MAP2110 - Projeto 1
Administração Florestal

Enzo Valentim Cappeloza Nº USP 12556736
Henrique Fujikawa Tokunaga NºUSP 12675207
Lucas Panfilo Donaire Nº USP 12556552
Marcos Martins Marchetti Nº USP 11910868
Ygor Peniche Maldonado Nº USP 10271558

Abril 2021

Questão 1

Citemos as variáveis do problema e a quais condições elas devem obedecer.

Vetor Configuração

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$$

Número de árvores por classe

$$G_x = \begin{pmatrix} (1-g)x_1 \\ (1-g_2)x_2 + g_1x_1 \\ (1-g_3)x_3 + g_2x_2 \\ \vdots \\ (1-(g_{n-1})(x_{n-1})) + (g_{n-2})(x_{n-2}) \\ x_n + x_{n-1}g_{n-1} \end{pmatrix}$$

$$\text{onde } \ 0 \leq g_i \leq 1, \ i = 1, \ 2, \ 3, \ \ldots \ n-1.$$

Vetor corte

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix}$$

Reposição

Definimos a matriz R_{nxn} para que possamos escrever uma equação matricial de reposição de árvores, que serão retiradas de acordo com a matriz Y.

Portanto:

$$R = \begin{pmatrix} 1 & 1 & \dots & 1 & 1 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Para definir a matriz reposição, devemos multiplicar a matriz R *por* Y, *onde*

$$R.Y = \begin{pmatrix} y_1 + \quad y_2 + \quad \dots \quad +y_{n-1} + \quad y_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Onde o somatório de y_1 *até* y_n *será o número de mudas plantadas.*

Equação para a Colheita Sustentável

A equação abaixo representa a política de corte sustentável,

$$\begin{bmatrix} configuração\ no \\ final\ do\ crescimento \end{bmatrix} \text{ - } \begin{bmatrix} corte \end{bmatrix} + \begin{bmatrix} reposição \end{bmatrix} = \begin{bmatrix} configuração\ no \\ início\ do\ crescimento \end{bmatrix}$$

Conforme demonstrado pela equação matricial, também podemos representar por $G_x - y + R_y = x \Rightarrow (I - R)y = (G - I)x$, *ou matricialmente:*

$$\begin{bmatrix} 0 & -1 & -1 & \dots & -1 & -1 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{(n-1)} \\ y_n \end{bmatrix} =$$

$$\begin{bmatrix} -g_1 & 0 & 0 & \dots & 0 & 0 \\ g_1 & -g_2 & 0 & \dots & 0 & 0 \\ 0 & g_2 & -g_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -g_{(n-1)} & 0 \\ 0 & 0 & 0 & \dots & -g_{(n-1)} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{(n-1)} \\ x_n \end{bmatrix} \tag{1}$$

onde $x_i > 0$ *e* $y_i > 0$.

Questão 2

Nota-se que, pelas condições do problema, a quantidade s de árvores deve se manter constante, portanto, temos que:

$$s = \sum_{i=1}^{n} x_i$$

Ademais, como o valor de venda das mudas é zero, definimos $y_1 = 0$.

Obtêm-se, transformando a equação (1) num sistema de equações lineares:

$$\begin{aligned} \sum_{i=1}^{n} y_i &= g_1 x_1 \\ y_2 &= g_1 x_1 - g_2 x_2 \\ y_3 &= g_2 x_2 - g_3 x_3 \\ &\vdots \\ y_{n-1} &= (g_{n-2})(x_{n-2}) - (g_{n-1})(x_{n-1}) \\ y_n &= g_{n-1} x_{n-1} \end{aligned} \tag{2}$$

