BIG DATA

LABDATA

FUNDAÇÃO INSTITUTO DE ADMINISTRAÇÃO

Introdução ao Big Data

Tema da Aula: **Visualização dos dados com Python**

Prof.: **Dino Magri**

Coordenação:

Prof. Dr. Adolpho Walter Pimazzi Canton

Profa. Dra. Alessandra de Ávila Montini

- Contatos:
  - E-mail: professor.dinomagri@gmail.com
  - Twitter: https://twitter.com/prof_dinomagri
  - LinkedIn:http://www.linkedin.com/in/dinomagri
  - Site: http://www.dinomagri.com

Coordenação:

Prof. Dr. Adolpho Walter Pimazzi Canton

Profa. Dra. Alessandra de Ávila Montini

Currículo

- **(2014-Presente)** – Professor no curso de Extensão, Pós e MBA na Fundação Instituto de Administração (FIA) – www.fia.com.br
- **(2018-Presente)** – Pesquisa e Desenvolvimento de Big Data e Machine Learning na Beholder (http://beholder.tech)
- **(2013-2018)** – Pesquisa e Desenvolvimento no Laboratório de Arquitetura e Redes de Computadores (LARC) na Universidade de São Paulo – www.larc.usp.br
- **(2012)** – Bacharel em Ciência da Computação pela Universidade do Estado de Santa Catarina (UDESC) – www.cct.udesc.br
- **(2009/2010)** – Pesquisador e Desenvolvedor no Centro de Computação Gráfica – Guimarães – Portugal – www.ccg.pt
- **Lattes:** http://lattes.cnpq.br/5673884504184733

Material das aulas

- Caso esteja utilizando seu próprio computador, realize o download de todos os arquivos e salve na **Área de Trabalho** para facilitar o acesso.
  - Lembre-se de instalar os softwares necessários conforme descrito no documento de Instalação (**`InstalaçãoPython3v1.2.pdf`**).

- Nos computadores da FIA os arquivos já estão disponíveis, bem como a instalação dos softwares necessários.

Conteúdo da Aula

- Objetivo
- Visualização dos dados com Python
  - Plotagem no Pandas
  - Matplotlib
  - Seaborn
- Exercícios
- Referências

Conteúdo da Aula

- **Objetivo**
- Visualização dos dados com Python
  - Plotagem no Pandas
  - Matplotlib
  - Seaborn
- Exercícios
- Referências

Objetivo

- Objetivo dessa aula é introduzir as **bibliotecas de visualização de dados** disponíveis no **Python**.

Conteúdo da Aula

- Objetivo
- **Visualização dos dados com Python**
  - Plotagem no Pandas
  - Matplotlib
  - Seaborn
- Exercícios
- Referências

Visualização dos dados com Python

- A visualização de dados faz parte do passo tanto do processamento quanto da apresentação dos dados.
- É mais fácil comparar valores quando estes são plotados do que comparar valores numéricos.
- Podemos identificar padrões ocultos nos dados que podemos explorar.
- Python tem **diversas bibliotecas** de visualização de dados.

Visualização dos dados com Python

- Existem bibliotecas para propósitos específicos, como rastrear o movimento dos olhos (**`GazeParser`**), até visualização em tempo real de redes neurais (**`pastalog`**).

- Uma lista das bibliotecas de visualização existente no PyPI - https://pypi.python.org/pypi?:action=browse&c=399

Visualização dos dados com Python

- Porém existem bibliotecas genéricas que podem ser utilizadas para diversos fins, como:
  - Matplotlib
  - Seaborn
  - Bokeh

matplotlib

- É uma das bibliotecas de visualização mais antiga do Python (2002), porém muito utilizada ainda.
- Funciona muito bem para realizarmos analises iniciais no dados, ter uma noção do que temos. Porém ela não é muito útil para a criação de gráficos com qualidade de publicação rápida e fácil.
- Ela é muito poderosa, porém complexa!
- Galeria de exemplos: http://matplotlib.org/examples/index.html

