

Insper

Wolfram Language

Computação para Ciências dos Dados

PÓS-GRADUAÇÃO EM DATA SCIENCE E DECISÃO

Prof. Daniel Carvalho @danielscarvalho

Dica do Dia: 010

Este é o ambiente notebook (UI) da Wolfram, criado em 1988 no Mathematica 1.0 para realização de computação científica exploratória, versão atual 12.1.

Use a combinação das teclas Shift + Enter para processar a célula de input.

Treinamento básico:

```
In[1]:= u = 10/2 + 1/9 - 4
```

```
Out[1]=  $\frac{10}{9}$ 
```

```
N[u, 20]
```

```
Out[1]= 1.111111111111111111111111
```

O kernel do sistema é simbólico, a maioria das linguagens de programação são numéricas.

Note abaixo que o valor de x não foi definido, mas o sistema é capaz de processar!

```
In[2]:= 1/(2 x) * 1/Sqrt[x]
```

```
Out[2]=  $\frac{1}{2 x^{3/2}}$ 
```

A Wolfram Language trabalha com precisão arbitrária não depende do tamanho da palavra do SO (32, 64 bits...) ou capacidade da variável numérica.

```
In[3]:= N[π, 100]
```

```
Out[3]= 3.1415926535897932384626433832795028841971693993751058209749445923078164062862089 $\cdot$   
98628034825342117068
```

Vamos ver algumas funções:

In[]:= **Plus[10, 3]**

Out[]= 13

In[]:= **Plus[x, 5]**

Out[]= 5 + x

In[]:= **Range[10]**

Out[]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

Table[x, {x, -10, 10}]

Out[]= {-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

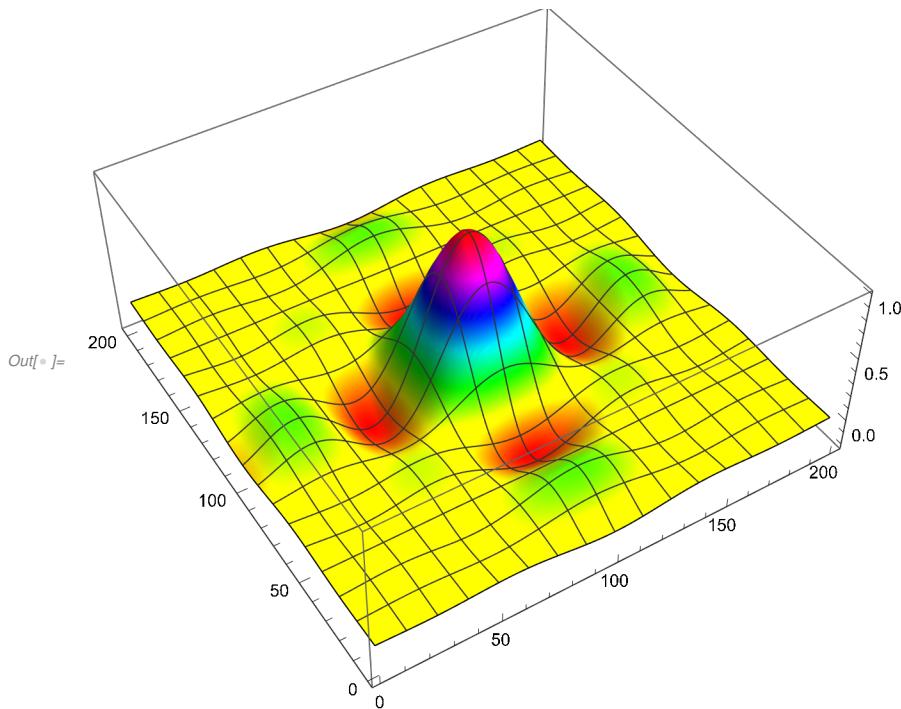
In[]:= **Solve[x^2 + 3 x + 2 == 0, x]**

Out[]= {{x → -2}, {x → -1}}

In[]:= **Table[Solve[x^2 + 3 x + 2 == z, x], {z, 0, 10}]**

Out[]= $\left\{ \left\{ \begin{array}{l} \{x \rightarrow -2\}, \{x \rightarrow -1\} \end{array} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{5}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{5}) \right\} \right\}, \left\{ \begin{array}{l} \{x \rightarrow -3\}, \{x \rightarrow 0\} \end{array} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{13}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{13}) \right\} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{17}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{17}) \right\} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{21}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{21}) \right\} \right\}, \left\{ \begin{array}{l} \{x \rightarrow -4\}, \{x \rightarrow 1\} \end{array} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{29}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{29}) \right\} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{33}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{33}) \right\} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{37}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{37}) \right\} \right\}, \left\{ \left\{ x \rightarrow \frac{1}{2}(-3 - \sqrt{41}) \right\}, \left\{ x \rightarrow \frac{1}{2}(-3 + \sqrt{41}) \right\} \right\} \right\}$

```
In[®] = ListPlot3D[Table[Sinc[x]*Sinc[y], {x, -10, 10, .1}, {y, -10, 10, .1}],
ColorFunction → Hue, PlotRange → All]
```



É a plataforma mais poderosa para resolver cálculos, recentemente todos os problemas dos principais livros acadêmicos de cálculos foram testados com sucesso:

```
Integrate[x ^ 2 + 3 x + Cos[x], x] // TraditionalForm
```

```
Out[®]//TraditionalForm=
```

