

Trabalho 01 - Pablo Correa Fonseca

April 8, 2017

0.1 Mestrado Profissional em Macroeconomia e Finanças - Trabalho Prático 01

Pablo Corrêa Fonseca para execução deste notebook, são necessárias as bibliotecas [pandas](#) e [cufflinks](#)

```
In [1]: import pandas as pd
import cufflinks as cf
cf.go_offline()
pd.set_option('display.max_columns', 7)
from IPython.display import display_pretty, HTML, display_html
```

**** Exercício 1 ****

**** Acesse o site do [IPEADATA](#) e obtenha os dados a seguir para o período de Janeiro de 2003 até Fevereiro de 2017: ****

- Inflação-IPCA
- IPCA-núcleo médias aparadas sem suavização-var.
- IPCA-alimentos e bebidas-var.
- IPCA-artigos de residência-var.
- IPCA-comunicação-var.
- IPCA-despesas pessoais-var.
- IPCA-preços livres-var.
- IPCA-preços monitorados-var. -IPCA-saúde e cuidados pessoais-var.

**** Após obter os dados, apresente tabelas das estatísticas descritivas apresentadas em aula para cada uma das variáveis e a matriz de correlação dos dados. Comente os resultados. ****

Os dados foram baixados do [IPEADATA](#), lidos e ajustados com os comandos a seguir:

```
In [2]: dados = pd.read_excel('IPEA_IPCA_DATA.xls', index_col='Data')
dados.index = pd.to_datetime(dados.index).to_period('M')
short_names = [m[-1].split('_')[-1] for m in dados.columns.str.split('-')]
col_map = dict(zip(short_names,dados.columns))
dados.rename_axis(dict(zip(dados.columns,zip(['IBGE'] * 6 + ['BACEN'] * 3,short_names)))
dados.tail().style.set_caption('Amostra dos dados baixados.')
```

Out[2]: <pandas.formats.style.Styler at 0xaccdac8>

Legenda para os dados :

```
In [3]: for c in dados.columns.get_level_values(1):
        display_html(HTML(u"<b>{}:</b> <br> {}".format(c,col_map.get(c))))
```

definimos as funções de estatística descritiva para análise dos dados a seguir

```
In [4]: def contagem(s):
        return len(s)
        dados.loc['2003-01':'2017-02'].apply(contagem)
```

```
Out[4]: IBGE      IPCAG12      170
        IPCAAB12      170
        IPCAAR12      170
        IPCACD12      170
        IPCACO12      170
        IPCASC12      170
        BACEN IPCA20N12  170
        IPCAPL12      170
        IPCAPM12      170
        dtype: int64
```

```
In [5]: def media(s):
        return sum(s)/contagem(s)
        dados.loc['2003-01':'2017-02'].apply(media)
```

```
Out[5]: IBGE      IPCAG12      0.506353
        IPCAAB12      0.590294
        IPCAAR12      0.233882
        IPCACD12      0.645882
        IPCACO12      0.283353
        IPCASC12      0.555235
        BACEN IPCA20N12  0.439412
        IPCAPL12      0.503118
        IPCAPM12      0.515706
        dtype: float64
```

```
In [6]: def variancia(s):
        m = sum((s - media(s))**2)/contagem(s)
        return m
        dados.loc['2003-01':'2017-02'].apply(variancia)
```

```
Out[6]: IBGE      IPCAG12      0.098211
        IPCAAB12      0.438476
        IPCAAR12      0.192566
        IPCACD12      0.150438
        IPCACO12      0.857352
        IPCASC12      0.126114
        BACEN IPCA20N12  0.038448
        IPCAPL12      0.095812
        IPCAPM12      0.378983
        dtype: float64
```

```
In [7]: def desvio_padrao(s):
        m = pd.np.sqrt(sum((s - media(s))**2)/contagem(s))
        return m
        dados.loc['2003-01':'2017-02'].apply(desvio_padrao)
```

```
Out[7]: IBGE    IPCAG12    0.313387
          IPCAAB12    0.662175
          IPCAAR12    0.438824
          IPCACD12    0.387864
          IPCAC012    0.925933
          IPCASC12    0.355126
BACEN    IPCA20N12    0.196081
          IPCAPL12    0.309535
          IPCAPM12    0.615616
dtype: float64
```

```
In [8]: def mediana(s):
        from math import ceil
        strd = sorted(s)
        if len(strd) % 2 == 0:
            return (strd[len(strd)/2-1] + strd[len(strd)/2])/2
        else:
            return strd[int(ceil(len(strd)/2.0))-1]
        dados.loc['2003-01':'2017-02'].apply(mediana)
```

```
Out[8]: IBGE    IPCAG12    0.470
          IPCAAB12    0.630
          IPCAAR12    0.265
          IPCACD12    0.575
          IPCAC012    0.060
          IPCASC12    0.450
BACEN    IPCA20N12    0.420
          IPCAPL12    0.500
          IPCAPM12    0.375
dtype: float64
```

```
In [9]: def escore_z(s):
        return (s - media(s))/pd.np.std(s)
        dados.loc['2003-01':'2017-02'].apply(escore_z).T
```

```
Out[9]: Data    2003-01    2003-02    2003-03    ...    2016-12    2017-01    \
IBGE IPCAG12    5.563878    3.394037    2.309116    ...    -0.658460    -0.403185
      IPCAAB12    2.355428    0.950966    1.615443    ...    -0.770633    -0.362886
      IPCAAR12    3.796783    3.044772    2.178820    ...    -1.239410    -0.760858
      IPCACD12    3.336526    -0.015166    0.191092    ...     0.938777    -0.505029
      IPCAC012    0.039579    3.193155    1.400368    ...    -0.284419    0.374376
      IPCASC12    2.491412    0.323166    5.166520    ...    -0.183696    -0.014742
BACEN IPCA20N12    6.377906    2.858958    2.705960    ...    -0.302996    -0.200997
      IPCAPL12    3.672869    1.992931    2.542142    ...    -0.333137    -0.817735
```

```

IPCAMP12    5.383702    3.564386    0.916633    ...    -0.853951    0.461804

Data                2017-02
IBGE IPCAG12    -0.562732
      IPCAAB12   -1.571026
      IPCAAR12   -0.122788
      IPCACD12   -0.865980
      IPCACO12    0.406776
      IPCASC12    0.266848
BACEN IPCA20N12 -0.812988
      IPCAPL12   -0.817735
      IPCAMP12    0.104439

