Ciência de Dados

Universidade Presbiteriana Mackenzie

Paradigmas de Linguagens de Programação para Ciência de Dados

Rogério de Oliveira

Este é um material de apoio para Atividade de Aprofundamento 2. Ele mostra como obter e tratar dados do [Gapminder]() e do [WID](). Você pode empregar uma ou ambas as fontes de dados.

Gapminder

Essa é uma importante fonte de dados aberta que contêm diversas informações e índices relacionados ao desenvolvimento dos países.

Acesse [aqui]() para extrair os dados.

Escolha os dados de seu interesse. Faça o download no formato `.csv` para o local que desejar.

```
ent.com/rodglins/Python/master/desafios/exploracaoDados/income_per_person_gdppercapita
```

	country	1799	1800	1801	1802	1803	1804	1805	1806	1807	1808	1809	181
0	Afghanistan	674	674	674	674	674	674	674	674	674	674	675	67
1	Angola	691	693	697	700	702	705	709	712	716	718	721	72

Preparação dos Dados

Vamos obter aqui dados de escolaridade e emissões de co2 do Brasil. O Gapminder fornece esses dados em conjuntos separados e vamos combinar esses dados para nossa análise.

5 rows × 252 columns

```
income_BR = income[ income.country == 'Brazil' ]
income_BR
```

	country	1799	1800	1801	1802	1803	1804	1805	1806	1807	1808	1809	1810
23	Brazil	997	997	997	997	997	997	997	997	997	997	997	997

1 rows × 252 columns

```
BR = pd.melt(income_BR, id_vars=['country'])
BR.head()
```

	country	variable	value
0	Brazil	1799	997
1	Brazil	1800	997
2	Brazil	1801	997
3	Brazil	1802	997
4	Brazil	1803	997

```
BR = BR.rename(columns={'variable':'year','value':'income'})
BR.head()
```

	country	year	income
0	Brazil	1799	997
1	Brazil	1800	997
2	Brazil	1801	997
3	Brazil	1802	997
4	Brazil	1803	997

```
sv('https://raw.githubusercontent.com/rodglins/Python/master/desafios/exploracaoDados/
ndustry.country == 'Brazil' ]
```

	country	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970
24	Brazil	31.8	36.6	29.8	34.8	32.0	29.9	29.8	29.2	30.7	31.3	32.2	32.7

```
ind_BR = pd.melt(ind_BR, id_vars=['country'])
ind_BR = ind_BR.rename(columns={'variable':'year','value':'industry'})
ind_BR.head()
```

	country	year	industry
0	Brazil	1959	31.8
1	Brazil	1960	36.6
2	Brazil	1961	29.8
3	Brazil	1962	34.8
4	Brazil	1963	32.0

```
BR·=·pd.merge(BR,ind_BR,on=['country','year'])
BR.head()
```

	country	year	income	industry
0	Brazil	1959	3910	31.8
1	Brazil	1960	4150	36.6
2	Brazil	1961	4320	29.8
3	Brazil	1962	4240	34.8
4	Brazil	1963	4280	32.0

```
display(BR.dtypes)
```

```
country      object
year         object
income       object
industry    float64
dtype: object
```

```
# BR.year·=·pd.to_datetime(BR.year,·format='%Y',errors='coerce')
# display(BR.dtypes)
```

```
BR.income = pd.to_numeric(BR.income,errors='coerce')
display(BR.dtypes)
```

```
country      object
year         object
income      float64
```