$$\frac{3 x^2}{2} + x x^2 + \sin(x)$$

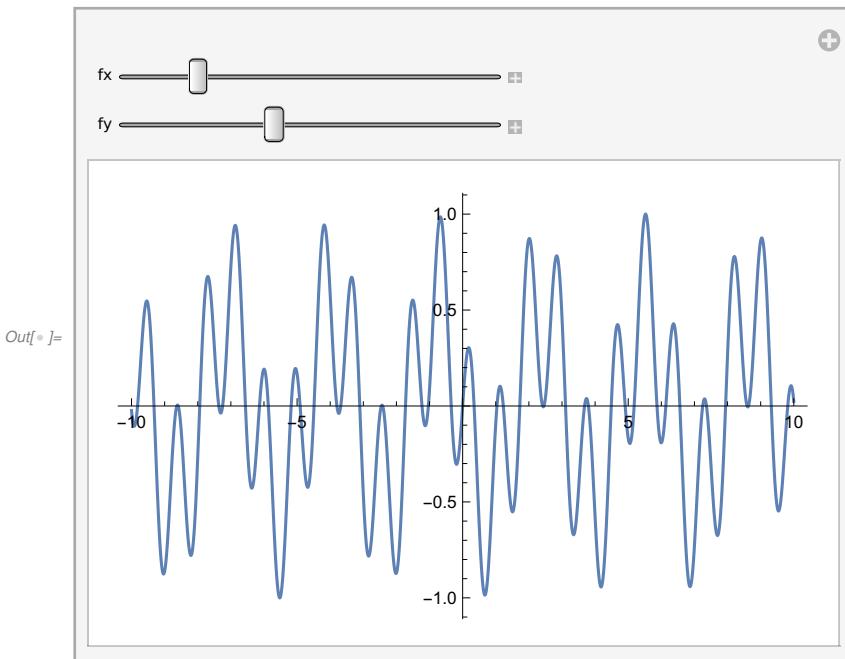
```
D[x ^ 2 + 3 x + Cos[x], x] // TraditionalForm
```

```
Out[®]//TraditionalForm=
```

$$3 - \sin(x)$$

Em uma linha de comando podemos criar uma interface:

```
Manipulate[Plot[Sin[a fx]*Cos[a fy], {a, -10, 10}],
{fx, 1, 10},
{fy, 1, 10}]
```



Como realizar uma transformada de fourier? Use a função Fourier, simples assim, a função tem o nome relacionado!

Pressione F1 para o help

In[1]:= ?Fourier*

System`

Fourier	FourierCosTransform	FourierDST	FourierSequenceTransform	FourierSinTransform
FourierCoefficient	FourierDCT	FourierDSTMatrix	FourierSeries	FourierTransform
FourierCosCoefficient	FourierDCTFilter	FourierMatrix	FourierSinCoefficient	FourierTrigSeries
FourierCosSeries	FourierDCTMatrix	FourierParameters	FourierSinSeries	

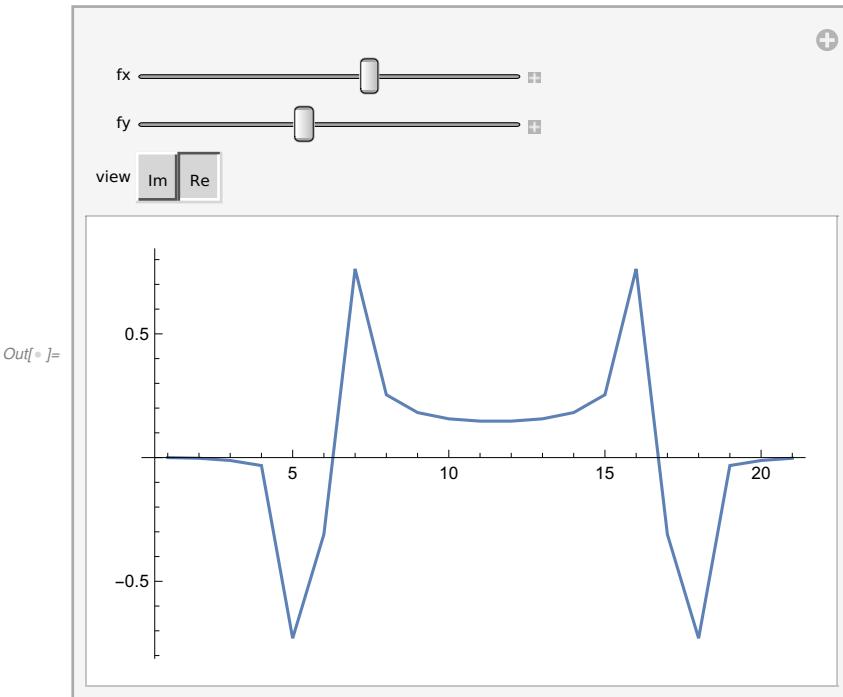
In[1]:= ?Fourier

Symbol i

Fourier[list] finds the discrete Fourier transform of a list of complex numbers.

Fourier[list, {p₁, p₂, ...}] returns the specified positions of the discrete Fourier transform.

```
In[1]:= Manipulate[
 ListLinePlot[view@Fourier[Table[Sin[a fx]*Cos[a fy], {a, -10, 10}]], PlotRange → All],
 {fx, 1, 10},
 {fy, 1, 10},
 {view, {Im, Re}}]
```



Pronto, treinamento acabou, agora vamos lá:

É possível realizar computação avançada neste ambiente, vamos iniciar com NLP, a plataforma tem dados acurados e computáveis.

Diversos sistemas populares utilizam estes recursos da Wolfram, mas você nem imagina!! :-)

Pressionando = no início de uma nova célula para realizar consultas em inglês, em linguagem natural
(Shift + Enter para processar)

E What is Sonia Braga age?

Out[1]= 69 yr 9 mo 28 days

E How many days the Brazil president is in charge?

Out[1]= Jair Bolsonaro (from January 1, 2019 to present)

E Distance from NYC to Paris

Out[1]= 3618.85 mi

E Distance from Diadema to Registro in kilometers

Out[⁶] = 153.273 km

In[⁷] := **E Whather in São Paulo now**

```
WeatherData[São Paulo CITY, "Temperature"]
```

Out[⁷] = 21 °C

E What is the boiling point of water?

