



# PUC-RIO

## INF1036 - Probabilidade Computacional

Material 1 - Introdução  
Professora - Ana Carolina Letichevsky\*  
2022.1

\*Material Adaptado de Professor Hélio Lopes

A Teoria das Probabilidades é uma área da matemática que desenvolve modelos que podem ser utilizados para estudar fenômenos aleatórios ou experimentos aleatórios.

No contexto de Teorias das Probabilidades os processos de observação são chamados de experimentos.

Experimentos aleatórios são aqueles que repetidos sobre as mesmas condições produzem geralmente resultados diferentes.

Em contraponto, um experimento é determinístico quando repetido em condições semelhantes produzem resultados exatamente iguais.

O modelo matemático utilizado para estudar um determinado fenômeno aleatório muda em sua complexidade de acordo com o fenômeno estudado.

# Variáveis Aleatórias



Dado um experimento aleatório ( $E$ ), descrito pelo espaço de probabilidades, uma função numérica

$X : S \rightarrow R$ , onde  $S$  é o conjunto de resultados possíveis

É uma variável aleatória do experimento.

Nos interessa estudar as distribuições de probabilidade das variáveis aleatórias.

# Processos Estocástico

Um processo estocástico é um modelo matemático utilizado para o estudo de fenômenos aleatórios que tem como resultados funções.

No contexto da teoria das probabilidades, um processo estocástico é uma família de variáveis aleatórias representando a evolução de um sistema de valores, em geral, com o tempo, mas pode ser a uma distância a um ponto fixo, um volume, etc.

Ao invés de um processo que possui um único modo de evoluir, em um processo estocástico, há uma indeterminação. Assim, mesmo que se conheça a condição inicial, existem diferentes, mesmo infinitas, direções nas quais o processo pode evoluir.



## Aplicações:

- Economia
- Atuária
- Engenharia
- Saúde
- Física
- Topografia
- Jogos
- Aprendizado de Máquinas

Uma simulação estocástica imita o comportamento de um determinado sistema buscando obter diferentes cenários das saídas desses sistemas.

Em uma simulação estocástica uma dada variável aleatória (ou família de variáveis aleatórias), cujo comportamento é representado matematicamente, pode ser reproduzida numericamente de modo que cada simulação tenha um resultado diferente.

Simular um grande número de cenários permite aproximar com precisão a distribuição de probabilidade de todas as saídas e essa precisão aumenta à medida que o número de cenários aumenta.

Os métodos de simulação estocástica são procedimentos que envolvem a geração de números aleatórios.

Dito isto, a geração de números aleatórios é o alicerce de qualquer sistema de simulação estocástica.

Entretanto, números realmente aleatórios são gerados por processos físicos. Para isso, são construídos dispositivos físicos que analisam fenômenos microscópicos ou quânticos e através de um conversor digital conseguem gerar um número aleatório.

Nos computadores digitais as conhecidas funções que geram números aleatórios não são efetivamente aleatórias.

Na prática o que se usa em simulação estocástica são os geradores de números pseudo-aleatórios.

Os métodos de Simulação Monte Carlo (SMC) utilizam geradores de números pseudo-aleatórios para simular possíveis valores da variável de interesse e assim obter a estimativa para o valor verdadeiro da mesma.

O nome “Monte Carlo” é uma referência ao famoso Cassino de Mônaco. A aleatoriedade e a repetição são as principais características dos métodos de Monte Carlo, que são análogas às atividades praticadas em um Cassino.

Os métodos de Monte Carlo são úteis para estudar:

- Sistemas não determinísticos;
- Sistemas determinísticos muito complexos para se modelar analiticamente;
- Sistemas determinísticos com alta dimensionalidade.



Principais requisitos para os métodos de SMC:

- Possuir o conhecimento das distribuições de probabilidade das variáveis de entrada no sistema;
- Possuir um gerador de números pseudo-aleatórios para criar cenários das variáveis de entrada no sistema.

