Python para Ciência de Dados

SQLAlchemy e Web Scraping

Luiz Alberto

Ciência da Computação

November 9, 2019

Contando Valores Distintos com SQLAlchemy

```
1
2  from sqlalchemy import create_engine, Table, MetaData, select,
       func
3
4  engine = create_engine('sqlite:///datasets/Northwind.sqlite')
5
6  metadata = MetaData()
7  orders   = Table('Orders', metadata, autoload=True,
       autoload_with=engine)
8
9  stmt   = select([func.count(orders.columns.ShipCountry.
       distinct())])
10
11 with engine.connect() as conn:
12   distinct_countries = conn.execute(stmt).scalar()
13
14 print(distinct_countries)
15 # 21
```

Contando Registros com SQLAlchemy (1)

```
1  import pandas as pd
2  from sqlalchemy import create_engine, Table, MetaData, select,
       func
3
4  engine = create_engine('sqlite:///datasets/Northwind.sqlite')
5
6  metadata  = MetaData()
7  customers = Table('Customers', metadata, autoload=True,
       autoload_with=engine)
8
9  stmt = select([customers.columns.Country
10   ,func.count(customers.columns.Id).label('Count')])
11 stmt = stmt.group_by(customers.columns.Country)
12
13 with engine.connect() as conn:
14   results   = conn.execute(stmt).fetchall()
15   countries = pd.DataFrame(results)
16   countries.columns = results[0].keys()
17
```

Contando Registros com SQLAlchemy (2)

```
18 print(countries)
19 #          Country    Count
20 # 0      Argentina        3
21 # 1        Austria        2
22 # 2        Belgium        2
23 # 3         Brazil        9
24 # 4         Canada        3
```

Somando Valores com SQLAlchemy (1)

```
1  import pandas as pd
2  from sqlalchemy import create_engine, Table, MetaData, select,
       func
3
4  engine = create_engine('sqlite:///datasets/Northwind.sqlite')
5
6  metadata  = MetaData()
7  details   = Table('OrderDetails', metadata, autoload=True,
       autoload_with=engine)
8
9  stmt = select([details.columns.ProductId
10   ,func.sum(details.columns.Id).label('Quantity')])
11 stmt = stmt.group_by(details.columns.ProductId)
12
13 with engine.connect() as conn:
14   results    = conn.execute(stmt).fetchall()
15   quantities = pd.DataFrame(results)
16   quantities.columns = results[0].keys()
17
```

Somando Valores com SQLAlchemy (2)

```
18 print(quantities)
19 #      ProductId  Quantity
20 # 0            1  406841.0
21 # 1            2  470965.0
22 # 2            3  128318.0
23 # 3            4  212436.0
24 # 4            5  106418.0
```

