e os rolamentos. Estes equipamentos devem ser sempre lubrificados para garantir sua eficiência e aumentar sua vida útil. Considere que o diâmetro dos orifícios dos mancais produzidos por uma indústria automobilística tem distribuição normal. Essa indústria trabalhava com um processo de desvio-padrão 0,1 mm, porém teve de realizar alterações em seu processo produtivo. Essas alterações certamente não afetaram o desvio padrão nem a distribuição das medidas (normal), contudo existe uma suspeita do engenheiro responsável de que o diâmetro médio tenha mudado. Foi feita uma amostra de 40 unidades e obtida uma medida de diâmetro médio da amostra igual a 5,426 mm. Encontre um intervalo de confiança para o diâmetro médio real, após a modificação, com nível de confiança de 90%.

Realizei essa questão particularmente com o auxílio de Python, cujo script está apresentado a seguir:

```python
from scipy.stats import norm
import math

media_amostral = 5.426
desvio_padrao = 0.1
n = 40
nivel_confianca = 0.90

Z = norm.ppf((1 + nivel_confianca) / 2)

erro_padrao = desvio_padrao / math.sqrt(n)

intervalo_inferior = media_amostral - Z * erro_padrao
intervalo_superior = media_amostral + Z * erro_padrao

print("Intervalo de confiança para o diâmetro médio real dos mancais após a modificação:")
print(f"({intervalo_inferior}, {intervalo_superior})")
```

```
Intervalo de confiança para o diâmetro médio real dos mancais após a modificação:
(5.399992580606222, 5.452007419393778)
```

Inicialmente, faz-se necessário encontrar o valor crítico Z para um nível de confiança equivalente a 90%; isso corresponde a um intervalo de 5% em cada

cauda da distribuição normal. Nesse contexto, o Python se faz muito útil, uma vez que, no script que apresentei, utilizamos uma função de distribuição acumulada inversa, que está disponível no módulo “scipy.stats”.

2) O voto nulo é utilizado como ferramenta de protesto político por muitos eleitores. Considere uma pesquisa eleitoral realizada em dois bairros, A e B. No bairro A, foram entrevistados 500 eleitores, sendo que 100 deles declararam que iriam anular seu voto. No bairro B, dos 1.000 eleitores entrevistados, 300 declararam a intenção de anular o voto.

   a) Construa um intervalo de confiança para a probabilidade de voto nulo no bairro A, considerando ά = 5%.

   b) Faço o mesmo para o bairro B.

   c) Calcule um intervalo de confiança para a diferença entre as probabilidades de intenção de voto nulo nos bairros A e B, considerando = 5%.

a) Intervalo de confiança para o bairro A:
   Proporção = número de eleitores que anularão seu voto / total de eleitores entrevistados no bairro A = 100/500 = 0.2

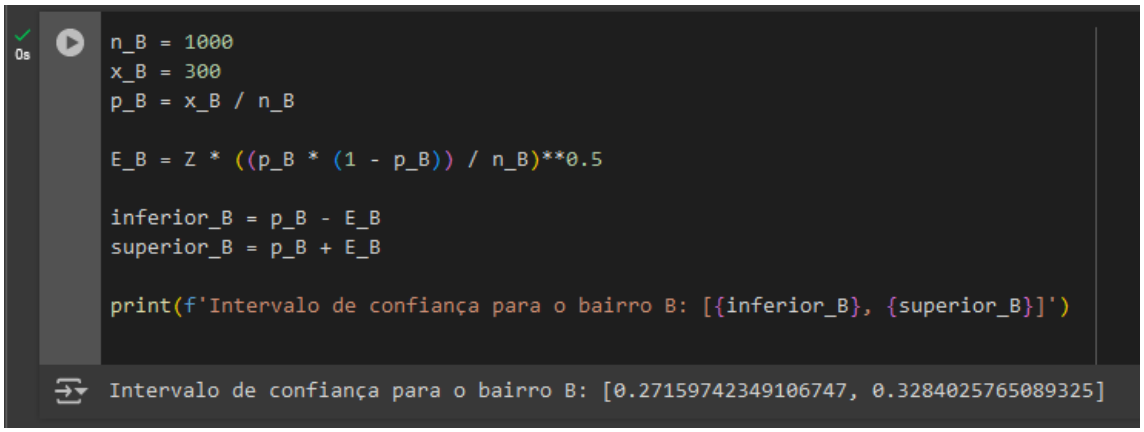
   É possível encontrar este valor com o seguinte script Python:

```python
import scipy.stats as stats

n_A = 500
x_A = 100
p_A = x_A / n_A

Z = stats.norm.ppf(0.975)
E_A = Z * ((p_A * (1 - p_A)) / n_A)**0.5

inferior_A = p_A - E_A
superior_A = p_A + E_A

print(f'Intervalo de confiança para o bairro A: [{inferior_A}, {superior_A}]')
```

```
Intervalo de confiança para o bairro A: [0.16493909837693674, 0.23506090162306328]
```

