Pró-reitora de Pós-Graduação, Pesquisa e Inovação

Especialização em Ciências de Dados e Analytics

Programação para Ciência de Dados

Parte 2 – Pandas

Aula 6

Agenda

- Biblioteca Pandas
  - Visão geral e instalação
  - *Series*
  - *DataFrame*
  - Dados ausentes
  - Agrupamento
  - Concatenação, junção e *mesclagem* de dados
  - I/O

Pandas

- Biblioteca escrita sobre o numpy

- Oferece diversos recursos para visualização e limpeza de dados, com funcionalidades similares aquelas que temos no Excel

Instalação

- Com o Anaconda já instalado, o processo de instalação do Pandas é muito simples:
  - conda install pandas

    OU
  - pip install pandas

Pandas :: *Series*

- Pandas possui essencialmente dois objetos, um deles é o *Series*
- Parece muito com o tipo de dados dicionário de Python

```
import numpy as np
import pandas as pd
```

```
rotulo = ['a', 'b', 'c']
```

```
valores = [3, 17, 30]
```

```
pd.Series(valores, rotulo)
```

```
a     3
b    17
c    30
dtype: int64
```

```
np_valores = np.array(valores)
```

```
pd.Series(np_valores, rotulo)
```

```
a     3
b    17
c    30
dtype: int32
```

Veja que ao criar a *Series* com numpy há uma economia de memória, por usar int32

Pandas :: *Series*

- Operações básicas

```
top5Paises1990 = pd.Series([89.5,66.6,42.4,34.6,26.9],
                           ['EUA','JP','AL','UK','FR'])
```

```
top5Paises1990
```

```
EUA    89.5
JP     66.6
AL     42.4
UK     34.6
FR     26.9
dtype: float64
```

```
top5Paises2000 = pd.Series([197.4,128.9,85.5,73.9,70.4],
                           ['EUA','JP','AL','CG','UK'])
```

```
top5Paises2010 = pd.Series([406,158.5,145.5,126.9,122],
                           ['CH','AL','EUA','CG','JP'])
```

```
top5Paises1990 + top5Paises2000 + top5Paises2010
```

```
AL     286.4
CG       NaN
CH       NaN
EUA    432.4
FR       NaN
JP     317.5
UK       NaN
dtype: float64
```

NaN ocorre porque os rótulos não foram localizados em algumas das séries somadas.

Pandas :: *DataFrame*

- O ***DataFrame*** corresponde a um conjunto de *Series*
- Podemos vê-lo como uma planilha Excel!

```
paises = ['EUA','JP','AL','UK','FR','CG','CH']
```

```
anos = ['1990','2000','2010']
```

```
dados = [[89.5,66.6,42.4,34.6,26.9,15,0],
         [197.4,128.9,85.5,70.4,58.7,73.9,41.7],
         [145.5,122,158.5,59.8,99.7,126.9,406]]
```

```
df = pd.DataFrame(dados, anos, paises)
```

```
df
```

	EUA	JP	AL	UK	FR	CG	CH
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0

Pandas :: *DataFrame*

- Acesso aos dados é similar a notação usada em Dicionários!

	EUA	JP	AL	UK	FR	CG	CH
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0

```
df['CH']
```

```
1990      0.0
2000     41.7
2010    406.0
Name: CH, dtype: float64
```

```
df[['CH','EUA']]
```

	CH	EUA
1990	0.0	89.5
2000	41.7	197.4
2010	406.0	145.5

```
df['CH']['2000']
```

```
41.7
```

Pandas :: *DataFrame*

- ***df.loc***: Para selecionar linhas ou células do DF usando os rótulos

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
df.loc['2000']
```

```
EUA    197.4
JP     128.9
AL      85.5
UK      70.4
FR      58.7
CG      73.9
CH      41.7
BR       5.9
Name: 2000, dtype: float64
```

```
df.loc[['2000','2010'],['UK','FR']]
```

	UK	FR
2000	70.4	58.7
2010	59.8	99.7

Pandas :: *DataFrame*

- ***df.iloc***: Para selecionar linhas ou células usando os índices

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
df.iloc[2]
```

```
EUA    145.5
JP     122.0
AL     158.5
UK      59.8
FR      99.7
CG     126.9
CH     406.0
BR       8.0
Name: 2010, dtype: float64
```

```
df.iloc[1:,3:5]
```

	UK	FR
2000	70.4	58.7
2010	59.8	99.7

Pandas :: *DataFrame*

- Inserção de novos valores também é similar ao que temos em Dicionários!

```
df['BR'] = [1,5.9,8]
```

```
df
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

Pandas :: *DataFrame*

- df.drop: Para remoção de linhas (0) ou colunas (1)

```
df
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
df.drop('UK', axis=1)
```

	EUA	JP	AL	FR	CG	CH	BR
1990	89.5	66.6	42.4	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	99.7	126.9	406.0	8.0

```
df
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
# para remover no objeto corrente, use inplace=true
# df.drop('UK', axis=1, inplace=True)
```

Pandas :: *DataFrame*

- Seleção condicional de valores no DF
  - Similar ao que temos com *numpy* array

```
mediaPorPais = df.mean()
```

```
mediaPorPais
```

```
EUA    144.133333
JP     105.833333
AL      95.466667
UK      54.933333
FR      61.766667
CG      71.933333
CH     149.233333
BR       4.966667
dtype: float64
```

```
mediaGeral = mediaPorPais.mean()
```

```
mediaGeral
```

```
86.03333333333333
```

```
indicesMaiorMediaGeral = df > mediaGeral
```

```
indicesMaiorMediaGeral
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	True	False	False	False	False	False	False	False
2000	True	True	False	False	False	False	False	False
2010	True	True	True	False	True	True	True	False

