

Equipe 148

Analizador de Desempenho do PBF

O Bolsa Família (PBF) é um programa que contribui para o combate à pobreza e à desigualdade no Brasil. Ele foi criado em outubro de 2003 e possui três eixos principais:

- Complemento da renda: todos os meses, as famílias atendidas pelo Programa recebem um benefício em dinheiro, que é transferido diretamente pelo governo federal. Esse eixo garante o alívio mais imediato da pobreza.
- Acesso a direitos: as famílias devem cumprir alguns compromissos (condicionalidades), que têm como objetivo reforçar o acesso à educação, à saúde e à assistência social. Esse eixo oferece condições para as futuras gerações quebrarem o ciclo da pobreza, graças a melhores oportunidades de inclusão social.
- Articulação com outras ações: o Bolsa Família tem capacidade de integrar e articular várias políticas sociais a fim de estimular o desenvolvimento das famílias, contribuindo para elas superarem a situação de vulnerabilidade e de pobreza.

A gestão do Bolsa Família é descentralizada, ou seja, tanto a União, quanto os estados, o Distrito Federal e os municípios têm atribuições em sua execução. Em nível federal, o Ministério do Desenvolvimento Social (MDS) é o responsável pelo Programa, e a Caixa Econômica Federal é o agente que executa os pagamentos.

O Programa Bolsa Família está previsto em lei — Lei Federal nº 10.836, de 9 de janeiro de 2004 — e é regulamentado pelo Decreto nº 5.209, de 17 de setembro de 2004, e outras normas.

Ministério do Desenvolvimento Social: <http://mds.gov.br/assuntos/bolsa-familia/o-que-e> (<http://mds.gov.br/assuntos/bolsa-familia/o-que-e>)

Sobre o Analizador de Desempenho do PBF

Este trabalho tem por finalidade demonstrar um ambiente para análise de desempenho e eficácia do PBF, a partir de cruzamento de dados e predição (baseada em regressão linear).

Apesar de ter como foco o PBF, este ambiente de análise pode ser aplicado para outros conjuntos de dados, de forma semelhante.

As simulações e análises realizadas neste ambiente são meramente ilustrativas e para fins didáticos, não podendo servir como base para tomadas de decisão por gestores ou administradores dos programas sociais relacionados.

Script do Analizador de Desempenho do PBF (Python / Jupyter Notebook)

```
In [1]: from sqlalchemy import create_engine
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing

%matplotlib inline
```

Conexão com banco IPEA

```
In [2]: ipea_engine = create_engine('postgresql://equipe148:XBHM2@gbd2.desafioinova  
cao.serpro:5432/ipea')
```