Note que a primeira equação é a soma de todas as demais $n-1$ *equações do sistema linear. Para que o modelo faça sentido devemos ter* $y_i \geq 0$, *caso contrário, haveria pelo menos uma classe cuja quantidade de árvores é negativa. De acordo com tal condição e com (2), podemos inferir que uma das condições exigidas é:*

$$g_1x_1 \geq g_2x_2 \geq \ldots \geq g_{n-1}x_{n-1} \geq 0 \tag{3}$$

É possível observar que é condição suficiente que o vetor coluna X seja integralmente composto de entradas não negativas a fim de satisfazer as equações (1) e (2). No entanto, deve-se respeitar a condição da equação (3), que é necessária, para que o modelo torne-se coerente. Caso a condição (3) não fosse respeitada, poderíamos dizer que a tendência do modelo seria a de haver uma concentração de árvores nas classes mais altas. Isso implica dizer que: Ou as árvores plantadas já estariam em uma classe mais elevada do que a 1ª, ou que as árvores podem subir mais de uma classe por ciclo de crescimento, ou que haveriam mais árvores crescendo para a classe $k+1$ do que as existentes na classe k.

Questão 3

O algoritmo fora construído no software Python usando a biblioteca NumPy para a realização dos cálculos matriciais. Nesse, o usuário é apresentado com uma interface onde deve inserir a quantidade de classes presentes no estudo, os valores de cada classe de árvore e a constante g_i de crescimento para cada classe. Fora adotado, em cada etapa da inserção dos dados, um loop, o qual permite ao usuário confirmar ou rejeitar os dados previamente inseridos. Tal medida fora feita para evitar situações em que dados colocados erroneamente comprometam a capacidade do usuário de reiniciar a etapa somente em que há erros, tendo assim que "começar do zero" a etapa de inserção de dados.

*O código-fonte do programa pode ser encontrado tanto no Apêndice desse documento, como também em anexo, com o nome **forest.py**, na plataforma em que o relatório fora entregue.*

Questão 4

O retorno sustentável ótimo relativo à k-ésima classe, de acordo com [1], pode ser obtido pela seguinte expressão:

$$Ret_k = \frac{P_k s}{\sum\limits_{i=1}^{k-1} \frac{1}{g_i}}$$

Questão 5

Para os dados informados, o retorno ótimo será para a 3ª classe, com o valor de $14.71 \times s$. Abaixo está uma imagem retirada do código desenvolvido em Python, na qual utiliza a equação apresentada na questão 4,

```
A matriz dos preços é representada por:
[  0.  50. 100. 150. 200. 250.]

A matriz das taxas de crescimento é representada por:
[0.28 0.31 0.25 0.23 0.37  nan]

A Matriz de retorno ótimo é representado por:
[ 0.         14.         14.71186441 13.89244558 13.20562516 14.00735703]

A Classe com retorno ótimo é a 3ª, com o valor de 14.7119
```

*Foram adotados os valores **p** em "unidades de moeda" e **g** adimensional:*

$$P_1 = 0, g_1 = 0.28$$
$$P_2 = 50, g_2 = 0.31$$
$$P_3 = 100, g_3 = 0.25$$
$$P_4 = 150, g_4 = 0.23$$
$$P_5 = 200, g_5 = 0.37$$
$$P_6 = 250$$

Questão 6

Já temos os retornos ótimos de todas as classes, calculados pelo software.

Denotemos o Retorno Sustentável Ótimo da classe k por Ret_k*, portanto:*

$$Ret_{3Max} = S \times 14,71186441$$
$$Ret_5 = S \times 13,20562516$$
$$P_3 = 100$$
$$P_5 = 200$$

Onde o Ret_k *é obtido através da equação apresentada na **Questão 3**.*

Para achar P_5*' (novo preço da classe 5 para ter o maior retorno sustentável ótimo) , precisamos achar o* K_5*, portanto:*

$S \times 13,20562516 = 200 \times S/K_5$

$1/k_5 = 0,66$ *(usaremos tal valor a fim de simplificar as contas)*

Realizando o processo inverso, obtêmos:

