

Measurement Research within the Python3 Ecosystem

Steffie Jacob Eravuchira and Vaibhav Bajpai
([s.eravuchira](mailto:s.eravuchira@jacobs-university.de)|[v.bajpai](mailto:v.bajpai@jacobs-university.de))@jacobs-university.de

Computer Networks and Distributed Systems Group,
Jacobs University Bremen,
Bremen, Germany

RIPE 69, London

November 2014

Leone Project: leone-project.eu

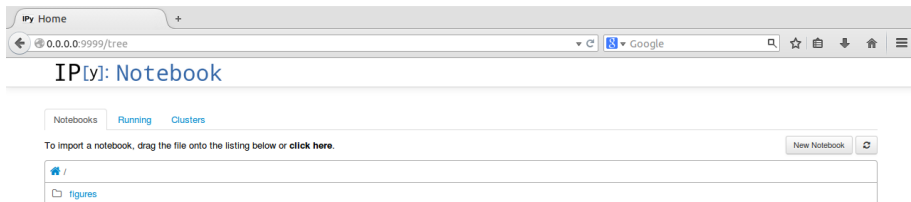
Table of Contents

- 1 Development Environment
- 2 Fetching Multi-Dimensional Data
- 3 Peeking at the Raw Data
- 4 Frictionless SQL storage
- 5 Frictionless Data Retrieval
- 6 Data Analysis
- 7 Data Visualization

Development Environment

ipython notebook

```
$ ipython notebook --pylab='inline'
--ip='0.0.0.0'
--port=9999
```



Fetching Multi-Dimensional Data

```
import requests
```



Requests

```
import requests
```

```
def get_json_from_url(url):  
    try: res = requests.get(url)  
    except Exception as e: print(e)  
    else: return(res.json())
```

Fetching Multi-Dimensional Data

requests + atlas.ripe.net

```
count = get_total_registered_atlas_probes()1  
print("Total Registered RIPE Atlas probes: %d"%total)
```

```
[Out] Total Registered RIPE Atlas probes: 10434
```

¹<http://goo.gl/FDqISn>

Fetching Multi-Dimensional Data

requests + stat.ripe.net

```
asn, holder = asn_from_endpoint('8.8.8.8')2  
print(asn, holder)
```

```
[Out] 15169 GOOGLE - Google Inc.,US
```

²<http://goo.gl/FDqISn>

Fetching Multi-Dimensional Data

requests + stat.ripe.net

```
holder = holder_from_asn(15169)3  
print(holder)
```

```
[Out] GOOGLE - Google Inc.,US
```

³<http://goo.gl/FDqISn>

Fetching Multi-Dimensional Data

requests + stat.ripe.net

```
node = create_pretty_node_names(holder, asn)4  
print(node)
```

```
[Out] GOOGLE (AS15169)
```

⁴<http://goo.gl/FDqISn>

Fetching Multi-Dimensional Data

requests + peeringdb

Peeking at the Raw Data

csvkit

```
$ csvcut -n data.csv
```

```
$ csvcut -c col1, col2,... data.csv
```

```
$ csvgrep -c col1 -m pattern data.csv
```

```
$ csvsort -c col1 data.csv
```

Frictionless SQL storage

Pandas Dataframe

```
import pandas as pd
...
url = 'https://atlas.ripe.net/api/v1/probe '
res = get_json_from_url(url)
df = pd.DataFrame(res['objects'])
df.head()
```

	address_v4	address_v6	asn_v4	asn_v6	country_code	id	is_anchor	is_public	latitude
0	NaN	NaN	6830	NaN	NL	1	False	False	52.3875
1	NaN	NaN	5615	NaN	NL	2	False	False	52.3085
2	213.93.36.211	None	6830	NaN	NL	3	False	True	52.3385
3	83.163.50.165	2001:980:57a4:1:220:4aff:fec8:244a	3265	3265	NL	4	False	True	52.3995
4	81.56.221.200	None	12322	NaN	NL	5	False	True	52.0595

Frictionless SQL storage

```
import pandas as pd; pd.to_sql(...)
```

```
import sqlite3
con = sqlite3.connect('$DB_LOCATION')
df.to_sql('$TABLENAME'
          , con
          , flavor='sqlite'
          , if_exists = 'append'
          , index_label = 'id'
          )
```

Frictionless Data Retrieval

```
import pandas as pd; pd.read(...)
```

```
query = '''select asn_v4,count(asn_v4) as  
           probe_count from %s where is_anchor = 0  
           and status = 1 group by asn_v4 having  
           count(asn_v4) > 50'''%TABLENAME
```

```
df = pd.read_sql(query, con)  
df.head()
```

	asn_v4	probe_count
0	701	65
1	2856	73
2	3215	94
3	3265	75
4	3320	191

Data Analysis

```
import ipaddress
```

```
import ipaddress
shared=ipaddress.ip_network('100.64.0.0/10')
myip=ipaddress.ip_address('192.168.0.1')
isPvt = myip.is_private
print('Is %s Private[RFC1918]? %s'%(myip,isPvt))
```

```
[Out] Is 192.168.0.1 Private[RFC1918]? True
```

Data Analysis

```
import socket
```

```
import socket
```

```
def if_ipv6(address):
```

```
    try:
```

```
        socket.inet_pton(socket.AF_INET6, address)
```

```
        return True
```

```
    except socket.error: return False
```

```
def if_ipv4(address):
```

```
    try:
```

```
        socket.inet_pton(socket.AF_INET, address)
```

```
        return True
```

```
    except socket.error: return False
```

Data Visualization

```
import matplotlib
```

```
df.head()
df = df.sort('probe_count', ascending=False)
ax = df.plot('asn_v4', 'probe_count', kind='bar')
ax.set_xlabel("ASN");
ax.set_ylabel("#Connected Probes");
```

	asn_v4	probe_count
0	701	65
1	2856	73
2	3215	94
3	3265	75
4	3320	191