Primeiro, é importada a biblioteca “scipy.stats” sob a renomeação dentro do código de “stats”. Após isso, é definido o tamanho da amostra, e, em seguida, é calculada a proporção de eleitores que pretendem anular o voto. A seguir, são realizados alguns cálculos necessários e por fim é impresso o resultado.

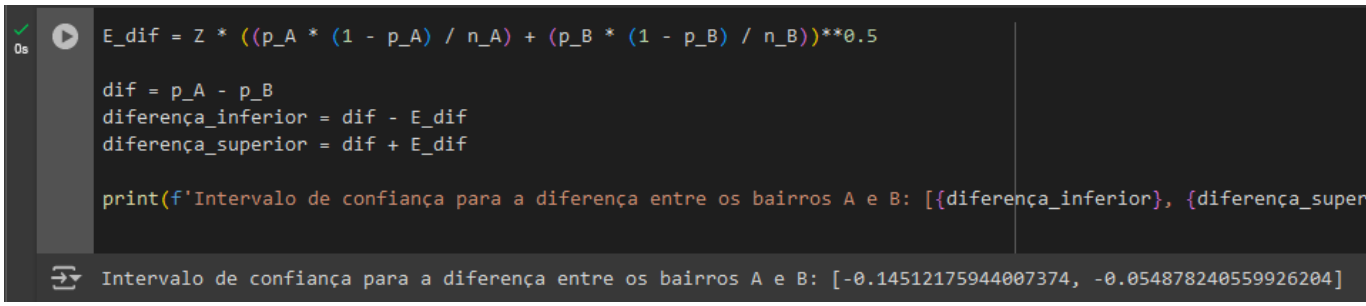
b) O cálculo para o bairro B, claramente, utiliza o mesmo script, porém com os valores apropriados.



```
n_B = 1000
x_B = 300
p_B = x_B / n_B

E_B = Z * ((p_B * (1 - p_B)) / n_B)**0.5

inferior_B = p_B - E_B
superior_B = p_B + E_B

print(f'Intervalo de confiança para o bairro B: [{inferior_B}, {superior_B}]')
```

```
Intervalo de confiança para o bairro B: [0.27159742349106747, 0.3284025765089325]
```

c) É possível obter isso com o seguinte script, em Python:



