Algoritmo e Lógica de Programação

Alunos: Daniel e Everton;

Curso: Análise e Desenvolvimento de Sistemas;

Período: 1º Semestre;

Turma: B;

Professor: Eduardo Masanori;

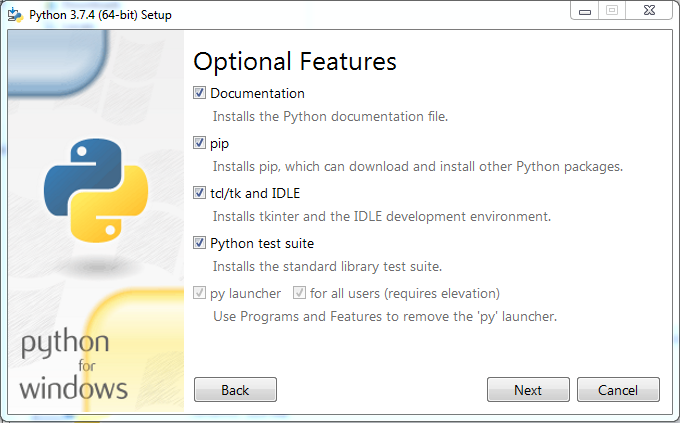
Atividade: EP1: Análise de Dados.

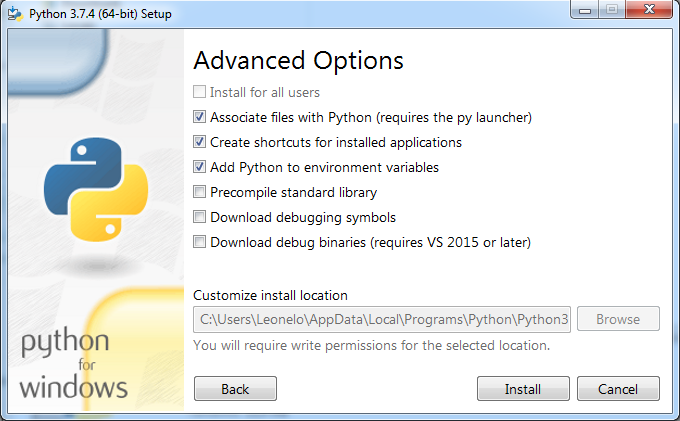
Equipamento utilizado para o trabalho:

* Nome do Sistema Operacional: Microsoft Windows 7 Professional;
* Versão: 6.1.7601 Service Pack 1 Compilação 7601;
* Tipo do sistema: x64-based PC;
* Processador: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz, 2501 Mhz, 2 Núcleo(s), 4 Processador(es) Lógico(s);
* Memória Física (RAM) Instalada: 4,00 GB;
* Memória física total: 3,87 GB.

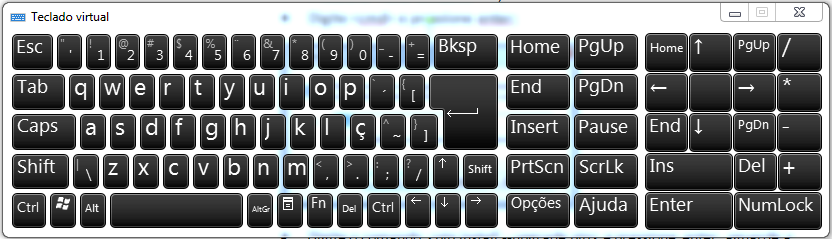
Dependências para a execução da análise e apresentação das informações:

* Python:
  + Versão: python-3.7.4-amd64;
  + Download: <https://www.python.org/downloads/>;
  + Instalação:
    - Na instalação, habilite a opção <pip> Figura1, pacote que permitirá a instalação de pacotes e suas dependências via prompt, e, habilite a opção <Add python to environment variables> Figura2, para que os comandos do Python sejam executar em qualquer diretório via prompt.