Out[⁸] = 99.9839 °C

In[⁹] := **E What is the boiling point of water at the mount Everest**

↳ Result

```
70.41 °C
```

Out[⁹] = 70.41 °C

In[¹⁰] := **E Apple stock price**

```
Apple FINANCIAL ENTITY [ price ]
```

Out[¹⁰] = { Day: Fri 3 Apr 2020 , \$241.41 }

In[¹¹] := **E IBM Employees**

```
IBM COMPANY [ total employees ]
```

Out[¹¹] = 381 100 people

In[¹²] := **E Number of students in Brazil**

```
Brazil COUNTRY [ students ]
```

Out[¹²] = 4.68293×10^7 people

In[¹³] := **E Number of college students in Brazil**

```
Brazil COUNTRY [ students + post-secondary education ]
```

Out[¹³] = 6.92932×10^6 people

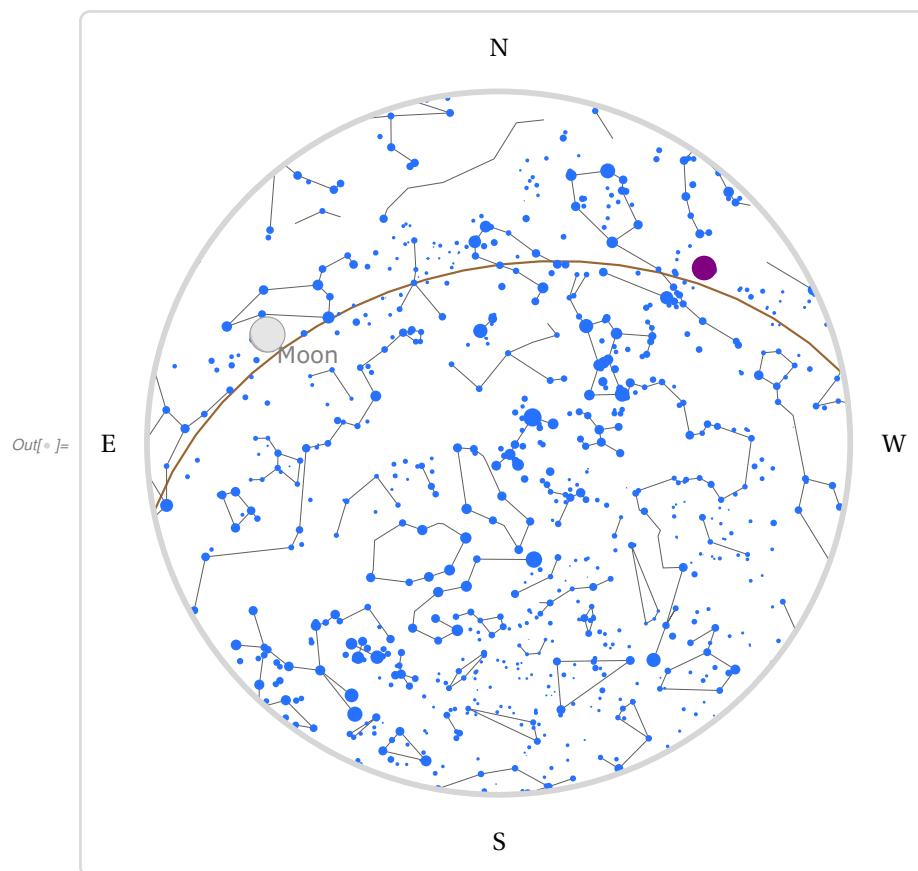
```
In[1]:= TextStructure["What is the boiling point of water at the mount Everest?"]
```

<i>Out[1]:=</i>	What	is	the	boiling	point	of	water	at	the	mount	Ever
	Wh-Pronoun	Verb	Determiner	Adjective	Noun	Preposition	Noun	Preposition	Determiner	Proper Noun	Proper
	Wh-Noun Phrase			Noun Phrase			Noun Phrase			Noun Phrase	
										Prepositional Phrase	
											Noun Phrase
											Prepositional Phrase
											Noun Phrase
											Clause

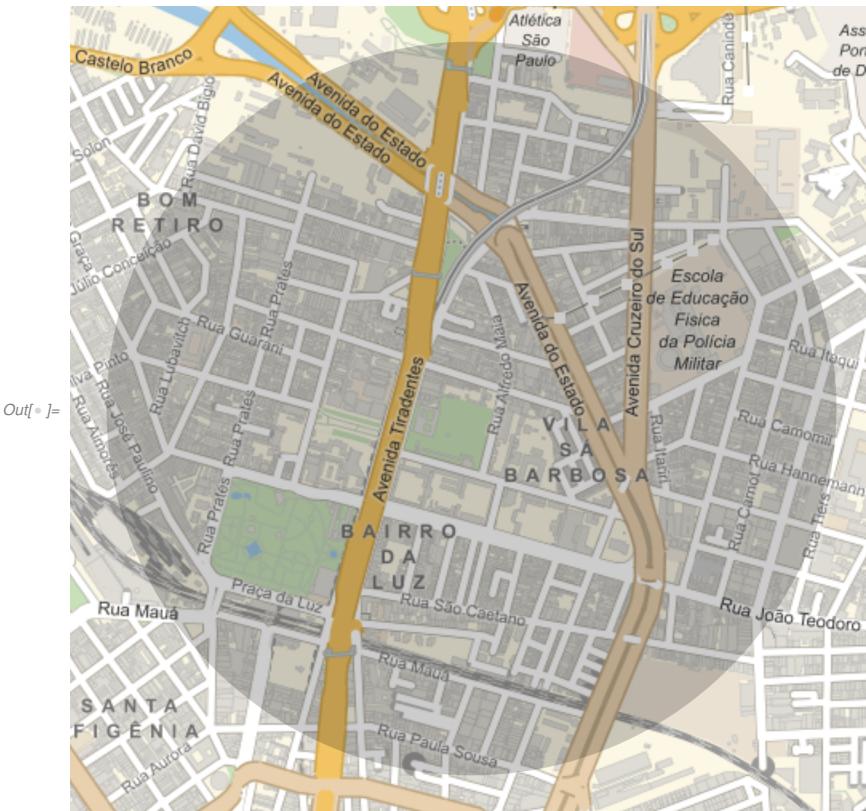
Como está o céu agora, onde estou?

```
In[2]:= Sky now > +
```