Query para banco IPEA

```
In [3]: sql = '''  
SELECT  
    ano,  
    (CASE WHEN UPPER(nome_unidade_federacao)='ACRE' THEN 'AC'  
          WHEN UPPER(nome_unidade_federacao)='ALAGOAS' THEN 'AL'  
          WHEN UPPER(nome_unidade_federacao)='AMAZONAS' THEN 'AM'  
          WHEN UPPER(nome_unidade_federacao)='AMAPÁ' THEN 'AP'  
          WHEN UPPER(nome_unidade_federacao)='BAHIA' THEN 'BA'  
          WHEN UPPER(nome_unidade_federacao)='CEARÁ' THEN 'CE'  
          WHEN UPPER(nome_unidade_federacao)='DISTRITO FEDERAL' THEN 'DF'  
          WHEN UPPER(nome_unidade_federacao)='ESPÍRITO SANTO' THEN 'ES'  
          WHEN UPPER(nome_unidade_federacao)='GOIÁS' THEN 'GO'  
          WHEN UPPER(nome_unidade_federacao)='MARANHÃO' THEN 'MA'  
          WHEN UPPER(nome_unidade_federacao)='MINAS GERAIS' THEN 'MG'  
          WHEN UPPER(nome_unidade_federacao)='MATO GROSSO DO SUL' THEN 'MS'  
          WHEN UPPER(nome_unidade_federacao)='MATO GROSSO' THEN 'MT'  
          WHEN UPPER(nome_unidade_federacao)='PARÁ' THEN 'PA'  
          WHEN UPPER(nome_unidade_federacao)='PARAÍBA' THEN 'PB'  
          WHEN UPPER(nome_unidade_federacao)='PERNAMBUCO' THEN 'PE'  
          WHEN UPPER(nome_unidade_federacao)='PIAUÍ' THEN 'PI'  
          WHEN UPPER(nome_unidade_federacao)='PARANÁ' THEN 'PR'  
          WHEN UPPER(nome_unidade_federacao)='RIO DE JANEIRO' THEN 'RJ'  
          WHEN UPPER(nome_unidade_federacao)='RIO GRANDE DO NORTE' THEN 'RN'  
          WHEN UPPER(nome_unidade_federacao)='RONDÔNIA' THEN 'RO'  
          WHEN UPPER(nome_unidade_federacao)='RORAIMA' THEN 'RR'  
          WHEN UPPER(nome_unidade_federacao)='RIO GRANDE DO SUL' THEN 'RS'  
          WHEN UPPER(nome_unidade_federacao)='SANTA CATARINA' THEN 'SC'  
          WHEN UPPER(nome_unidade_federacao)='SERGIPE' THEN 'SE'  
          WHEN UPPER(nome_unidade_federacao)='SÃO PAULO' THEN 'SP'  
          WHEN UPPER(nome_unidade_federacao)='TOCANTINS' THEN 'TO'  
          ELSE NULL  
        END) AS uf,  
    UPPER(municipio) AS municipio,  
    perc_crianças_6_14_fora_escola,  
    perc_extremamente_pobres, perc_pobres,  
    perc_mulheres_10_14_anos_que_tiveram_filhos, perc_mulheres_15_17_anos_que  
    _tiveram_filhos  
FROM  
    {}  
WHERE  
    municipio IS NOT NULL AND  
    perc_crianças_6_14_fora_escola IS NOT NULL AND  
    perc_extremamente_pobres IS NOT NULL AND  
    perc_pobres IS NOT NULL AND  
    perc_mulheres_10_14_anos_que_tiveram_filhos IS NOT NULL AND  
    perc_mulheres_15_17_anos_que_tiveram_filhos IS NOT NULL  
ORDER BY  
    uf, municipio  
;'''
```

Dados econômicos e sociais de 1991

```
In [4]: df = pd.read_sql_query(sql.format('idhm1991'), con=ipea_engine)
ipea1991 = df
ipea1991.head()
```

Out[4]:

	ano	uf	municipio	perc_crianças_6_14_fora_escola	perc_extremamente_pobres	perc_po
0	1991	AC	ACRELÂNDIA	40.56	26.93	55.59
1	1991	AC	ASSIS BRASIL	44.69	24.55	58.49
2	1991	AC	BRASILÉIA	47.48	33.96	63.85
3	1991	AC	BUJARI	54.12	28.12	51.74
4	1991	AC	CAPIXABA	69.53	29.05	56.27

Dados econômicos e sociais de 2000

```
In [5]: df = pd.read_sql_query(sql.format('idhm2000'), con=ipea_engine)
ipea2000 = df
ipea2000.head()
```

Out[5]:

	ano	uf	municipio	perc_crianças_6_14_fora_escola	perc_extremamente_pobres	perc_po
0	2000	AC	ACRELÂNDIA	12.42	27.43	51.78
1	2000	AC	ASSIS BRASIL	15.39	37.77	57.08
2	2000	AC	BRASILÉIA	21.54	27.64	48.49
3	2000	AC	BUJARI	17.94	32.16	61.62
4	2000	AC	CAPIXABA	36.93	22.87	56.03

