

O asterisco `''` logo depois de `\section` serve para evitar exibir o número antes do nome da seção.

Texto (sem número)

1 Texto (com número)

Aí Jhow quê içu donde tu vai com essa arma na tua mão meu parça ß Veja que esse texto é do tipo 'inline'. Já esse texto começa em uma nova linha como parágrafo.

Esse texto sofreu um 'line break' e está sem indentação.

Esse texto possui indentação por causa de `\par`.

Mudando a Fonte

Utilizando a fonte Times.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Fórmulas Matemáticas

Os 3 tipos de comandos que podemos usar para exibir fórmulas matemáticas.

1. Formato 'display' clássico:

$$F = G \left(\frac{m_1 m_2}{r^2} \right)$$

```
\[
F = G \left( \frac{m_1 m_2}{r^2} \right)
\]
```

2. Equivalente ao primeiro modo, mas é mais difícil de passar despercebido:

$$F = G \left(\frac{m_1 m_2}{r^2} \right)$$

```
\begin{displaymath}
F = G \left( \frac{m_1 m_2}{r^2} \right)
\end{displaymath}
```

3. A forma abaixo utilizando `\begin` enumera as fórmulas automaticamente:

$$F = G \left(\frac{m_1 m_2}{r^2} \right) \tag{1}$$

```
\begin{equation}
F = G \left( \frac{m_1 m_2}{r^2} \right)
\end{equation}
```

Ordered, Unordered, Nested list

Ordered

1. The labels consists of sequential numbers.
2. The numbers starts at 1 with every call to the `enumerate` environment.

Unordered

- The individual entries are indicated with a black dot, a so-called bullet.
 - The text in the entries may be of any length.
-

Nested

1. The labels consists of sequential numbers.
 - The individual entries are indicated with a black dot, a so-called bullet.
 - The text in the entries may be of any length.
2. The numbers starts at 1 with every call to the enumerate environment.

Colando códigos

Using verbatim

Usando `\beginverbatim` para colar pedaços de códigos:

```
<html>
<head> </head>
<body>
<h1> Hello World</h1>
</body>
</html>
```

Using listings to highlight code

```
import numpy as np

def incmatrix(genl1, genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j]):
                for k in range(len(r)):
                    VT[(i)*n + r[k]] = 1;
                    VT[(i)*n + c[k]] = 1;
                    VT[(j)*n + r[k]] = 1;
                    VT[(j)*n + c[k]] = 1;

    if M is None:
        M = np.copy(VT)
    else:
        M = np.concatenate((M, VT), 1)

    VT = np.zeros((n*m,1), int)

    return M
```

Using minted

```
var http = require("http");
http.createServer(function(request, response) {
    response.writeHead(200, {"Content-Type": "text/plain"});
    response.end();
}).listen(8080);
```

```
import numpy as np

def incmatrix(genl1, genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m, 1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2), 1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r, c] = np.where(M2 == M1[i, j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;

    if M is None:
        M = np.copy(VT)
    else:
        M = np.concatenate((M, VT), 1)

    VT = np.zeros((n*m, 1), int)

    return M
```

```
var http = require("http");
http.createServer(function(request, response) {
    response.writeHead(200, {"Content-Type": "text/plain"});
    response.end();
}).listen(8080);
```