Pandas :: *DataFrame*

- Seleção condicional de valores no DF
  - Similar ao que temos com *numpy* array

```
df[indicesMaiorMediaGeral]
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2000	197.4	128.9	NaN	NaN	NaN	NaN	NaN	NaN
2010	145.5	122.0	158.5	NaN	99.7	126.9	406.0	NaN

```
df[indicesMaiorMediaGeral]['JP']
```

```
1990      NaN
2000    128.9
2010    122.0
Name: JP, dtype: float64
```

```
df[df['JP'] > mediaGeral]['JP']
```

```
2000    128.9
2010    122.0
Name: JP, dtype: float64
```

Pandas :: *DataFrame*

- Operadores de seleção úteis
  - &
  - |

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
desvioPadraoGeral = df.std().std()
```

```
desvioPadraoGeral
```

```
68.52119478973259
```

```
mediaGeral
```

```
86.03333333333333
```

```
df[(df > mediaGeral+desvioPadraoGeral)]
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2000	197.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2010	NaN	NaN	158.5	NaN	NaN	NaN	406.0	NaN

```
df[(df > mediaGeral)
   & (df <= mediaGeral+desvioPadraoGeral)]
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2000	NaN	128.9	NaN	NaN	NaN	NaN	NaN	NaN
2010	145.5	122.0	NaN	NaN	99.7	126.9	NaN	NaN

```
df[(df < mediaGeral-desvioPadraoGeral)
   | (df > mediaGeral+desvioPadraoGeral)]
```

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	NaN	NaN	NaN	NaN	NaN	15.0	0.0	1.0
2000	197.4	NaN	NaN	NaN	NaN	NaN	NaN	5.9
2010	NaN	NaN	158.5	NaN	NaN	NaN	406.0	8.0

Pandas :: *DataFrame*

- Alterando o índice do DF
  - ***df.reset_index***

	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
df.reset_index(inplace=True)
```

```
df
```

	index	EUA	JP	AL	UK	FR	CG	CH	BR
0	1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
1	2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2	2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

Pandas :: *DataFrame*

- Alterando o índice do DF

	index	EUA	JP	AL	UK	FR	CG	CH	BR
0	1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
1	2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2	2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
df['Ano'] = ['1990', '2000', '2010']
```

```
df
```

	index	EUA	JP	AL	UK	FR	CG	CH	BR	Ano
0	1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0	1990
1	2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9	2000
2	2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0	2010

```
df.set_index('Ano',inplace=True)
```

```
df
```

Ano	index	EUA	JP	AL	UK	FR	CG	CH	BR
1990	1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

```
df = df.iloc[:,1:]
```

```
df
```

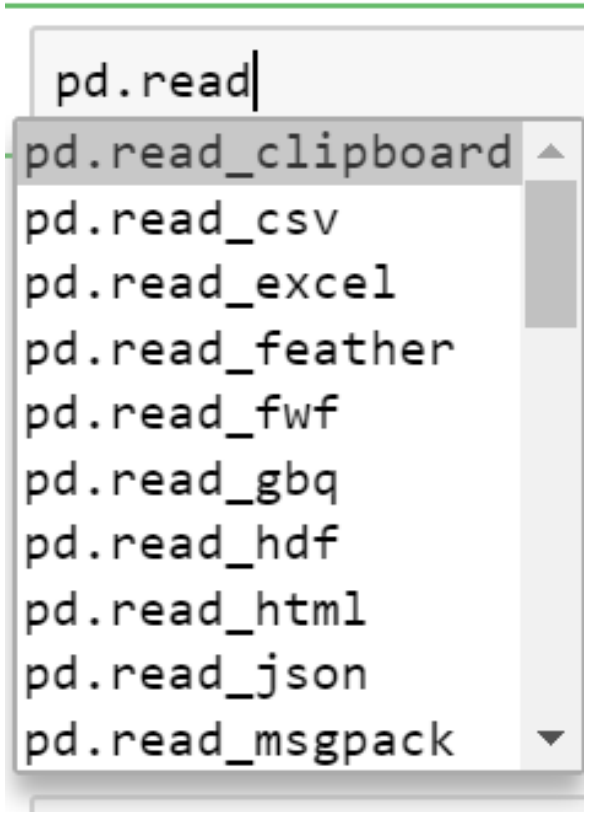
Ano	EUA	JP	AL	UK	FR	CG	CH	BR
1990	89.5	66.6	42.4	34.6	26.9	15.0	0.0	1.0
2000	197.4	128.9	85.5	70.4	58.7	73.9	41.7	5.9
2010	145.5	122.0	158.5	59.8	99.7	126.9	406.0	8.0

