UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
DEPARTAMENTO DE INFORMÁTICA

**Filipe Gomes Arante de Souza**

# 2º Trabalho de Algoritmos Numéricos
# Regressão Linear Múltipla e Polinomial

**Julho / 2022**

# 1 Impressões de Tela do Editor do Octave

## 1.1 Geração da matriz de variáveis explicativas

Segue abaixo implementação:

```
% Gera a matriz de variáveis explicativas de acordo com o modelo especificado.
% n: Número de pontos.
% v: Quantidade de variáveis explicativas.
% p: Quantidade de parâmetros do modelo.
% x: Submatriz com variáveis explicativas originais.
% Retorna a matriz de variáveis explicativas e a informação sobre erro.
function [MatX, info] = matriz_explicativas(n, v, p, x)
    if p < v % Modelo não permitido.
        info = 1; % Erro.
        MatX = []; % Erro.
        return;
    endif

    info = 0;
    if v == 1 % Regressão polinomial.
        for i = (1: 1: n)
            MatX(i, 1) = 1;
        endfor

        for j = (2: 1: p)
            for i = (1: 1: n)
                MatX(i, j) = MatX(i, j - 1) * x(i, 1);
            endfor
        endfor
    else % Regressão linear múltipla.
        for i = (1: 1: n)
            MatX(i, 1) = 1;
        endfor

        for j = (2: 1: p)
            for i = (1: 1: n)
                MatX(i, j) = x(i, j - 1);
            endfor
        endfor
    endif
endfunction
```

Figura 1: Implementação da geração da matriz explicativa.

## 1.2 Regressão linear múltipla e polinomial

Segue abaixo implementação: Obs: Não coloquei foto das funções auxiliares de eliminação de Gauss com pivotação parcial e substituições retroativas pois elas já foram mostradas no trabalho 1. Contudo, os arquivos .m referentes a estes algoritmos foram enviados no classroom.

```
% Calcula os parâmetros do modelo de regressão linear.
% n: Número de pontos.
% v: Quantidade de variáveis explicativas.
% p: Quantidade de parâmetros do modelo.
% x: Variáveis explicativas originais.
% y: Variáveis respostas.
% Retorna o vetor solução b com os parâmetros do modelo, o coeficiente de determinação r2
% e a informamação sobre erro. Caso ocorra erro, convencionei b ser vazio e r2 igual a -1.
function [b, r2, s2, info] = reg_linear_mult(n, v, p, x, y)
    [MatX, info] = matriz_explicativas(n, v, p, x)

    if info != 0 % Erro.
        b = [] % Erro.
        r2 = -1 % Erro.
        return
    endif

    for i = (1: 1: p)
        for j = (1: 1: i)
            soma = 0;
            for k = (1: 1: n)
                soma = soma + MatX(k, i) * MatX(k, j);
            endfor

            Sxx(i, j) = soma;

            if i != j
                Sxx(j, i) = soma;
            endif

            soma = 0;
            for k = (1: 1: n)
                soma = soma + MatX(k, i) * y(k);
            endfor

            Sxy(i) = soma; % Vetor dos termos independentes.

        endfor
    endfor

    [U, d] = elim_gauss_pivotacao(p, Sxx, Sxy);
    b = subs_retroativas(p, U, d);

    info = 0;
    S = 0;
    Sy2 = 0;

    for i = (1: 1: n)
        u = 0;
        for j = (1: 1: p)
            u = u + MatX(i, j) * b(j);
        endfor

        S = S + (y(i) - u) * (y(i) - u);
        Sy2 = Sy2 + y(i) * y(i);
    endfor

    r2 = 1 - S / (Sy2 - Sxy(1) * Sxy(1) / n);
    s2 = S / (n - p);
endfunction
```

Figura 2: Implementação da resolução de regressão linear múltipla e polinomial.