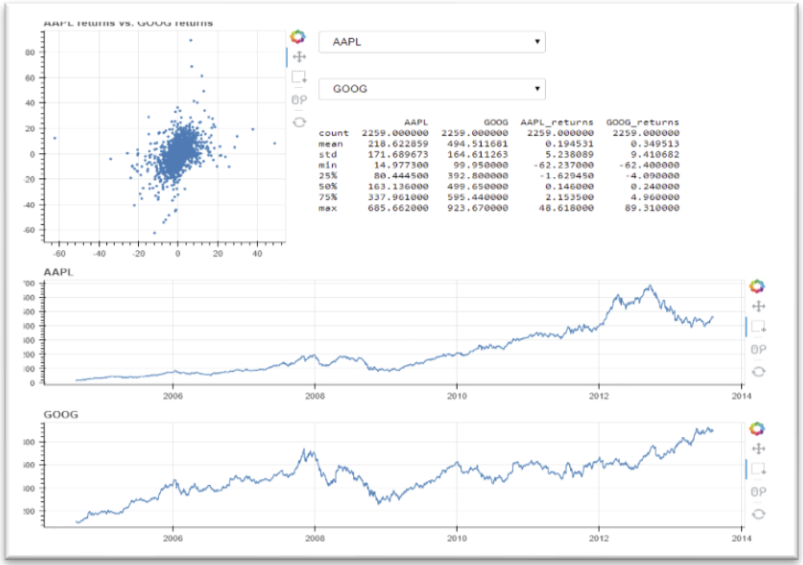
seaborn

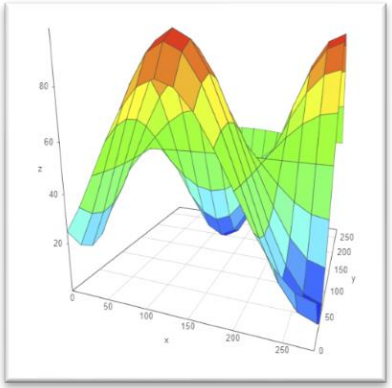
- O **seaborn** é baseado na biblioteca matplotlib e oferece uma interface de nível mais alto para gráficos estatísticos.
- Fornece uma interface para criar visualizações mais bonitas e complexas com menos linhas de código.
- Criado por Michael Waskom em meados de 2012.
- Galeria: https://seaborn.pydata.org/examples/index.htm
- Documentação: https://seaborn.pydata.org/tutorial.html

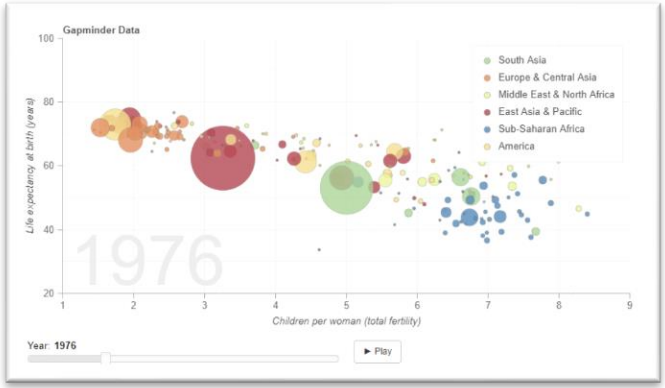
bokeh

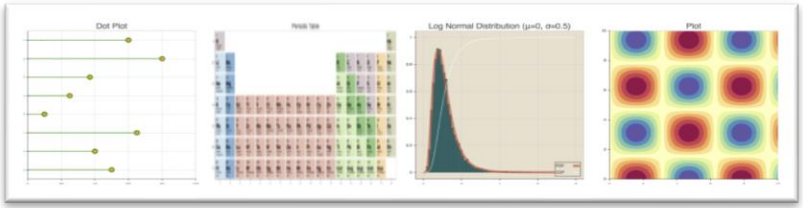
- **Bokeh** é uma biblioteca de visualização interativa que roda nos principais navegadores.
- Foi criado pela Continuum Analytics.
- Possibilita construir de forma simples e rápida gráficos elegantes, concisos e de alto desempenho em grandes volumes de dados.