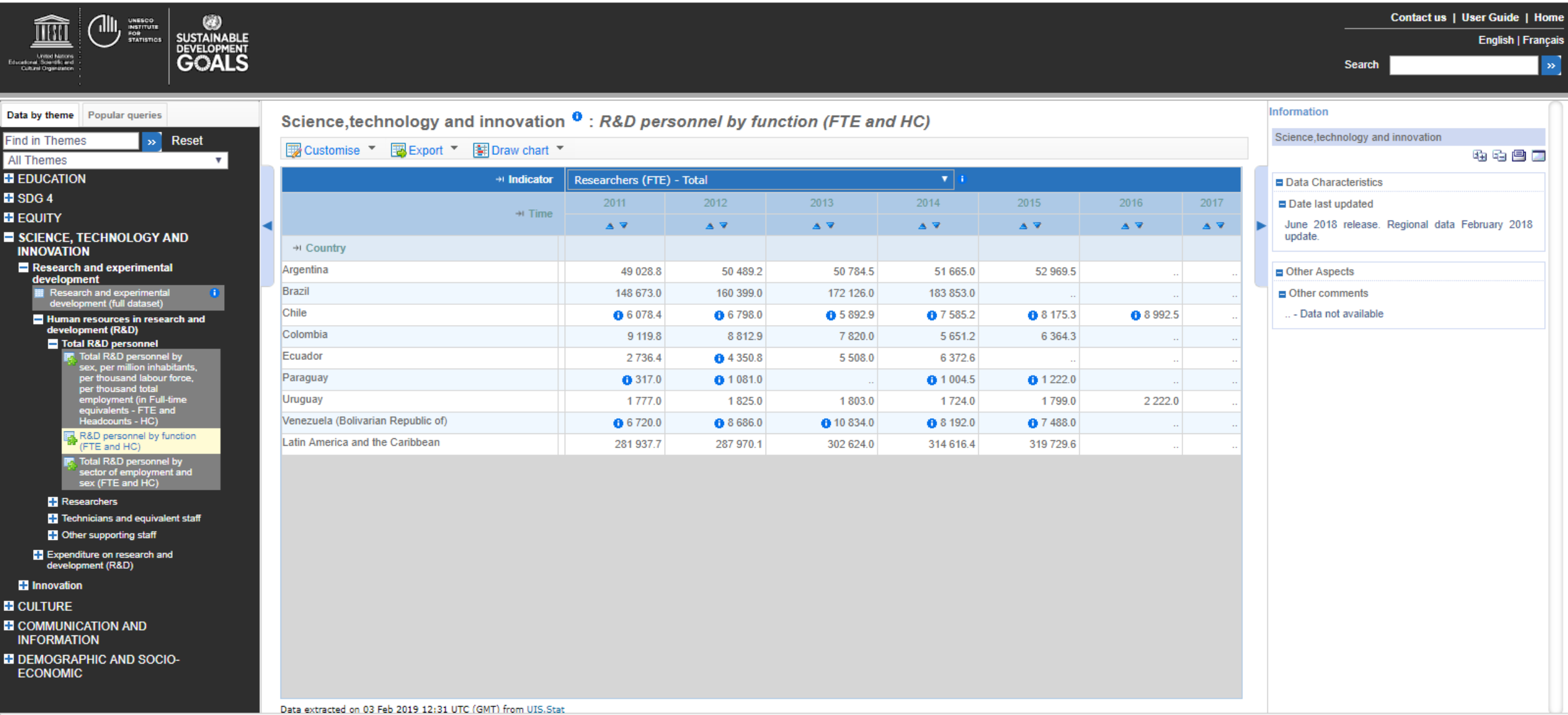
Agenda

- Biblioteca Pandas
  - ~~Visão geral e instalação~~
  - ~~*Series*~~
  - ~~*DataFrame*~~
  - I/O
  - Multiníveis
  - Dados ausentes
  - Agrupamento
  - Concatenação, junção e *mesclagem* de dados

Pandas :: *I/O*

- Importando dados com o Pandas
  - ***pd.read ...***





UNESCO Institute for Statistics | Sustainable Development Goals

Contact us | User Guide | Home

English | Français

Search

Data by theme | Popular queries

Find in Themes | Reset

All Themes

- EDUCATION
- SDG 4
- EQUITY
- SCIENCE, TECHNOLOGY AND INNOVATION
  - Research and experimental development
    - Research and experimental development (full dataset)
    - Human resources in research and development (R&D)
      - Total R&D personnel
        - Total R&D personnel by sex, per million inhabitants, per thousand labour force, per thousand total employment (in Full-time equivalents - FTE and Headcounts - HC)
        - R&D personnel by function (FTE and HC)
        - Total R&D personnel by sector of employment and sex (FTE and HC)
      - Researchers
      - Technicians and equivalent staff
      - Other supporting staff
    - Expenditure on research and development (R&D)
  - Innovation
- CULTURE
- COMMUNICATION AND INFORMATION
- DEMOGRAPHIC AND SOCIO-ECONOMIC

Science,technology and innovation : *R&D personnel by function (FTE and HC)*

Customise | Export | Draw chart

Indicator	Researchers (FTE) - Total						
Time	2011	2012	2013	2014	2015	2016	2017
Country							
Argentina	49 028.8	50 489.2	50 784.5	51 665.0	52 969.5	..	..
Brazil	148 673.0	160 399.0	172 126.0	183 853.0	..	..	..
Chile	6 078.4	6 798.0	5 892.9	7 585.2	8 175.3	8 992.5	..
Colombia	9 119.8	8 812.9	7 820.0	5 651.2	6 364.3	..	..
Ecuador	2 736.4	4 350.8	5 508.0	6 372.6	..	..	..
Paraguay	317.0	1 081.0	..	1 004.5	1 222.0	..	..
Uruguay	1 777.0	1 825.0	1 803.0	1 724.0	1 799.0	2 222.0	..
Venezuela (Bolivarian Republic of)	6 720.0	8 686.0	10 834.0	8 192.0	7 488.0	..	..
Latin America and the Caribbean	281 937.7	287 970.1	302 624.0	314 616.4	319 729.6	..	..

Data extracted on 03 Feb 2019 12:31 UTC (GMT) from UIS.Stat

Information

Science,technology and innovation

Data Characteristics

Date last updated

June 2018 release. Regional data February 2018 update.

Other Aspects

Other comments

.. - Data not available

Fonte: http://data.uis.unesco.org/

C:\Users\byron\Google Drive\UPE\especializacao\residencia-ia\python\dados\SCN_DS_03022019073414299-ResearchersFTE.csv - Notepad++

Arquivo Editar Localizar Visualizar Formatar Linguagem Configurações Tools Macro Executar Plugins Janela ?

