Processo seletivo para a vaga de

Hacker de Fiscalização e Análise de Dados

do *Gabinete Compartilhado*

SEGUNDA PARTE

Nosso objetivo com a segunda parte, composta pelas cinco questões abaixo, é conhecer melhor as suas especialidades e pontos de destaque. Quanto mais você puder responder, melhor, mas não se preocupe caso não consiga responder tudo. Lembre-se de enviar as respostas para o e-mail henrique.xavier@senado.leg.br em **até três horas a partir do preenchimento do formulário Google e até o final do dia 8 de maio**.

Questão 1. Imaginemos que, em um certo município que possui dois distritos, a prefeitura anunciou que pretende construir 16 creches. Para evitar brigas e disputas entre os distritos, a prefeitura propôs o seguinte: para cada creche, ela irá sortear, com reposição, o distrito onde será feita a construção. Acontece que um dos distritos foi escolhido 12 vezes. É possível afirmar que a escolha não foi feita de maneira aleatória? Existiria algum resultado do sorteio para o qual sua resposta mudaria? Explique seu raciocínio.

Questão 2. Em uma determinada cidade do Brasil, as crianças recém-nascidas do sexo masculino nascem com peso segundo uma distribuição normal com média 2,4kg e desvio padrão igual a 0,040kg. Nesse cenário, os 25% mais leves possuem, no máximo, quantos quilos?

Questão 3. Preencha a docstring da função abaixo, substituindo as tags <...> pelo que elas pedem.

```
def f(text, sep=';', end='\n'):
    """
    <Descrição do que a função faz>

    Parâmetros
    ----------
    text : <Tipo>
        <Descrição desse parâmetro>
    sep : <Tipo>
        <Descrição desse parâmetro>
    end : <Tipo>
        <Descrição desse parâmetro>

    Retorna
    -------
    final : <Tipo>
        <Descrição da saída da função>
    """

    for x in [text, sep, end]:
        assert type(x) == str

    header = text.split(end)[0].split(sep)
    header = [item.strip().lower().replace(' ', '_') for item in header]

    data = text.split(end)[1:]
    if data[-1].strip() == '':
        data = data[:-1]

    final = []
    for instance in data:
        instance = instance.split(sep)
        entrada = dict(zip(header, instance))
        final.append(entrada)

    return final
```

Questão 4. Considere as seguintes tabelas e o código SQL abaixo:

Transacoes		
cartao	data	valor
3213	02/04/2018	20
3213	11/05/2018	30
3213	23/04/2018	50
7624	18/05/2018	10
7624	09/04/2018	5
2121	27/05/2018	25

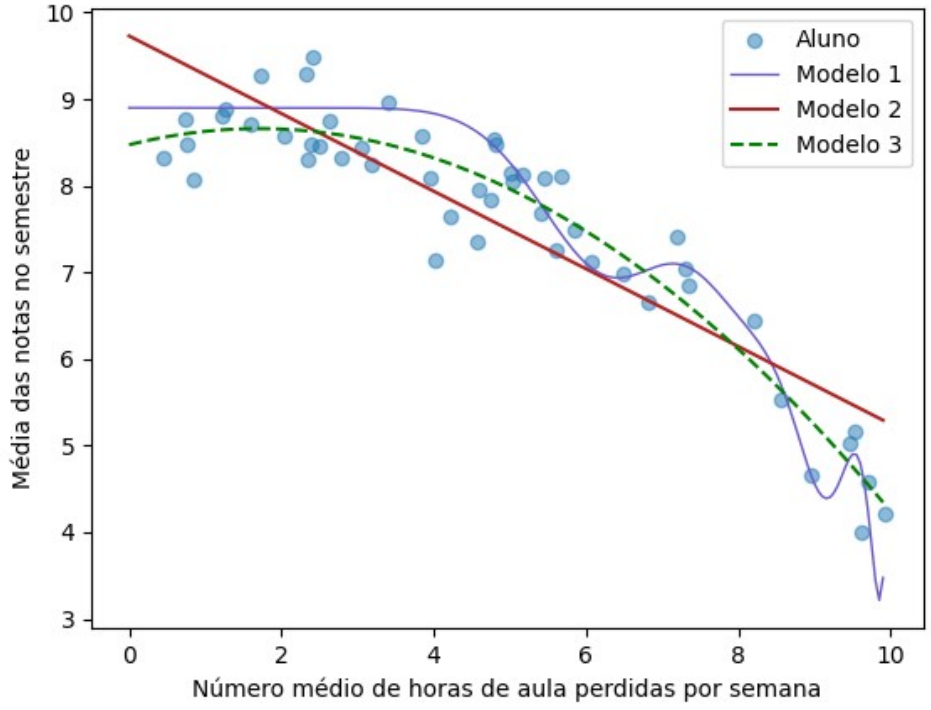
Cartoes	
id_cliente	cartao
4433	3213
2134	7624
9987	2121
8765	9864

```
SELECT
        c.id_cliente AS id_cliente,
        SUM(t.valor) AS soma_gastos_cartao
FROM cartoes AS c
LEFT JOIN transacoes AS t
ON c.cartao = t.cartao
GROUP BY c.id_cliente
ORDER BY id_cliente
```

Sobre a tabela retornada pela consulta, responda:
a) Qual é o valor da coluna soma_gastos_cartao para a segunda linha?

b) Considere que a cláusula LEFT JOIN é substituída por INNER JOIN. Quais serão os valores, em ordem, da coluna id_cliente na tabela resultante? Separar os valores por vírgula (ex.: valor1, valor2, …)

Questão 5. Considere uma faculdade fictícia com 1000 estudantes. Com a intenção de estimar como a quantidade de horas de aula perdidas pelos alunos impactam nos seus rendimentos, a faculdade selecionou 50 alunos de maneira aleatória e monitorou as horas perdidas e as médias das notas no semestre de cada um. Aos resultados observados foram ajustados três modelos preditivos, apresentados na figura abaixo.



Se utilizarmos os três modelos para prever a média das notas dos outros 950 estudantes a partir do número médio de horas perdidas, qual deles deve apresentar o melhor desempenho (i.e. o menor erro médio)? Explique porque os outros dois modelos devem apresentar um desempenho pior.