```

[9 rows x 170 columns]

com isso é possível descrever os dados, obter a correlação e um boxplot:

```

In [24]: print('Sumário dos dados:')
         dados.loc['2003-01':'2017-02'].describe().append(pd.Series(dados.apply(mediana),name=

```

Sumário dos dados:

```

Out[24]: <pandas.formats.style.Styler at 0x49f7ba8>

```

```

In [11]: %matplotlib inline
         dados.loc['2003-01':'2017-02'].T.reset_index(level=0,drop=True,).T.iplot(kind='box', t

```

```

In [12]: print('Correlação entre as séries de Índices de inflação')
         dados.loc['2003-01':'2017-02'].corr().style

```

Correlação entre as séries de Índices de inflação

```

Out[12]: <pandas.formats.style.Styler at 0xa954e10>

```

Os dados são compostos por índices de inflação gerados por duas instituições diferentes, o IBGE e o BACEN. Podemos notar uma correlação positiva de 0.89 entre os dois índices gerais de cada agência (IPCAG12 e IPCA20N12)

Dentre os índices com maior variação, destaca-se o IPCACO12 que tem média para o período de 0.28 e máxima de 8.33 em julho de 2003

**** Exercício 2. ****

No dia 14 de abril de 2016, a ex-Presidente Dilma Rousseff disse, em um discurso em Belo Horizonte, que *a inflação é a grande conquista dos 10 anos do PT no governo*. Nos laboratórios seguintes poderemos verificar esta afirmação testando se a diferença entre as médias é estatisticamente significativa ou não. Por hora, vamos comparar o período FHC de 1995 até 2002, com o período Lula/Dilma. Primeiro entre no site do IPEADATA e baixe a série mensal da taxa de inflação medida pelo IPCA de janeiro de 1995 até o afastamento da Presidente Dilma. Após isso, responda os itens a seguir:

```
In [13]: IPCA_95_2016 = pd.Series(dados.ix['1995-01':'2016-03'], ('IBGE', 'IPCAG12')), name='INFLACAO
IPCA_FHC = pd.Series(dados.ix['1995-01':'2003-01'], ('IBGE', 'IPCAG12')), name='INFLACAO
IPCA_PT = pd.Series(dados.ix['2003-01':'2016-03'], ('IBGE', 'IPCAG12')), name='INFLACAO
pd.concat([IPCA_95_2016, IPCA_FHC, IPCA_PT], axis=1).describe().style.set_caption(u'Dados de inflação')
```

```
Out[13]: <pandas.formats.style.Styler at 0xbb28f28>
```

- **Faça um gráfico de linha para todo o período. Analise o comportamento dos dados ao longo do tempo. Faça o histograma dos dados e apresente as estatísticas descritivas dos dados analisando os resultados.**

```
In [14]: pd.concat([IPCA_FHC, IPCA_PT], axis=0).iplot(title='IPCA durante os governos FHC e Dilma')
```

```
In [15]: pd.concat([IPCA_FHC, IPCA_PT], axis=0).iplot(title='Distribuição do IPCA durante os governos FHC e Dilma')
```

Os dados do período de janeiro de 1995 à março de 2016 demonstram uma inflação média de 0.596% ao mês, com um pico de 3.02 em novembro de 2002, época da campanha presidencial que marcava o fim da era FHC. Nota-se que o período onde houve a redução da inflação para níveis próximos à média se dá logo após o início do plano real em 94 e se estende até meados de 1998.

