

# R para *Machine Learning*



**Prof. Dr. Diego Bruno**

Education Tech Lead na DIO

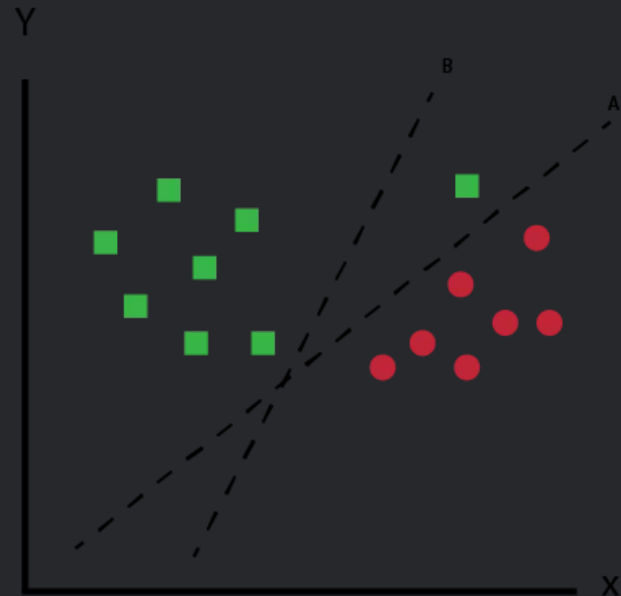
Doutor em Robótica e *Machine Learning* pelo ICMC-USP



# Vamos começar a programar...

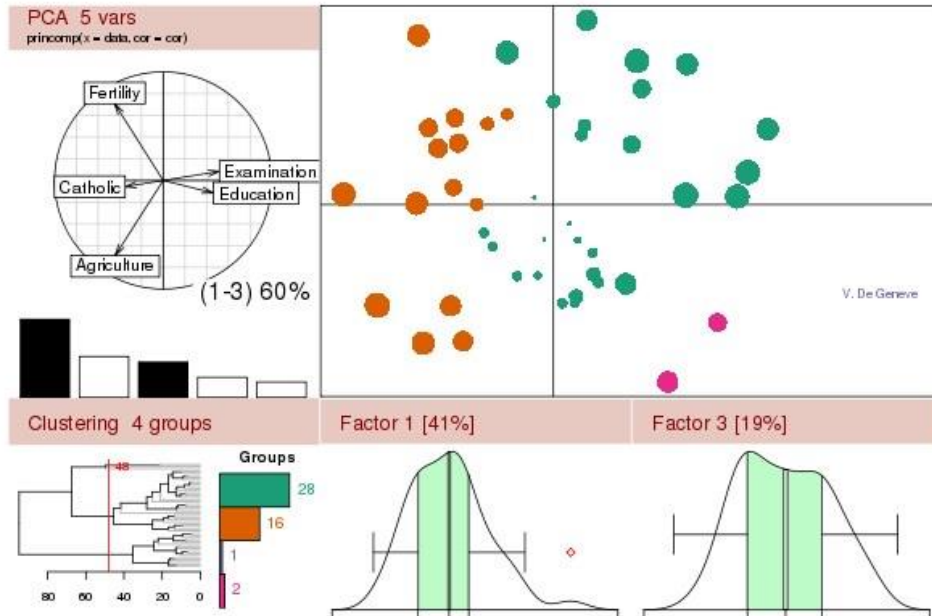
Prof. Dr. Diego Bruno

*Machine Learning*



# Linguagens de Programação

R é uma linguagem de programação multi-paradigma orientada a objetos, programação funcional, dinâmica, voltada à manipulação, análise e visualização de dados:

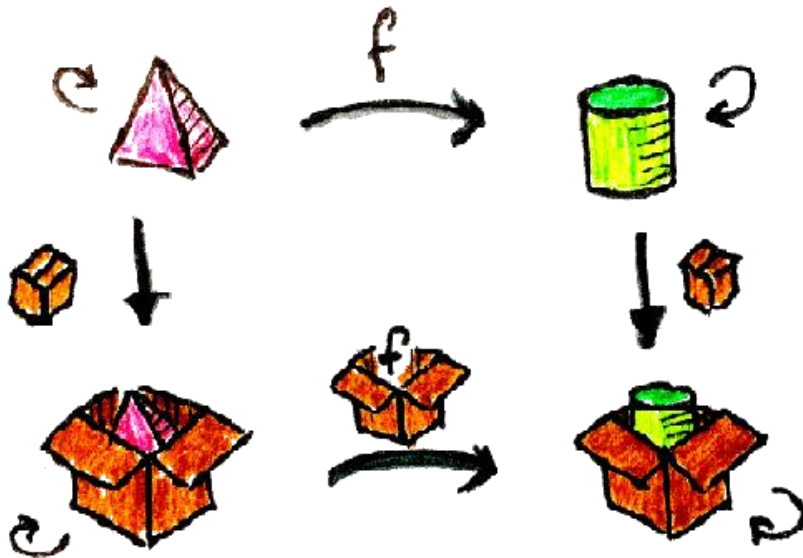


# Paradigmas de Programação



# Programação Funcional

Programação funcional é um paradigma de programação que trata a computação como uma avaliação de funções matemáticas.



$$2+2 \times 3 = ?$$

# Programação Funcional

Programação funcional é um paradigma de programação que trata a computação como uma avaliação de funções matemáticas.

## Scheme

```
1 ((lambda (x) (+ x x)) (* 3 4))
```

Nesse caso, seria isso que aconteceria:

$3 * 4 = 12;$

$x = 12;$

$x + x = 12 + 12 = 24;.$



Linguagem funcional



Suporte para funcional

# Programação em R



A versão base do R possui uma coleção enorme de funções:

- Modelos Estatísticos
- Algoritmos Computacionais
- Métodos Matemáticas
- Visualização de Dados

# Programação em R



## Pacotes:

Uma coleção de funções que podem ser escritas em R, C++, Fortran e C e que são chamadas diretamente de dentro do R.

Qualquer pessoa pode desenvolver seus pacotes e então submeter ao CRAN, disponibilizar através do *GitHub* ou *standalone*.



# Programação em R



Mas as vezes não é suficiente:

Assim como alguns softwares estatísticos, o R também é extensível através de "módulos". Em R estes módulos são chamados de pacotes, bibliotecas ou packages.

# Programação em R



As funcionalidades do R, podem ser ampliadas carregando estes pacotes, tornando um software ainda mais poderoso, capaz de realizar inúmeras tarefas:

- Análise multivariada;
- Análise Bayesiana;
- Manipulação de dados;
- Gráficos a nível de publicação;
- Big Data, Deep Learning;
- Processamento de imagens.

# Programação em R



## Alguns pacotes

- **maptools**: Funções para leitura, exportação e manipulação de estruturas espaciais.
- **cluster**: Funções para análise de clusters.
- **ggplot2**: Criação de gráficos elegantes.
- **rmarkdown**: criação de documentos (dinâmicos) em PDF, Word, HTML.
- **nlme**: Modelos lineares e não-lineares de efeitos mistos.

# Obrigado!

Prof. Dr. Diego Bruno

## *Machine Learning*

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense

class MyModel(Model):
    def __init__(self, hidden_units, outputs, **kwargs):
        super(MyModel, self).__init__(**kwargs)
        self.dense = Dense(hidden_units, activation='sigmoid')
        self.linear = LinearMap(hidden_units, outputs)

    def call(self, inputs):
        h = self.dense(inputs)
        return self.linear(h)

my_model = MyModel(64, 12, name='my_custom_model')
```