$Ret_{5'} > Ret_3$

$P_{5'} \times S \times 0,66 > 14,71186441 \times S$

$P_{5'} > 229,901$

Usando um valor inteiro, o novo valor para a classe 5 para que essa tenha o maior retorno sustentável ótimo e, portanto, seja retirada, é de:

$P_{5'}$ *= 223*

$Ret_{5'} = 14,724272 \times S$

Questão 7

As equações que regem a condição são:

$$\frac{P_2.S}{\sum_{i=1}^{1}\frac{1}{g_i}} = \frac{P_3.S}{\sum_{i=1}^{2}\frac{1}{g_i}} = \frac{P_4.S}{\sum_{i=1}^{3}\frac{1}{g_i}} = \frac{P_5.S}{\sum_{i=1}^{4}\frac{1}{g_i}} = \frac{P_6.S}{\sum_{i=1}^{5}\frac{1}{g_i}}$$

Adotando, então, o valor do Retorno Ótimo proveniente da Questão 5, temos: $P_k = P_3 = 100$. *Onde Pk é o valor de mercado da classe que será totalmente colhida, tal que:*

$$Ret_{3Max} = [P_3 s/(1/g_1 + 1/g_2)] = 14,7s$$

e para que a condição exposta seja verdade queremos:

$$\begin{aligned} P_2 &= 52,5; \\ P_3 &= 100; \\ P_4 &= 158,72; \\ P_5 &= 222,64; \\ P_6 &= 262,37 \end{aligned}$$

Dessa forma, qualquer que seja a classe que se escolha, o Retorno Sustentável Ótimo é obtido. É importante ressaltar que esses não são os únicos valores que proporcionam uma igualdade dos retornos: multiplicar todos esses valores por uma mesma constante irá manter a igualdade, pois a proporção dos preços não se alteraria.

Questão 8

Em cada colheita, devem ser removidas todas as árvores pertencentes à classe que provê o retorno ótimo, sendo a equação que determina a quantidade de árvores: $y_k = x_{k-1}g_{k-1} - x_k y_k$
*Em especial, para a classe da **Questão 5** temos que a equação é:*

$$y_3 = g_2 x_2 - g_3 x_3$$

Já que esse número tem que ser igual ao número de mudas que crescem por época de colheita (a quantidade de mudas que crescem é igual à quantidade retirada, mantendo a colheita sustentável), podemos também dizer que $y_k = g_1 x_1$, *o que é confirmado pela equação (2).*

Questão 9

Através do algoritmo, encontramos os seguintes retornos ótimos para cada uma das classes do exemplo 5:

$$Ret_1 = 0 \times s$$
$$Ret_2 = 14 \times s$$
$$Ret_3 = 14.71186441 \times s$$
$$Ret_4 = 13.89244558 \times s$$
$$Ret_5 = 13.20562616 \times s$$
$$Ret_6 = 14.00735703 \times s$$

Considerando uma situação com 100 árvores, teríamos:

$$Ret_1 = 0.00$$
$$Ret_2 = 1400.00$$
$$Ret_3 = 1471.186441$$
$$Ret_4 = 1389.244558$$
$$Ret_5 = 1320.562616$$
$$Ret_6 = 1400.735703$$

Com 500 árvores, aproximadamente:

$$Ret_1 = 0.00$$
$$Ret_2 = 7000.00$$
$$Ret_3 = 7355.93$$
$$Ret_4 = 6946.22$$
$$Ret_5 = 6602.81$$
$$Ret_6 = 7003.68$$

Com 2000 árvores, aproximadamente:

$$Ret_1 = 0.00$$
$$Ret_2 = 28000.00$$
$$Ret_3 = 29423.76$$
$$Ret_4 = 27784.88$$
$$Ret_5 = 26411.24$$
$$Ret_6 = 28014.70$$

É nítido que o maior retorno ótimo é o da classe 3, como previsto.
Para testar o algoritmo com outros valores, fizemos as contas com

$$g_1 = 0.35 \qquad p_2 = 100$$
$$g_2 = 0.25 \qquad p_3 = 200$$
$$g_3 = 0.40 \qquad p_4 = 300$$
$$g_4 = 0.50 \qquad p_5 = 400$$
$$g_5 = 0.15 \qquad p_6 = 500$$