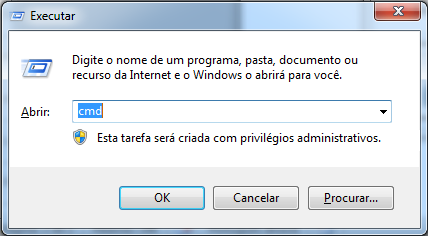




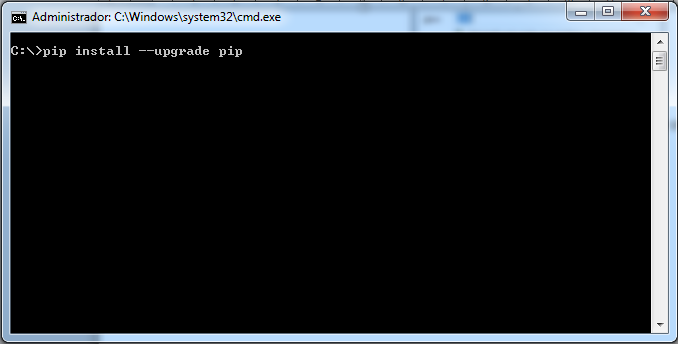
* + Atualização pip:
    - Abra o prompt de comando:
      * Pressione Tecla Windows + R;



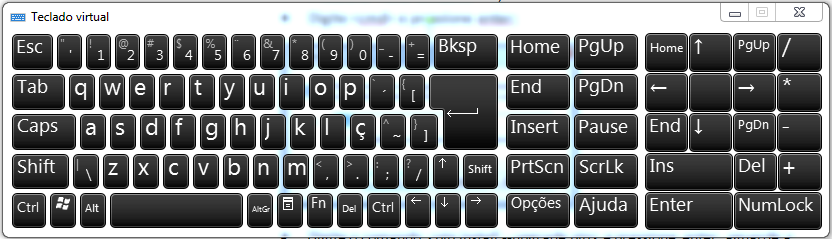
* + - * Digite <cmd> e pressione enter;



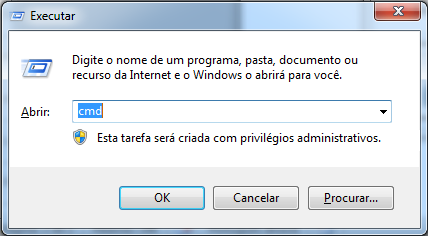
* + - Digite o comando <pip install --upgrade pip> e pressione enter, aguarde a atualização.



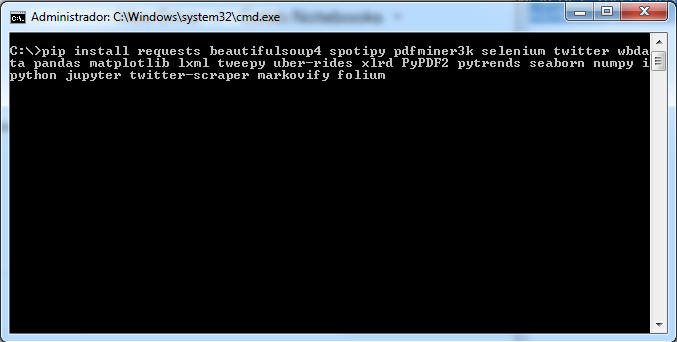
* + Instalação das bibliotecas e dependências:
    - Abra o prompt de comando:
      * Pressione Tecla Windows + R;



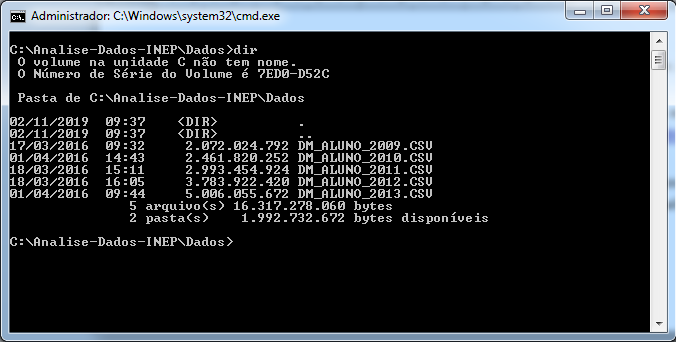
* + - * Digite <cmd> e pressione enter;



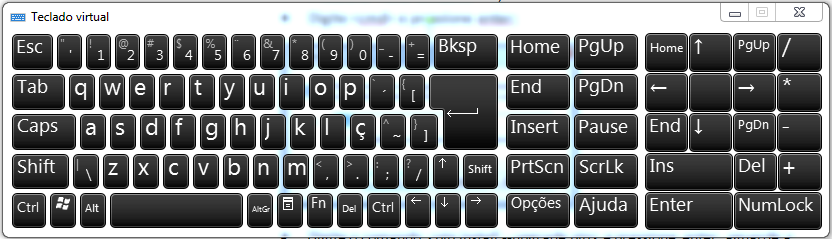
* + - Digite o comando <pip install requests beautifulsoup4 spotipy pdfminer3k selenium twitter wbdata pandas matplotlib lxml tweepy uber-rides xlrd PyPDF2 pytrends seaborn numpy ipython jupyter twitter-scraper markovify folium> e pressione enter, aguarde a instalação dos pacotes.



