**Problemas em Equipe 06**

Estudantes:

Eduardo Eiji Goto; Gustavo Hammerschmidt; João Vitor Andrioli de Souza.

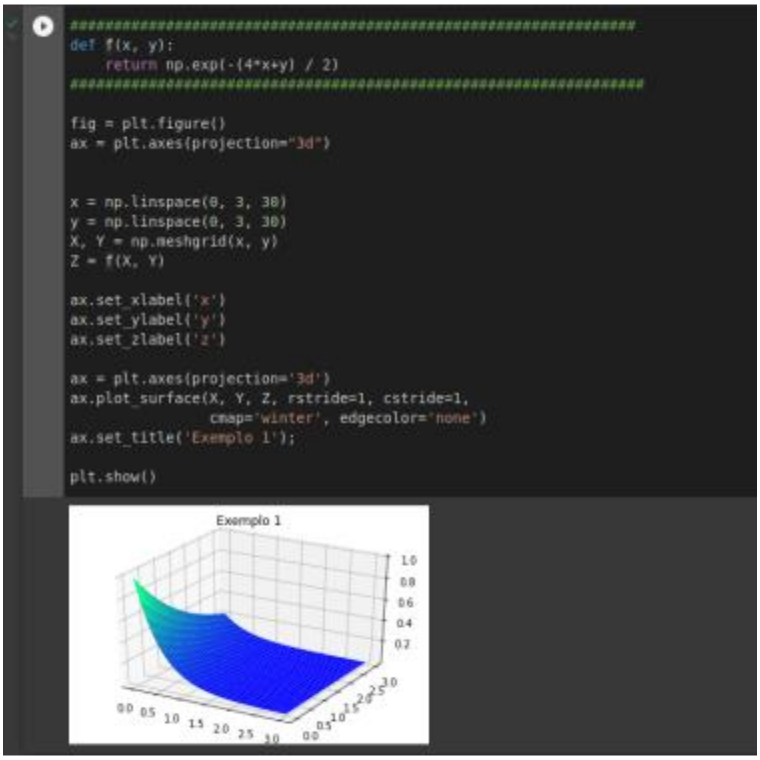
## 1 – Distribuições Multivariadas Contínuas

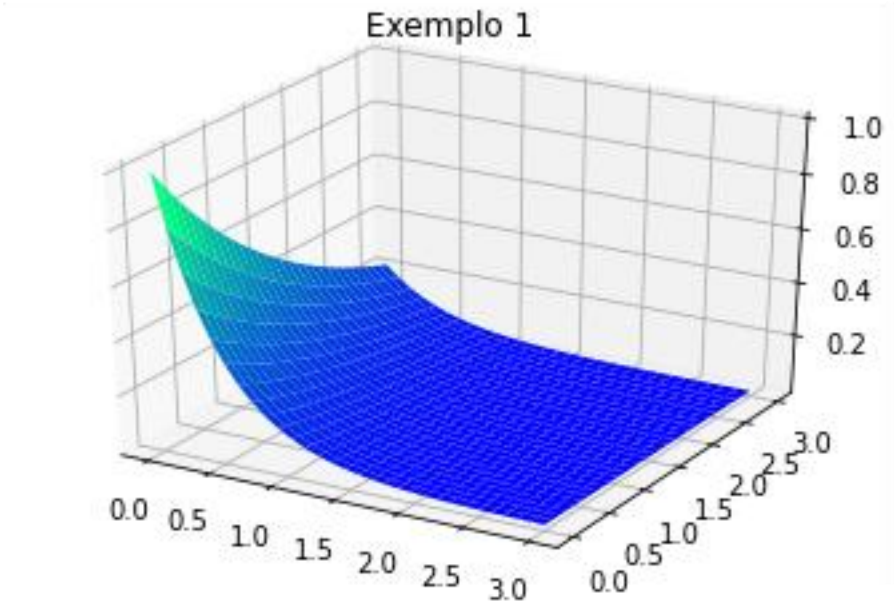
1. Gráfícos 3D de funções de duas variáveis
   O notebook Jupyter Plot3D.ipynb contém um mini tutorial para plotar gráficos em 3D com o MatPlotLib. Você aprenderá como as funções mostradas em sala de aula são plotadas em 3D, incluído a Gaussiana Bivariada Padrão. Você também pode usar os arquivos Plot3D.py e GaussianaBivariadaPadrao.py.

2. A função de densidade conjunta de duas variáveis aleatórias $X$ e $Y$ é dada por:

$$f_{X,Y}(x,y) = \begin{cases} e^{\frac{-(4x+y)}{2}} & \begin{matrix} 0 \le x \le \infty \\ 0 \le y \le \infty \end{matrix} \end{cases}$$

a) Plotar o gráfico de $f_{X,Y}(x, y)$
   Veja as dicas em Plot3D.ipynb e copiar o gráfico aqui






b) Encontrar $f_X(x)$ e $f_Y(y)$

Para encontrar $f_X(x)$ integrar $f_{X,Y}(x, y)$ em $y$

$$f_X(x) = \int_0^{\infty} f_{X,Y}(x, y)\, dy = 2e^{-2x}$$

(usar o comando abixo no Wolfram e copiar o resultado)

integrate[exp[-(4*x+y)/2]] {y from 0 to infinity}

$$\int_0^{\infty} \exp\left(-\frac{1}{2}(4x + y)\right) dy = 2e^{-2x}$$

Series expansion of the integral at x=0

$$2 - 4x + 4x^2 - \frac{8x^3}{3} + \frac{4x^4}{3} + O(x^5)$$

(Taylor series)

$$\int \exp\left(-\frac{1}{2}(4x + y)\right) dy = -2e^{-2x-y/2} + \text{constant}$$

