

Construindo um DashBoard Interativo usando Bokeh e Pandas

A Importância da Visualização de Dados

Uma grande quantidade de dados está sendo gerada a cada momento devido as atividades de negócios na globalização. As empresas estão extraíndo informações úteis de tais dados gerados para tomarem importantes decisões de negócios. A análise Exploratória de Dados pode ajudá-los a visualizarem a situação atual do mercado e projetarem as tendências futuras, entenderem os que seus clientes dizem e esperam do seu produto, melhorias no produto tomando medidas adequadas, e muito mais.

Para conseguir isso, a visualização de dados (Data visualization) é a solução, i.é., para criar uma representação visualmente apelativa dos dados que diga uma história interessante rapidamente ainda é simples para todos leitores entenderem.

Pode-se usar o Pandas para a análise de dados dita acima no Python através de suas funções de gráficos embutidas. Mas não gostaríamos de ser grandes se você pode interagir com o gráfico através de funções como zoom ou hover to dig a little deeper nos dados?

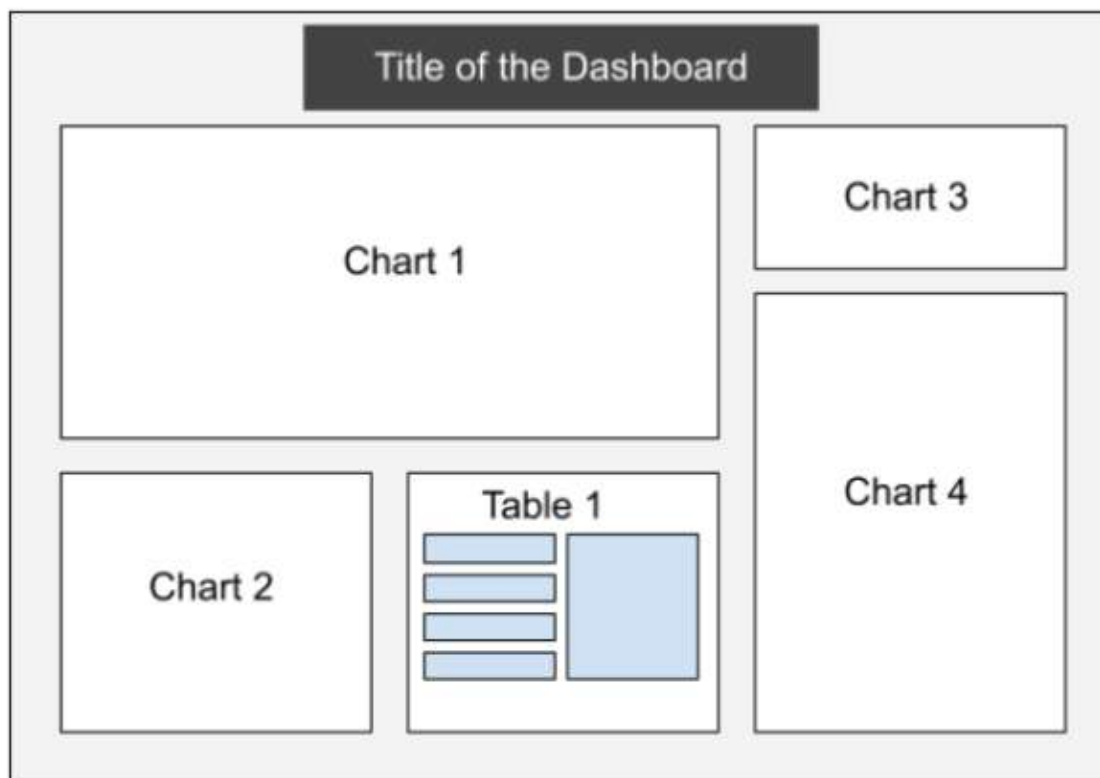
Benefícios dos Plots e Dashboards Interativos usando o Boken

A visualização interativa de dados permite o usuário instantaneamente modificar os elementos numa plot gráfico em vez de mudar o código no background. Imagine você interagindo com um plot que mostra um preço de produto por uma década. Agora, se existia um slider ou um menu drop-down para selecionar os preços para um ano ou mês particular, então você como um leitor terá insights muito mais rápidos do gráfico e isso aumentará muito rapidamente sem edição de códigos. Isto é exatamente o que plots interativos oferecem.