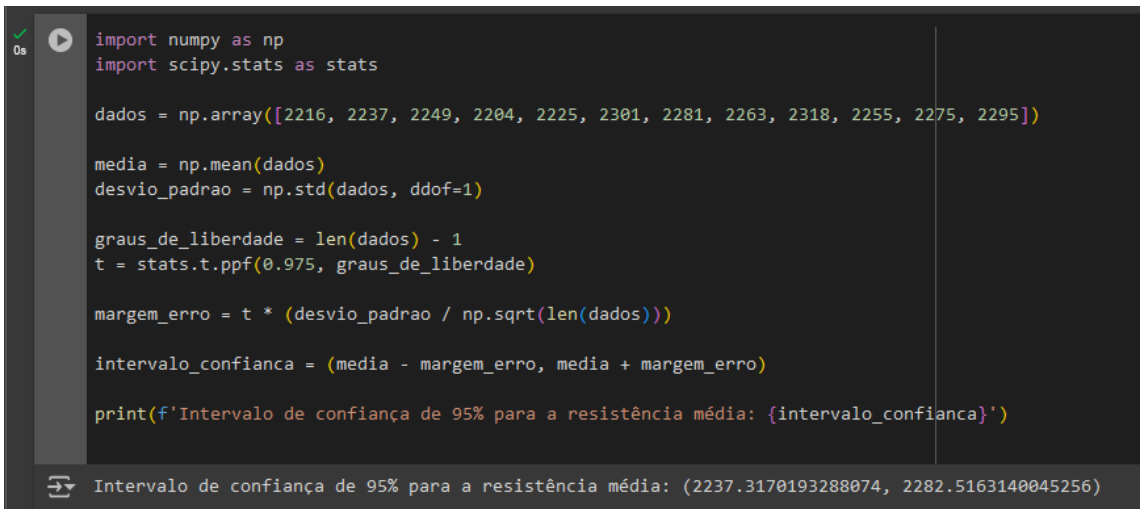
```
E_dif = Z * ((p_A * (1 - p_A) / n_A) + (p_B * (1 - p_B) / n_B))**0.5

dif = p_A - p_B
diferença_inferior = dif - E_dif
diferença_superior = dif + E_dif

print(f'Intervalo de confiança para a diferença entre os bairros A e B: [{diferença_inferior}, {diferença_super
```

```
Intervalo de confiança para a diferença entre os bairros A e B: [-0.14512175944007374, -0.054878240559926204]
```

A primeira linha realiza o cálculo da margem de erro para a diferença entre as proporções de intenção de voto nulo nos bairros A e B. Após isso, são calculadas as diferenças entre as proporções de intenção de voto, e, por fim, é impresso o resultado.

3) A resistência do concreto à compressão está sendo testada por um engenheiro. Ele testa 12 corpos de prova e obtém os dados abaixo. Construa um intervalo de confiança de 95% para a resistência média.

2216	2237	2249	2204
2225	2301	2281	2263
2318	2255	2275	2295



```python
import numpy as np
import scipy.stats as stats

dados = np.array([2216, 2237, 2249, 2204, 2225, 2301, 2281, 2263, 2318, 2255, 2275, 2295])

media = np.mean(dados)
desvio_padrao = np.std(dados, ddof=1)

graus_de_liberdade = len(dados) - 1
t = stats.t.ppf(0.975, graus_de_liberdade)

margem_erro = t * (desvio_padrao / np.sqrt(len(dados)))

intervalo_confianca = (media - margem_erro, media + margem_erro)

print(f'Intervalo de confiança de 95% para a resistência média: {intervalo_confianca}')
```

```
Intervalo de confiança de 95% para a resistência média: (2237.3170193288074, 2282.5163140045256)
```

Inicialmente, é necessário importar os módulos necessários para a questão. Após isso, usamos NumPy para fazer uma matriz com os valores fornecidos. Após isso, há as funções de cálculo das propriedades que a atividade pede. Por fim, o programa imprime os resultados encontrados.