# 2 Impressões da janela de comandos do Octave

## 2.1 Problema 1: Regressão Linear Múltipla

1) $\hat{y}(x_1, x_2) = b_0 + b_1 x_1 + b_2 x_2$ (Campos, 2018; Exercício 4.6, Página 247)

| $i$ | $x_{i1}$ | $x_{i2}$ | $y$ |
|---|---|---|---|
| 1 | -1 | -2 | 13 |
| 2 | 0 | -1 | 11 |
| 3 | 1 | 0 | 9 |
| 4 | 2 | 1 | 4 |
| 5 | 4 | 1 | 11 |
| 6 | 5 | 2 | 9 |
| 7 | 5 | 3 | 1 |
| 8 | 6 | 4 | -1 |

Figura 3: Primeiro problema proposto.

```
octave:32> n = 8;
octave:33> v = 2;
octave:34> p = 3;
octave:35> x = [-1 -2; 0 -1; 1 0; 2 1; 4 1; 5 2; 5 3; 6 4]
x =

  -1  -2
   0  -1
   1   0
   2   1
   4   1
   5   2
   5   3
   6   4

octave:36> y = [13; 11; 9; 4; 11; 9; 1; -1]
y =

   13
   11
    9
    4
   11
    9
    1
   -1

octave:37> [b, r2, s2, info] = reg_linear_mult(n, v, p, x, y)
MatX =

   1  -1  -2
   1   0  -1
   1   1   0
   1   2   1
   1   4   1
   1   5   2
   1   5   3
   1   6   4

info = 0
b =

   4.2393
   3.4000
  -6.4643

r2 = 0.9771
s2 = 0.8479
info = 0
octave:38>
```

Figura 4: Linha de comando do Octave na resolução do primeiro problema.

## 2.2 Problema 2: Regressão Linear Polinomial

2) $\hat{y}(x) = b_0 + b_1 x + b_2 x^2$ (Campos, 2018; Exercício 4.7, Página 247)

| $i$ | $x_1$ | $y$ |
|---|---|---|
| 1 | -2,0 | -30,5 |
| 2 | -1,5 | -20,2 |
| 3 | 0,0 | -3,3 |
| 4 | 1,0 | 8,9 |
| 5 | 2,2 | 16,8 |
| 6 | 3,1 | 21,4 |

Figura 5: Segundo problema proposto.

```
octave:39> n = 6;
octave:40> v = 1;
octave:41> p = 3;
octave:42> x = [-2; -1.5; 0; 1; 2.2; 3.1]
x =

  -2.0000
  -1.5000
        0
   1.0000
   2.2000
   3.1000

octave:43> y = [-30.5; -20.2; -3.3; 8.9; 16.8; 21.4]
y =

  -30.5000
  -20.2000
   -3.3000
    8.9000
   16.8000
   21.4000

octave:44> [b, r2, s2, info] = reg_linear_mult(n, v, p, x, y)
MatX =

   1.0000  -2.0000   4.0000
   1.0000  -1.5000   2.2500
   1.0000        0        0
   1.0000   1.0000   1.0000
   1.0000   2.2000   4.8400
   1.0000   3.1000   9.6100

info = 0
b =

   -2.0177
   11.3315
   -1.2222

r2 = 0.9974
s2 = 1.8841
info = 0
octave:45> _
```

Figura 6: Linha de comando do Octave na resolução do segundo problema.

# 3 Resultados do ajuste de curvas pelo método dos quadrados mínimos para os modelos 1 e 2

| Modelo | Parâmetros do Modelo | Coeficiente de Determinação |
|---|---|---|
| **1** | **b0** = 4.2393<br>**b1** = 3.4<br>**b2** = -6.4643 | **r²** = 0.9771 |
| **2** | **b0** = -2.0177<br>**b1** = 11.3315<br>**b2** = -1.2222 | **r²** = 0.9974 |

Figura 7: Tabela de resultados dos problemas 1 e 2.