

合同会社 緑 I T 事務所

Midori IT Office, LLC

データビジュアライゼーション(2)

この記事は1年以上前に書かれました。
内容が古くなっている可能性がありますのでご注意ください。

[LODとSPARQL入門\(1\)](#)で簡単なSPARQLクエリを書けるようになりましたので、今回はSgvizlerというオープンソース（MITライセンス）のJavaScriptライブラリを使用してSPARQLクエリの結果をビジュアライズします。

HTMLファイルの完全なリストは以下のとおりです。

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Sgvizler Sample</title>
  <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.js"></script>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript" src="http://beta.data2000.no/sgvizler/release/0.6/sgvizler.js"></script>
  <script type="text/javascript">
    $(document).ready(function (){ sgvizler.containerDrawAll();});
  </script>
</head>
<body>
  <div id="sample"
    data-sgvizler-endpoint="http://ja.dbpedia.org/sparql"
    data-sgvizler-query="
      SELECT DISTINCT ?kenmei COUNT(?shi) AS ?cities WHERE {
        ?ken rdf:type schema:AdministrativeArea ;
        dbpedia-owl:country dbpedia-ja:日本 ;
        prop-ja:name ?kenmei .
        ?shi rdf:type dbpedia-owl:City ;
        dbpedia-owl:location ?ken .
      }
      GROUP BY ?kenmei
      ORDER BY DESC(?cities)"
    data-sgvizler-chart="google.visualization.BarChart"
    data-sgvizler-chart-options="legend=none|fontSize=10"
    style="width:900px; height:900px;">
  </div>
</body>
</html>
```

たったこれだけです。

まず、ヘッダ部の

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.0/jquery.js"></script>
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript" src="http://beta.data2000.no/sgvizler/release/0.6/sgvizler.js"></script>
```

で必要なライブラリを読み込みます。今回は、すべて外部のサイトから読み込むことにします。そして、

```
<script type="text/javascript">
  $(document).ready(function (){ sgvizler.containerDrawAll();});
</script>
```

DOMを構築したらSgvizlerのcontainerDrawAll()メソッドを実行するようにします。

Sgvizlerでは、SPARQLクエリや描画方法の指定を<div>タグの属性として記述することができます。

```
<div id="sample"
  data-sgvizler-endpoint="http://ja.dbpedia.org/sparql"
  data-sgvizler-query="
    SELECT DISTINCT ?kenmei COUNT(?shi) AS ?cities WHERE {
```

```
?ken rdf:type schema:AdministrativeArea ;
dbpedia-owl:country dbpedia-ja:日本 ;
prop-ja:name ?kenmei .
?shi rdf:type dbpedia-owl:City ;
dbpedia-owl:location ?ken .
}
GROUP BY ?kenmei
ORDER BY DESC(?cities)"
data-sgvizler-chart="google.visualization.BarChart"
data-sgvizler-chart-options="legend=none|fontSize=10"
style="width:900px; height:900px;">
</div>
```

まずは、SPARQLエンドポイントの指定です。

```
data-sgvizler-endpoint="http://ja.dbpedia.org/sparql"
```

今回も、DBpedia Japaneseを使用します。

```
data-sgvizler-query="
SELECT DISTINCT ?kenmei COUNT(?shi) AS ?cities WHERE {
  ?ken rdf:type schema:AdministrativeArea ;
dbpedia-owl:country dbpedia-ja:日本 ;
prop-ja:name ?kenmei .
?shi rdf:type dbpedia-owl:City ;
dbpedia-owl:location ?ken .
}
GROUP BY ?kenmei
ORDER BY DESC(?cities)"
```

グラフを描くために数値データが欲しいと思い、苦慮した結果がこのSPARQLクエリです。県別の市町村数を取得しようというものです。

PREFIXはDBpediaであらかじめ定義されているものを使用しています。

まず、「rdf:type」が「schema:AdministrativeArea」ということで県（?ken）を取得できます。「dbpedia-owl:country」を「dbpedia-ja:日本」に限定しておきます。前の行末が「;」ですので、主語の?kenは省略しています。県の「prop-ja:name」で県の名前（?kenmei）が得られます。