Dados econômicos e sociais de 2010

```
In [6]: df = pd.read_sql_query(sql.format('idhm2010'), con=ipea_engine)
ipea2010 = df
ipea2010.head()
```

Out[6]:

	ano	uf	municipio	perc_crianças_6_14_fora_escola	perc_extremamente_pobres	perc_po
0	2010	AC	ACRELÂNDIA	4.89	18.09	36.56
1	2010	AC	ASSIS BRASIL	14.93	30.66	44.09
2	2010	AC	BRASILÉIA	9.81	17.09	35.22
3	2010	AC	BUJARI	8.89	20.68	37.61
4	2010	AC	CAPIXABA	7.40	17.89	35.00

Dados econômicos e sociais agregados

```
In [7]: ipea = ipea1991.append(ipea2000).append(ipea2010)
del df, ipea1991, ipea2000, ipea2010
ipea.head()
```

Out [7]:

	ano	uf	município	perc_crianças_6_14_fora_escola	perc_extremamente_pobres	perc_po
0	1991	AC	ACRELÂNDIA	40.56	26.93	55.59
1	1991	AC	ASSIS BRASIL	44.69	24.55	58.49
2	1991	AC	BRASILÉIA	47.48	33.96	63.85
3	1991	AC	BUJARI	54.12	28.12	51.74
4	1991	AC	CAPIXABA	69.53	29.05	56.27

Dados do PBF

Dados em CSV baixados do site <https://aplicacoes.mds.gov.br/sagi/vis/data/data-table.php> (<https://aplicacoes.mds.gov.br/sagi/vis/data/data-table.php>)

Pagamento de benefícios por município, no período entre 2004 e 2010.

```
In [8]: pbf = pd.read_csv('pbf.csv', sep=';')

pbf2004 = pbf[['uf', 'município', 'qtd_2004', 'vlr_2004']].rename(columns={'uf': 'uf', 'município': 'município', 'qtd_2004': 'qtd', 'vlr_2004': 'valor'}).assign(ano=2004)
pbf2005 = pbf[['uf', 'município', 'qtd_2005', 'vlr_2005']].rename(columns={'uf': 'uf', 'município': 'município', 'qtd_2005': 'qtd', 'vlr_2005': 'valor'}).assign(ano=2005)
pbf2006 = pbf[['uf', 'município', 'qtd_2006', 'vlr_2006']].rename(columns={'uf': 'uf', 'município': 'município', 'qtd_2006': 'qtd', 'vlr_2006': 'valor'}).assign(ano=2006)
pbf2007 = pbf[['uf', 'município', 'qtd_2007', 'vlr_2007']].rename(columns={'uf': 'uf', 'município': 'município', 'qtd_2007': 'qtd', 'vlr_2007': 'valor'}).assign(ano=2007)
pbf2008 = pbf[['uf', 'município', 'qtd_2008', 'vlr_2008']].rename(columns={'uf': 'uf', 'município': 'município', 'qtd_2008': 'qtd', 'vlr_2008': 'valor'}).assign(ano=2008)
pbf2009 = pbf[['uf', 'município', 'qtd_2009', 'vlr_2009']].rename(columns={'uf': 'uf', 'município': 'município', 'qtd_2009': 'qtd', 'vlr_2009': 'valor'}).assign(ano=2009)
pbf2010 = pbf[['uf', 'município', 'qtd_2010', 'vlr_2010']].rename(columns={'uf': 'uf', 'município': 'município', 'qtd_2010': 'qtd', 'vlr_2010': 'valor'}).assign(ano=2010)

pbf = pbf2004.append(pbf2005).append(pbf2006).append(pbf2007).append(pbf2008).append(pbf2009).append(pbf2010)
del pbf2004, pbf2005, pbf2006, pbf2007, pbf2008, pbf2009, pbf2010
pbf.head()
```