↳ Results (1 of 2)



In[⁶]:= **GeoGraphics[GeoDisk[**São Paulo CITY**, 1 km]]**



In[⁶]:= **Here**

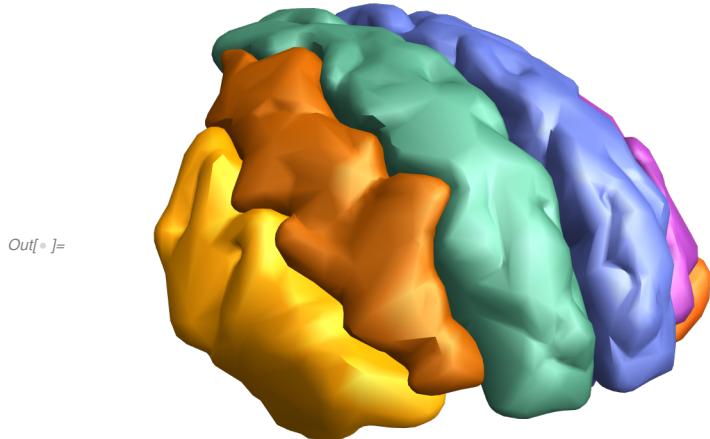
Out[⁶]= **GeoPosition[{-23.63, -46.63}]**



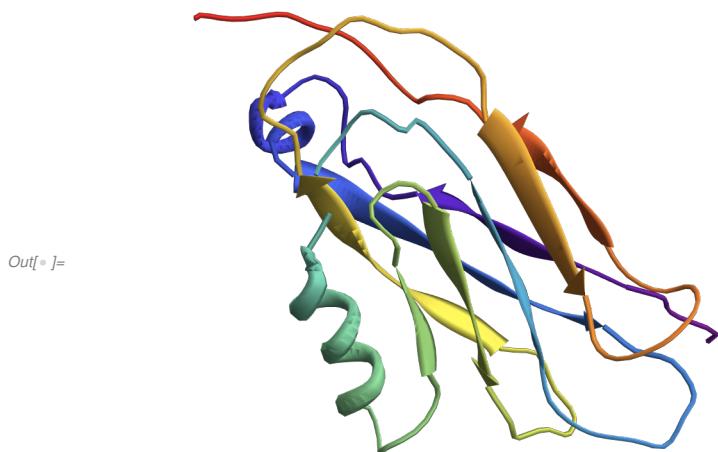
Out[⁶]= **GeoPosition[{-23.63, -46.63}]**

Há dados computáveis anatômicos, de química, humano, cachorro, gato, cavalo, genoma, medicamentos, doenças etc...

```
In[1]:= AnatomyPlot3D[Entity["AnatomicalStructure", "PrefrontalCortex"],  
"PlotTheme" → "Scientific"]
```

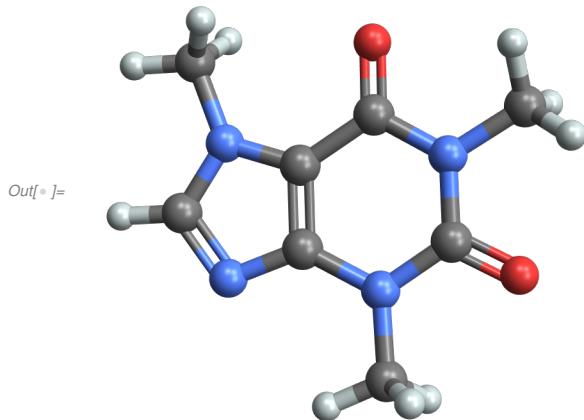


```
In[2]:= ProteinData["A2M", "MoleculePlot"]
```



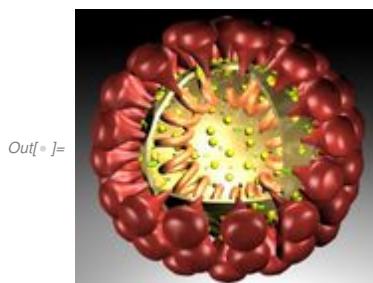
In[1]:= **3D caffeine molecule**

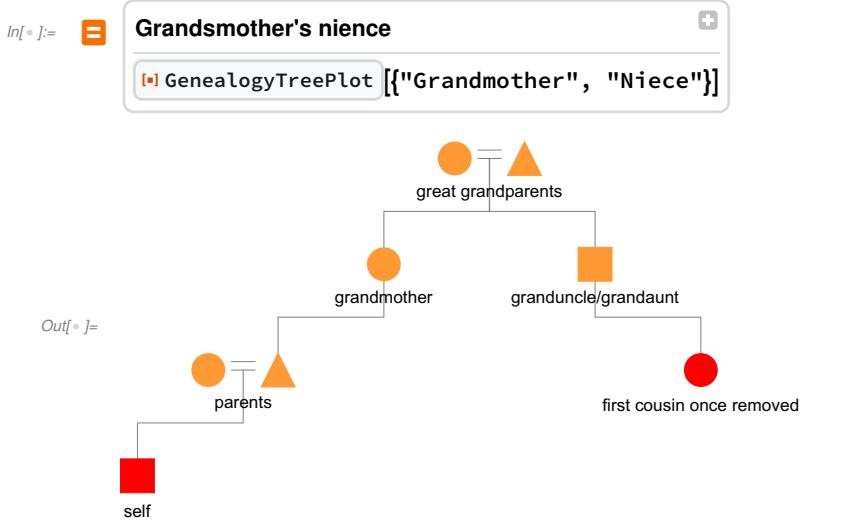
caffeine CHEMICAL [**molecule plot**]