「rdf:type」が「dbpedia-owl:City」ということで市町村（?shi）を取得できます。市町村（?shi）の「dbpedia-owl:location」は県（?ken）でなければなりません。

COUNTやGROUP BY、ORDER BYの使い方はSQLと同様です。

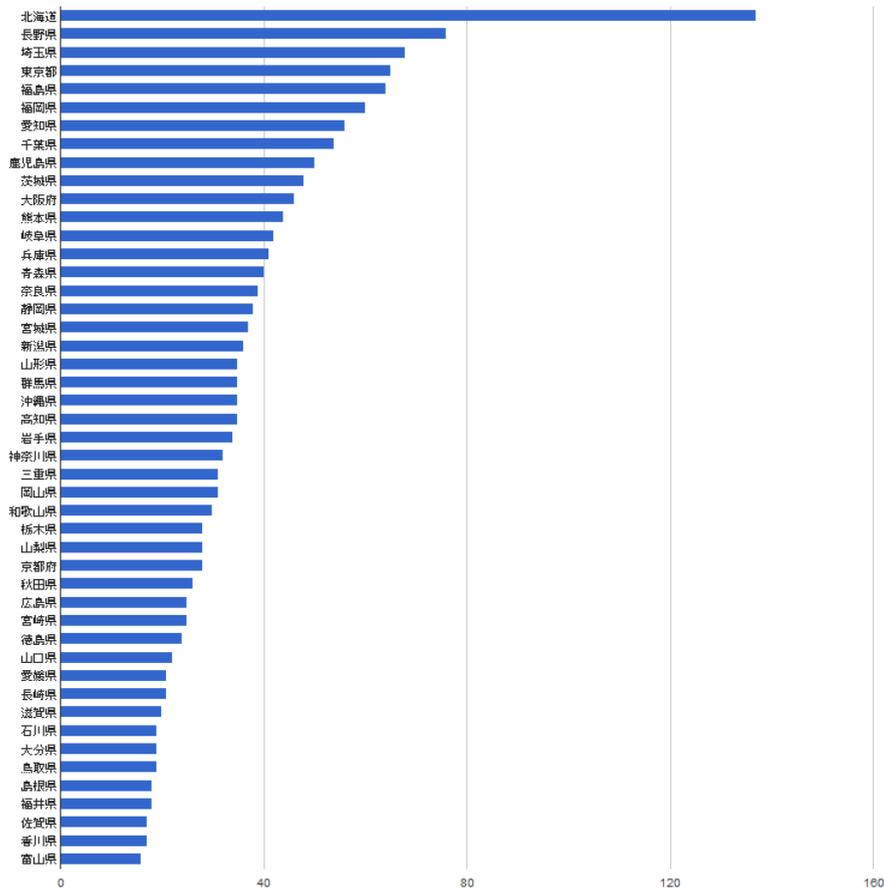
県名（?kenmei）でグループ化し、県ごとの市町村の数（COUNT(?shi)）を数えて?citiesという別名を付け、?citiesの値で降順にソートした結果を取得します。

最後に

```
data-sgvizler-chart="google.visualization.BarChart"
data-sgvizler-chart-options="legend=none|fontSize=10"
style="width:900px; height:900px;">
```

Sgvizlerで表示するチャートの種類を選び、オプションを指定し、領域のサイズを指定しておしまいです。

実行結果は、以下のようになります（画像キャプチャです。実際のHTMLの表示は[こちら](#)）。



今回のSPARQLクエリでは、廃止された市町村も（Wikipediaに記事があれば）カウントしてしまうため、正確な市町村数ではありませんが、これで良いこととします。

データビジュアライゼーション [1](#) [2](#) [3](#) [4](#) [5](#) [6](#)

カテゴリー: オープンデータ, ビジュアライゼーション | タグ: LOD, SPARQL | 投稿日: 2014年3月6日

[<https://midoriit.com/2014/03/%e3%83%87%e3%83%bc%e3%82%bf%e3%83%93%e3%82%b8%e3%83%a5%e3%82%a2%e3%83%a9%e3%82%a4%e3%82%bc%e3%83%bc%e3%82%b7%e3%83%a7%e3%83%b32.html>] | 投稿者: 小池隆