



Felipe Galvão

Web Development / Desenvolvimento Web

Mais Visualização de Dados com Python (agora com Bokeh)

Felipe Galvão – [Python \(Português\)](#) – março 15, 2016

[English version of this post](#) / [Versão em inglês deste post](#)

Introdução

Depois de falarmos da porta de entrada para visualização de dados em Python que é o matplotlib ([neste post](#)), vamos falar agora sobre o Bokeh.

Bokeh ([página oficial](#)) é uma biblioteca do Python para visualização interativa, que busca seguir o estilo do D3.js. Seu objetivo é permitir a criação de gráficos interativos, painéis / dashboards, aplicações que mexam com dados.

Instalação

O Bokeh não vem instalado junto com o Anaconda, mas para instala-lo é bem simples. Se você estiver usando Anaconda, a instalação se resume ao comando:

```
conda install bokeh
```

Se você tiver todas as dependência instaladas (NumPy, Pandas, Redis), também pode instalar o Bokeh pelo pip.

Se quiser informações mais completas sobre instalação do Bokeh, [clique aqui](#).

Começando a usar

Bem, vamos começar a usar o Bokeh. Começando pelos exemplos bem simples, como sempre 🤔

Primeiro, vamos ao nosso gráfico de linha, mais básico possível. Vamos preparar os dados para o gráfico, definir o arquivo para o qual vai o output com a função “output_file” e criar uma figure para a plotagem dos gráficos com a função “fig”, junto com seu título e os títulos dos eixos. Então iremos traçar a linha passando para a função “line” da figure os dados do gráfico, e então executar o comando “show” para mostrar a figure:

```
import pandas as pd
import numpy as np
from bokeh.plotting import figure, output_file, show

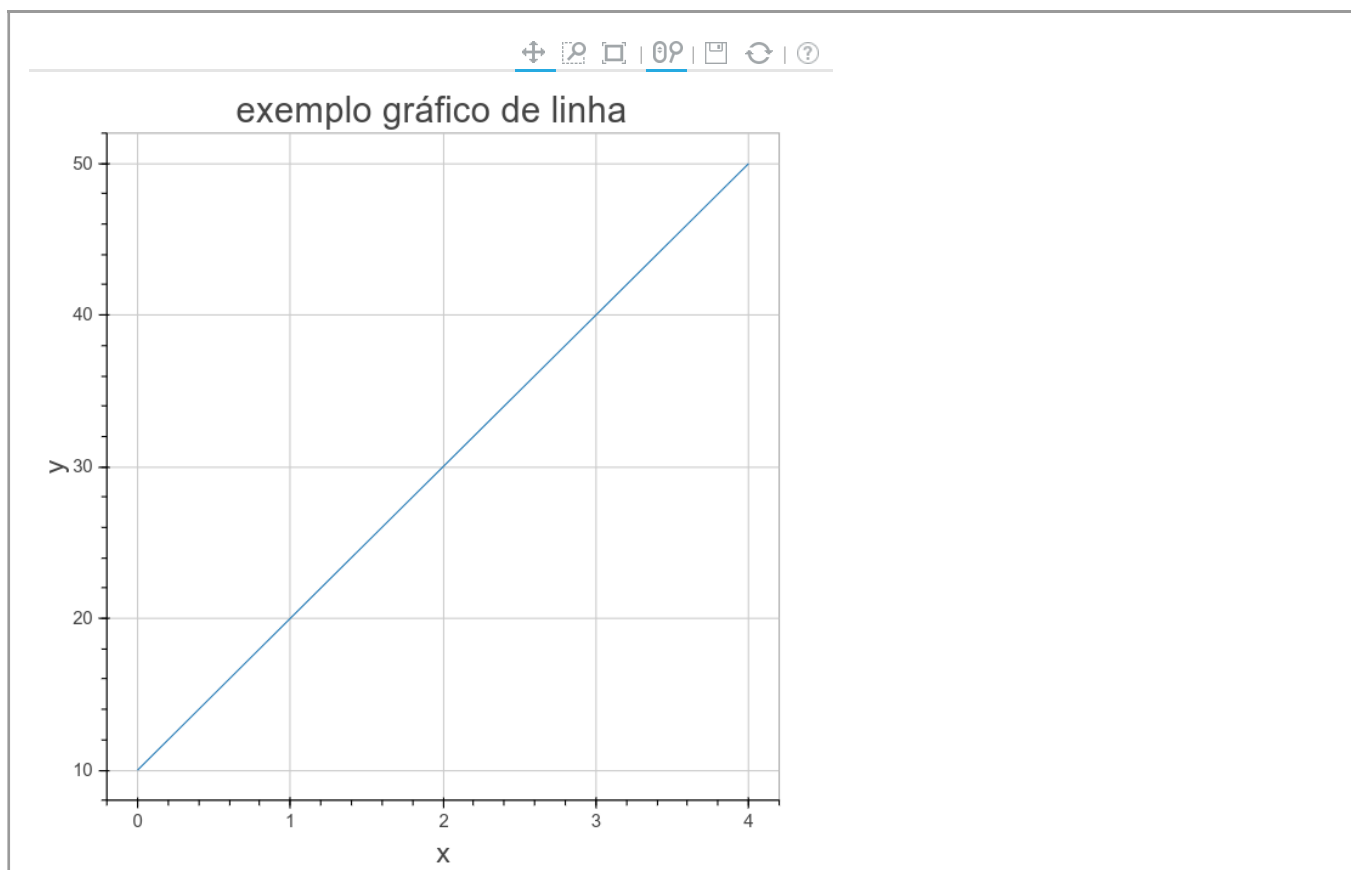
# preparação dos dados
y = [10, 20, 30, 40, 50]
x = range(len(y))

# Configurar o destino do gráfico
output_file("bokeh_exemplo_1.html", title="Exemplo gráfico de linha")

# Cria a figure e define algumas propriedades
fig = figure(title="exemplo gráfico de linha", x_axis_label='x', y_axis_label='y')

# Adiciona a linha
fig.line(x, y)
```

```
# Mostra os resultados, semelhante ao matplotlib  
show(fig)
```