Colando imagens

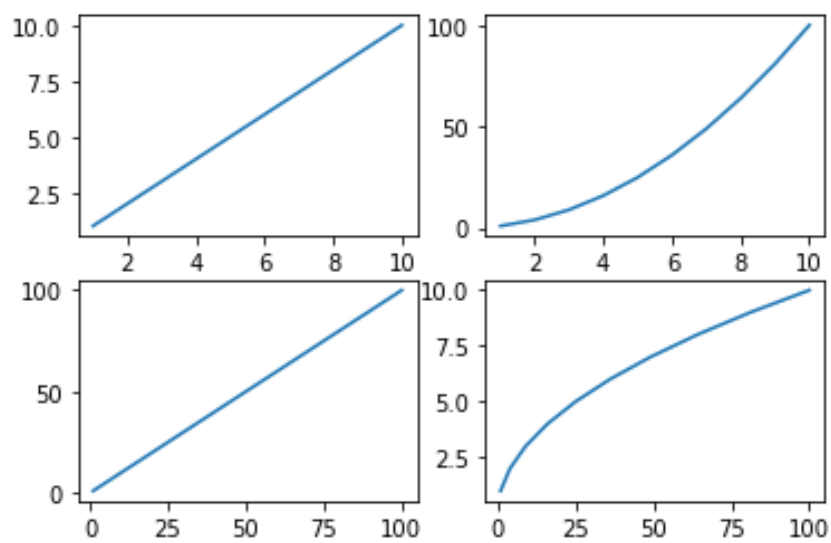


Figura 1: Gráfico de linhas

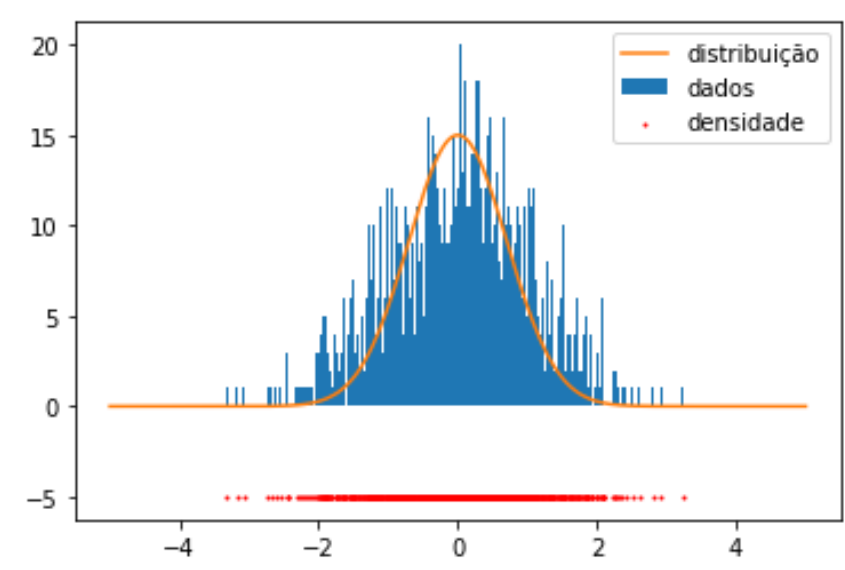


Figura 2: Gráfico de distribuição normal

Insira um texto qualquer aqui só para teste. Insira um texto qualquer aqui só para teste Insira um texto qualquer aqui só para teste Insira um texto qualquer aqui só para teste Insira um texto qualquer aqui só para teste.

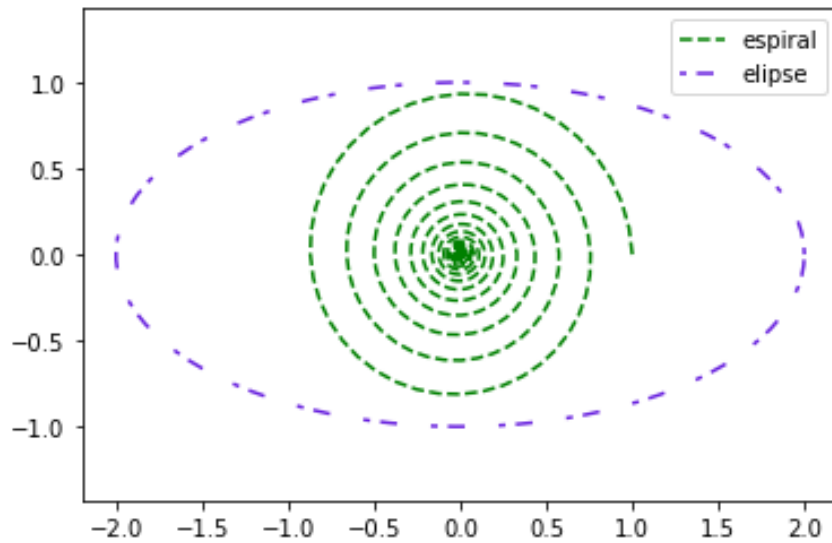


Figura 3: Espiral insano

Criando comandos

One way of save some time when writing too many expressions with exponents is by defining a new command to make simpler:

```
\newcommand{\plusbinomial}[3][2]{(#2 + #3)^#1}
```

Assim:

```
\[ \plusbinomial{x}{y} \]
```

$$(x + y)^2$$

And even the exponent can be changed:

```
\[ \plusbinomial[4]{a}{b} \]
```

$$(a + b)^4$$

Explaining:

- `\plusbinomial`: This is the name of the new command.
- `[3]`: The number of parameters the command will take, in this case 3.
- `[2]`: Is the default value for the first parameter. This is what makes the first parameter optional, if not passed it will use this default value.