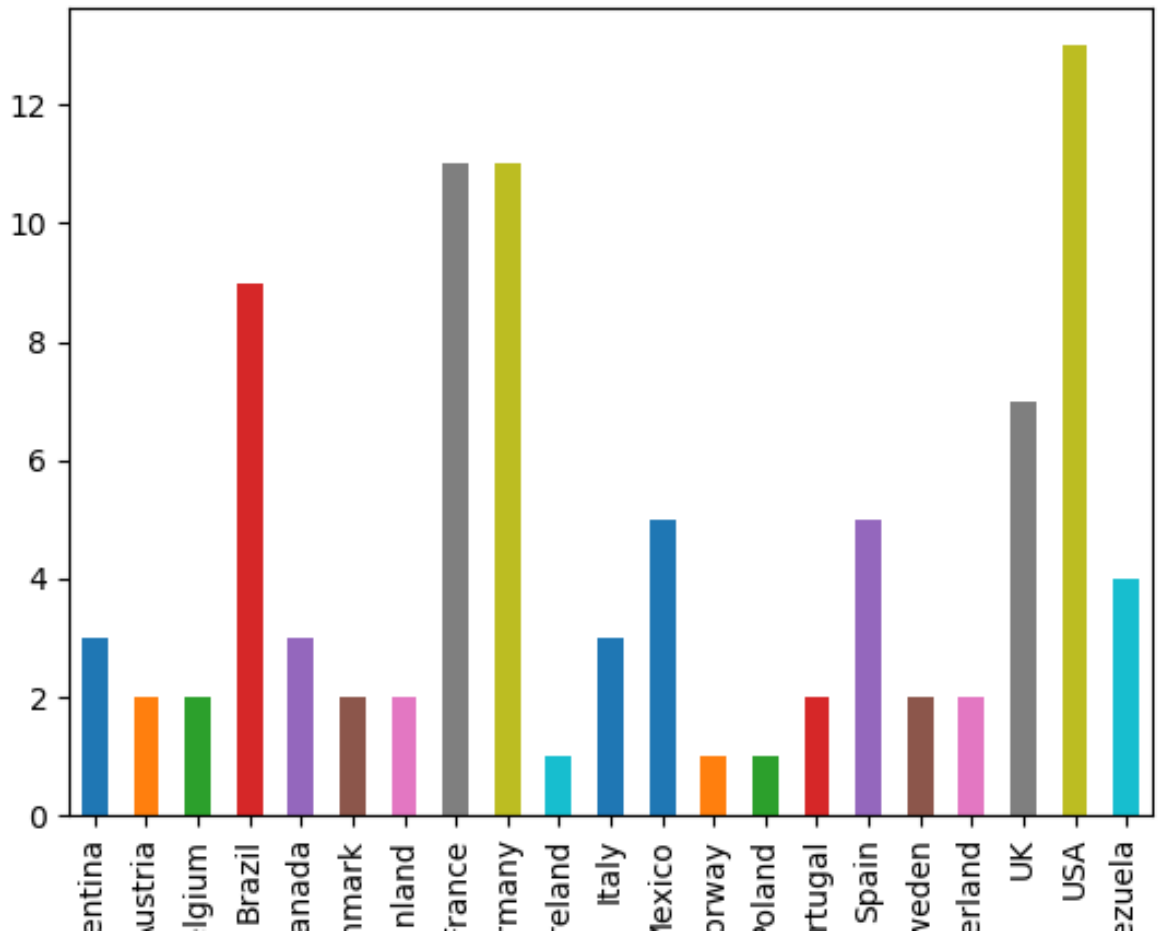
SQLAlchemy, Pandas e Matplotlib (1)

```
1  import pandas as pd
2  import matplotlib.pyplot as plt
3  from sqlalchemy import create_engine, Table, MetaData, select,
       func
4
5  engine = create_engine('sqlite:///datasets/Northwind.sqlite')
6
7  metadata  = MetaData()
8  customers = Table('Customers', metadata, autoload=True,
       autoload_with=engine)
9
10 stmt = select([customers.columns.Country
11   ,func.count(customers.columns.Id).label('Count')])
12 stmt = stmt.group_by(customers.columns.Country)
13
14 with engine.connect() as conn:
15   results   = conn.execute(stmt).fetchall()
16   countries = pd.DataFrame(results)
17   countries.columns = results[0].keys()
```

SQLAlchemy, Pandas e Matplotlib (2)

```
18
19 countries.plot(kind="bar", x='Country', y='Count', legend=None
       )
20 plt.show()
```

SQLAlchemy, Pandas e Matplotlib (3)



Hora de colocar as mãos na massa

- Na pasta `python-para-ciencia-de-dados/notebooks`, abra o arquivo `aula-4-maos-na-massa-1.ipynb`
- Faça todos os exercícios neste notebook.