Com plots interativos, podemos entender melhor a história por detrás dos dados. Estes plots permite-nos ampliar ou reduzir numa variação interessante as tendências spot e variações como também encontrar correlações e conexões entre as variáveis. Tudo isso torna o processo de exploração de dados mais significativo

Introdução aos Dashboards

Dashboards são ferramentas visuais que diz a história contida no dataset e permite aos usuários entenderem rapidamente a figura maior. Esses são coleções de diferentes plots ligadas juntas num layout de estilo de grade como mostrado abaixo e são uma parte da história. Assim, podemos dizer que os dashboards são uma forma comum para apresentar insights valiosos num único lugar.



Bibliotecas para Plots Interativos

Os gráficos criados usando o Matplotlib lib e o Seaborn são gráficos estáticos, i.é., um usuário não pode atualizá-lo sem a atualização dos códigos e rodá-lo novamente. Assim, as bibliotecas de plots interativos – D3 e chart.js deveriam ser usadas, mas elas esperam que o usuário tenha algum conhecimento prévio de JavaScript.

Atualmente, existem duas bibliotecas populares Open-Source para construção de visualizações interativas – Bokeh e Plotty. Neste artigo, apresentaremos um simples tutorial usando Bokeh.

A Bokeh é uma biblioteca Open_Source para visualizações interativas que renderizam os gráficos usando HTML e JavaScript. Ela é uma poderosa ferramenta EDA que pode ser usada para construir dashboards e aplicações baseadas na web.

Bokeh suporta gráficos de linhas, pie charts, gráficos de Barras e Stacked Bar charts, histogramas e plots scatter. A fonte de dados é convertida a uma arquivo JSON que torna-se uma entrada ao BokehJS (biblioteca JavaScript) e isto torna possível renderizar plots e visualizações interativas suportadas pelo browse.

A biblioteca Bokeh exige um entendimento básico do código JavaScript para escrever funções personalizadas para atualizar os plots dependendo das entradas do usuário. Para esse artigo para iniciantes, eu usei uma biblioteca chamada Pandas-Bokeh que é mais fácil de usar para os novatos e permite renderizar da mesma plots Bokeh através do seu suporte Back-end para Pandas. Para visualizações avançadas, pode-se sempre usar a biblioteca Bokeh para definir visualizações personalizadas.

Uma simples linha de código é exigida para cada plot interativo. Demonstrarei a funcionalidade da biblioteca Pandas-Bokeh e como podemos usá-la para construir um dashboard simples do dataset.

CONSTRUINDO UM DASHBOARD INTERATIVO USANDO BOKEH

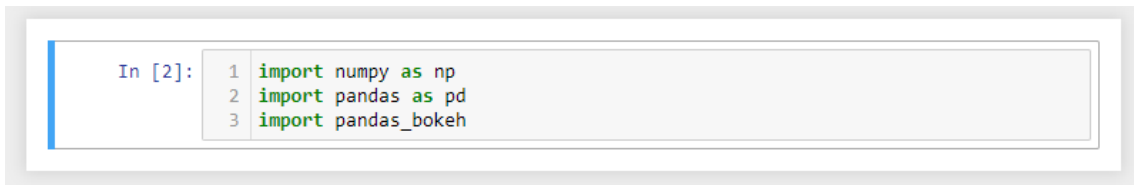
Vamos começar instalando primeiro a biblioteca usando pip do PyPI.

#



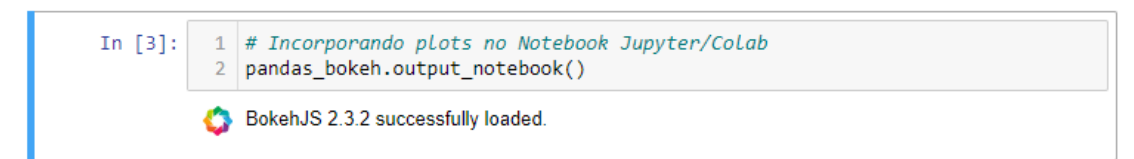
```
Anaconda Prompt (anacondaNavigator)
(base) C:\Users\Bertolo>pip install pandas_bokeh
```