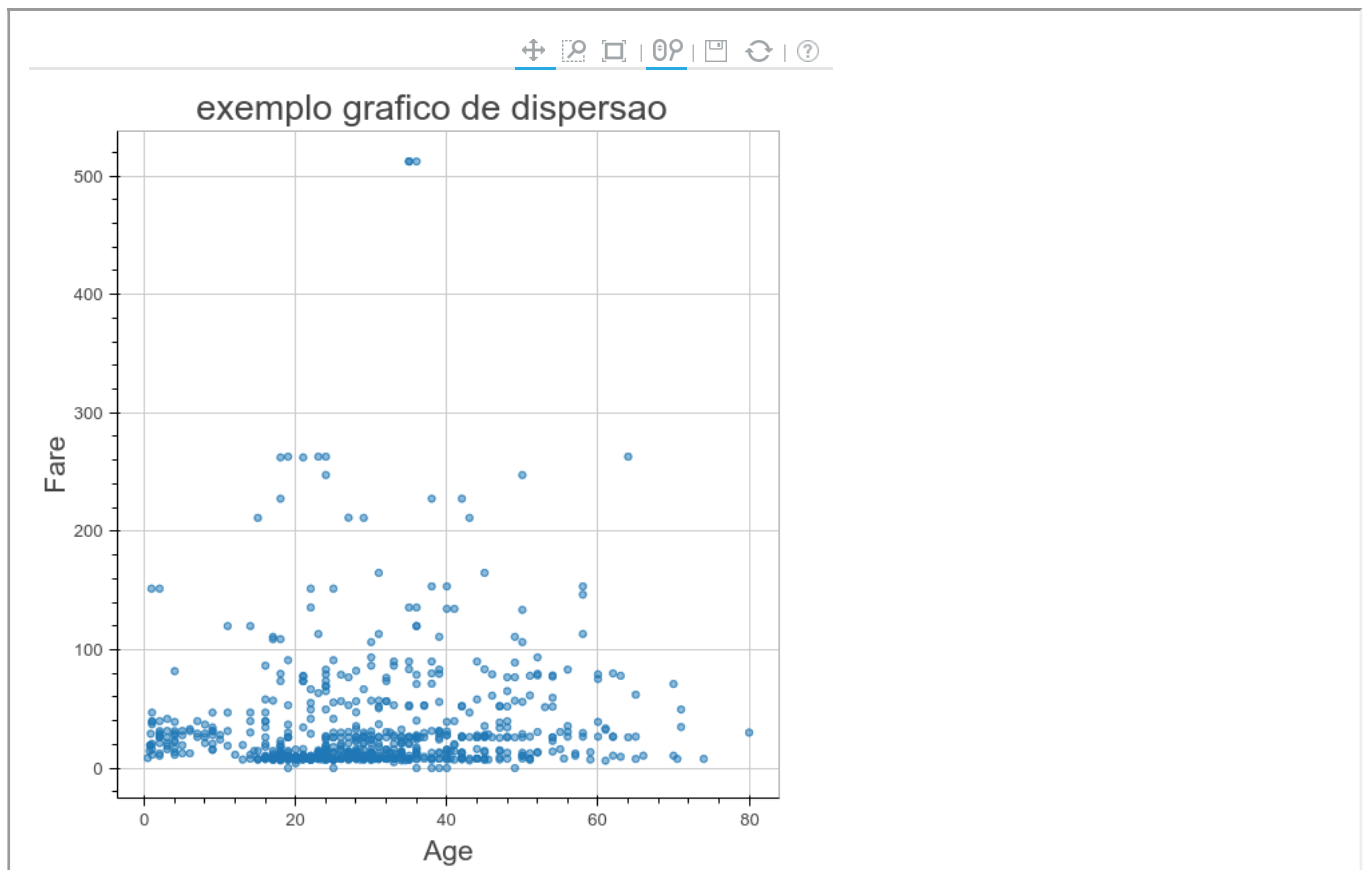


Repare como você pode mover o gráfico, salvar, dar zoom com o scroll do mouse. Essa interatividade é bem legal quando você quer criar alguma aplicação web que envolva gráficos.

Gráfico de Dispersão (Scatter Plot)

Vamos ver agora como fica o gráfico de dispersão que criamos no primeiro post de Visualização de Dados, quando feito com o Bokeh. Da mesma forma que no primeiro exemplo, configuramos os dados que criarão o gráfico, extraindo-os da coluna do Dataset do Titanic. Depois, configuramos o arquivo de destino e a figure, mas agora usaremos a função `circle` da figure para criar os pontos do gráfico. Vamos configurar um "alpha", para transparência dos pontos, e o tamanho dos mesmos:

```
train_df = pd.read_csv('train.csv')  
  
ages = train_df['Age']  
fare = train_df['Fare']  
  
output_file("bokeh_exemplo_dispersao.html", title="Exemplo Gráfico de Dispersão")  
  
fig2 = figure(title="exemplo grafico de dispersao", x_axis_label='Age',  
              y_axis_label='Fare')  
  
fig2.circle(ages, fare, size=5, alpha=0.5)  
  
show(fig2)
```



Bonito, não é? Vamos agora mostrar um gráfico de barras. O gráfico de barras do Bokeh funciona de forma um pouco diferente.

Gráficos de Barras

Os dados para o gráfico de barras no Bokeh são organizados em um Dictionary do Python, composto por Listas com os valores que usaremos no gráfico. Vamos ver os exemplos de sobrevivência por sexo com gráficos de barras no Bokeh. Adicionalmente, iremos fazer gráficos múltiplos e então podemos ver como fazer também gráficos de barras empilhadas.

Primeiro, vamos definir os valores que precisamos. Vamos precisar da quantidade de sobreviventes e não sobreviventes para cada sexo. Vamos usar o `pivot_table` do Pandas para calcular esta informação. Precisamos então transformar os valores em uma lista. A lista terá primeiro a contagem para “female”, “Não Sobreviveu” e “Sobreviveu”, e depois “male”, na mesma ordem. As listas para “sexo” e “sobrevivencia” devem indicar a quais destas categorias os valores pertencem. Se na lista de valores o primeiro item se refere a não sobreviventes do sexo feminino, o primeiro item da lista “sexo” deve ser “female”, o primeiro da lista de “sobrevivencia” deve ser “Não Sobreviveu”, e assim por diante para os outros valores das listas.