Números aleatórios:

- São gerados por um processo físico;
- Uma sequência de números é aleatória quando a sua menor representação é ela mesma.

Números pseudo-aleatórios:

- Podem ser gerados em computadores digitais a partir de funções que não são efetivamente aleatórias;
- Os geradores de números pseudo-aleatórios produzem uma sequência determinística de números na precisão do computador, que imita uma sequência de variáveis aleatórias independentes e uniformemente distribuídas entre 0 e 1;
- Uma sequência de números pseudo-aleatórios é imprevisível na medida que ao ver a sequência não é possível explicitar a regra determinística que a produz.

Propriedades desejáveis de um bom gerador de números pseudo-aleatórios:

- Possuir um padrão aleatório: ele deve passar em testes estatísticos de aleatoriedade;
- Possuir um período longo: passar por um grande número de estados antes de voltar a um estado já produzido;
- Ser eficiente: ele deve ser executado rapidamente e requerer um baixo armazenamento;
- Ser de fácil reprodução: a partir de determinadas condições iniciais ele deve produzir sempre a mesma sequência;
- Ser portátil: a partir de determinadas condições iniciais a sequência gerada deve ser a mesma em qualquer computador.

## Python e R

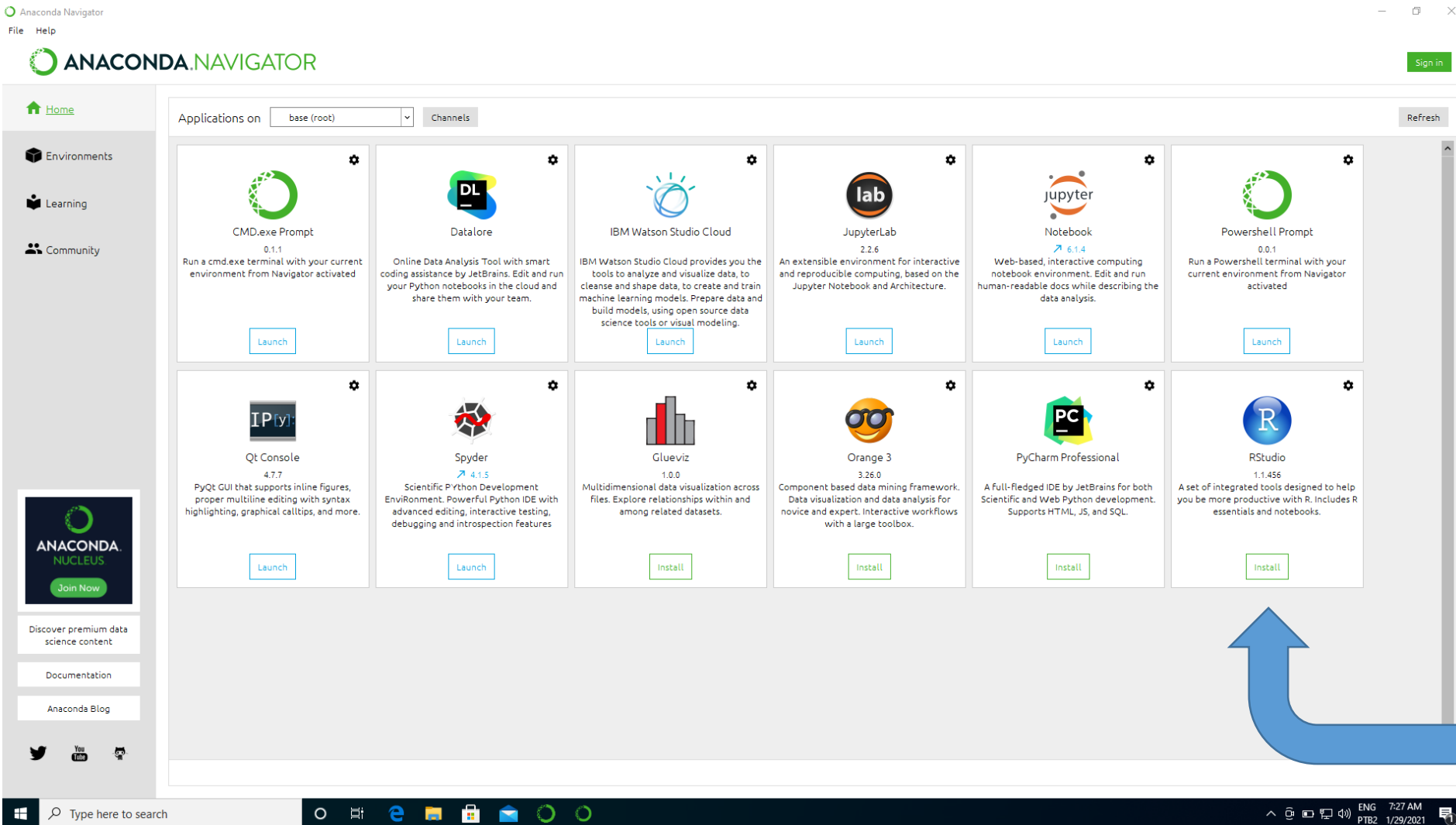
Apesar de existirem outras possibilidades, estas duas linguagens tem polarizado as discussões sobre qual ferramenta utilizar para análise de dados.