Out [8]:

	uf	município	qtd	valor	ano
0	GO	ABADIA DE GOIÁS	182.0	84844.0	2004
1	MG	ABADIA DOS DOURADOS	454.0	175967.0	2004
2	GO	ABADIÂNIA	633.0	408591.0	2004
3	MG	ABAETÉ	1260.0	373919.0	2004
4	PA	ABAETETUBA	7503.0	6513626.0	2004

Função para traçar gráficos de análise por município

```
In [9]: def plotData(uf,municipio,p=pbf,i=ipea):
        ax1 = p[(p["uf"] == uf) & (p["municipio"] == municipio)].plot(x="ano",y
        =["valor"],color='gray',title='{ } ( { })'.format(municipio,uf))
        ax1.set_ylabel('Valor (Milhões R$)')
        ax1.tick_params(axis='y')
        ax1.legend(loc='upper center', bbox_to_anchor=(0.5, -0.15),fancybox=True
        e, shadow=True, ncol=1)
        ax1.get_xaxis().set_major_formatter(matplotlib.ticker.FuncFormatter(lambda
        x, p: int(x)))
        ax1.get_yaxis().set_major_formatter(matplotlib.ticker.FuncFormatter(lambda
        x, p: x/1000000.0))
        ax1.grid()

        ax2 = ax1.twinx()
        ax2.set_ylabel('Percentual (%)')
        ax2.tick_params(axis='y')

        ax3 = i[(i["uf"] == uf) & (i["municipio"] == municipio)].plot(ax=ax2,x=
        "ano",y=["perc_crianças_6_14_fora_escola","perc_extremamente_pobres","perc_
        pobres","perc_mulheres_10_14_anos_que_tiveram_filhos","perc_mulheres_15_17_
        anos_que_tiveram_filhos"])
        ax3.legend(loc='upper center', bbox_to_anchor=(0.5, -0.25),fancybox=True
        e, shadow=True, ncol=5)
        ax3.grid()
```

Algumas análises de dados

O PBF inicia os pagamentos em 2004. Os dados sociais do IPEA mapeiam o período entre 1991 e 2010.

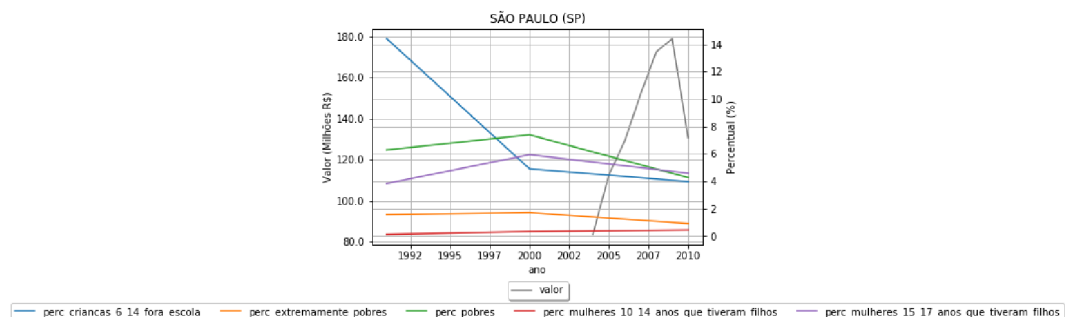
Através dos gráficos, é possível perceber se o PBF tem contribuído desde 2004 para melhorar os índices de renda e escolaridade.

Aqui são demonstradas algumas análises.