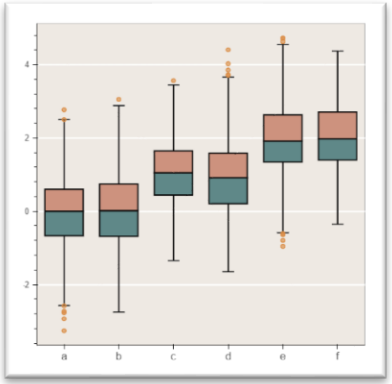
bokeh

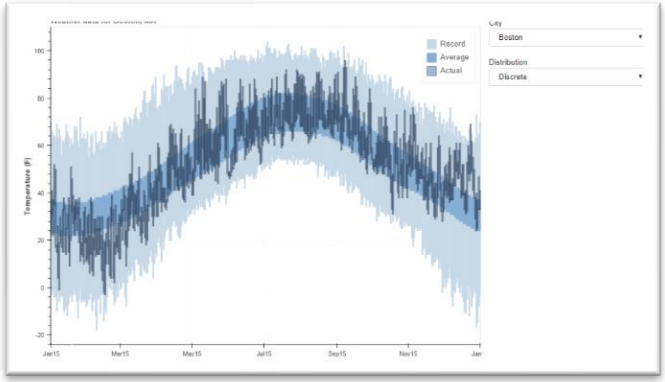

AAPL
GOOG
AAPL
GOOG



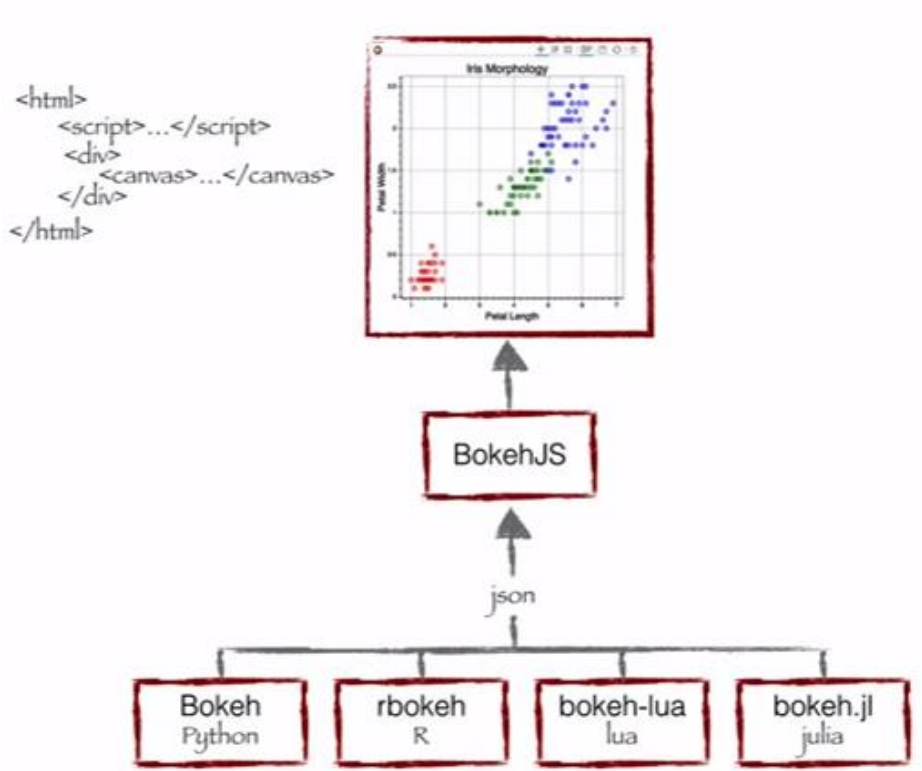

Gapminder Data
South Asia
Europe & Central Asia
Middle East & North Africa
East Asia & Pacific
Sub-Saharan Africa
America
1976
Children per woman (total fertility)
Year: 1976
Play


Dot Plot
Log Normal Distribution (μ=0, σ=0.5)
Plot




Record
Average
Actual
Boston
Discrete

bokeh



Fonte: https://goo.gl/esSgAo

Visualização de dados com Python

- Para instalar essas bibliotecas:
  - `pip install matplotlib`
  - `pip install seaborn`
  - `pip install bokeh`

Visualização de dados com Python

- Iremos abordar:
  - Plotagem do Pandas
  - Matplotlib
  - Seaborn

Conteúdo da Aula

- Objetivo
- Visualização dos dados com Python
  - **Plotagem no Pandas**
  - Matplotlib
  - Seaborn
- Exercícios
- Referências

Plotagem no Pandas

- Os objetos do Pandas também têm as próprias funções de plotagem com base na biblioteca `matplotlib`.

- Em geral, a plotagem utilizando Pandas é feita de acordo com as funções `DataFrame.plot.TIPO_PLOTAGEM` ou `Series.plot.TIPO_PLOTAGEM`.