Isso resultou nos retornos ótimos de:

$$Ret_1 = 0.00 \times s$$
$$Ret_2 = 35.00 \times s$$
$$Ret_3 = 29.16666667 \times s$$
$$Ret_4 = 32.0610687 \times s$$
$$Ret_5 = 35.22012579 \times s$$
$$Ret_6 = 27.74108322 \times s$$

Com 100 árvores, os retornos seriam:

$$Ret_1 = 0.00$$
$$Ret_2 = 3500.00$$
$$Ret_3 = 2916.666667$$
$$Ret_4 = 3206.10687$$
$$Ret_5 = 3522.012579$$
$$Ret_6 = 2774.108322$$

Com 500 árvores, aproximadamente:

$$Ret_1 = 0.00$$
$$Ret_2 = 17500.00$$
$$Ret_3 = 14583.33$$
$$Ret_4 = 16030.53$$
$$Ret_5 = 17610.06$$
$$Ret_6 = 13870.54$$

Com 2000 árvores, aproximadamente:

$$Ret_1 = 0.00$$
$$Ret_2 = 70000.00$$
$$Ret_3 = 58333.33$$
$$Ret_4 = 64122.14$$
$$Ret_5 = 70440.25$$
$$Ret_6 = 55482.17$$

Nesse caso, a classe 5 é a com maior retorno, seguida por perto pela classe 2. O aumento nos preços desse exemplo em relação ao anterior, além de leves variações para cima nas taxas de crescimento, fizeram esses valores terem retornos ótimos superiores.

Questão 10

Nesta seção realizaremos uma análise crítica acerca da aplicabilidade do modelo, tendo em vista alguns fatores mercadológicos e temporais.

Flutuação de preços de mercado no período intra-plantio.

O modelo não leva em conta as flutuações no preço de quaisquer itens intrínsecos ao plantio e manutenção da plantação. Parte-se do pressuposto que os valores das mudas de árvores jovens, mão de obra, custo de manutenção e operação de maquinário, insumos, fertilizantes e bio-defensores agrícolas, são todos constante ao longo de todos os anos da plantação. Ou seja, ainda que o modelo seja capaz de contabilizar tais variações durante, e somente durante, o período de corte das árvores do ano t *e plantio das árvores jovens do ano* $t+1$*, através de um incremento monetário nos valores de venda* y_i*, não é possível realizar quaisquer alterações durante o período de crescimento das árvores, visto que, uma vez fixados, os valores se mantêm até Dezembro, quando as árvores da classe* y_k *são cortadas.*

Portanto, caso ocorra alguma variação abrupta em alguns dos fatores citados, durante o período Janeiro-Dezembro, o agricultor pode sofrer perdas financeiras.

Ademais, caso o modelo necessite de alterações para adequar o valor de venda das árvores da classe y_k*, toda e qualquer alteração deverá ser feita manualmente. Isso implica dizer que o tempo gasto pelo modelador do problema, no ato de contabilizar todos os custos de operação, será demasiadamente grande. Deverão ser contabilizados, manualmente, todas as variações de preço dos itens necessários ao plantio.*

Árvores que não morrem.

O modelo supõe que as árvores não morrem, seja de causa natural ou ocasionada por alguma praga ou doença.

Estabilidade e variação dos fatores de crescimento.

Os fatores de crescimento g_i *são ditos como constantes durante todo o período Janeiro-Dezembro. Exime-se de qualquer influência sob o crescimento das árvores os fatores climáticos, ainda que a plantação seja exposta às quatro estações do ano. Também não é contabilizado o manejo não ideal do solo ou efeitos de lixiviação devido a períodos de alta pluviosidade, os quais podem retardar ou até mesmo inviabilizar o crescimento das árvores por falta de nutrientes.*

O risco da sustentabilidade e mudança parâmetros intra-plantio.