Visualização e Análise dos Dados

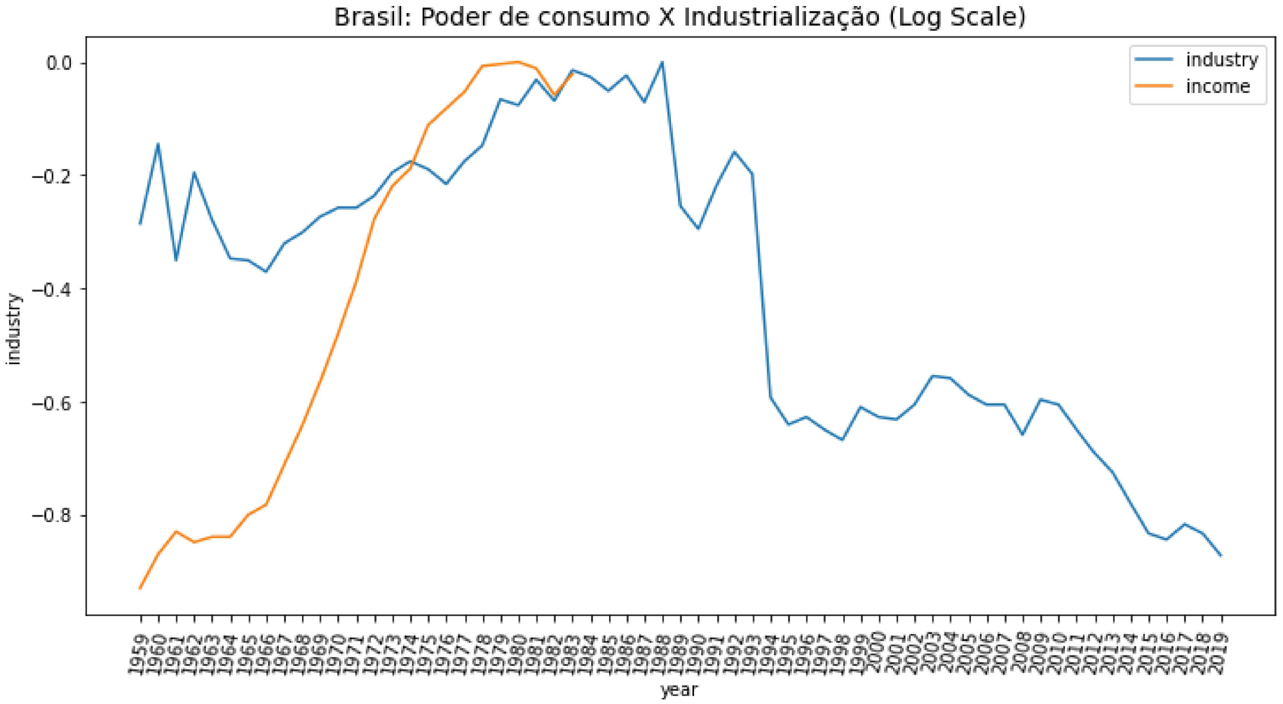
```
import·matplotlib.pyplot·as·plt
import·numpy·as·np·
import·seaborn·as·sns
%matplotlib·inline·
```

```
for·c·in·BR[['industry','income']]:
··BR[c]·=·BR[c]·/·BR[c].max()
BR.head()
```

	country	year	income	industry
0	Brazil	1959	0.394949	0.751773
1	Brazil	1960	0.419192	0.865248
2	Brazil	1961	0.436364	0.704492
3	Brazil	1962	0.428283	0.822695
4	Brazil	1963	0.432323	0.756501

```
plt.figure(figsize=(12,6))

sns.lineplot(x=BR.year, y=BR.income, label='income')
sns.lineplot(x=BR.year, y=BR.industry, label='industry')

plt.title('Brasil: Poder de consumo X Industrialização', fontsize=14)
plt.legend()
plt.xticks(rotation=90)
plt.show()
```


```
import numpy as np 
plt.figure(figsize=(12,6))

sns.lineplot(x=BR.year, y=np.log( BR.industry ), label='industry')
sns.lineplot(x=BR.year, y=np.log( BR.income ), label='income')

plt.title('Brasil: Poder de consumo X Industrialização (Log Scale)', fontsize=14)
plt.legend()
plt.xticks(rotation=80)
plt.show()
```



World Inequality Database

Vamos ver como combinar dados de outras fontes?

O World Inequality Database é uma base de dados aberta que mantem informações sobre desigualdade e concentração de renda no mundo. Esses dados são a base do livro **Capital in the Twenty-First Century** de *Thomas Piketty*.

```
from IPython.display import IFrame
IFrame('https://wid.world/data/', width='1000', height=400)
```

Se você for empregar essa base escolha os índices de interesse, a estrutura da tabela e dê preferência para o formato `.xlsx` para download.

Preparação dos Dados

Vamos selecionar dados do Brasil de 2000-2019 sobre a renda per capta (gpd) e o percentual da renda concentrado nos 10% mais ricos da população (percentil 10), e combinar esses dados com as informações que já coletamos do gapminder.

```
d.read_excel('https://github.com/rodglins/Python/raw/master/desafios/exploracaoDados/W
```

	0	1	2	3	4
0	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1980	0.2521
1	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1981	0.2521
2	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1982	0.2521

```
gpd_BR.columns = ['country', 'ind_description',  'ind_code', 'year', 'value']
gpd_BR.year·=·gpd_BR.year.astype(str)
display(gpd_BR)
display(gpd_BR.dtypes)
```

	country	ind_description	ind_code	year	value
0	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1980	0.2521
1	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1981	0.2521
2	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1982	0.2521
3	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1983	0.2521
4	Brazil	sptinc_p99p100_z_BR\nPre-tax national income \...	p99p100	1984	0.2521
...	...	...	...	...	...
75	Brazil	sptinc_p0p50_z_BR\nPre-tax national income \nB...	p0p50	2015	0.1062
76	Brazil	sptinc_p0p50_z_BR\nPre-tax national income \nB...	p0p50	2016	0.0991
77	Brazil	sptinc_p0p50_z_BR\nPre-tax national income \nB...	p0p50	2017	0.0991
78	Brazil	sptinc_p0p50_z_BR\nPre-tax national income \nB...	p0p50	2018	0.1015
79	Brazil	sptinc_p0p50_z_BR\nPre-tax national income \nB...	p0p50	2019	0.1007