</> Abra o arquivo **"aula8-parte1-plot-pandas.ipynb"**

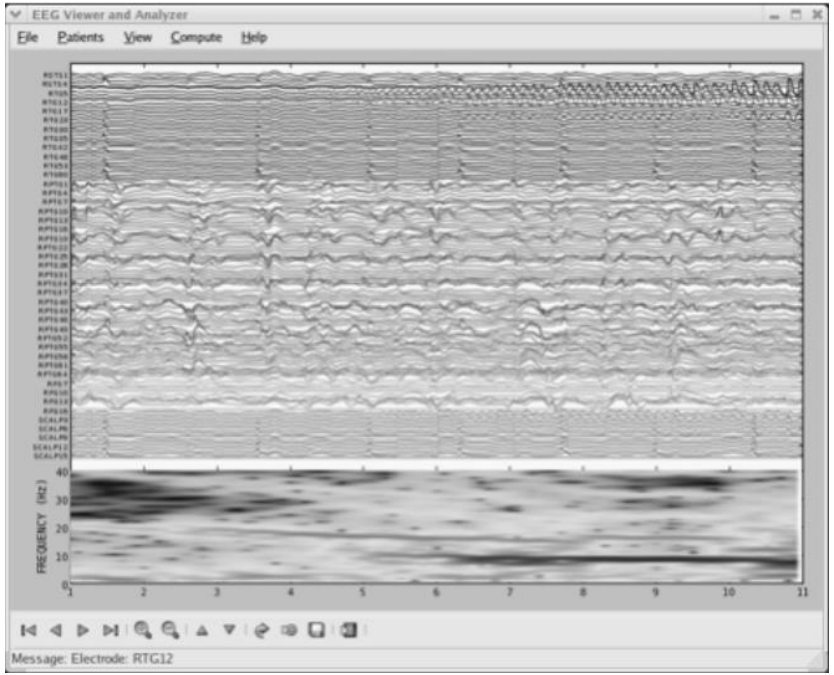
Conteúdo da Aula

- Objetivo
- Visualização dos dados com Python
  - Plotagem no Pandas
  - **Matplotlib**
  - Seaborn
- Exercícios
- Referências

matplotlib

- Como vimos anteriormente, o Pandas fornece um objeto de plotagem que facilita bastante a criação de alguns gráficos.
- Isso é possível, pois esse objeto de plotagem foi construído em cima do Matplotlib.
- A biblioteca Matplotlib é uma das mais antigas e ela surgiu como uma alternativa ao MATLAB.