Suponha que a classe com retorno ótimo e de regime de plantio sustentável seja a de y_j*. Suponha também que um número suficientemente grande de árvores de alguma classe morra. Note que o algoritmo continuará a apontar a classe* y_j *como a de retorno sustentável ótimo devido à hipótese anterior de que as árvores não morrem. O problema não é de fato quando as árvores morrem já estando na classe* y_j *pois, de acordo com [2] ainda seria possível vender a madeira, mesmo* *que de uma árvore morta. A problemática está quando uma árvore de uma classe $i < j$ morre, pois o algoritmo não é capaz de apresentar uma metodologia de corte a fim de minimizar prejuízos financeiros e/ou ambientais.*

Falta de melhores parâmetros de controle de preço.

*Em [2] as classes são definidas com base na **circunferência**, diferentemente da classe de **altura** das árvores. Cremos que é possível obter um modelo de maior retorno financeiro caso forem aliados ambos fatores. Uma árvore cuja circunferência do tronco é maior, por exemplo, teria usos mais variados no mercado. Portanto, o preço das árvores cuja classe de circunferência é maior poderia ser mais elevada tanto pelo fator **massa total de madeira disponível** quando pelo fator **variedade de usos no mercado**. A altura poderia ser usada como um fator secundário, exclusivo para classificar a massa total de madeira por árvore. Sendo assim, duas árvores que ocupem a mesma classe de altura, com circunferências de tronco diferentes, teriam preços diferentes, sendo aquela com o tronco de maior diâmetro a de valor mais elevado.*

Questão 11

Primeiramente, é possível estimar o parâmetro S (número total de árvores) com base na área do terreno, ao passo que, o número de classes de árvores é determinado previamente. O parâmetro preço, por sua vez, mantém certa constância, mas em períodos de colheita muito longos, é ideal estimar variações baseadas na inflação para evitar problemas. A grande questão está em como estimar a taxa de crescimento G. O ideal seria uma pesquisa aprofundada sobre a espécie de árvore (no caso da Questão 5, pinheiros da Escócia) e sobre seu desenvolvimento em determinadas situações.Quanto maior o tempo de colheita, os desequilíbrios naturais entre uma árvore e outra acabam sendo homogeneizados, pois pequenos fenômenos (chuva fora de época, dias de seca, etc.) se equilibram a longo prazo. Enquanto isso, em colheitas curtas, a chance de imprevistos é maior. Por outro lado, é preciso observar se o clima e o solo interferem decisivamente no crescimento das árvores, pois baseando-se somente nas informações da espécie (sem levar em conta o ambiente), imprevistos podem ocorrer.

Com uma boa base de dados (relativamente fácil de encontrar na internet, em revistas, bibliotecas), será possível obter informações precisas sobre o crescimento das espécies em situações não excepcionais. Uma média aritmética entre os tempos de crescimento de determinada faixa de altura (de preferência já levando em consideração o solo e clima em que serão futuramente plantados) seria suficiente para uma boa estimativa, com um olhar atento sobre dados discrepantes. Ainda assim, o ideal para o início da plantação seria a formação de um grupo controle: não cortar algumas árvores até chegarem em sua altura máxima, pois assim os dados estimados seriam empiricamente comprovados e ganhariam maior precisão. Importante ressaltar que sempre se deve adequar o modelo ao que se observa na realidade (se os dados observados na floresta *são diferentes dos anteriormente coletados, devem substituí-los. Usar a média aritmética entre as observações deve resultar medidas precisas.). Após algumas colheitas, o modelo atingiria eficiência máxima na estimativa das taxas de crescimento, facilitando a sustentabilidade e aumentando o retorno.*

1 Bibliografia

1 - Anton Rorres, Álgebra Linear com Aplicações, 10ª Ed., Bookman, 2012
2 - M.B. Usher, "A Matrix Approach to the Management of Renewable Resources, with Special Reference to Selection Forest" , Journal of Applied Ecology, Vol. 3, 1966, pág. 355-367