4) O conteúdo de açúcar na calda de pêssegos em lata é normalmente distribuído. Uma amostra aleatória de n=10 latas resulta em um desvio padrão amostral de S=4,8 miligramas. Calcule o intervalo de confiança de 95% para o desvio padrão.

```python
import scipy.stats as stats

n = 10
S = 4.8

graus_de_liberdade = n - 1

chi2_alpha_2 = stats.chi2.ppf(0.025, graus_de_liberdade)
chi2_1_minus_alpha_2 = stats.chi2.ppf(0.975, graus_de_liberdade)

lower = (n - 1) * S**2 / chi2_1_minus_alpha_2
upper = (n - 1) * S**2 / chi2_alpha_2

print(f'Intervalo de confiança de 95% para o desvio padrão: [{lower**0.5}, {upper**0.5}]')
```

```
Intervalo de confiança de 95% para o desvio padrão: [3.3016089923951983, 8.762928876093953]
```

Iniciamos realizando o mesmo processo da questão anterior com relação à importação de módulos. Após isso, são definidos o tamanho da amostra e o desvio padrão como as variáveis “n” e “S”. Após isso, é calculado o grau de liberdade; que será usado posteriormente no código para o cálculo dos valores críticos. Por fim, é impresso o resultado.

5) Uma fábrica de celulares quer testar a hipótese de haver diferença significativa de peso médio de seus aparelhos entre duas amostras. Amostra 1: n1=10 celulares e média 184,6 g. Amostra 2: n2=20 celulares e média 188,9 g. Sabemos que o desvio padrão é constante e igual a 5 g. A)Qual a conclusão considerando α=5% R. Zcrit=1,96 e Zcalc= -2,22 B)Qual a conclusão considerando α=1%

```python
import math

n1 = 10
media1 = 184.6

n2 = 20
media2 = 188.9

desvio_padrao = 5

alpha_1 = 0.05  # 5%
alpha_2 = 0.01  # 1%

Z = (media1 - media2) / (desvio_padrao * math.sqrt((1 / n1) + (1 / n2)))
```

```
Z_critico_05 = 1.96

Z_critico_01 = 2.576

if Z < -Z_critico_05 or Z > Z_critico_05:
    print("Devemos rejeitar a hipótese nula para alpha=5%, pois há diferença significativa de peso médio entre as amostras.")
else:
    print("Não devemos rejeitar a hipótese nula para alpha=5%, pois não há diferença significativa de peso médio entre as amostras.")
```

```
if Z < -Z_critico_01 or Z > Z_critico_01:
    print("Devemos rejeitar a hipótese nula para alpha=1%, pois há diferença significativa de peso médio entre as amostras.")
else:
    print("Não devemos rejeitar a hipótese nula para alpha=1%, pois não há diferença significativa de peso médio entre as amostras.")
```

Retorno do script:

```
Devemos rejeitar a hipótese nula para alpha=5%, pois há diferença
significativa de peso médio entre as amostras.
Não devemos rejeitar a hipótese nula para alpha=1%, pois não há
diferença significativa de peso médio entre as amostras.
```

Código explicado:

É importado o módulo necessário, são os definidos os dados das duas amostras, sendo elas o tamanho e a média. É informado o desvio padrão conhecido, são definidos os níveis de significância para ambas as situações, são definidos os valores críticos Z, e, por fim, é realizada a comparação de dados que nos retorna a resposta final.

6) Uma rede de lojas de sapato está estudando se deve implantar ou não uma nova ação de marketing. Para isso, computou os dados de venda de uma de suas lojas antes e depois de efetuar a ação. Antes da ação: Média = R$ 68; Variância = 50; nA=12. Após a ação Média = R$ 76; Variância =75; nB=15. Ao nível de significância de 5%, o que podemos concluir a respeito da ação de marketing? R. tcal=-2,582 e tcrit=-1,708

Resposta na próxima página.

```python
import math

media_A = 68
variancia_A = 50
n_A = 12

media_B = 76
variancia_B = 75
n_B = 15

alpha = 0.05

pooled_variancia = ((n_A - 1) * variancia_A + (n_B - 1) * variancia_B) / (n_A + n_B - 2)

t = (media_A - media_B) / (math.sqrt(pooled_variancia * ((1 / n_A) + (1 / n_B))))

graus_liberdade = n_A + n_B - 2

valor_critico_t = -1.708
```

```python
if t < valor_critico_t:
    print("Rejeitamos a hipótese nula. Há evidências estatísticas para suportar que a ação de marketing teve um efeito significativo nas vendas.")
else:
    print("Não rejeitamos a hipótese nula. Não há evidências estatísticas suficientes para suportar que a ação de marketing teve um efeito significativo nas vendas.")
```