Municípios mais ricos e com mais frequência escolar

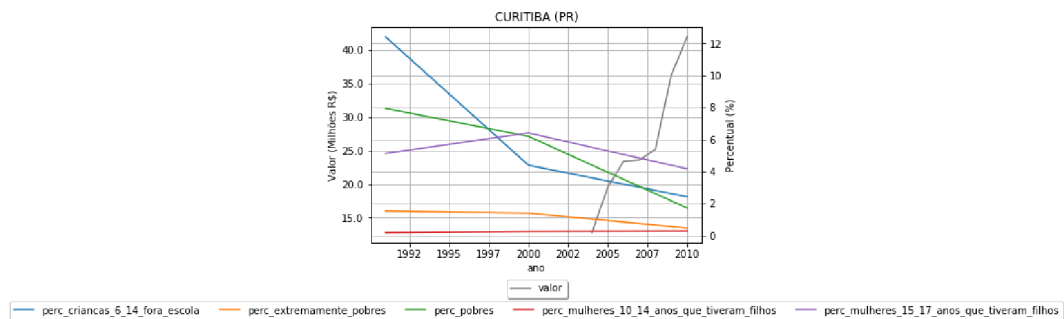
Nesses dois exemplos, os municípios possuem baixo percentual de pessoas pobres e extremamente pobres, e maior frequência escolar.

```
In [10]: plotData('SP','SÃO PAULO')
```



Reflexão: Será que o alto valor investido no PBF para o município de São Paulo tem contribuído de fato para diminuição da pobreza e para o aumento da frequência escolar das crianças?

```
In [11]: plotData('PR', 'CURITIBA')
```

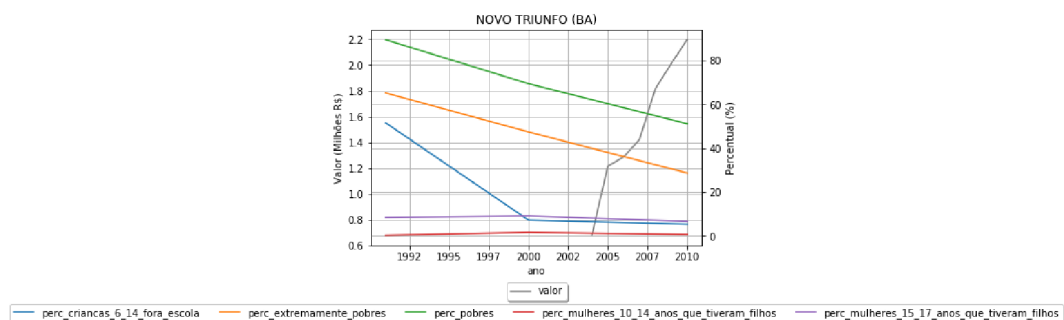


Reflexão: A alta de investimento no PBF em 2010 em Curitiba irá contribuir com a futura redução dos já baixos índices de pobreza do município?

Municípios mais pobres e com menor frequência escolar

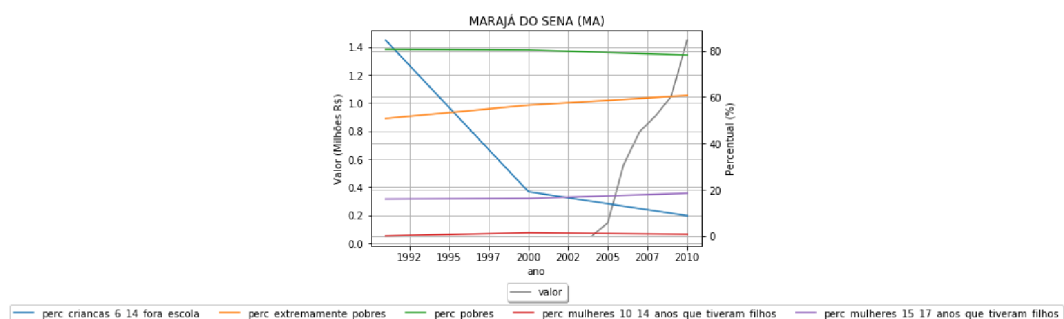
Nesses dois exemplos, os municípios possuem altíssimo percentual de pessoas pobres e extremamente pobres, e baixa frequência escolar.

```
In [12]: plotData('BA', 'NOVO TRIUNFO')
```



Reflexão: Seria necessário maior investimento no PBF para o município de Novo Triunfo, visto que os índices de pobreza são muito elevados?

```
In [13]: plotData('MA', 'MARAJÁ DO SENA')
```



Reflexão: Em Marajá do Sena, que é um dos municípios mais pobres do País, os índices de pobreza / extrema pobreza são absurdamente elevados, e aqui, estão aumentando. Além disso, o índice de meninas em idade escolar que já são mães, é muito alto. Qual a eficácia do PBF nesse município?