DSDLModelsInfo.txt | new 69 | DSIPPreProcArguments.txt | new 70 | new 45 | SCN_DS_03022019073414299.csv | SCN_DS_03022019073414299-ResearchersFTE.csv

```
"INDICATOR","Indicator","LOCATION","Country","TIME","Time","Value","Flag Codes","Flags"CRLF
"20001","Researchers (FTE) - Total ","ARG","Argentina","2011","2011",49028.83,,CRLF
"20001","Researchers (FTE) - Total ","ARG","Argentina","2012","2012",50489.18231,,CRLF
"20001","Researchers (FTE) - Total ","ARG","Argentina","2013","2013",50784.51421,,CRLF
"20001","Researchers (FTE) - Total ","ARG","A
"20001","Researchers (FTE) - Total ","ARG","A
"20001","Researchers (FTE) - Total ","BRA","B
"20001","Researchers (FTE) - Total ","BRA","B
"20001","Researchers (FTE) - Total ","BRA","B
"20001","Researchers (FTE) - Total ","BRA","B
"20001","Researchers (FTE) - Total ","CHL","C
"20001","Researchers (FTE) - Total ","CHL","C
"20001","Researchers (FTE) - Total ","CHL","C
"20001","Researchers (FTE) - Total ","CHL","C
"20001","Researchers (FTE) - Total ","CHL","C
"20001","Researchers (FTE) - Total ","CHL","C
"20001","Researchers (FTE) - Total ","COL","C
"20001","Researchers (FTE) - Total ","COL","C
"20001","Researchers (FTE) - Total ","COL","C
"20001","Researchers (FTE) - Total ","COL","C
"20001","Researchers (FTE) - Total ","COL","C
"20001","Researchers (FTE) - Total ","ECU","E
"20001","Researchers (FTE) - Total ","ECU","E
"20001","Researchers (FTE) - Total ","ECU","E
"20001","Researchers (FTE) - Total ","ECU","E
"20001","Researchers (FTE) - Total ","PRY","P
"20001","Researchers (FTE) - Total ","PRY","P
"20001","Researchers (FTE) - Total ","PRY","P
"20001","Researchers (FTE) - Total ","PRY","P
"20001","Researchers (FTE) - Total ","URY","U
"20001","Researchers (FTE) - Total ","URY","U
```

Normal text file

```
dfIn = pd.read_csv('C:\\Users\\byron\\Google Drive\\UPE\\especializacao\\residencia-ia\\python\\dados\\SCN_DS_03022019073414299-R
```

```
dfIn
```

	INDICATOR	Indicator	LOCATION	Country	TIME	Time	Value	Flag Codes	Flags
0	20001	Researchers (FTE) - Total	ARG	Argentina	2011	2011	49028.83000	NaN	NaN
1	20001	Researchers (FTE) - Total	ARG	Argentina	2012	2012	50489.18231	NaN	NaN
2	20001	Researchers (FTE) - Total	ARG	Argentina	2013	2013	50784.51421	NaN	NaN
3	20001	Researchers (FTE) - Total	ARG	Argentina	2014	2014	51665.00000	NaN	NaN
4	20001	Researchers (FTE) - Total	ARG	Argentina	2015	2015	52969.50000	NaN	NaN
5	20001	Researchers (FTE) - Total	BRA	Brazil	2011	2011	148673.00000	NaN	NaN
6	20001	Researchers (FTE) - Total	BRA	Brazil	2012	2012	160399.00000	NaN	NaN
7	20001	Researchers (FTE) - Total	BRA	Brazil	2013	2013	172126.00000	NaN	NaN
8	20001	Researchers (FTE) - Total	BRA	Brazil	2014	2014	183853.00000	NaN	NaN
9	20001	Researchers (FTE) - Total	CHL	Chile	2011	2011	6078.40000	NaN	NaN
10	20001	Researchers (FTE) - Total	CHL	Chile	2012	2012	6798.01000	NaN	NaN
11	20001	Researchers (FTE) - Total	CHL	Chile	2013	2013	5892.89900	NaN	NaN
12	20001	Researchers (FTE) - Total	CHL	Chile	2014	2014	7585.24000	NaN	NaN
13	20001	Researchers (FTE) - Total	CHL	Chile	2015	2015	8175.32700	NaN	NaN
14	20001	Researchers (FTE) - Total	CHL	Chile	2016	2016	8992.52200	NaN	NaN
15	20001	Researchers (FTE) - Total	COL	Colombia	2011	2011	9119.80982	NaN	NaN
16	20001	Researchers (FTE) - Total	COL	Colombia	2012	2012	8812.94661	NaN	NaN
17	20001	Researchers (FTE) - Total	COL	Colombia	2013	2013	7820.02607	NaN	NaN
18	20001	Researchers (FTE) - Total	COL	Colombia	2014	2014	5651.22725	NaN	NaN
19	20001	Researchers (FTE) - Total	COL	Colombia	2015	2015	6364.25000	NaN	NaN
20	20001	Researchers (FTE) - Total	ECU	Ecuador	2011	2011	2736.39444	NaN	NaN

Pandas :: *I/O*

- Exportando dados com o Pandas
  - ***pd.to_csv***

```
del dfIn['INDICATOR']
```

```
del dfIn['LOCATION']
```