- **Refaça o item anterior dividindo a amostra em sub amostras determinadas pelos diferentes governos do período. Com as informações que você tem até o momento, você concorda com a afirmação da ex-Presidente Dilma?**

```
In [16]: pd.concat([IPCA_FHC, IPCA_PT], axis=1).iplot(title='IPCA durante os governos FHC e Dilma')
```

```
In [17]: pd.concat([IPCA_FHC, IPCA_PT], axis=1).iplot(title='Distribuição do IPCA durante os governos FHC e Dilma')
```

Embora a média do período FHC seja superior à média do período do PT à frente do governo, nota-se que a grande redução da inflação ocorreu após a implantação do plano Real, com um pico durante as eleições de 2002, retornando ao nível médio durante o governo Lula e dando sinais de aceleração durante o segundo governo Dilma. Com base nos dados, eu discordo da afirmação da ex-presidente Dilma.

Exercício 3.

Dados do censo americano mostram que nos Estados Unidos, a idade mediana dos americanos que se identificam como brancos é de 36,3 anos, mas a idade mediana dos americanos negros é de 30,1 anos.

(a) Baseando-se nesta informação, dê duas razões que poderiam explicar porque a mediana de idade é menor para os negros.

(b) A mediana mais baixa de idade para os americanos negros significa necessariamente que nos Estados Unidos os negros vivem menos tempo do que os brancos?

(c) Como seria possível negros e brancos morrerem aproximadamente (em média) na mesma idade, mas com uma diferença na mediana de idade?

(a) A diferença da mediana da idade pode ser explicada por:

- Uma população maior de jovens negros, devido a um número maior de filhos negros por mulher.
- Uma população maior de brancos mais velhos, devido a uma maior expectativa de vida.

(b) Não, pode significar que nascem mais bebês negros que brancos.

(c) Essa condição pode ser verificada pela assimetria das distribuições de idades por etnia, com a distribuição dos negros sendo mais assimétrica à direita.

Exercício 4.

Este primeiro exercício visa ilustrar a Lei dos Grandes Números e o Teorema Central do Limite via simulação de Monte Carlo. Para tal, vamos construir uma simulação da seguinte forma:

1. Gere uma amostra de 10 observações a partir de uma distribuição exponencial com parâmetro 0.5. Calcule a média da distribuição. Armazene o resultado.
2. Repita o procedimento acima 10000 vezes. Ao final do processo você terá 10000 estimativas da média da distribuição. Desenhe o histograma e calcule a média e a variância dos estimadores. Guarde os resultados.
3. Repita os passos 1 e 2 acima para os seguintes tamanhos de amostra: 50, 100, 500, 1000, 5000 e 10000 observações. Desenhe um gráfico da média e da variância dos estimadores em função do tamanho da amostra. O que você pode concluir.
4. Repita os passos anteriores para pelo menos mais duas distribuições. Comente os resultados.

```
In [18]: import numpy as np
def gera_medias(repeticoes, observacoes, distribuicao):
    medias = []
    for i in range(repeticoes):
        media = []
        for j in range(observacoes):
            media.append(distribuicao())
        medias.append(np.mean(media))
    return medias

In [19]: experimento_1 = gera_medias(1, 10, lambda: np.random.exponential(scale=0.5))
experimento_2 = gera_medias(10000, 10, lambda: np.random.exponential(scale=0.5))
print('Média 1x10 itens: {}'.format(experimento_1))
print('Média 10000x10 itens: {}'.format(np.mean(experimento_2)))
print('Variância 10000x10 itens: {}'.format(np.var(experimento_2)))
```

```
Média 1x10 itens: [0.39346201257470054]
Média 10000x10 itens: 0.502034526373
Variância 10000x10 itens: 0.0246713150209
```

```
In [20]: pd.Series(experimento_2).plot(kind='histogram')
```

```
In [ ]:
```

```
In [21]: experimento_3 = {'n'+str(o):gera_medias(10000,o, lambda: np.random.exponential(scale=0.5)) for o in range(1,11)}
medias = pd.DataFrame(experimento_3).mean()
var = pd.DataFrame(experimento_3).var()
pd.concat([medias,var], axis=1, keys=['Media','Variancia']).iloc[[2,0,3,1,4]].plot(kind='line')
```