Predições

Nesta seção são demonstrados exemplos de predição de índices socio-econômicos através de Regressão Linear.

Classe e métodos para Predição

```
In [14]: class Predictor:
    def __init__(self):
        self.__leUf = preprocessing.LabelEncoder()
        self.__leMunicipio = preprocessing.LabelEncoder()

    def predictIpea(self, ano):
        pred = pd.DataFrame(ipea.groupby(['uf', 'municipio'], as_index=False)
            .mean(), copy=True)
        pred.ano = ano
        pred.perc_crianças_6_14_fora_escola = self.__doPredict
            (ipea, pred, ipea.perc_crianças_6_14_fora_escola)
        pred.perc_mulheres_10_14_anos_que_tiveram_filhos = self.__doPredict
            (ipea, pred, ipea.perc_mulheres_10_14_anos_que_tiveram_filhos)
        pred.perc_mulheres_15_17_anos_que_tiveram_filhos = self.__doPredict
            (ipea, pred, ipea.perc_mulheres_15_17_anos_que_tiveram_filhos)
        pred.perc_pobres = self.__doPredict
            (ipea, pred, ipea.perc_pobres)
        pred.perc_extremamente_pobres = self.__doPredict
            (ipea, pred, ipea.perc_extremamente_pobres)
        return pred

    def predictPbf(self, ano):
        pred = pd.DataFrame(pbf.groupby(['uf', 'municipio'], as_index=False)
            .mean(), copy=True)
        pred.ano = ano
        pred.valor = self.__doPredict(pbf, pred, pbf.valor)
        pred.qtd = self.__doPredict(pbf, pred, pbf.qtd)
        return pred

    def __doPredict(self, training, pred, target):
        X = self.__predictPrepareTraining(training, target)
        y = self.__predictPrepareTarget(target)
        X2 = self.__predictPreparePredict(pred)
        model = LinearRegression(fit_intercept=False)
        model.fit(X, y)
        return model.predict(X2)

    def __predictPrepareTraining(self, data, target):
        X = pd.DataFrame(data[['ano', 'uf', 'municipio']], copy=True)
        self.__leUf.fit(X.uf)
        self.__leMunicipio.fit(X.municipio)
        X.uf = self.__leUf.transform(X.uf)
        X.municipio = self.__leMunicipio.transform(X.municipio)
        return X

    def __predictPreparePredict(self, data):
        X2 = pd.DataFrame(data[['ano', 'uf', 'municipio']], copy=True)
        X2.uf = self.__leUf.transform(X2.uf)
        X2.municipio = self.__leMunicipio.transform(X2.municipio)
        return X2

    def __predictPrepareTarget(self, data):
        y = pd.DataFrame(data, copy=True)
        y.fillna(0, inplace=True)
        return y
```

Exemplo de Predição de índices sócio-econômicos para o ano de 2030

Os índices são estimados para o futuro se baseando em regressão linear, com o modelo treinado com a série temporal da própria variável (target) em questão. Uma melhoria seria a correlação de variáveis no treino (por exemplo: um modelo treinado simultaneamente com as séries de pobreza e valor pago do PBF ao longo do tempo).


```
In [15]: predictor = Predictor()
pbf2030 = predictor.predictPbf(2030)
pbf2030.head()
```

Out[15]:

	uf	municipio	qtd	valor	ano
0	AC	ACRELÂNDIA	3028.569257	2.773801e+06	2030
1	AC	ASSIS BRASIL	3032.217394	2.776348e+06	2030
2	AC	BRASILÉIA	3035.594897	2.778707e+06	2030
3	AC	BUJARI	3036.060386	2.779032e+06	2030
4	AC	CAPIXABA	3038.810020	2.780952e+06	2030

```
In [16]: ipea2030 = predictor.predictIpea(2030)
ipea2030.head()
```