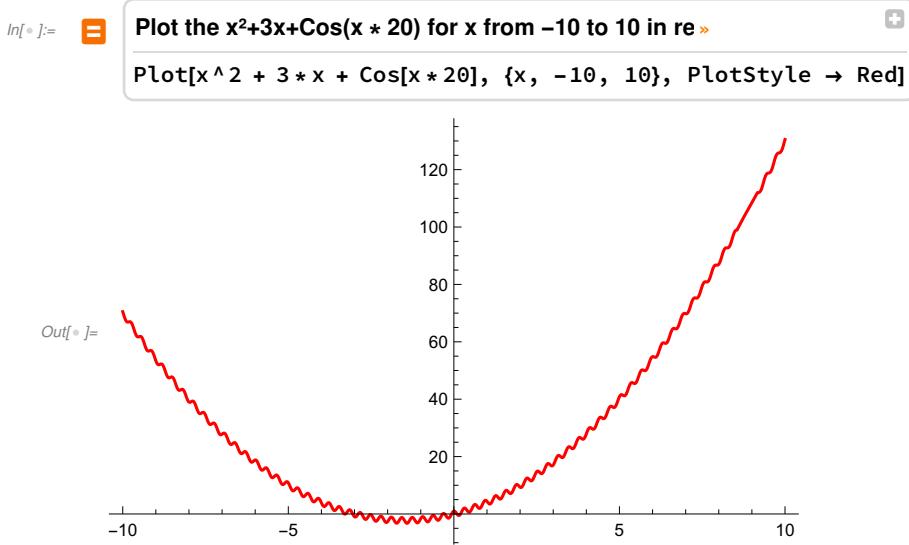
In[2]:= **Coronavirus image**

Coronavirus SPECIES SPECIFICATION [**image**]





Faça um pedido bem definido e o sistema transforma em código:



Há dados acurados computáveis como parte da plataforma, isso aliado a capacidade de processamento simbólico, e programação funcional, torna o ambiente ideal para realizar IA e ML:

```
In[6]:= FacialFeatures[] // Dataset
```

Image	Age	Gender	Emotion
	29	Female	happiness
	12	Male	happiness
	52	Male	happiness
	22	Male	happiness

```
In[1]:= ImageIdentify[
```

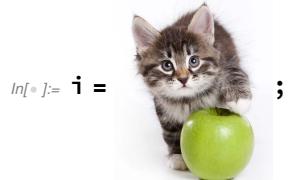
```
Out[•]= terrier
```

In[1]:= WordCloud[%]

ÅBÆ Å



Â Ç Ă Ă Ă Ă Ă Ă Ă Ă



In[3]:= ImageBoundingBoxes[i]

Out[3]:= <| domestic cat → {Rectangle[{50.0503, 48.6728}, {381.29, 504.017}]}|>

```
In[0] = HighlightImage[i, %]
```

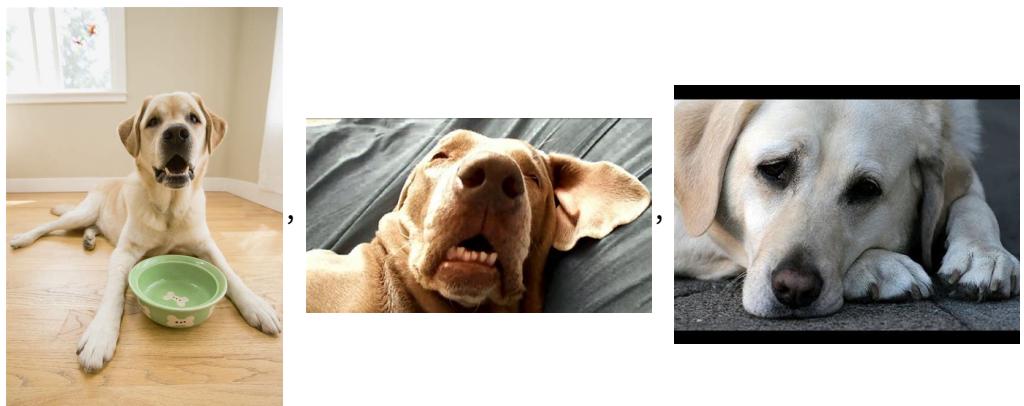
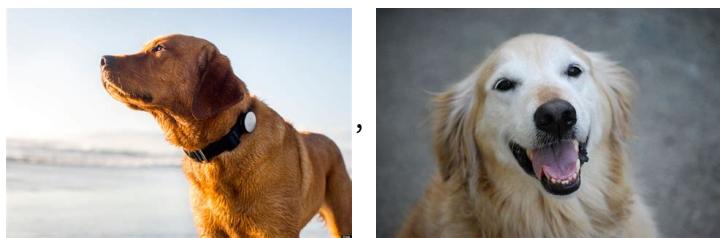
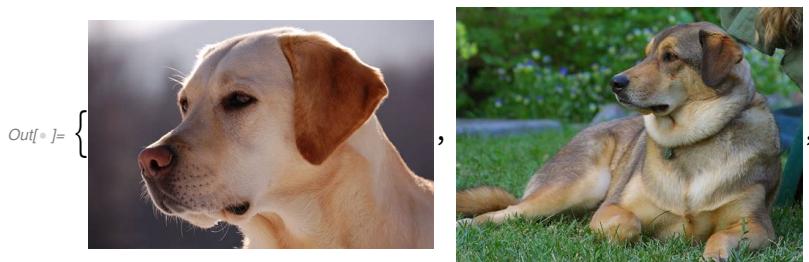
Out[⁰] =