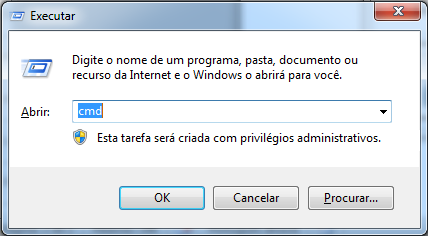
* + Preparação do ambiente para análise dos dados:
    - Crie um diretório, selecione o local que desejar. Neste trabalho utilizado o diretório <C:\Analise-Dados-INEP>;
    - Baixe os arquivos ‘.CSV’ para análise. Arquivos estão disponíveis em:
      * <https://drive.google.com/file/d/1IOG8BEshJLGOQG2Eg84v8UMPeSATbiJ4/view?usp=sharing>
      * Neste trabalho serão analisados os anos de 2009 há 2013.
    - Extraia os arquivos dentro do diretório <C:\Analise-Dados-INEP\Dados\>;



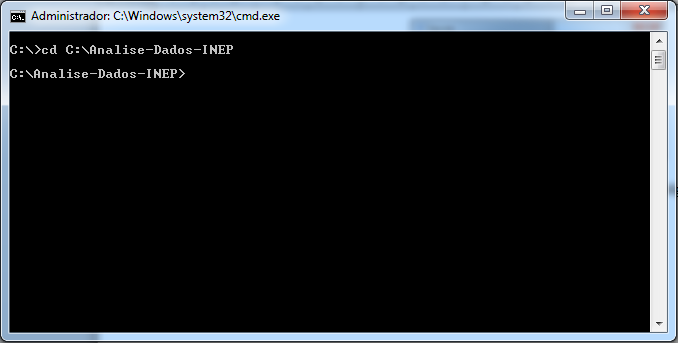
* + Executar o Jupyter Notebook:
    - Abra o prompt de comando:
      * Pressione Tecla Windows + R;



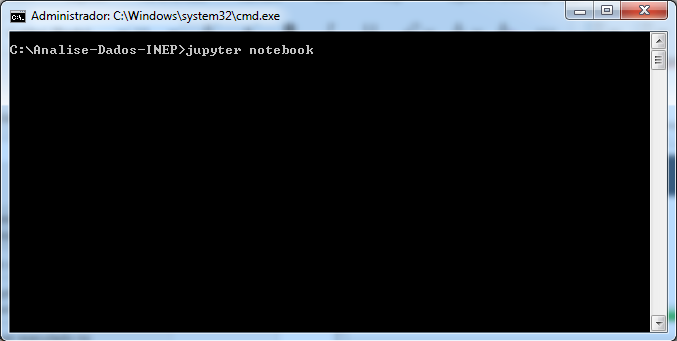
* + - * Digite <cmd> e pressione enter;



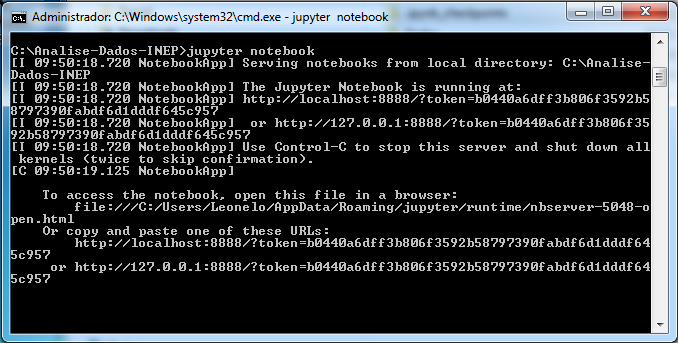
* + - Acesse o diretório <C:\Analise-Dados-INEP\>, digite o comando <cd C:\Analise-Dados-INEP> e pressione enter;