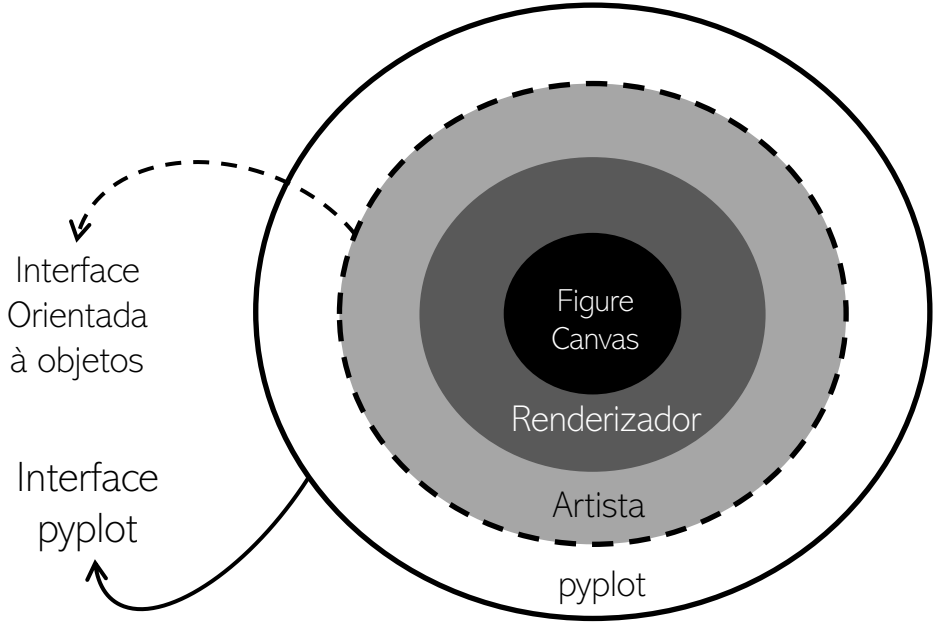
matplotlib



Matplotlib 1.0

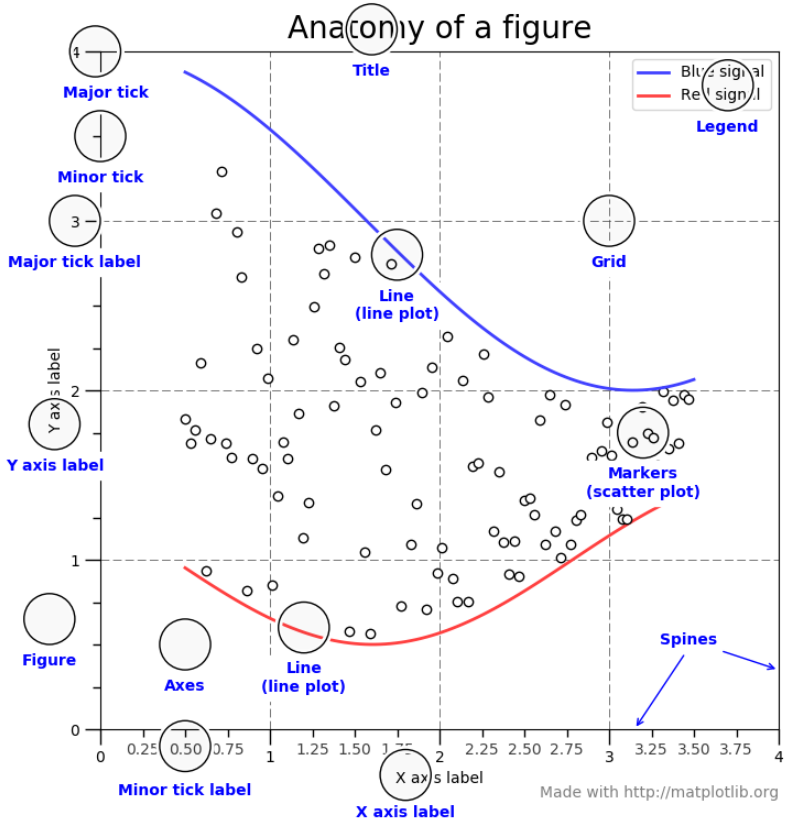
matplotlib

- Existem duas formas de criar gráficos no matplotlib
  - Pyplot API
  - Object Oriented API



matplotlib

- Além das APIs de utilização, precisamos entender como as figuras são construídas e quais elementos podemos utilizar nas figuras.



matplotlib

</> Abra o arquivo **"aula8-parte2-matplotlib.ipynb"**

Conteúdo da Aula

- Objetivo
- Visualização dos dados com Python
  - Plotagem no Pandas
  - Matplotlib
  - **Seaborn**
- Exercícios
- Referências

seaborn

- Como vimos a biblioteca matplotlib como principal ferramenta de plotagem em Python.
- O `seaborn` é baseado na matplotlib e oferece uma interface de nível mais alto para gráficos estatísticos.
- Ele provê uma interface para criar visualizações mais bonitas e complexas com menos linhas de códigos.

seaborn

- A biblioteca `seaborn` está altamente integrada ao Pandas, facilitando que o processo de análise de dados seja muito simples.
- Como o `seaborn` foi desenvolvido com base na matplotlib, o usuário continua tenda a capacidade de fazer ajustes finos nas visualizações.

seaborn

- Iremos criar os seguintes gráficos:
  - Histograma
  - Densidade
  - Rugs
  - Hexbin
  - Plotagem de densidade 2D
  - Dados multivariados

seaborn

- Iremos criar os seguintes gráficos:
  - **Histograma**
  - **Densidade**
  - **Rugs**
  - Hexbin
  - Dados multivariados

seaborn

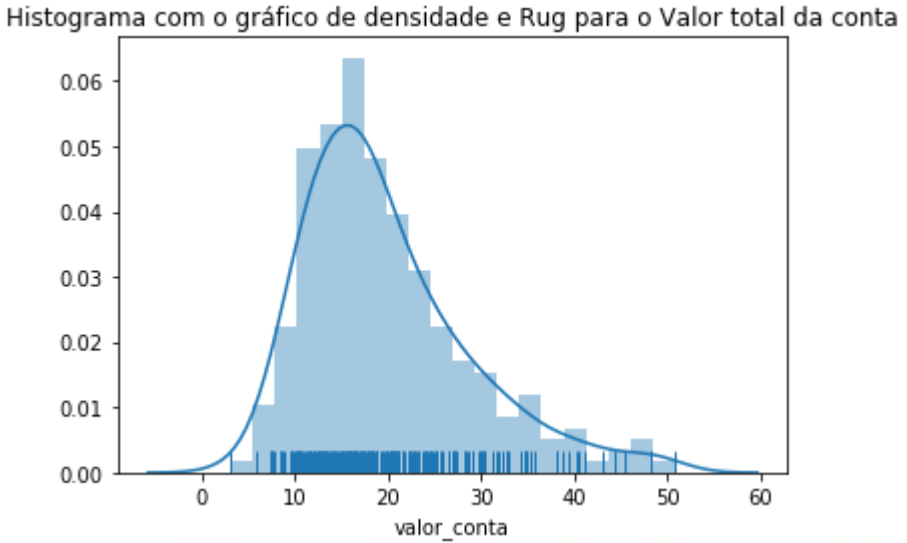
- Por padrão, o método `displot` irá plotar um histograma bem como um gráfico de densidade.
- Um gráfico de densidade permite visualizar a distribuição de dados em um intervalo ou período de tempo contínuo. Este gráfico é uma variação de um histograma que usa a suavização de kernel para plotar os valores, permitindo visualizar as distribuições mais suaves quando existe o ruído.
- Os picos no gráfico de densidade ajudam a exibir onde os valores são concentrados no intervalo.