Depois iremos usar a função `Bar` que importamos para criar os dois gráficos (um empilhado e o outro não empilhado). Para a função vamos passar o Dict (que chamamos de “dados”), vamos indicar quais os valores que devem ser agregados, quem vai ser o label das colunas e o título. Para o gráfico não empilhado, definiremos dois labels, e aí o Bokeh utilizará esta informação para formar quatro barras, com todas as combinações possíveis entre estas listas. Para os gráficos empilhados, vamos indicar através de qual variável o Bokeh deve empilhar, no caso, por “sobrevivencia”. Passaremos esta key do Dict para o parâmetro “stack”, e a outra key, “sexo”, passaremos para a parâmetro “label”. Para exibição, vamos usar a função “hplot”, que cria gráficos múltiplos na horizontal. O resultado fica assim:

```
from bokeh.charts import Bar, output_file, show, hplot

tabela = pd.pivot_table(data=train_df, values='PassengerId', index='Sex',
                        columns='Survived', aggfunc='count')

valores_grafico = list(tabela.ix['female'].values)
for item in (list(tabela.ix['male'].values)):
    valores_grafico.append(item)
```

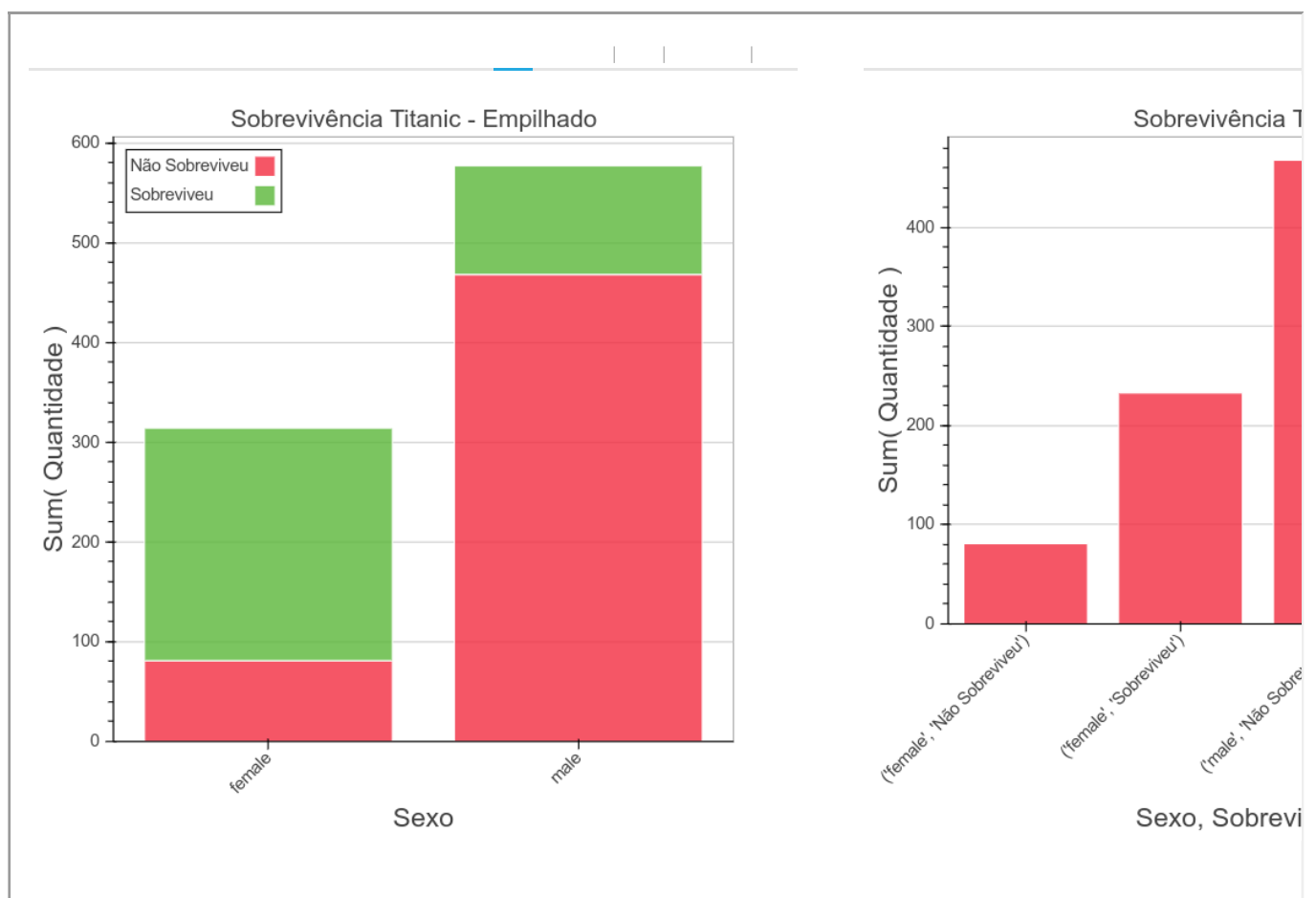
```
output_file("bokeh_exemplos_barra.html", title="Exemplos de Gráficos de Barra")
```

```
dados = {
    'sobrevivencia': ['Not Survived', 'Survived', 'Not Survived', 'Survived'],
    'sexo': ['female', 'female', 'male', 'male'],
    'quantidade': valores_grafico
}
```

```
bar = Bar(dados, values='quantidade', label='sexo', stack='sobrevivencia',
          title="Sobrevivência Titanic - Empilhado", legend='top_left')
```

```
bar2 = Bar(dados, values='quantidade', label=['sexo', 'sobrevivencia'],
           title="Sobrevivência Titanic")
```

```
show(hplot(bar, bar2))
```