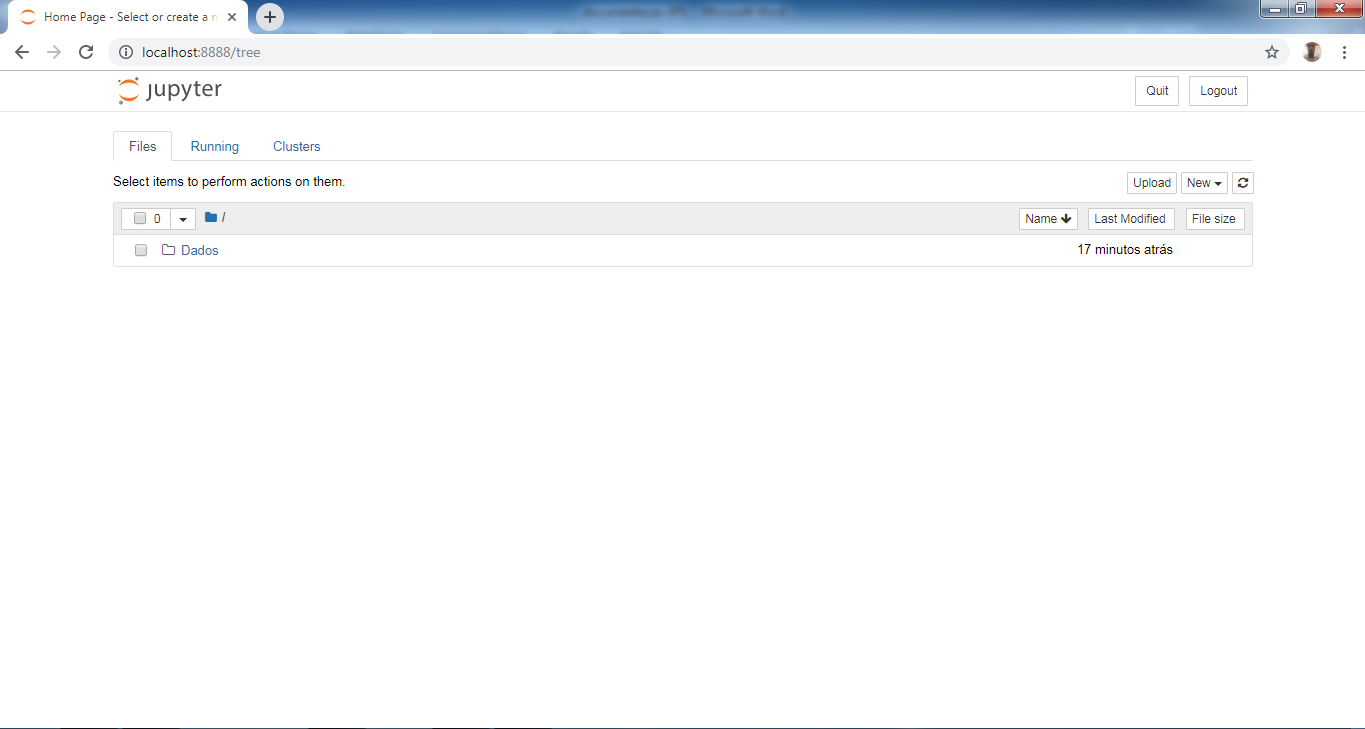
* + - Digite o comando <jupyter notebook>, será executado em servidor de web com os recursos para a realização das análises, aguarde a execução do servidor e não feche o cmd, abrirá a janela do navegador padrão com o serviço Jupyter Notebook.



