



Configuração Anaconda

Introdução

O Pandas disponibiliza estruturas de dados e ferramentas para tratar tais estruturas que forma idealizados para facilitar o processo de tratamento e análise de informações utilizando a linguagem Python.

Para trabalharmos com alguns projetos, precisamos utilizar bibliotecas específicas. Com o tempo, perceberemos que o Pandas está sempre atrelado a tais bibliotecas. Um exemplo é uma biblioteca de processamento numérico como Numpy e SciPy, o Pandas pode nos ajudar fornecendo informações a elas. No caso de bibliotecas de análise como scikit-learn o Pandas também estará atuante no fornecimento de informações.

- Anaconda é a principal distribuição para cientistas de dados que usam Python
- Jupyter é uma ferramenta para executar código Python e visualizar os dados

Conhecendo o Anaconda

É uma distribuição do Python própria para Data Science por conter todo o conjunto de ferramentas de que precisamos para resolver problemas.

O Anaconda ainda fornece uma IDE padrão denominada Spider, não usaremos este recurso no projeto, mas sim o Jupyter, uma ferramenta mais interativa.

Configuração

- Na etapa "Advanced Options" assinalaremos apenas a opção "Register Anaconda as my default Python 3.6", para que posteriormente aprendermos como criar ambientes virtuais.
- Ao lado esquerdo da tela, haverá o menu de ferramentas em que temos, por exemplo, a opção "Enviromnments" que nos permite acessar todas as bibliotecas instaladas no ambiente padrão, como o próprio Pandas e Scikit-Learn.
- Iremos estabelecer dois caminhos em nosso path. Primeiro, coletaremos o endereço em que salvamos a ferramenta Anaconda, no caso C:\Users\Meu nome\Anaconda3.
- Ainda no explorer, clicaremos sobre "Este Computador" com o botão direito e selecionaremos a opção "Propriedades". Com a janela do sistema aberta, clicaremos em "Configurações Avançadas do Sistema".
- Será aberta uma nova janela de "Propriedades do Sistema". Selecionaremos "Variáveis de Ambiente".
- Na área "Variáveis de usuário para Alura", selecionaremos "Path" e clicaremos sobre a opção "Editar". Em seguida, inserimos o C:\Users\Anaconda3 caminho na área "Editar a variável de ambiente". Em seguida, clicaremos sobre o botão "Novo" e adicionaremos o caminho C:\Users\Alura\Anaconda3\Scripts.

Ambientes Virtuais no Anaconda

- ambientes virtuais ajudam a isolar um projeto para definir a versão das bibliotecas e do próprio Python
- as bibliotecas são atualizadas, e muitas funcionalidades se modificam ou deixam de existir.
- por isso, quando temos um projeto que foi desenvolvido com essas bibliotecas, depois delas serem atualizadas, o projeto passa a não ser mais funcional
- para isso são criados os **ambientes virtuais**

- quando criamos um ambiente virtual, a manutenção de um código é mais simples:
 - podemos inserir a versão da biblioteca que queremos utilizar, e assim o código será executado no ambiente da mesma maneira que antes da atualização

Criação de Ambientes Virtuais

- no Anaconda Navigator, no menu principal, se clicarmos sobre "Environments" poderemos criar um novo ambiente virtual, em **create**
- contudo, por essa via teremos um ambiente mais engessado, com menos recursos de gostaríamos
- criando pelo prompt de comando podemos escolher as versões do python e do pandas que serão utilizadas:

```
conda create --name elton_pandas python=3.6 pandas=0.22.0
```

- para selecionar o ambiente pelo prompt:

```
activate nome_do_ambiente
```

- para ver a versão de uma biblioteca no prompt, com o ambiente ativo, abrir o python:

```
python
import nome_da_biblioteca
biblioteca.__version__
```

- para sair

```
exit() # sai do python
conda deactivate # sai do ambiente
```

- visualizar ambientes

```
conda info --envs
```

- apagar ambiente

```
conda remove --name nome_do_ambiente --all
```