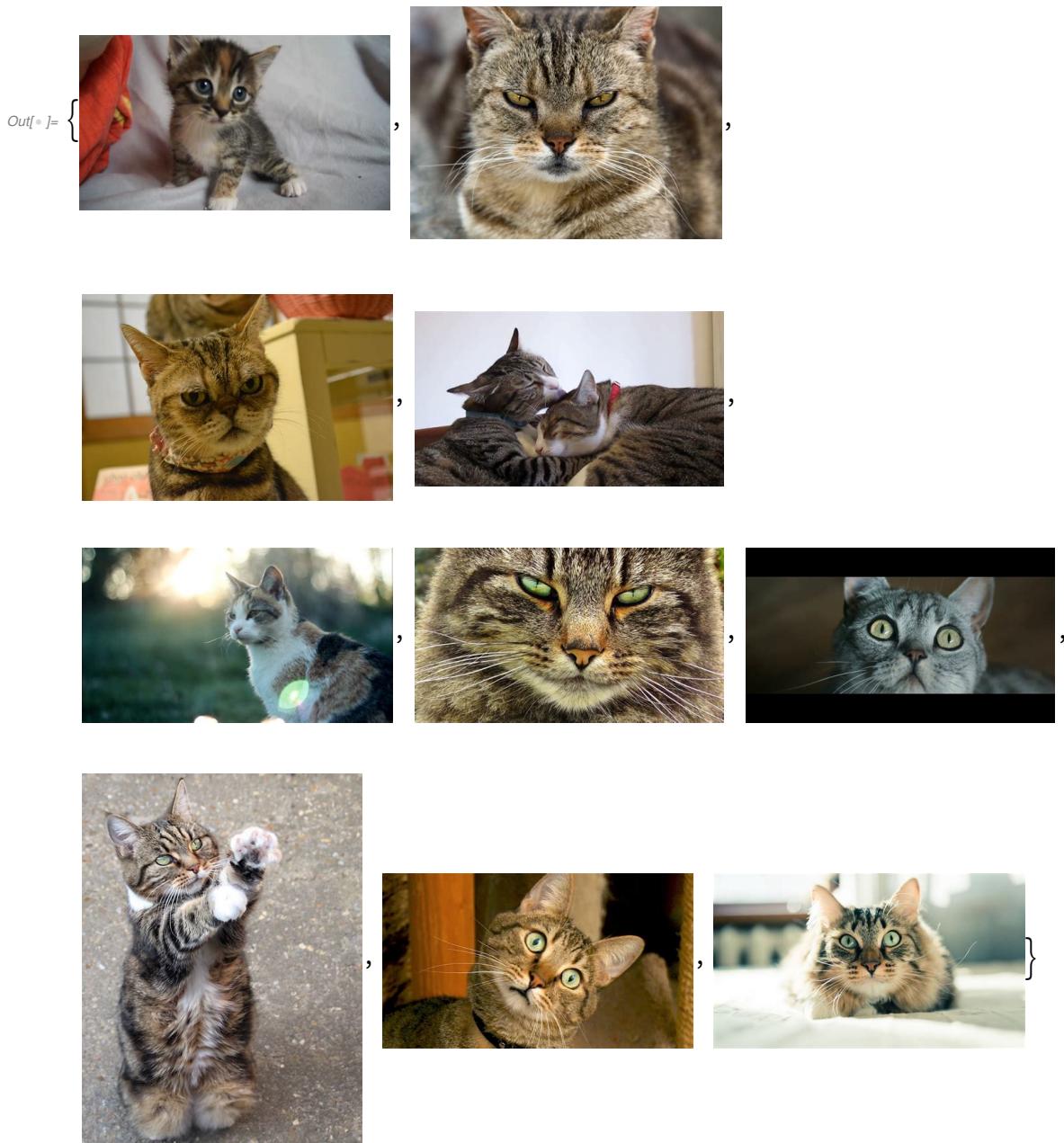
Ops! Cade a maçã! Ver o help da função `ImageBoundingBoxes`...

Vamos buscar imagens na Internet para treinamento:

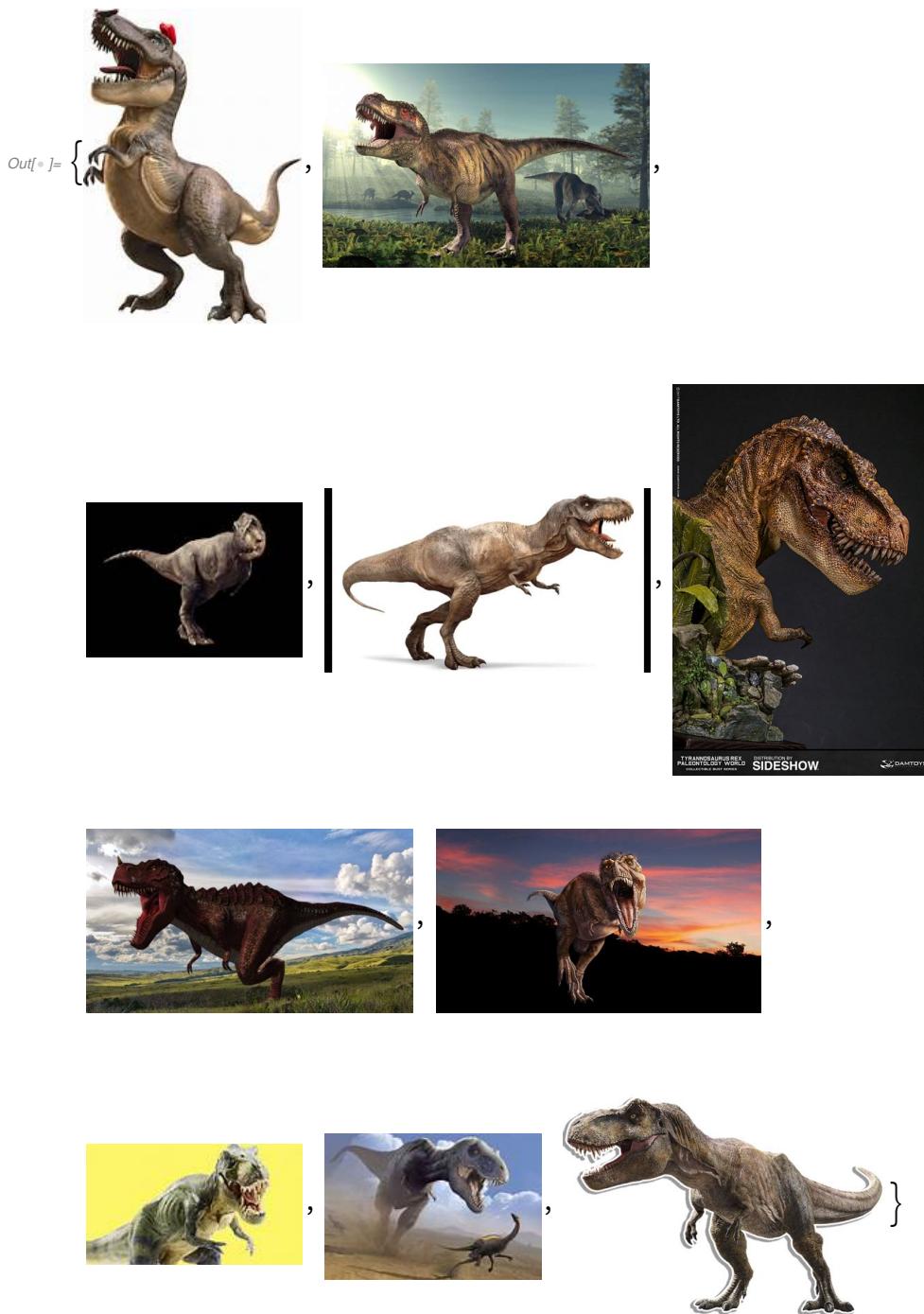
```
In[1]:= dog = WebImageSearch["dog", "Thumbnails"]
```



```
In[1]:= cat = WebImageSearch["cat", "Thumbnails"]
```



```
In[1]:= t rex = WebImageSearch["t-rex", "Thumbnails"]
```