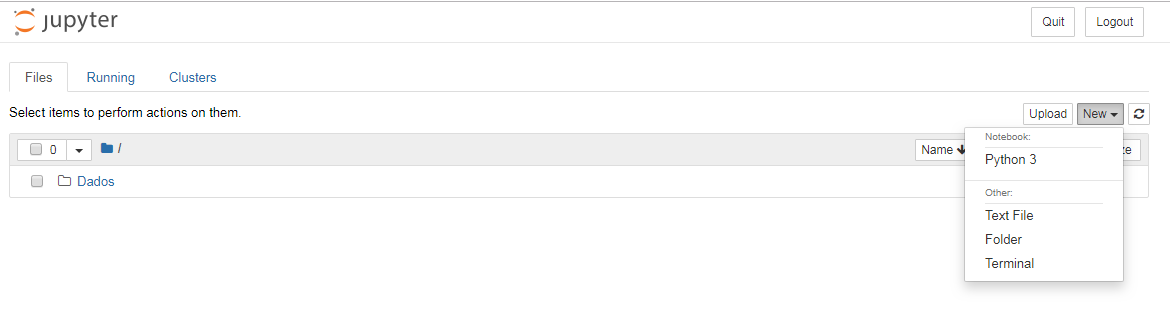
Obs. Caso não abra o navegador padrão com o acesso ao Jupyter Notebook, copie e cole no navegador padrão e URL e de acesso, conforme figura abaixo:



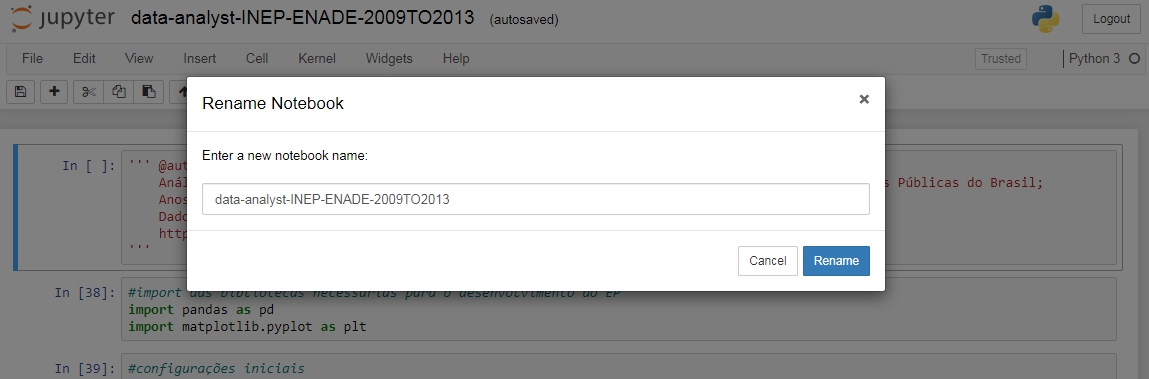
No computador gerou a URL “http://localhost:8888/?token=b0440a6dff3b806f3592b58797390fabdf6d1dddf645c957”, copie a URL gerada e cole no navegador padrão e pressione enter que abrirá os recursos do Jupyter Notebook, conforme imagem abaixo:



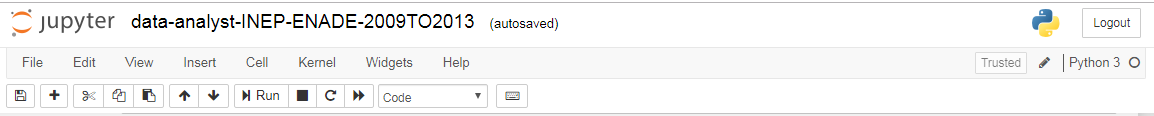
* + Iniciar a análise de dados:
    - Clique em <New\Python 3>, conforme figura abaixo:



* + - Na nova janela que abrira renomeie o arquivo para o nome desejado, neste trabalho utilizado o nome “data-analyst-INEP-ENADE-2009TO2013”, conforme figura abaixo:



* + - Inserir as células e executá-las, para adicionar pressione o botão (+) e para executar pressione o botão (>| Run), da paleta de comandos, conforme figura abaixo:

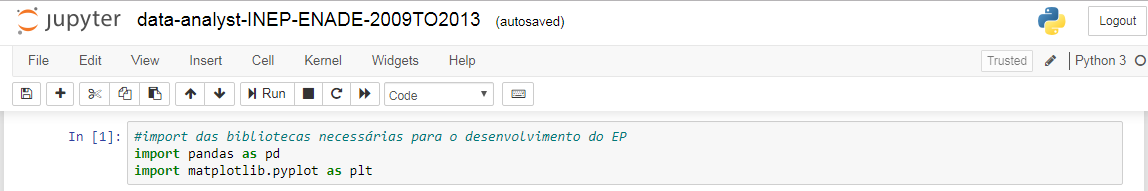


* + - * Import das bibliotecas:

#import das bibliotecas necessárias para o desenvolvimento do EP

import pandas as pd

import matplotlib.pyplot as plt



Atenção: Caso apareça um asterisco entre colchetes, significa que o comando está em execução, caso seja representado numericamente, significa que o comando foi executado, caso apareça em branco, significa que os comandos contidos na célula não foram executados.