A seguir, importamos as bibliotecas pandas e numpy. Lembre-se de importar estes antes da biblioteca pandas_bokeh.



```
In [2]: 1 import numpy as np
        2 import pandas as pd
        3 import pandas_bokeh
```

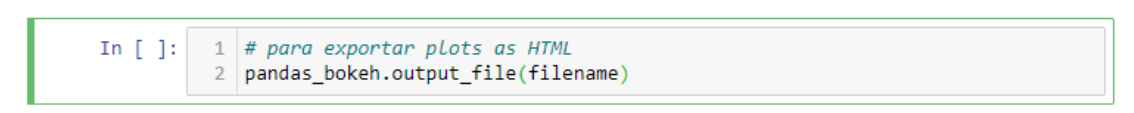
Precisamos, também, o comando seguinte para mostrar as saídas dos gráficos no notebook:



```
In [3]: 1 # Incorporando plots no Notebook Jupyter/Colab
        2 pandas_bokeh.output_notebook()

BokehJS 2.3.2 successfully loaded.
```

Para exibir os gráficos numa HTML separada, usamos este comando:



```
In [ ]: 1 # para exportar plots as HTML
        2 pandas_bokeh.output_file(filename)
```

Para este tutorial para iniciantes, estamos gerando um dataset randômico usando a biblioteca NumPy e usando-a para construir o dashboard.

Vamos assumir que o dataset contenha amostra de valores medidos de 4 sensores durante um período de 12 meses e cada valor tenha um número e uma categoria de identificação únicos associado a ele. Isto significa que existem um total de 6 características, i.é., “id”, “mês”, “sensor_1”, “sensor_2”, “sensor_3”, e “categoria”. Por simplicidade, estamos considerando somente 15 amostras de linhas de dados.

Para gerar este dataset, usamos a função np.random da biblioteca NumPy como segue:

(Link da documentação oficial do NumPY:

<https://numpy.org/doc/stable/reference/random/generator.html>

```

In [5]: 1 # define as variáveis categóricas
        2 categoria = ['A', 'B', 'C']
        3 # define a semente aleatória para tornar o dataset reproduzível
        4 np.random.seed(42)
        5 # cria um dataset
        6 df_random = pd.DataFrame({
        7     'id': np.arange(0,15),
        8     'mes': np.random.randint(1,12,15),
        9     'sensor_1': np.random.uniform(0,1,15),
       10     'sensor_2': np.random.uniform(10,15,15),
       11     'sensor_3': np.random.randint(0,20,15),
       12     'categoria': np.random.choice(categoria, 15, p=[0.2, 0.4, 0.4])
       13 })

```

```

In [6]: 1 # define index para coluna id
        2 df_random = df_random.set_index('id')

```

Verificando a forma (shape) do dataframe:

```

In [7]: 1 df_random.shape
Out[7]: (15, 5)

```

Geramos com sucesso um dataset com 15 amostras 6 características.

Vamos olhar para alguns valores amostras dos dataset.

```

In [12]: 1 df_random.head()
Out[12]:

```

	mes	sensor_1	sensor_2	sensor_3	categoria
id					
0	7	0.708073	12.280350	14	B
1	4	0.020584	13.925880	6	A
2	11	0.969910	10.998369	11	C
3	8	0.832443	12.571172	7	B
4	5	0.212339	12.962073	14	B

Agora podemos plotar o gráfico num dashboard. Para propósitos de demonstração, vamos plotar os seguintes gráficos usando a biblioteca pandas_bokeh:

1. Line plot
2. Bar Chart
3. Stacked Bar chart
4. Scatter plot
5. Pie chart
6. Histogram

```
# Plot1 - Line plot
```

```

p_line =
df_random.groupby(['mes']).mean().plot_bokeh(kind="line",
y="sensor_2", color='#d01c8b',plot_data_points=True,
show_figure=False)