Vamos observar as imagens em forma de cluster, note que as imagens são organizadas na arvore não o conteúdo (animais)...

Há diversas funções para realizar classificação, regressão, clusterização entre outros algoritmos que estão na moda.

```
In[①]:= ClusteringTree[Flatten[{cat, dog, trex}]]
```



Agora vamos criar e treinar um classificador, em uma linha de comando:

```
In[②]:= animal = Classify[<|
  "cão" → dog,
  "gato" → cat,
  "t-rex" → trex
  |>]
```

```
Out[②]= ClassifierFunction[ Input type: Image
Classes: cão, gato, t-rex
Method: LogisticRegression
Number of training examples: 30]
```

Data not in notebook. Store now 

Teste com imagem de um lagarto...

```
In[③]:= animal[

```

```
Out[③]= t-rex
```

O que está pegando no NYT hoje?

```
In[•]:= WordCloud[DeleteStopwords[Import["http://www.nyt.com"]]]
```



Função em Wolfram Language que implementei e foi publicada:

```
In[•]:= ResourceFunction["SequenceGraph"]
```

Out[•]= SequenceGraph +

In[•]:= Range[10]

[■] SequenceGraph + [%]

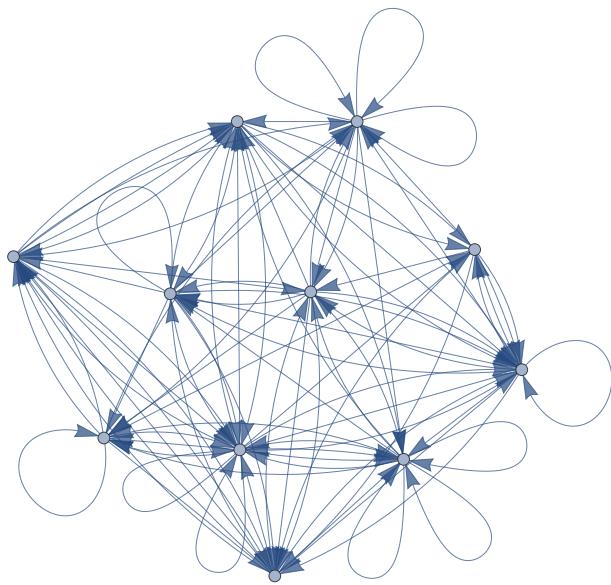
```
Out[•]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Outflow =

```
In[1]:= RandomInteger[10, 100]
```

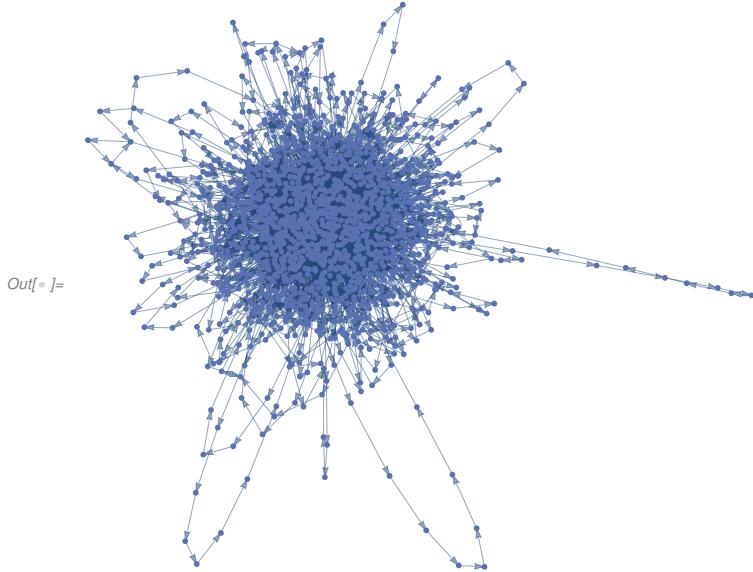
[SequenceGraph] [%]

```
Out[1]:= {2, 3, 6, 9, 1, 6, 0, 2, 2, 10, 6, 4, 7, 6, 2, 4, 3, 10, 0, 7, 0, 5, 2, 0,
3, 8, 7, 10, 5, 5, 9, 8, 0, 3, 2, 7, 0, 1, 4, 6, 2, 9, 7, 0, 3, 4, 5, 6, 10,
9, 7, 4, 7, 4, 8, 1, 4, 2, 8, 5, 6, 3, 2, 8, 8, 5, 5, 8, 3, 1, 3, 3, 5, 5, 0,
8, 8, 0, 1, 3, 8, 9, 4, 3, 1, 10, 10, 10, 10, 6, 4, 7, 8, 5, 9, 10, 7, 8, 9, 9}
```



Agora vamos gerar um grafo do livro “Alice no país das maravilhas”, em uma única linha de código, cada nó é uma palavra na ordem que aparece no livro.