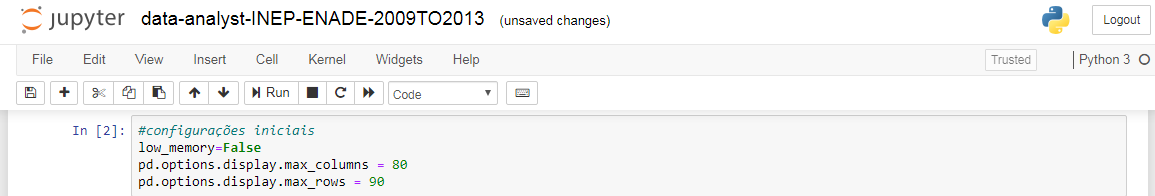
* + - * Configurações iniciais:

#configurações iniciais

low\_memory=False

pd.options.display.max\_columns = 80

pd.options.display.max\_rows = 90



* + - * Declaração das variáveis:

#declaração variáveis

d = './Dados/'

pN = 'DM\_ALUNO\_'

anos = ['2009', '2010', '2011', '2012', '2013']

e = '.CSV'

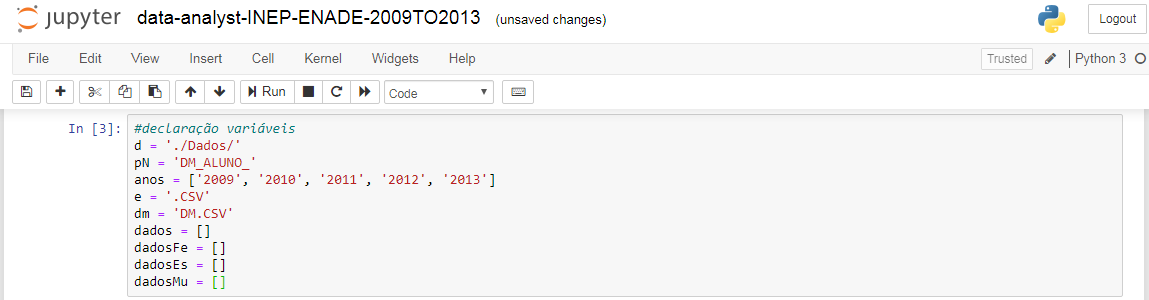
dm = 'DM.CSV'

dados = []

dadosFe = []

dadosEs = []

dadosMu = []



* + - * Criação do Data Frame e Leitura dos arquivos, quantificando os dados e armazenando-os em suas respectivas listas:

for ano in anos:

df = pd.read\_csv(d+pN+ano+e, usecols=['CO\_COR\_RACA\_ALUNO', 'CO\_CATEGORIA\_ADMINISTRATIVA'], delimiter = '|', encoding = 'iso-8859-1')

#contabilização e consolidação das informações - Geral

dados.append(df.query('(CO\_COR\_RACA\_ALUNO == 2 or CO\_COR\_RACA\_ALUNO == 3) and (CO\_CATEGORIA\_ADMINISTRATIVA == 1 | CO\_CATEGORIA\_ADMINISTRATIVA == 2 | CO\_CATEGORIA\_ADMINISTRATIVA == 3)')['CO\_COR\_RACA\_ALUNO'].count())