Importando dados da web

- O pacote **urllib** fornece uma interface para ler dados da web
- `urlopen` - aceita uma url ao invés de um nome de arquivo

Importando dados da web

```
1
2 import pandas as pd
3 from urllib.request import urlretrieve
4 url = "https://archive.ics.uci.edu/ml/machine-learning-
      databases/wine-quality/winequality-red.csv"
5 urlretrieve(url, 'winequality-red.csv')
6 df = pd.read_csv('winequality-red.csv', sep=';')
7 print(df.head())
```

URL

- Uniform/Universal Resource Locator
- Referencia recursos na web
- Foco: endereço web
- Ingredientes:
  - identificador do protocolo - http:
  - nome do recurso - `python.org`

HTTP

- HyperText Transfer Protocol
- Fundação da comunicação de dados na Web
- HTTPS - forma mais segura do HTTP
- `urlretrieve()` executa uma requisição GET

Requisição GET utilizando **urllib**

```
1
2 import pandas as pd
3 from urllib.request import urlopen, Request
4 url = "https://www.wikipedia.org"
5 request  = Request(url)
6 response = urlopen(request)
7 html = response.read()
8 response.close()
```

Requisição GET utilizando **requests**

```
1 import requests
2 url = "https://www.wikipedia.org"
3 r = requests.get(url)
4 texto = r.text
```

Web Scraping com Python
HTML

- Mistura dados estruturados e não-estruturados
- Dados estruturados:
  - tem um modelo de dados pré-definido
  - organizado de uma forma definida
- Dados não estruturados: nenhum propriedade definida

Parser e Extração Usando **BeautifulSoup**

HTML

- Mistura dados estruturados e não-estruturados
- Dados estruturados:
  - tem um modelo de dados pré-definido
  - organizado de uma forma definida
- Dados não estruturados: nenhum propriedade definida

Parser e Extração Usando **BeautifulSoup**

```
1  from bs4 import BeautifulSoup
2  import requests
3  url='https://www.crummy.com/software/BeautifulSoup/'
4  r = requests.get(url)
5  html_doc = r.text
6  soup = BeautifulSoup(html_doc)
7  print(soup.prettify())
8  # <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 ...
9  # <html>
10 # <head>
11 #  <meta content="text/html; charset=utf-8" http-equiv="
       Content-Type"/>
12 #  <title>
13 #   Beautiful Soup: We called him Tortoise because he taught
       us.
14 #  </title>
15 # ...
```

Explorando o **BeautifulSoup** (1)

- Fornece vários métodos tais como:

```
1  from bs4 import BeautifulSoup
2  import requests
3  url='https://www.crummy.com/software/BeautifulSoup/'
4  r = requests.get(url)
5  html_doc = r.text
6  soup = BeautifulSoup(html_doc)
7  print(soup.title)
8  #  <title>
9  #   Beautiful Soup: We called him Tortoise because he taught
       us.
10 #  </title>
```

Explorando o **BeautifulSoup** (2)

```
1  from bs4 import BeautifulSoup
2  import requests
3  url='https://www.crummy.com/software/BeautifulSoup/'
4  r = requests.get(url)
5  html_doc = r.text
6  soup = BeautifulSoup(html_doc)
7  print(soup.get_text())
8  # Beautiful Soup: We called him Tortoise because he taught us.
9  #
10 #
11 # You didn't write that awful page. You're just trying to get
       some
12 # data out of it. Beautiful Soup is here to help. Since 2004,
       it's been
13 # saving programmers hours or days of work on quick-turnaround
```

Explorando o **BeautifulSoup** (3)

```
1  from bs4 import BeautifulSoup
2  import requests
3  url='https://www.crummy.com/software/BeautifulSoup/'
4  r = requests.get(url)
5  html_doc = r.text
6  soup = BeautifulSoup(html_doc)
7  for link in soup.find_all('a'):
8    print(link.get('href'))
9  # bs4/download/
10 # #Download
11 # bs4/doc/
12 # #HallOfFame
13 # https://code.launchpad.net/beautifulsoup
14 # https://bazaar.launchpad.net/%7Eleonardr/beautifulsoup/bs4/
       view/head:/CHANGELOG
```

Hora de colocar as mãos na massa

- Na pasta `python-para-ciencia-de-dados/notebooks`, abra o arquivo `aula-4-maos-na-massa-2.ipynb`
- Faça todos os exercícios neste notebook.