Para encontrar $f_Y(y)$ integrar $f_{X,Y}(x, y)$ em $x$

$$f_Y(y) = \int_0^{\infty} f_{X,Y}(x, y)\, dx = \frac{1}{2}e^{--\frac{y}{2}}$$

(usar o comando abixo no Wolfram e copiar o resultado)

integrate[exp[-(4*x+y)/2]] {x from 0 to infinity}

$$\int_0^{\infty} \exp\left(-\frac{1}{2}(4x + y)\right) dx = \frac{e^{-y/2}}{2}$$

Series expansion of the integral at y=0

$$\frac{1}{2} - \frac{y}{4} + \frac{y^2}{16} - \frac{y^3}{96} + \frac{y^4}{768} + O(y^5)$$

(Taylor series)

$$\int \exp\left(-\frac{1}{2}(4x + y)\right) dx = -\frac{1}{2} e^{-2x - y/2} + \text{constant}$$

c) Calcular $E[X,Y]$

$$E[X,Y] = \begin{bmatrix} E[X] \\ E[Y] \end{bmatrix}$$

$$E[X] = \int_0^\infty x \cdot f_X(x)\, dx$$

(usar o comando abaixo no Wolfram e copiar o resultado)
integrate[x*2*exp(-2*x)] {x from 0 to infinity}

$$\int_0^\infty x\, 2 \exp(-2x)\, dx = \frac{1}{2} = 0.5$$

$$\int x\, 2 \exp(-2x)\, dx = 2 e^{-2x} \left(-\frac{x}{2} - \frac{1}{4}\right) + \text{constant}$$

$$E[Y] = \int_0^\infty y \cdot f_Y(y)\, dy$$

(usar o comando abaixo no Wolfram e copiar o resultado)
integrate[y*1/2*exp(-y/2)] {y from 0 to infinity}

$$\int_0^\infty \frac{1}{2} y \exp\left(-\frac{y}{2}\right) dy = 2$$

$$\int \frac{1}{2} y \exp\left(-\frac{y}{2}\right) dy = \frac{1}{2} e^{-y/2} (-2y - 4) + \text{constant}$$

$$E[X,Y] = \begin{bmatrix} 0.5 \\ 2 \end{bmatrix}$$

d) Verificar se as duas variáveis são independentes.

$$f_X(x) \cdot f_X(x) = 2e^{-2x} \cdot \frac{1}{2} e^{-\frac{y}{2}} = e^{-\frac{(4x+y)}{2}} = f_{X,Y}(x, y)$$

Portanto as variáveis são independentes.

Exercício já está resolvido. Apenas para reforçar o método de veificação de independência de distribuições contínuas.

## 2 – Detecção de Outliers

Usar um modelo baseado na Gaussiana Multivariada para detectar observações anômalas. O exercício é dividido em duas partes. Na primeira, ajustar a distribuição Gaussiana Multivariada para um dataset com duas características (bidimensional), o que permite visualizar os dados e os resultados, para então encontrar valores com baixa probabilidade que serão classificados como anomalias. Na segunda parte, aplicar o algoritmo de detecção de anomalia para um dataset com maior número de dimensões.

Utilizar o Livescript DetAnomalia.

a) Modificar o arquvo estimarGaussiana.m e copiar o código modificado aqui.

```
function [mu, sigma] = estimarGaussiana(X)

    n = size(X, 2);
    mu = mean(X);
    sigma = cov(X);

end
```

Obs.: no arquivo .mlx, eu modifiquei a chamada da função para: estimarGaussiana(...); sem o Resposta no nome.

b) Modificar o arquivo selecionarLimiar.m e copiar o código modificado aqui.

```
function [melhorEpsilon, melhorF1] = selecionarLimiar(yval, pval)

melhorEpsilon = 0;
melhorF1 = 0;
F1 = 0;

passo = (max(pval) - min(pval)) / 1000;

for epsilon = min(pval):passo:max(pval)

    prev = (pval < epsilon);

    vp = sum((prev==1) & (yval==1));
    fp = sum((prev==1) & (yval==0));
    fn = sum((prev==0) & (yval==1));

    prec = vp/(vp+fp);
    rec = vp/(vp+fn);
    F1 = (2*prec*rec) / (prec+rec);
```

```
        if F1 > melhorF1
            melhorF1 = F1;
            melhorEpsilon = epsilon;
        end

    end

end
```

Obs.: no arquivo .mlx, eu modifiquei a chamada da função para: selecionarLimiar(...); sem o Resposta no nome.

c) Copiar os valores de acuracidade, e das métricas prec, rec e F1

```
acuracidade = 0.1175
prec = 0.1153
rec = 1
F1 = 0.2067
```