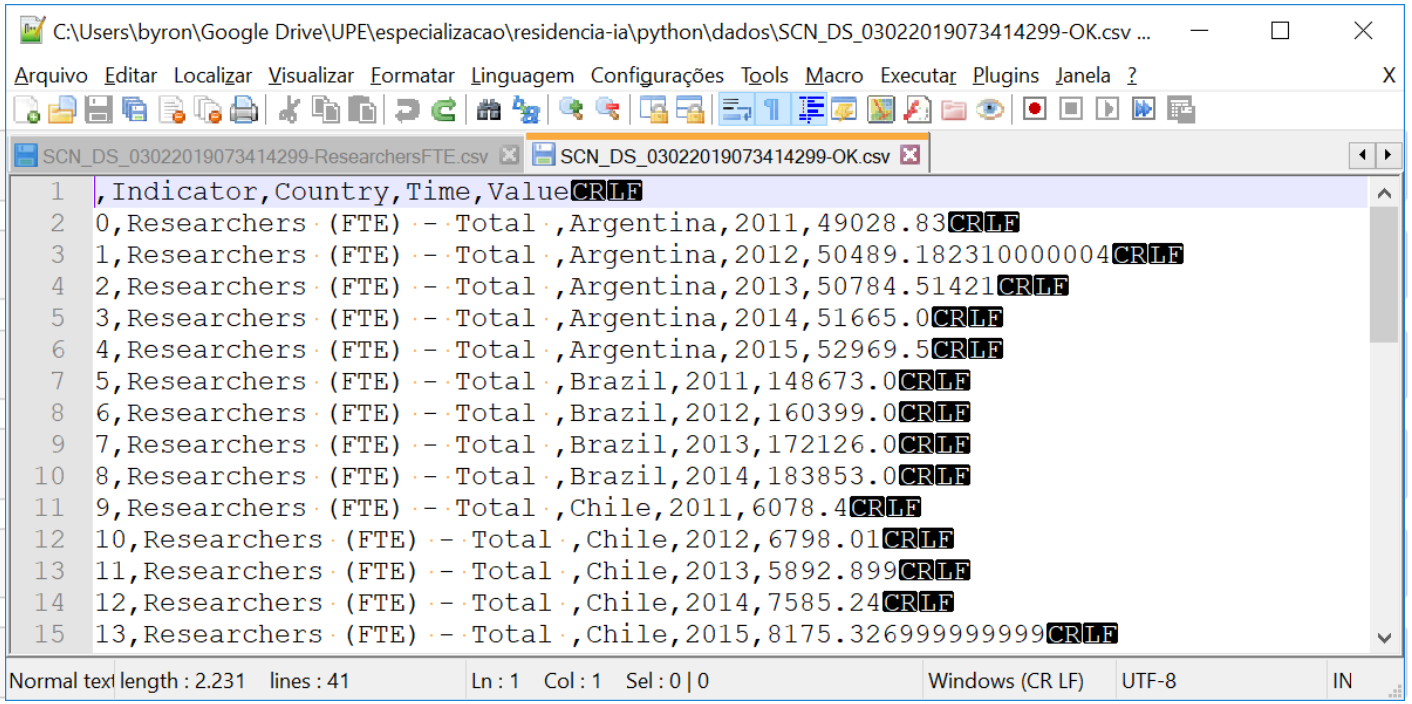
```
del dfIn['TIME']
```

```
del dfIn['Flag Codes']
```

```
del dfIn['Flags']
```

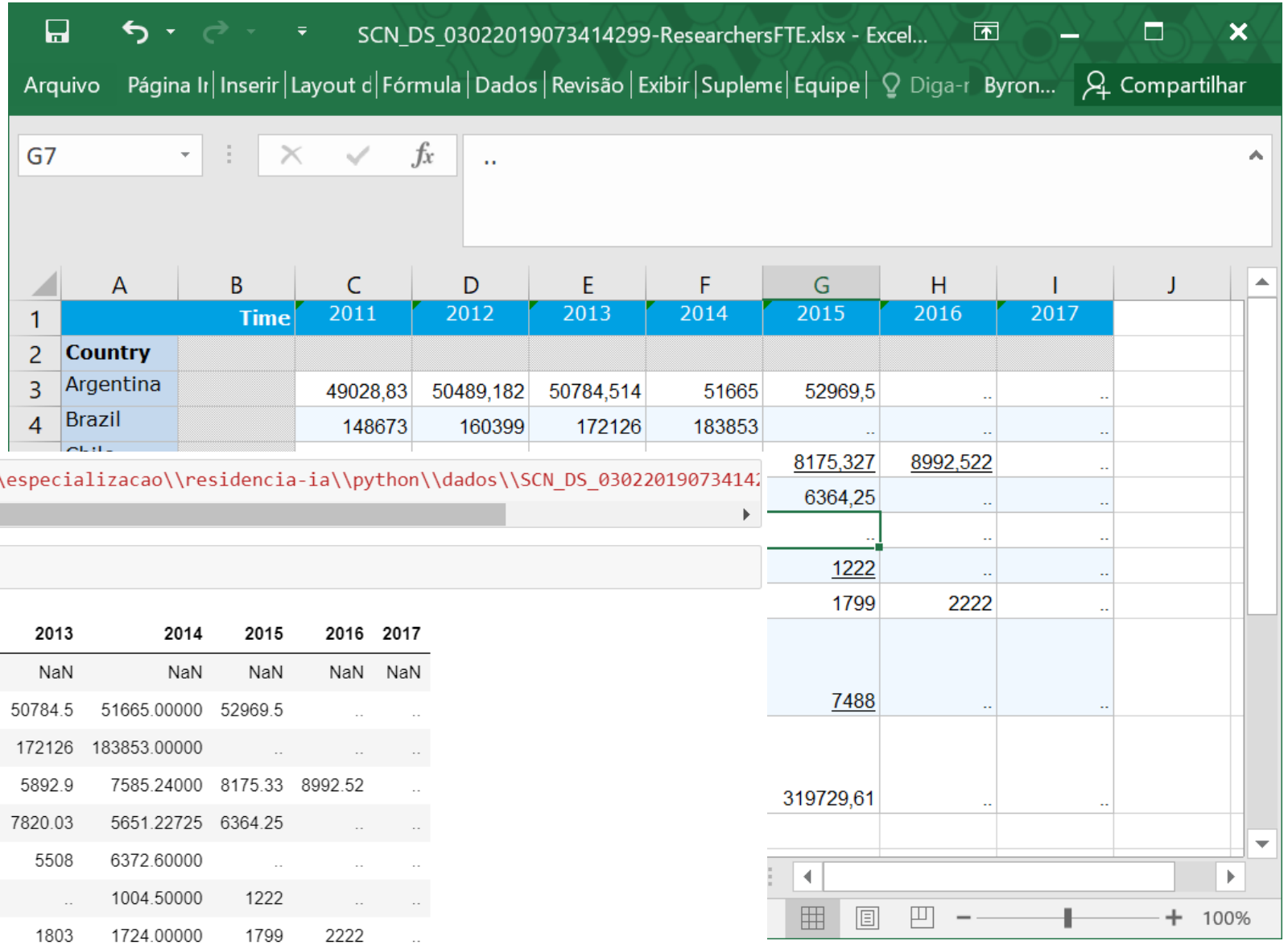
```
dfIn.columns
```

```
Index(['Indicator', 'Country', 'Time', 'Value'], dtype='object')
```




```
dfIn.to_csv('C:\\Users\\byron\\Google Drive\\UPE\\especializacao\\residencia-ia\\python\\dados\\SCN_DS_03022019073414299-OK.csv')
```