As duas linguagens são gratuitas, podem ser instaladas em diversos sistemas operacionais, possuem grande comunidade, muita documentação e ambas possuem vantagens e desvantagens, pontos fortes e pontos fracos.

As duas linguagens podem ser utilizadas em conjunto, ou instalando um pacote no R, para chamada de scripts Python ou instalando um pacote no Python, para executar funções e comandos R.

# Ambiente de desenvolvimento

<https://www.anaconda.com/products/individual> (distribuição open source)



## Instalação padrão

- Spyder 4.1.5
- Python 3.8.5

Após a instalação do Anaconda, deve ser realizada a instalação do RStudio através do próprio ambiente Anaconda.

- RStudio 1.1.456
- R 3.6.1

# Introdução ao Ambiente R



O R (<http://www.r-project.org/>) é ao mesmo tempo uma linguagem de programação e um ambiente para cálculos estatísticos e gráficos.

Foi criada por Ross Ihaka e Robert Gentleman na Universidade de Auckland, Nova Zelândia, e atualmente é desenvolvido pela equipe de Desenvolvimento do R.

É uma linguagem especializada em computação de dados, sendo gratuita e estando disponível em vários sistemas operacionais.

É uma linguagem interpretada, tipicamente utilizada através de um interpretador de comandos.

É também altamente expansível com o uso dos pacotes, que são bibliotecas para sub-rotinas específicas ou áreas de estudo específicas.

Pode ser aplicada nas mais diversas áreas e para os mais diversos fins, tais como: pesquisa científica, business analytics, desenvolvimento de software, relatórios estatísticos, econometria e análise financeira, ciência sociais e big data analytics.

# RStudio



The screenshot displays the RStudio integrated development environment (IDE) with the following components:

- Source Editor:** A single file named 'Untitled1' is open, showing a single line of code with the number '1'.
- Environment Pane:** Located on the right, it shows the 'Global Environment' and states 'Environment is empty'.
- File Explorer:** Also on the right, it shows a file tree with a context menu open over the 'Ho' folder, offering options like 'New Folder', 'Delete', 'Rename', and 'More'. The file list includes folders like 'Graphics', 'IISExpress', and 'Integration Services Script Component', as well as files like 'script.sql'.
- Console:** At the bottom left, it displays the R startup message, including copyright information for 2018 and instructions on how to use R, such as 'Type "license()" or "licence()" for distribution details.'