seaborn

- Uma vantagem que os gráficos de densidade têm é que eles não são afetados pelo parâmetro bins que ocorre nos Histogramas.
- Um histograma composto por apenas 4 bins não produziria uma forma de distribuição suficientemente distinta, como faria um histograma de 20.
- No entanto, **com o gráfico de densidade isso não é um problema**.

seaborn

```
>>> hist, ax = plt.subplots()

>>> ax = sns.distplot(rest['valor_conta'], rug=True, bins=20)

>>> ax.set_title('Histograma com o gráfico de densidade e Rug')

>>> plt.show()
```



seaborn

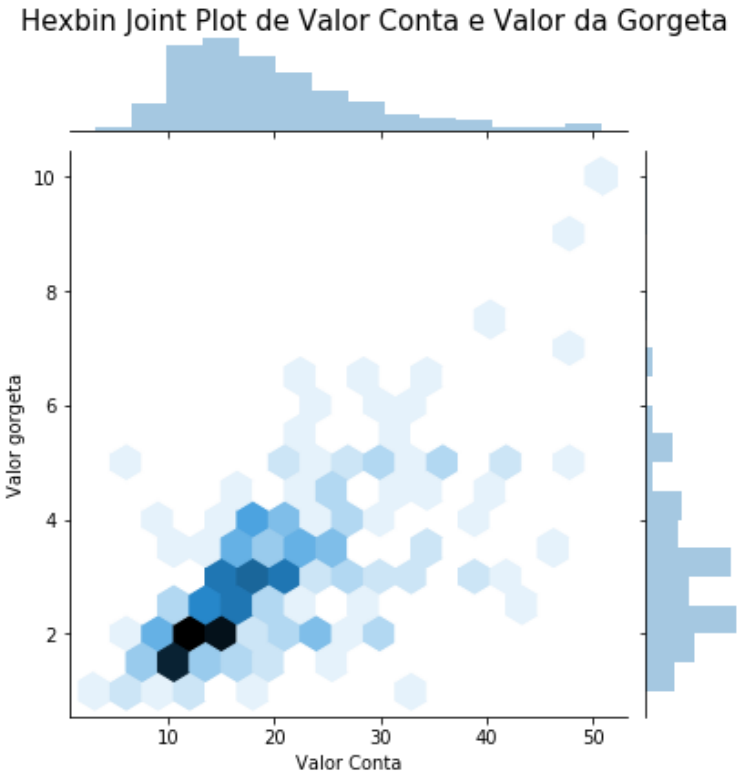
- Iremos criar os seguintes gráficos:
  - Histograma
  - Densidade
  - Rugs
  - **Hexbin**
  - Dados multivariados

seaborn

- Os gráficos de dispersão são ótimos para comparar duas variáveis.
- Porém, as vezes, há pontos demais para que um gráfico de dispersão seja significativo. Uma forma de contornar esse problema é **reunir pontos no gráfico**.
- Assim como os histogramas podem reunir dados de uma variável para criar uma barra, o **`hexbin`** pode fazer o mesmo com duas variáveis.
- Um hexágono é usado com essa finalidade, pois é o formato mais eficiente para cobrir uma superfície 2D arbitrária.
- Utiliza-se o método **`joinplot`** para criar o hexbin.