80 rows × 5 columns

```
country            object
ind_description    object
ind_code           object
year               object
value             float64
dtype: object
```

```
gpd_BR_all = gpd_BR[ gpd_BR.ind_code == 'p99p100' ][['country', 'year', 'value']]
gpd_BR_perc = gpd_BR[ gpd_BR.ind_code == 'p0p50' ][['country', 'year', 'value']]

display(gpd_BR_all)
display(gpd_BR_perc)
```

	country	year	value
0	Brazil	1980	0.2521
1	Brazil	1981	0.2521
2	Brazil	1982	0.2521
3	Brazil	1983	0.2521
4	Brazil	1984	0.2521
5	Brazil	1985	0.2521
6	Brazil	1986	0.2521
7	Brazil	1987	0.2521
8	Brazil	1988	0.2521
9	Brazil	1989	0.2521
10	Brazil	1990	0.2521
11	Brazil	1991	0.2521
12	Brazil	1992	0.2521
13	Brazil	1993	0.2521
14	Brazil	1994	0.2521
15	Brazil	1995	0.2521
16	Brazil	1996	0.2521
17	Brazil	1997	0.2521
18	Brazil	1998	0.2521
19	Brazil	1999	0.2521
20	Brazil	2000	0.2468
21	Brazil	2001	0.2468
22	Brazil	2002	0.2373
23	Brazil	2003	0.2462
24	Brazil	2004	0.2555
25	Brazil	2005	0.2531
26	Brazil	2006	0.2653
27	Brazil	2007	0.2280
28	Brazil	2008	0.2678
29	Brazil	2009	0.2792
30	Brazil	2010	0.2805
31	Brazil	2011	0.2819
32	Brazil	2012	0.2988

32	Brazil	2012	0.2988
33	Brazil	2013	0.2705
34	Brazil	2014	0.2627
35	Brazil	2015	0.2625
36	Brazil	2016	0.2649
37	Brazil	2017	0.2742
38	Brazil	2018	0.2471
39	Brazil	2019	0.2660

	country	year	value
40	Brazil	1980	0.1086
41	Brazil	1981	0.1086
42	Brazil	1982	0.1086
43	Brazil	1983	0.1086
44	Brazil	1984	0.1086
45	Brazil	1985	0.1086
46	Brazil	1986	0.1086
47	Brazil	1987	0.1086
48	Brazil	1988	0.1086
49	Brazil	1989	0.1086
50	Brazil	1990	0.1086
51	Brazil	1991	0.1086
52	Brazil	1992	0.1086
53	Brazil	1993	0.1086
54	Brazil	1994	0.1086
55	Brazil	1995	0.1086
56	Brazil	1996	0.1086
57	Brazil	1997	0.1086
58	Brazil	1998	0.1086
59	Brazil	1999	0.1086
60	Brazil	2000	0.1094
61	Brazil	2001	0.1094
62	Brazil	2002	0.1201
63	Brazil	2003	0.1097
64	Brazil	2004	0.1109

65	Brazil	2005	0.1114
66	Brazil	2006	0.1101
67	Brazil	2007	0.1226
68	Brazil	2008	0.1083
69	Brazil	2009	0.1070
70	Brazil	2010	0.1058
71	Brazil	2011	0.1046
72	Brazil	2012	0.1098
73	Brazil	2013	0.1091
74	Brazil	2014	0.1090
75	Brazil	2015	0.1062
76	Brazil	2016	0.0991

```
BR = pd.merge(BR,gpd_BR_all,on=['country','year'])
BR = BR.rename(columns={'value':'gpd_perc1'})
BR.head()
```

	country	year	income	industry	gpd_perc1
0	Brazil	1980	1.000000	0.926714	0.2521
1	Brazil	1981	0.988889	0.969267	0.2521
2	Brazil	1982	0.943434	0.933806	0.2521
3	Brazil	1983	0.978788	0.985816	0.2521
4	Brazil	1984	NaN	0.973995	0.2521

```
BR = pd.merge(BR,gpd_BR_perc,on=['country','year'])
BR = BR.rename(columns={'value':'gpd_perc50'})
BR.head()
```

	country	year	income	industry	gpd_perc1	gpd_perc50
0	Brazil	1980	1.000000	0.926714	0.2521	0.1086
1	Brazil	1981	0.988889	0.969267	0.2521	0.1086
2	Brazil	1982	0.943434	0.933806	0.2521	0.1086
3	Brazil	1983	0.978788	0.985816	0.2521	0.1086
4	Brazil	1984	NaN	0.973995	0.2521	0.1086