Para conseguir ajuda sobre um comando, pode ser usada a função help:

```
help(comando) ou ?comando
```

Para instalar um pacote no seu ambiente, você pode usar o comando `install.packages()` com o nome do pacote entre parênteses "":

```
install.packages("nomepacote")
```

Para usar o pacote em seu programa, use o comando `library()` com o nome do pacote como parâmetro (sem ""):

```
library(nomepacote)
```



# Algumas características do R



- R é **case-sensitive**: então “A” e “a” são símbolos diferentes e se referem a diferentes variáveis.
- Comandos diferentes são separados por ponto e vírgula “;”.
- Todos os símbolos alfanuméricos são permitidos, incluindo “.” e “\_”.
- Comentários começam com “#”.
- Como a maioria das linguagens de programação, R permite atribuir valores a variáveis.

```
> # Comentário
> A = 2; a = "uva";
> print(A);
[1] 2
> print(a);
[1] "uva"
> .C=3;
> print(.C+2);
[1] 5
>
```

# Constantes armazenadas no R

```
> pi
[1] 3.141593
>
> letters
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q"
   "r" "s" "t" "u" "v" "w" "x" "y" "z"
>
> LETTERS
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q"
   "R" "S" "T" "U" "V" "W" "X" "Y" "Z"
>
> month.abb
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "
Dec"
>
> month.name
[1] "January"    "February"   "March"      "April"      "May"        "June"
     "July"      "August"    "September"
[10] "October"    "November"  "December"
```

# Tipos de dados no R



- character (“aula”)
- numeric (1.)
- integer (3)
- logical (TRUE or FALSE)
- vector (tipos homogêneos)
- list (parecidos com vectors, mas heterogêneos)
- matrix
- missing values (NA)
- ...

# Operadores Matemáticos

A lista abaixo apresenta alguns dos principais operadores matemáticos do R.

Operador	Descrição
<code>+</code>	Operador de adição
<code>-</code>	Operador de subtração
<code>*</code>	Operador de multiplicação
<code>/</code>	Operador de divisão
<code>:</code>	Operador de sequência
<code>^</code>	Operador exponencial
<code>%%</code>	Operador de módulo

# Operadores Lógicos

São operados binários para realização de testes entre duas variáveis (objetos). Estas operações retornam o valor TRUE (1) ou FALSE (0).

Operador	Descrição
==	Operador de igualdade
>	Operador “maior que”
<	Operador “menor que”
<=	Operador “menor ou igual”
!	Operador “Não”
&	Operador lógico “E”
	Operador lógico “OU”

# Ementa da Disciplina



- Introdução à Probabilidade.
- Algoritmos para geração de números pseudo-aleatórios.
- Algoritmos para simulação estocástica de variáveis aleatórias discretas e aplicações.
- Algoritmos para simulação estocástica de variáveis aleatórias contínuas e aplicações.
- Algoritmos para simulação de eventos discretos: filas, redes, etc.

1. Sheldon Ross, Simulation, Academic Press. 5th Edition, 2012.
2. Jane Horgan, Probability with R: An Introduction with Computer Science Applications, Willey, 2009.
3. Douglas C. Montgomery, George C. Runger. Estatística Aplicada e Probabilidade Para Engenheiros, LTC, 2016.

1. Donald E. Knuth, Art of Computer Programming, Volume 2: Seminumerical Algorithms, Addison-Wesley Professional, 1997.
2. W. Celes, R. Cerqueira, J.L. Rangel. Introdução à estruturas de dados com técnicas de programação em C. Elsevier, 2016.
3. A. Downey. Pense em Python. Novatec, 2016.
4. I. Malta, S. Pesco, H. Lopes. Cálculo a uma Variável, Volume I. Edição Loyola, Editora PUC Rio, 2002.
5. I. Malta, S. Pesco, H. Lopes. Cálculo a uma Variável, Volume II. Edição Loyola, Editora PUCRio, 2002.