- Importando dados do Excel
  – ***pd.read_excel***




```
dfInXls = pd.read_excel('C:\\Users\\byron\\Google Drive\\UPE\\especializacao\\residencia-ia\\python\\dados\\SCN_DS_03022019073414
```

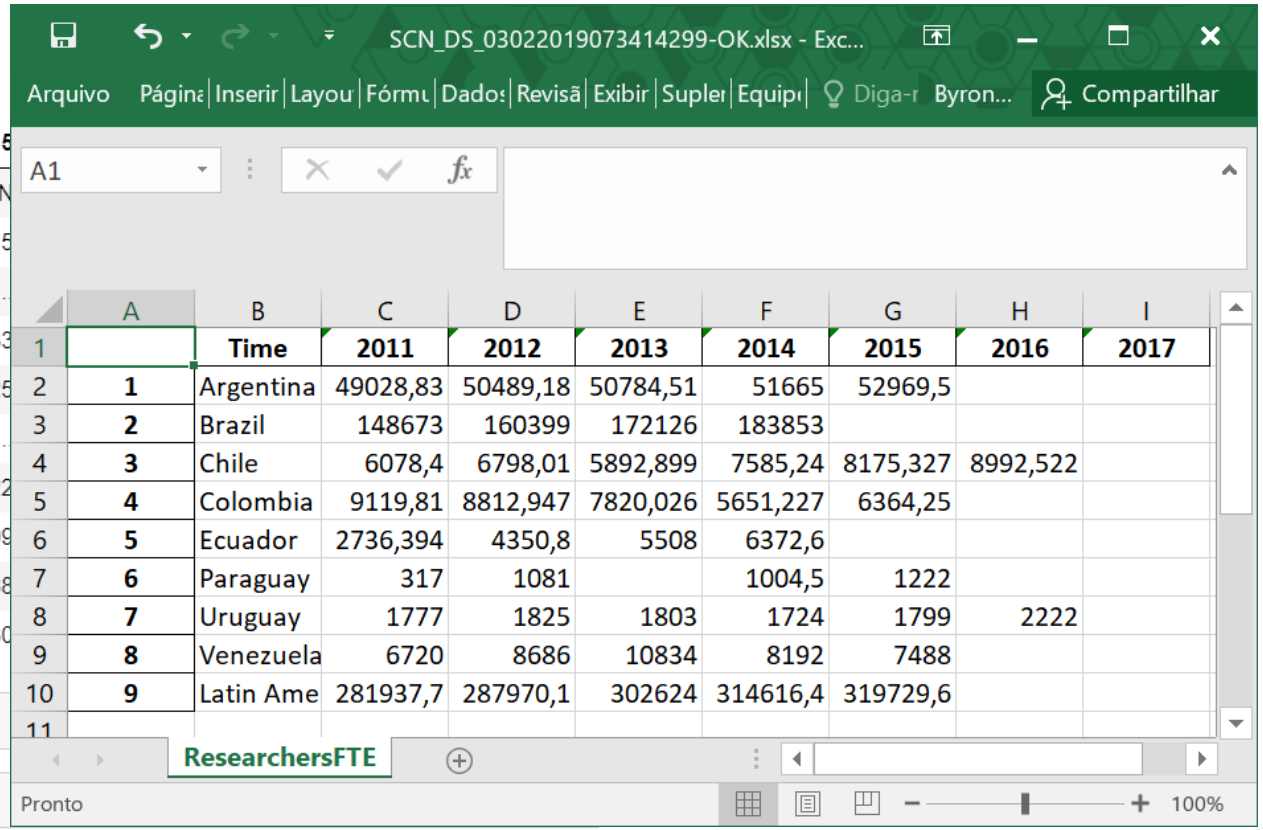
```
dfInXls
```

	Time	Unnamed: 1	2011	2012	2013	2014	2015	2016	2017
0	Country	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Argentina	NaN	49028.83000	50489.18231	50784.5	51665.00000	52969.5	..	..
2	Brazil	NaN	148673.00000	160399.00000	172126	183853.00000	..	..	..
3	Chile	NaN	6078.40000	6798.01000	5892.9	7585.24000	8175.33	8992.52	..
4	Colombia	NaN	9119.80982	8812.94661	7820.03	5651.22725	6364.25	..	..
5	Ecuador	NaN	2736.39444	4350.80000	5508	6372.60000	..	..	..
6	Paraguay	NaN	317.00000	1081.00000	..	1004.50000	1222	..	..
7	Uruguay	NaN	1777.00000	1825.00000	1803	1724.00000	1799	2222	..
8	Venezuela (Bolivarian Republic of)	NaN	6720.00000	8686.00000	10834	8192.00000	7488	..	..
9	Latin America and the Caribbean	NaN	281937.68509	287970.10495	302624	314616.39137	319730	..	..