```
In[1]:= ResourceFunction["SequenceGraph"][[StringSplit[
DeleteStopwords[ResourceData["Alice in Wonderland"]], RegularExpression["\\W+"]]]]
```



Usando outra função que publiquei que consome a API do DuckDuckGo:

```
In[2]:= ResourceFunction["DuckDuckGoQuery"]
```

Out[2]= DuckDuckGoQuery +

```
In[3]:=  DuckDuckGoQuery + ["Alan Turing"]["Image"]
```

Out[3]= <https://duckduckgo.com/i/a377e75e.jpg>

```
In[4]:= atImage = Import[%]
```



```
In[④]:= Classify["NotablePerson", atImage]
```

```
Out[④]:= Alan Turing
```

Alan Turing é uma Entity com semântica, ou seja, o sistema sabe o que é “Alan Turing” assim como “Brazil”, “São Paulo”, “Moon”, etc....:

```
In[⑤]:= Alan Turing PERSON ["Dataset"]
```

```
Out[⑤]:=
```



```
In[⑥]:= FindFaces[atImage, "Image"]
```

```
Out[⑥]:= {}
```

Processamento de imagens:

In[1]:= **ColorNegate[EdgeDetect[Blur[atImage, 5], 5]]**



Out[1]:=

In[2]:= **ImageIdentify[atImage]**

Out[2]:= person

Entity, um elemento que tem semântica... Ctrl = para definir uma entity...

Brazil COUNTRY

In[3]:= Brazil COUNTRY ["Flag"]



Out[3]:=

In[4]:= Brazil COUNTRY ["GDP"]

Out[4]:= $\$1.86863 \times 10^{12}$ per year

In[5]:= China COUNTRY [GDP + nominal] / Brazil COUNTRY [GDP + nominal]

Out[5]:= 7.28244

In[6]:= United States COUNTRY [dependencies]

Out[6]:= $\{$ American Samoa, Guam, Northern Mariana Islands,
Puerto Rico, United States Minor Outlying Islands, United States Virgin Islands $\}$

In[1]:= **Brazil** COUNTRY [dependencies]

Out[1]:= {}

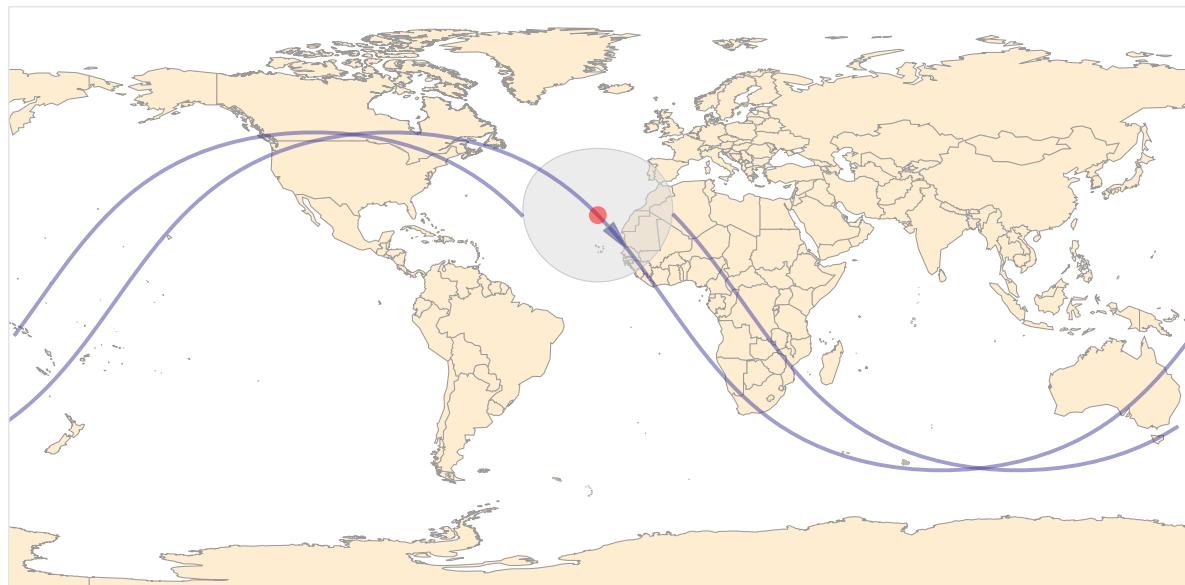
É possível definir “entities” junto a banco de dados da instituição. Desta forma as colunas do seu banco de dados relacional passam a ter propriedades semânticas e se integram ao contexto da Wolfram Language.

Para finalizar por hoje:

In[2]:= **Where is the ISS now?**

↳ Results (1 of 3)

Out[2]:=



In[3]:= **What is the meaning of life?**

↳ Result

Out[3]:= 42

(according to the book The Hitchhiker's Guide to the Galaxy, by Douglas Adams)

Agora vamos publicar este notebook na nuvem, gerar um short URL e um QRCode para acesso fácil:

In[4]:= **CloudPublish[]**

Ou no client (Mathematica) File, Publish to Cloud...

In[5]:= **URLShorten["www.wolframcloud.com/obj/dcarvalho/Published/Insper-Dica-010.nb"]**

Out[5]:= <https://wolfr.am/LBbZEwAP>

```
In[1]:= BarcodeImage[ "https://wolfr.am/LBbZEwAP" , "QR"]
```



```
Out[1]=
```

Vou também exportar como PDF...

Notou que não foram importadas bibliotecas externas...

Para saber mais:

- www.wolfram.com
- www.wolframalpha.com
- <https://www.wolfram.com/language/fast-introduction-for-programmers/en/>
- <https://www.wolfram.com/language/elementary-introduction/2nd-ed/>
- <https://www.wolfram.com/wolfram-u/>