#contabilização e consolidação das informações - Federal

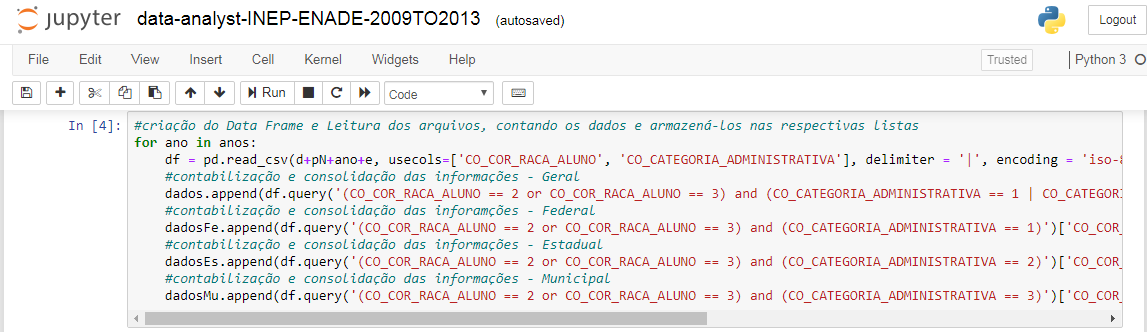
dadosFe.append(df.query('(CO\_COR\_RACA\_ALUNO == 2 or CO\_COR\_RACA\_ALUNO == 3) and (CO\_CATEGORIA\_ADMINISTRATIVA == 1)')['CO\_COR\_RACA\_ALUNO'].count())

#contabilização e consolidação das informações - Estadual

dadosEs.append(df.query('(CO\_COR\_RACA\_ALUNO == 2 or CO\_COR\_RACA\_ALUNO == 3) and (CO\_CATEGORIA\_ADMINISTRATIVA == 2)')['CO\_COR\_RACA\_ALUNO'].count())

#contabilização e consolidação das informações - Municipal

dadosMu.append(df.query('(CO\_COR\_RACA\_ALUNO == 2 or CO\_COR\_RACA\_ALUNO == 3) and (CO\_CATEGORIA\_ADMINISTRATIVA == 3)')['CO\_COR\_RACA\_ALUNO'].count())



* + - * Plotar dados sintéticos obtidos no gráfico:

#plotar dados gráfico

fig, ax = plt.subplots()

plt.rcParams['figure.figsize'] = (10,8)

ax.plot(anos, dados, label='Total')

ax.plot(anos, dadosFe, label='Federal')

ax.plot(anos, dadosEs, label='Estadual')

ax.plot(anos, dadosMu, label='Municipal')

ax.grid()

plt.legend()

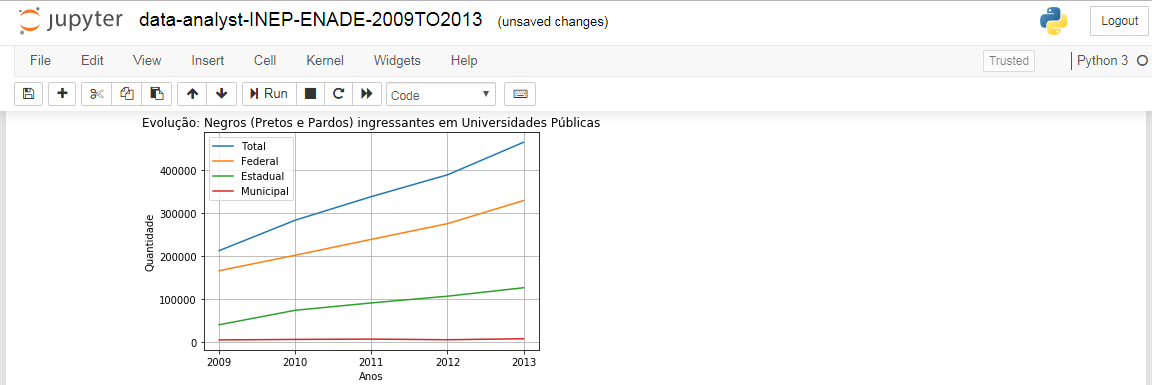
plt.title('Evolução: Negros (Pretos e Pardos) ingressantes em Universidades Públicas')

plt.xlabel('Anos')

plt.ylabel('Quantidade')

plt.show()





* + - * Consolidação dos resultados:

#organização dos dados em tabelas, para melhor visualização dos dados

consolidacao = {

'2009': [dadosFe[0], dadosEs[0], dadosMu[0], dados[0]],

'2010': [dadosFe[1], dadosEs[1], dadosMu[1], dados[1]],

'2011': [dadosFe[2], dadosEs[2], dadosMu[2], dados[2]],

'2012': [dadosFe[3], dadosEs[3], dadosMu[3], dados[3]],

'2013': [dadosFe[4], dadosEs[4], dadosMu[4], dados[4]],

}

df\_DC = pd.DataFrame(consolidacao, columns=anos, index=['Federal','Estadual','Municipal','Total'])

df\_DC