```

```

# Plot2- Barplot

p_bar =
df_random.groupby(['mes']).mean().plot_bokeh(kind="bar", colormap
=colors, show_figure=False)

# Plot3- stacked bar chart

df_sensor=df_random.drop(['month'],axis=1)

p_stack=df_sensor.groupby(['category']).mean().plot_bokeh(kind='
barh', stacked=True,colormap=colors, show_figure=False)

#Plot4- Scatterplot

p_scatter = df_random.plot_bokeh(kind="scatter", x="month",
y="sensor_2",category="category", colormap=colors, show_figure=False)

#Plot5- Pie chart

p_pie=
df_random.groupby(['category']).mean().plot_bokeh.pie(y='sensor_
1', colormap=colors, show_figure=False)

#Plot6- Histogram

p_hist=df_sensor.plot_bokeh(kind='hist',
histogram_type="stacked",bins=6, colormap=colors,
show_figure=False)

```

Executando estes comandos geraremos os plots mas eles não serão exibidos pois temos definido “show_figure=False”. Como queremos estes que esses gráficos apareçam no dashboard, usamos essa opção.

A seguir, definimos o layout da grade para o dashboard usando o comando “pandas_bokeh.plot_grid”. Plotamos os três primeiros plots na primeira linha e os três restantes na segunda linha.

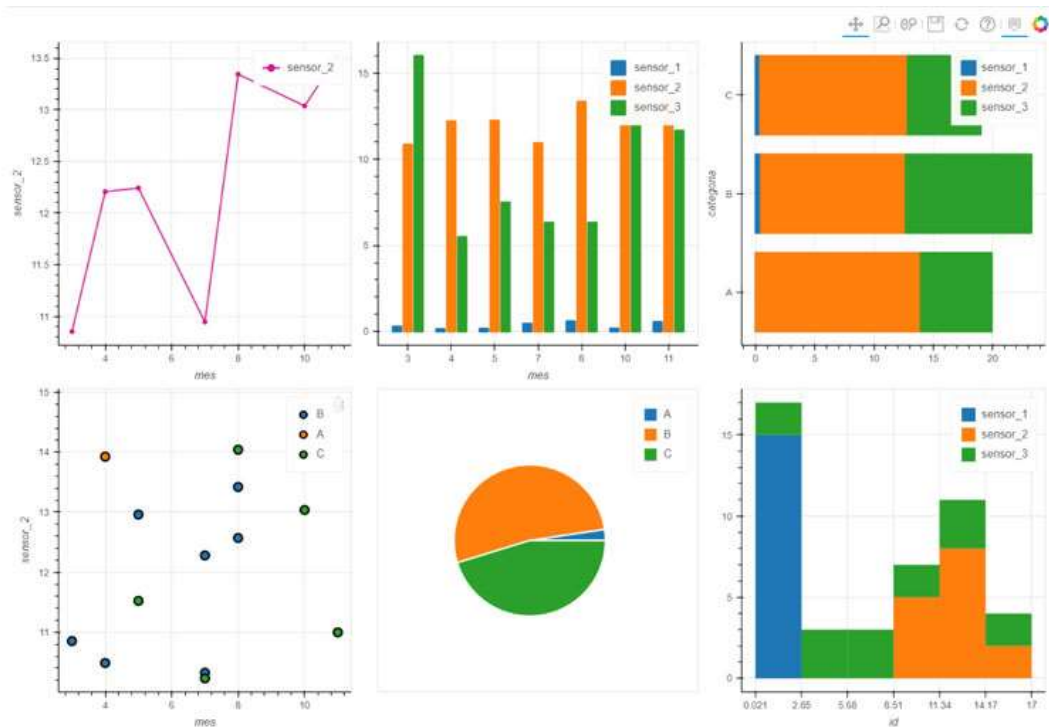
#Fazer o Dashboard com Layout Grid:

```

pandas_bokeh.plot_grid([[p_line, p_bar,p_stack],[p_scatter, p_pie,p_hist]],
plot_width=400)

```

O dashboard ficará assim:



Todos esses plots são interativos e permitem você usar as funções hover e zoom como também filtrar as categorias.

Conclusão

Através deste artigo, mostramos como gerar diretamente plots interativos Bokeh dentro do Pandas e montar um dashboard simples usando a biblioteca Pandas-Bokeh. A biblioteca Pandas-Bokeh é extremamente fácil de usar pelos iniciantes com entendimento básico da sintaxe de plotagem pandas. Esta biblioteca pode ajudar definitivamente a tornar as visualizações mais apresentáveis sem a necessidade de aprender códigos JavaScripts adicionais para geração de plots interativos. Espero ter satisfeito você explorando essa biblioteca tanto quanto eu fiz!