Pandas :: *I/O*

- Exportando dados com o Pandas
  - ***pd.to_excel***

	Time	Unnamed: 1	2011	2012	2013	2014	2015
0	Country	NaN	NaN	NaN	NaN	NaN	NaN
1	Argentina	NaN	49028.83000	50489.18231	50784.5	51665.00000	52969.5
2	Brazil	NaN	148673.00000	160399.00000	172126	183853.00000	
3	Chile	NaN	6078.40000	6798.01000	5892.9	7585.24000	8175.33
4	Colombia	NaN	9119.80982	8812.94661	7820.03	5651.22725	6364.25
5	Ecuador	NaN	2736.39444	4350.80000	5508	6372.60000	
6	Paraguay	NaN	317.00000	1081.00000	..	1004.50000	1222
7	Uruguay	NaN	1777.00000	1825.00000	1803	1724.00000	1799
8	Venezuela (Bolivarian Republic of)	NaN	6720.00000	8686.00000	10834	8192.00000	7488
9	Latin America and the Caribbean	NaN	281937.68509	287970.10495	302624	314616.39137	319730



	A	B	C	D	E	F	G	H	I
1		Time	2011	2012	2013	2014	2015	2016	2017
2	1	Argentina	49028,83	50489,18	50784,51	51665	52969,5		
3	2	Brazil	148673	160399	172126	183853			
4	3	Chile	6078,4	6798,01	5892,899	7585,24	8175,327	8992,522	
5	4	Colombia	9119,81	8812,947	7820,026	5651,227	6364,25		
6	5	Ecuador	2736,394	4350,8	5508	6372,6			
7	6	Paraguay	317	1081		1004,5	1222		
8	7	Uruguay	1777	1825	1803	1724	1799	2222	
9	8	Venezuela	6720	8686	10834	8192	7488		
10	9	Latin Ame	281937,7	287970,1	302624	314616,4	319729,6		
11									

```
del dfInXls['Unnamed: 1']
```

```
dfInXls.drop(index=0, axis=1, inplace=True)
```

```
dfInXls.replace('..', np.nan, inplace=True)
```

```
dfInXls.to_excel('C:\\Users\\byron\\Google Drive\\UPE\\especializacao\\residencia-ia\\python\\dados\\SCN_DS_03022019073414299-O
                 sheet_name='ResearchersFTE')
```

Failed Bank List

The FDIC is often appointed as receiver for failed banks. This page contains useful information for the customers and vendors of these banks. This includes information on the acquiring bank (if applicable), how your accounts and loans are affected, and how vendors can file claims against the receivership.

This list includes banks which have failed since October 1, 2000. To search for banks that failed prior to those on this page, visit this link: Failures and Assistance Transactions

Click arrows next to headers to sort in Ascending or Descending order.

Download Data

Search:

Show 25 **entries**

Showing 1 to 25 of 555 entries

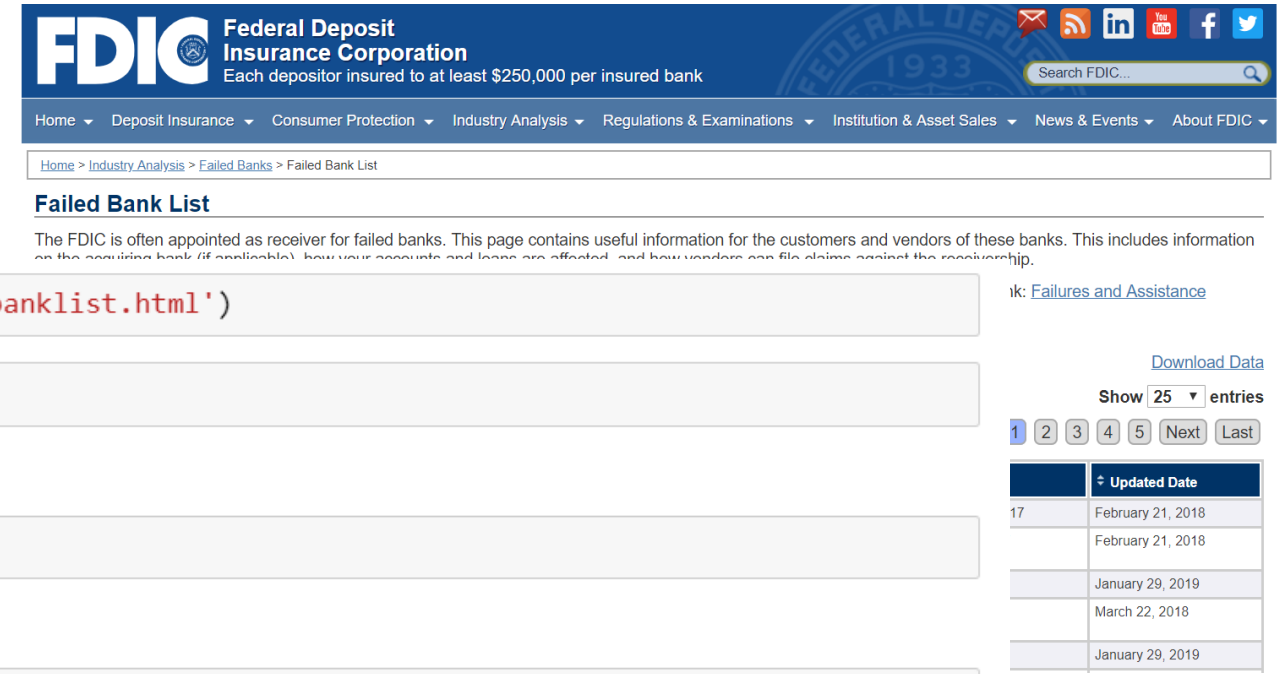
First Previous 1 2 3 4 5 Next Last

Bank Name	City	ST	CERT	Acquiring Institution	Closing Date	Updated Date
Washington Federal Bank for Savings	Chicago	IL	30570	Royal Savings Bank	December 15, 2017	February 21, 2018
The Farmers and Merchants State Bank of Argonia	Argonia	KS	17719	Conway Bank	October 13, 2017	February 21, 2018
Fayette County Bank	Saint Elmo	IL	1802	United Fidelity Bank, fsb	May 26, 2017	January 29, 2019
Guaranty Bank, (d/b/a BestBank in Georgia & Michigan)	Milwaukee	WI	30003	First-Citizens Bank & Trust Company	May 5, 2017	March 22, 2018
First NBC Bank	New Orleans	LA	58302	Whitney Bank	April 28, 2017	January 29, 2019