Histogramas

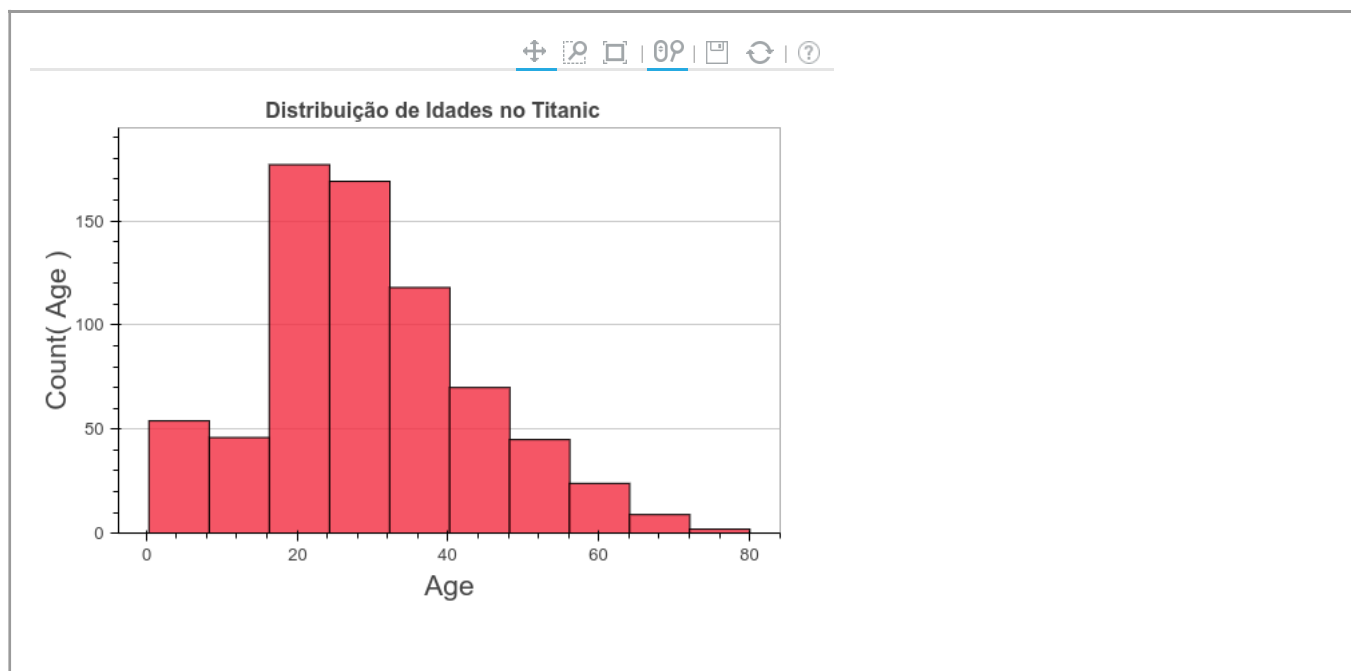
Histogramas no Bokeh são bem simples. Importamos a função `Histogram` e podemos passar o próprio Dataframe para ela. Basta definir então a coluna que contém os dados que serão utilizados no Histograma. Adicionalmente, podemos definir o número de divisões através do parâmetro "bins". Vamos usar o Dataset do Titanic e fazer um histograma das idades, com 10 barras (bins).

```
from bokeh.charts import Histogram
```

```
hist = Histogram(train_df, values="Age",
                  title="Distribuição de Idades no Titanic", bins=10)
```

```
output_file("bokeh_exemplo_histograma.html", title="Exemplo de Histograma")
```

```
show(hist)
```