seaborn

```
>>> hexbin = sns.jointplot(x='valor_conta',
y='valor_gorgeta', data=rest, kind='hex’)

>>> hexbin.set_axis_labels(xlabel='Valor
Conta', ylabel='Valor gorgeta’)

>>> hexbin.fig.suptitle('Hexbin Joint Plot
de Valor Conta e Valor da Gorgeta',
fontsize=15, y=1);
```



seaborn

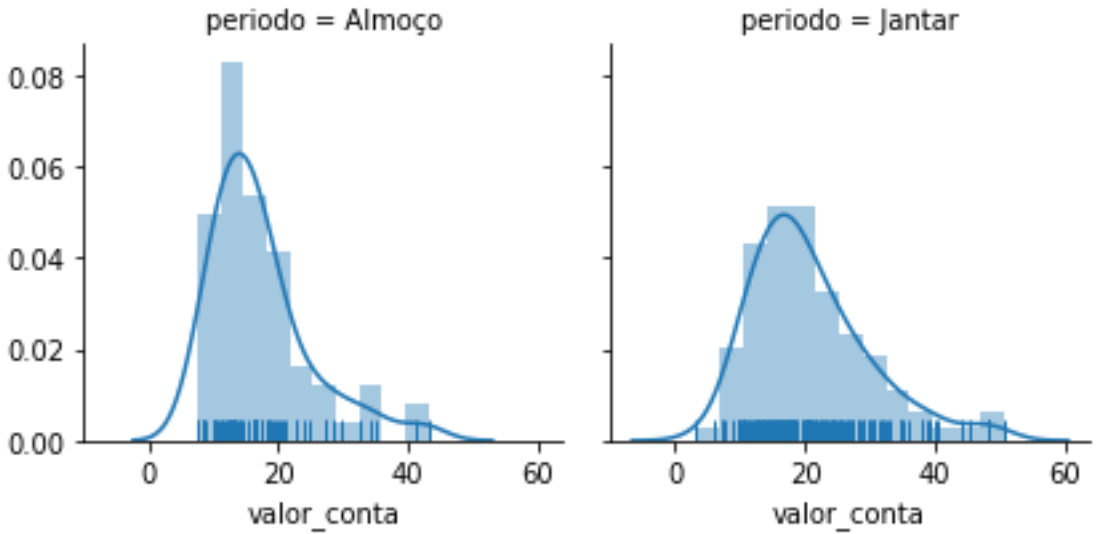
- Iremos criar os seguintes gráficos:
  - Histograma
  - Densidade
  - Rugs
  - Hexbin
  - **Dados multivariados**

seaborn

- Podemos utilizar diferentes maneiras para incluir mais informações no gráficos, utilizando cores, tamanho e formato para distinguir os dados na plotagem.
- Podemos utilizar as Facetas geram os subconjuntos e adicionam na figura de forma simples e rápida.

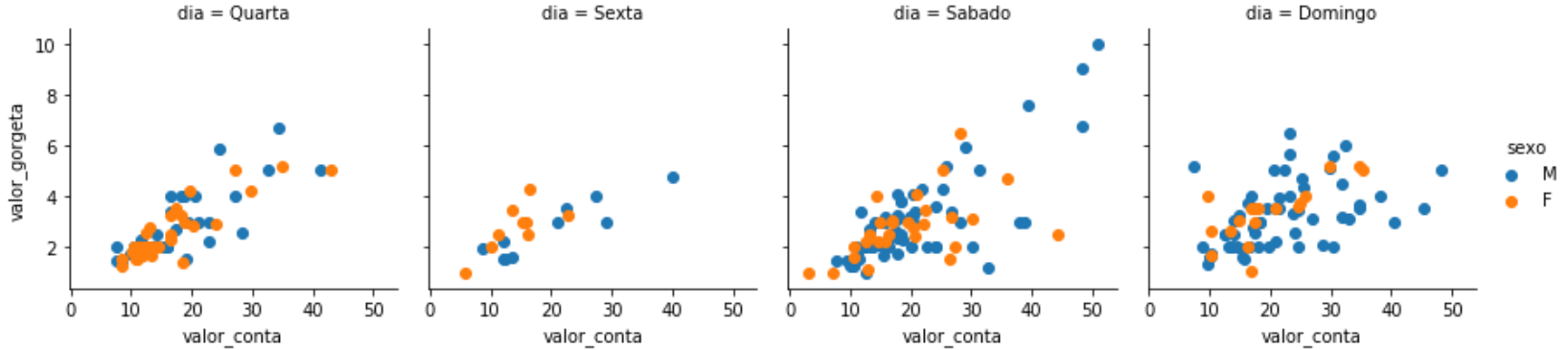
seaborn

```
>>> faceta = sns.FacetGrid(rest, col='periodo')

>>> faceta.map(sns.distplot, 'valor_conta', rug=True)

>>> plt.show()
```