Fonte: https://www.fdic.gov/bank/individual/failed/banklist.html

Pandas :: *I/O*

- Importando dados da Internet
  – ***pd.read_html***




```
dfInHtml = pd.read_html('https://www.fdic.gov/bank/individual/failed/banklist.html')
```

```
type(dfInHtml)
```

```
list
```

```
len(dfInHtml)
```

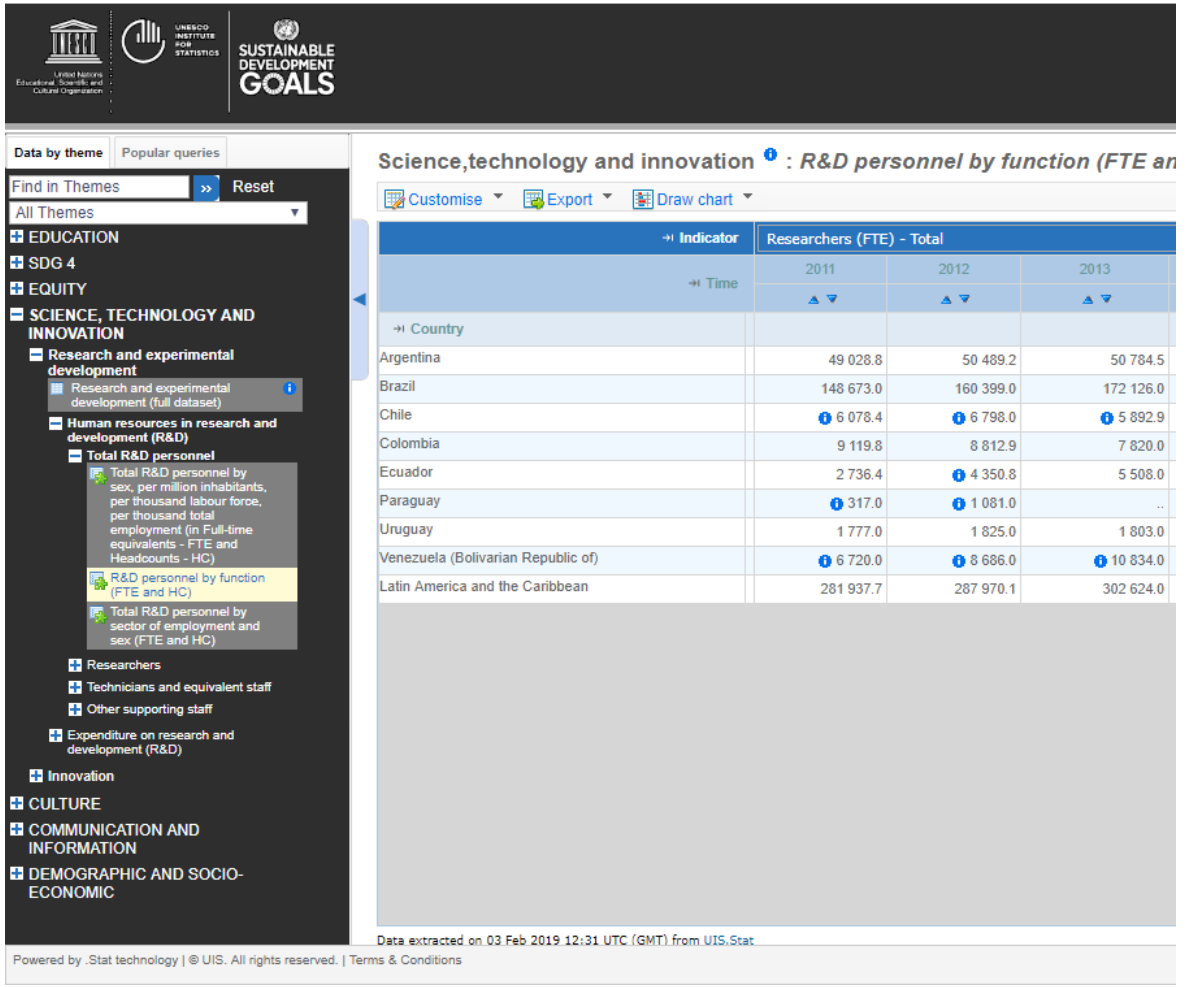
```
1
```

```
dfInHtml[0]
```

	Bank Name	City	ST	CERT	Acquiring Institution	Closing Date	Updated Date
0	Washington Federal Bank for Savings	Chicago	IL	30570	Royal Savings Bank	December 15, 2017	February 21, 2018
1	The Farmers and Merchants State Bank of Argonia	Argonia	KS	17719	Conway Bank	October 13, 2017	February 21, 2018
2	Fayette County Bank	Saint Elmo	IL	1802	United Fidelity Bank, fsb	May 26, 2017	January 29, 2019
3	Guaranty Bank, (d/b/a BestBank in Georgia & Mi...	Milwaukee	WI	30003	First-Citizens Bank & Trust Company	May 5, 2017	March 22, 2018
4	First NBC Bank	New Orleans	LA	58302	Whitney Bank	April 28, 2017	January 29, 2019
5	Proficio Bank	Cottonwood Heights	UT	35495	Cache Valley Bank	March 3, 2017	January 29, 2019

Exercício

- Crie um DataFrame para representar os dados da tabela abaixo obtidos a partir do site http://data.uis.unesco.org/ conforme tela ao lado, utilizando algum método de importação
- Limpe as linhas e colunas do DataFrame importado que não tem relevância e escolha apenas 5 países em um período de 5 anos
- Construa uma função que recebe o DataFrame criado e um ano e retorna o país com melhor colocação no ano



Dica: ***https://pandas.pydata.org/pandas-docs/stable/reference/series.html***

Fonte: ***http://data.uis.unesco.org/***