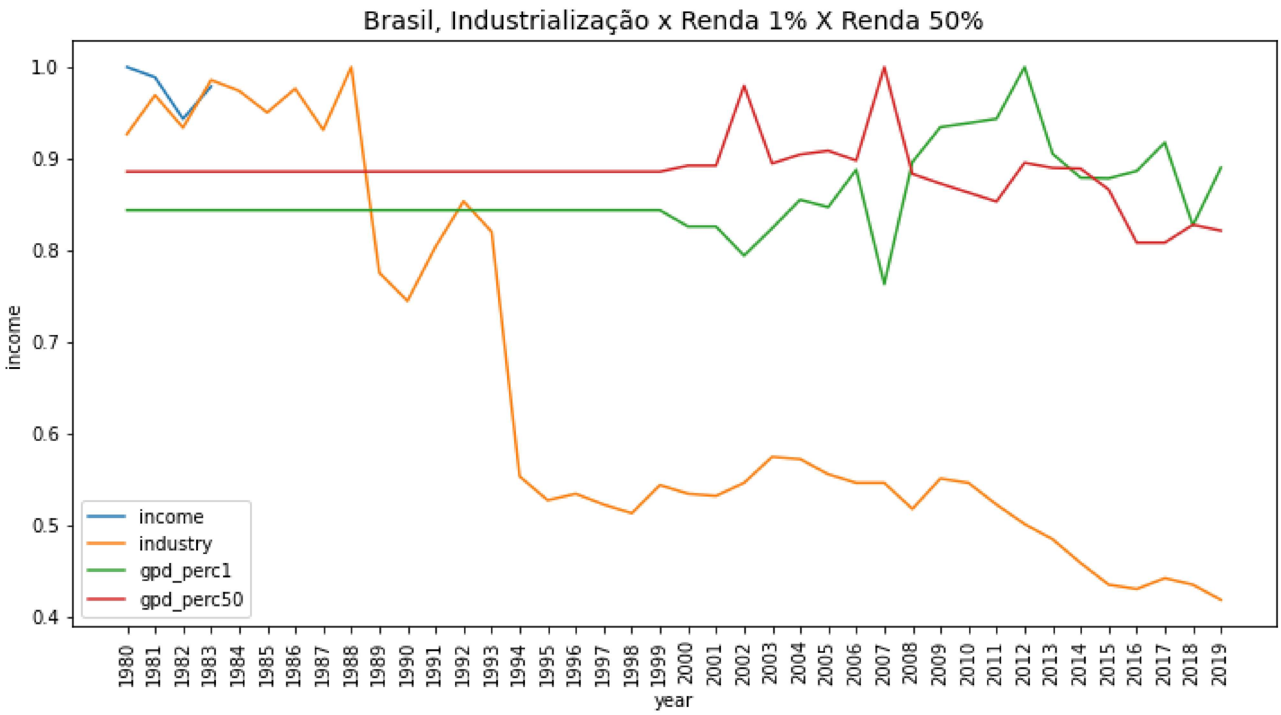
Visualização e Análise dos Dados

Como queremos comparar dados em escalas muito diferentes uma sugestão é empregarmos dados normalizados.

```
for c in BR[['income','industry','gpd_perc1','gpd_perc50']]:
  BR[c] = BR[c] / BR[c].max()
BR.head()
```

	country	year	income	industry	gpd_perc1	gpd_perc50
0	Brazil	1980	1.000000	0.926714	0.843708	0.885808
1	Brazil	1981	0.988889	0.969267	0.843708	0.885808
2	Brazil	1982	0.943434	0.933806	0.843708	0.885808
3	Brazil	1983	0.978788	0.985816	0.843708	0.885808
4	Brazil	1984	NaN	0.973995	0.843708	0.885808

```
plt.figure(figsize=(12,6))

for c in BR[['income','industry','gpd_perc1','gpd_perc50']]:
  sns.lineplot(x=BR.year, y=BR[c], label=c)

plt.title('Brasil, Industrialização x Renda 1% X Renda 50%', fontsize=14)
plt.legend()
plt.xticks(rotation=90)
plt.show()
```



Conclusões