Diagramas de Caixa (Boxplots)

Diagramas de Caixa, ou Boxplots, são interessantes quando você quer visualizar variações de valores dentro de categorias e possíveis pontos fora da curva, muito acima ou abaixo dos valores médios. Vamos criar um para ver como variam os valores do tíquete (coluna "Fare") de acordo com as classes em que viajam os passageiros (coluna "Pclass").

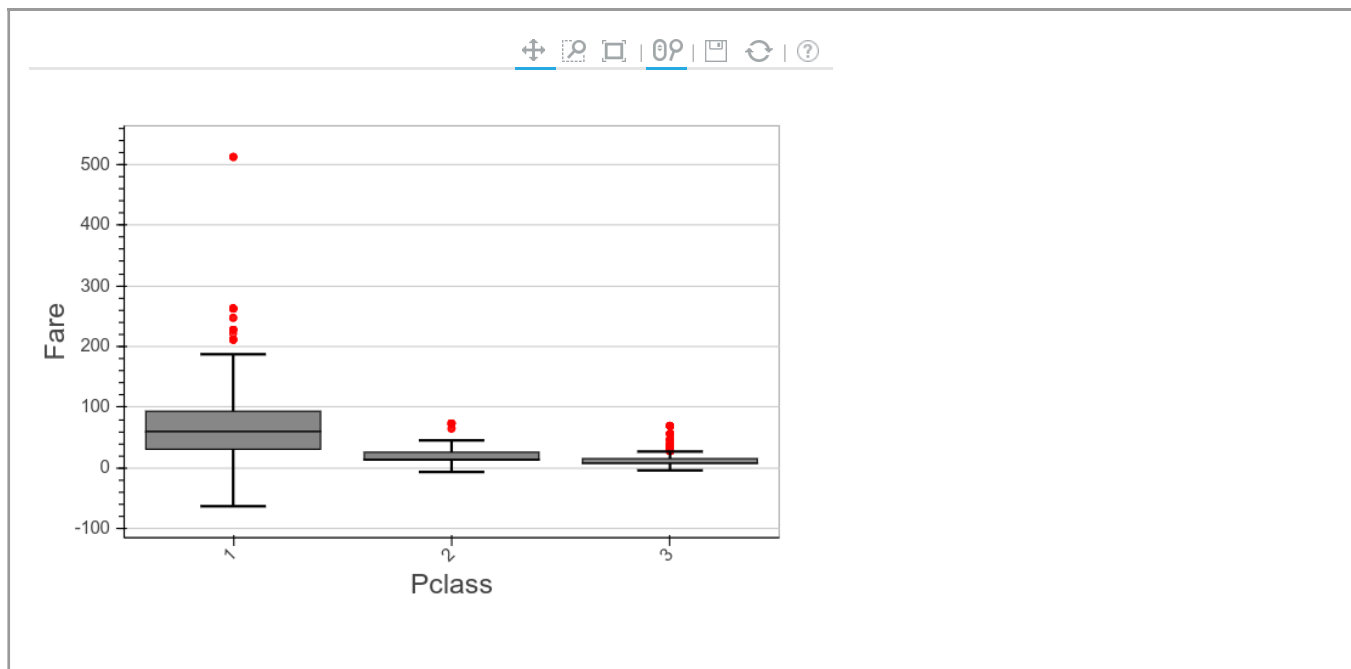
Vamos importar a função BoxPlot, passar o Dataframe para ela e definir em "values" a variável a ser agregada e em "label" a variável que definirá a divisão a ser feita no gráfico, ou seja, cada classe será uma caixa:

```
from bokeh.charts import BoxPlot
```

```
box = BoxPlot(train_df, label="Pclass", values="Fare")
```

```
output_file("bokeh_boxplot.html")
```

```
show(box)
```



Estamos chegando ao final e você pode estar se perguntando onde está o gráfico de pizza. Até onde sei, este não tem um suporte adequado no Bokeh, não sendo nem mencionado na documentação oficial. Talvez eles adicionem esta feature em versões futuras, mas por enquanto temos que viver sem ele no Bokeh.