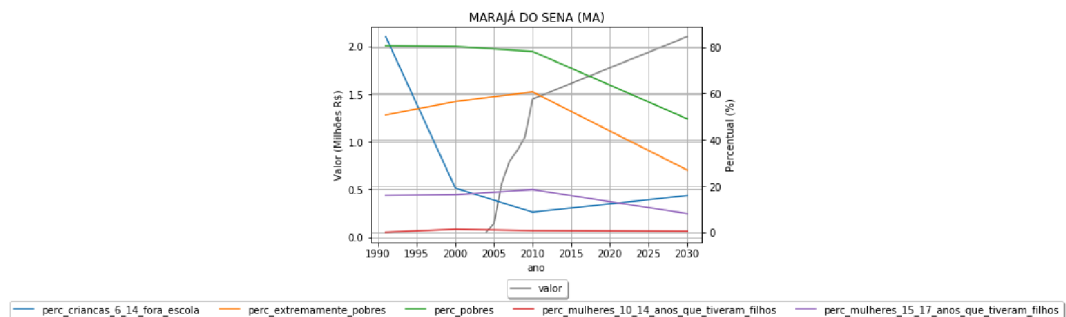
Out[16]:

	uf	municipio	ano	perc_crianças_6_14_fora_escola	perc_extremamente_pobres	perc_po
0	AC	ACRELÂNDIA	2030	19.152734	35.324726	61.3461
1	AC	ASSIS BRASIL	2030	19.141386	35.354846	61.3789
2	AC	BRASILÉIA	2030	19.130914	35.382643	61.4092
3	AC	BUJARI	2030	19.129466	35.386486	61.4133
4	AC	CAPIXABA	2030	19.120913	35.409188	61.4380

Análise gráfica dos dados da predição

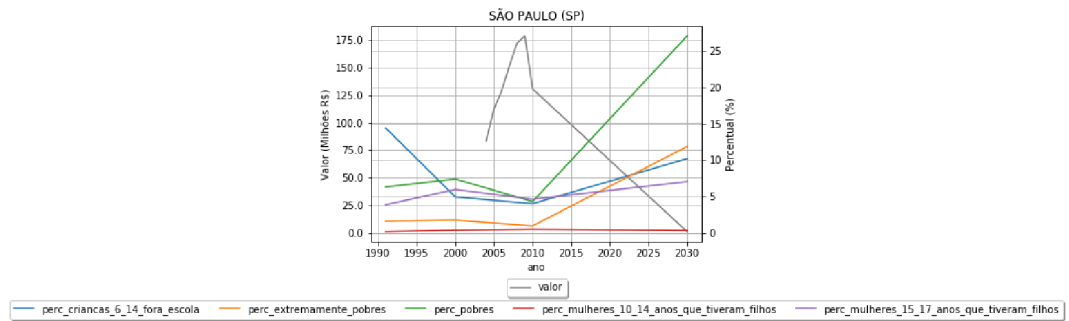
```
In [18]: ipea = ipea.append(ipea2030,sort=False)
pbf = pbf.append(pbf2030,sort=False)
```

```
In [19]: plotData('MA', 'MARAJÁ DO SENA', pbf, ipea)
```



Esse é um exemplo de predição dos índices sócio-econômicos (incluindo o investimento no PBF), no Município de Marajá do SENA, estendendo a análise de dados para o previsto no ano de 2030.

```
In [20]: plotData('SP','SÃO PAULO',pbf,ipea)
```



A mesma análise, para o Município de São Paulo, com o previsto para o ano de 2030.

Pode-se notar aqui que a predição dos índices sócio-econômicos aponta para um aumento na pobreza e redução da frequência escolar, em contrapartida à redução (tendendo a zero) no investimento do PBF até 2030.