Retorno do script:

```
Rejeitamos a hipótese nula. Há evidências estatísticas para suportar
que a ação de marketing teve um efeito significativo nas vendas.
```

Código explicado:

É importado o módulo necessário para a atividade, são definidos os dados antes e depois a ação de marketing, é definido o nível de significância. A partir disso, são realizados variados cálculos envolvendo os graus de liberdade, e, no fim do código, é realizada uma comparação que retorna a conclusão acima.

7) Uma empresa que desenvolve softwares precisa garantir a qualidade do seu processo de desenvolvimento e, para isso, está testando dois métodos de trabalho diferentes. A seguir, apresentam-se os valores encontrados para o número de erros produzidos por duas equipes independentes, uma atuando em Recife, seguindo o método 1, outra atuando em Bangalore, seguindo o método 2.

Valores:

Método1	**2748**	**2700**	**2655**	**2822**	**2511**	**3149**	**3257**	**3213**	**3220**	**2753**
Método 2	2727	3706	3709	3547	3275	2560	2589	2652		

a) Teste, ao nível de significância de 10%, se a variância do número de erros produzidos seguindo o método 2 é maior do que a variância de acordo com o método

1. R. Fcalc = 3,456 e Fcrit=2,51

```
import numpy as np
from scipy.stats import f

erros_metodo1 = np.array([2748, 2700, 2655, 2822, 2511, 3149, 3257, 3213, 3220, 2753])
erros_metodo2 = np.array([2727, 3706, 3709, 3547, 3275, 2560, 2589, 2652])

variancia_metodo1 = np.var(erros_metodo1, ddof=1)
variancia_metodo2 = np.var(erros_metodo2, ddof=1)

F = variancia_metodo2 / variancia_metodo1

alpha = 0.10

df1 = len(erros_metodo2) - 1
df2 = len(erros_metodo1) - 1
F_critico = f.ppf(1 - alpha, df1, df2)
```

```
if F > F_critico:
    print("Rejeitamos a hipótese nula. A variância do método 2 é maior do que a variância do método 1.")
else:
    print("Não rejeitamos a hipótese nula. Não há evidências suficientes para afirmar que a variância do método 2 é maior do que a variância do método 1.")
```

Retorno do script:

```
Rejeitamos a hipótese nula. A variância do método 2 é maior do que a
variância do método 1.
```

Código explicado:

Neste código, é usado o teste F. Primeiramente, é calculada a estatística F e depois é realizada uma comparação com o valor crítico F para, assim, poder determinar se a variância do método 2 é de forma significativa maior que a do método 1.

b) Com base no resultado do teste do item anterior, realize um outro teste, para verificar se, ao nível de significância de 5%, o método 2 é menos eficaz que o método

1. R. tcal = - 0,953 e tcrit=1,746

```python
from scipy.stats import t

t_calculado = (np.mean(erros_metodo1) - np.mean(erros_metodo2)) / np.sqrt((variancia_metodo1 / len(erros_metodo1)) + (variancia_metodo2 / len(erros_metodo2)))

alpha = 0.05
```

```python
graus_liberdade = len(erros_metodo1) + len(erros_metodo2) - 2
t_critico = t.ppf(alpha / 2, graus_liberdade)

if t_calculado < -t_critico:
    print("Rejeitamos a hipótese nula. O método 2 é menos eficaz que o método 1.")
else:
    print("Não rejeitamos a hipótese nula. Não há evidências suficientes para afirmar que o método 2 é menos eficaz que o método 1.")
```

Retorno do script:

```
Rejeitamos a hipótese nula. O método 2 é menos eficaz que o método 1.
```

Código explicado:

Baseado no teste anterior, realizamos outro teste, porém, assumimos que as variâncias são diferentes.