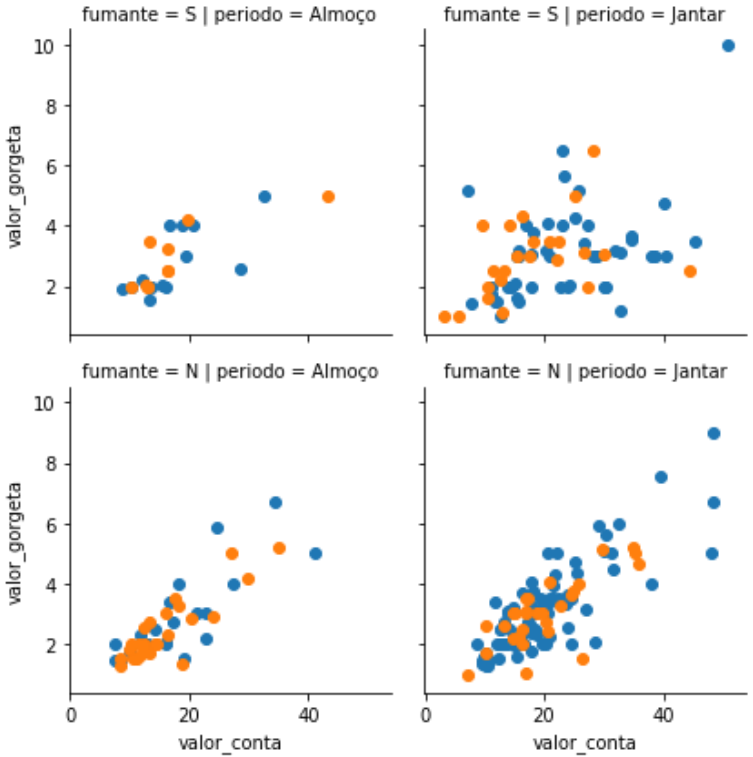
seaborn

```
>>> faceta = sns.FacetGrid(rest, col='dia', hue='sexo')

>>> faceta = faceta.map(plt.scatter, 'valor_conta', 'valor_gorgeta’)

>>> faceta = faceta.add_legend()
```



seaborn

```
>>> faceta = sns.FacetGrid(rest,
col='periodo', row='fumante', hue='sexo')


>>> faceta.map(plt.scatter, 'valor_conta',
'valor_gorgeta')
```



seaborn

</> Abra o arquivo **"aula8-parte3-seaborn.ipynb"**

Conteúdo da Aula

- Objetivo
- Visualização dos dados com Python
  - Plotagem no Pandas
  - Matplotlib
  - Seaborn
- **Exercícios**
- Referências

seaborn

</> Abra o arquivo **"aula8-parte4-exercicios.ipynb"**

Conteúdo da Aula

- Objetivo
- Visualização dos dados com Python
  - Plotagem no Pandas
  - Matplotlib
  - Seaborn
- Exercícios
- **Referências**

Referências Bibliográficas

- **Learning Python Data Visualization** – Shai Vaingast – Apress, 2014.
- **Beginning Python Visualization** – Shai Vaingast – Apress, 2014.
- **Mastering pandas** – Femi Anthony – Packt Publishing, 2015.
- Referência Bokeh - http://bokeh.pydata.org/en/latest/docs/reference.html

Referências Bibliográficas

- **Data Science from Scratch** – Joel Grus – O'Reilly, 2015.

- **Python for Data Analysis** – Wes McKinney – USA: O'Reilly, 2013.

- As referências de links utilizados podem ser visualizados em http://urls.dinomagri.com/refs

Material Extra

Material Extra

- Utilizando os dados do Twitter, iremos criar um mapa para plotar os tweets e os usuários que tem algum local definido, seja via coordenadas ou nome do local ☺
- A ideia é utilizar a biblioteca Bokeh + Google Maps API

Material Extra

- Utilizando o arquivo `tweets_10min.csv` iremos utilizar a latitude e longitude para plotar no Google Maps a localização dos usuários.
- Também iremos adicionar informações complementares (localização e usuário) em cada ponto gerado no gráfico.

Material Extra

- Para isso, temos que realizar 5 passos:
  1. Carregar o conjunto de dados
  2. Criar um mapa
  3. Gerar a API para acessar o Google Maps
  4. Adicionar os tweets no mapa (lat, long)
  5. Adicionar as informações complementares

</> Abra o arquivo **"aula8-parte5-extra.ipynb"**