No próximo post, Seaborn para complementar o Matplotlib. Fiquem ligados :)

Abraços!

Tags [ANALISE DE DADOS](#) [APRENDER](#) [CIÊNCIA DE DADOS](#) [DADOS](#) [DATA SCIENCE](#) [DATAFRAMES](#) [PANDAS](#) [PROGRAMACAO](#)
[PYTHON](#) [PYTHON BASICO](#) [VISUALIZAÇÃO DE DADOS](#)

[More Data Visualization with Python \(now with Bokeh\)](#)

[Descriptive Statistics with Python](#)

6 comments on “Mais Visualização de Dados com Python (agora com Bokeh)”

Pingback: [More Data Visualization with Python \(now with Bokeh\) - Felipe Galvão](#)



Silvio disse:

março 15, 2016 às 16:47

Sensacional, simplesmente isso.

[Responder](#)



Felipe Galvão disse:

março 15, 2016 às 18:21

Valeu Silvio, muito obrigado!

[Responder](#)

Pingback: [Interactive Data Visualization In Python With Bokeh | 神刀安全网](#)

**Lucas** disse:

abril 27, 2016 às 10:08

Parabéns pelo blog, Felipe. Achei agora e estou vários posts!! Muito obrigado por compartilhar seus conhecimentos!!

[Responder](#)**Felipe Galvão** disse:

abril 27, 2016 às 11:32

Fala Lucas! Que isso, precisa agradecer não. Escrever já me ajuda a aprender melhor, e se ajudar alguém já é um bônus legal. Abraços!

[Responder](#)

Deixe uma resposta

O seu endereço de e-mail não será publicado. Campos obrigatórios são marcados com *

Comentário

Nome *

E-mail *

Site

[PUBLICAR COMENTÁRIO](#)

ABOUT ME / SOBRE MIM

Hi, i'm Felipe Galvão. Here, I'll talk about web development and everything I learned in my journey as a programmer.

Olá, sou Felipe Galvão. Aqui, vou falar de desenvolvimento web e tudo que aprendi na minha jornada como programador.

SOCIAL MEDIA



PESQUISAR

TÓPICOS RECENTES

- Primeiros Passos com React
- React First Steps
- Understanding webpack and creating a React application with it
- Entendendo webpack e criando uma aplicação em React com ele
- Creating development environments and installing Django with Anaconda

COMENTÁRIOS

- React First Steps - Felipe Galvão em Primeiros Passos com React
- Primeiros Passos com React - Felipe Galvão em React First Steps
- React First Steps - Felipe Galvão em Understanding webpack and creating a React application with it
- Understanding webpack and creating a React application with it - Felipe Galvão em Entendendo webpack e criando uma aplicação em React com ele
- Entendendo webpack e criando uma aplicação em React com ele - Felipe Galvão em Understanding webpack and creating a React application with it


ARQUIVOS

- abril 2017
- março 2017
- janeiro 2017
- junho 2016
- maio 2016
- abril 2016
- março 2016
- fevereiro 2016
- dezembro 2015
- novembro 2015

CATEGORIAS

- Basic R
- Django (English)
- Django (Português)
- Javascript (English)
- Javascript (Português)
- Python (English)
- Python (Português)
- R (English)
- R (Português)
- R Básico
- React (English)
- React (Português)

META

- Cadastre-se
- Fazer login
- Posts 
-  dos comentários
- WordPress.org

Powered by WordPress | Theme: Talon by aThemes.

[Ciência / Análise de Dados com Python – Básico](#) [Contatos / Contact Me](#) [Data Science / Analysis with Python – Basics](#) [Guides](#)