Apêndice: Código do Algoritmo

Listing 1: forest.py

```
import numpy as np


def get_matrices():
    """
    Obtem do usu rio os valores para:
    Classes: Agrupamento das avores conforme as suas faixas de alturas.
    Pre os: Pre os das  rvores  que est o em cada classe.
    Crescimeto: Taxa de crescismento de quantas avores passam de uma classe pa
    :return: o valor das classes, pre os e taxa de crescimento.
    """
    # variavel que armazena a quantidade de classes das alturas das avores
    classes = None

    # dicionario que armazena os pre os de cada classe
    precos = {}

    # dicion rio que armazena as taxas de crescimento de cada classe
    crescimento = {}

    # variaveis que controlam as entradas do usuario
    classes_flag = True
    precos_flag = False
    crescimento_flag = False
    while True:

        # recebe do usuario as entradas da quantidade de classe
        if classes_flag:
            classes = int(input('\nDigite quantas classes de avores existem: '
            correto = input(f'\nConfirma a quantidade de {classes} classes? [S

            if correto == 's':
                classes_flag = False
                precos_flag = True

        # recebe do usuario as entradas referente aos pre os de cada classe
        if precos_flag:

            # o pre o da primeira classe recebe o valor de $0,00, conforme ex
            precos['classe_1'] = 0.00

            for i in range(2, classes + 1):
```

```
                precos[f'classe_{i}'] = float(input(f'\nInforme o pre o das 

            correto = input(f'\nOs pre os est o corretos? \n{precos} \n[S/N]
            if correto == 's':
                precos_flag = False
                crescimento_flag = True

        # recebe do usuario as taxas de crescimento de cada classe
        if crescimento_flag:
            for i in range(1, classes):
                crescimento[f'classe_{i}'] = float(input(
                    f'\nInforme a taxa de crescimento das  rvores  na classe {

            correto = input(f'\nAs taxas de crescimento est o corretas? \n{cr
            if correto == 's':
                break

    return classes, precos, crescimento


def calculate_matrices(classes, preco, crescimento):
    """
    Realiza o calculo do retorno otimo sustentavel.
    :param classes: Quantidade de classes.
    :param preco: Matriz com os pre os das arvores.
    :param crescimento: Matriz contendo as taxas de crescimento.
    :return: Exibe na tela a matriz resultado, o valor e classe do retorno otin
    """
    # converte em numpy array para podermos realizar operacoes com matrizes
    matrix_price = np.array(list(preco.values()))
    matrix_growth = np.array(list(crescimento.values()))

    # lista que armazena os resultados
    matrix_result = [.0]
    for i in range(1, classes):

        # itera em cada classe e realiza os c lculos das matrizes.
        # c lculo demonstrado na quest o 3
        result = (matrix_price[i] / np.sum(1 / matrix_growth[:i]))
        matrix_result.append(result)

    # salva em numpy array os resultado do calculo anterior
    matrix_result = np.array(matrix_result)

    # armazena o maior valor obtido no calculo
    optimal_return = np.max(matrix_result)
```

```
    # armazena o n mero da classe que obteve o maior valor
    optimal_class = np.argmax(matrix_result) + 1

    print(f'\nA matriz dos pre os   representada por:\n{matrix_price}')
    matrix_growth = np.append(matrix_growth, np.NAN)
    print(f'\nA matriz das taxas de crescimento   representada por:\n{matrix_g
    print(f'\nA Matriz de retorno  timo     representado por:\n{matrix_result}
    print(f'\nA Classe com retorno  timo     a {optimal_class} , com o valor


if __name__ == '__main__':
    # inicia o codigo
    print('Sistema de Retorno M ximo para um Plantio Sustent vel.')
    cl, pr, cr = get_matrices()
    calculate_matrices(cl, pr, cr)
```