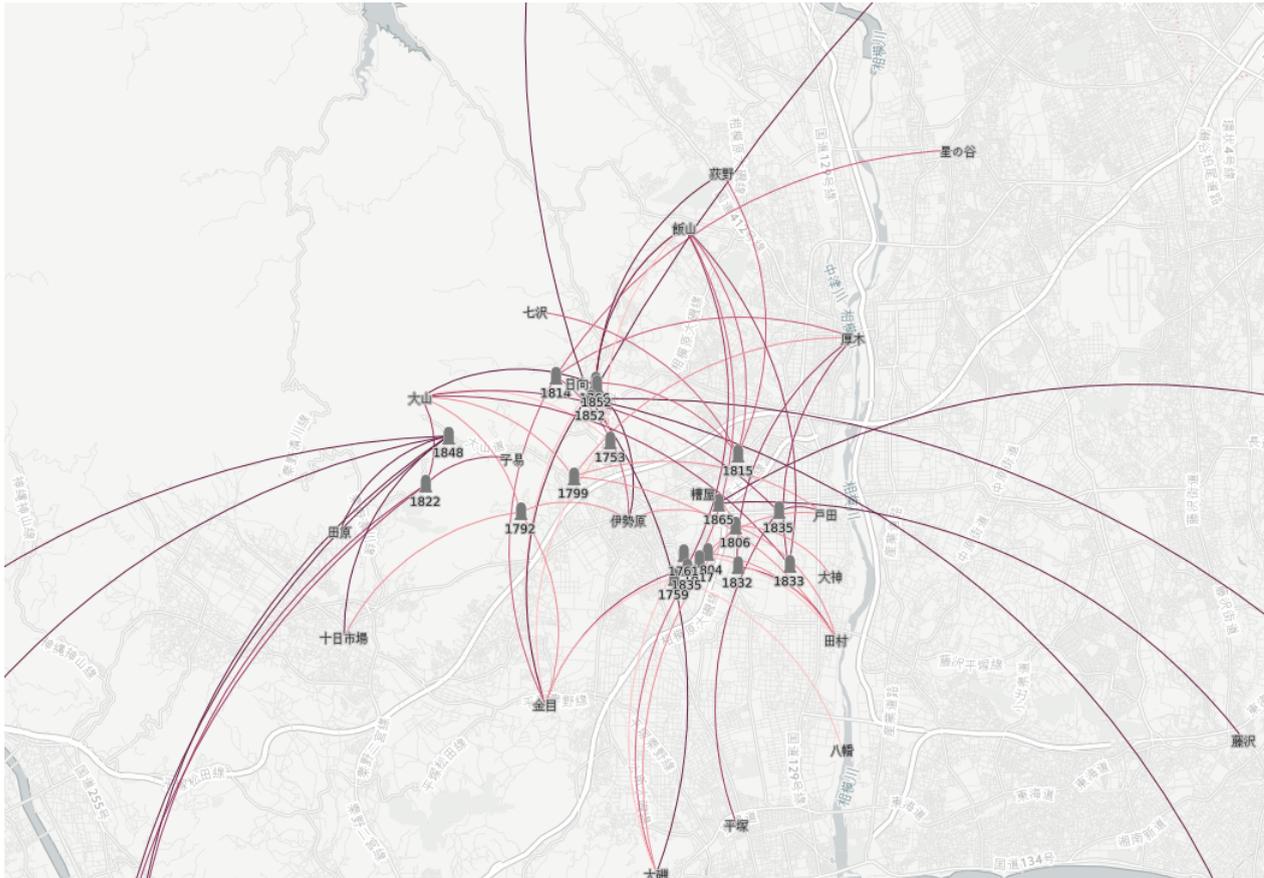


CARTOによる花火図の作成

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。

データビジュアライゼーションの手法の一つに、地図上の点と点を（曲）線でつないだConnection Map（Link Map）があります。同様のものをRESASでは〇〇花火図などと呼んでいます。この手法を用いて、江戸時代に造立された道標の所在地と道標に刻まれた標示地名の位置を曲線でつないだ道標花火図をCARTO上で作成しました。



データの作成

伊勢原市教育委員会が2012年に発行した「伊勢原市内の大山道と道標 第二版」という書籍を参考に、

- ・江戸時代に造立された
- ・当初の造立地近くに現存する
- ・市外の2つ以上の地名が標示されている

という条件に合う20基の道標についてデータ化しました。

道標の所在地の緯度経度は、書籍を参考に地理院地図から取得し、道標に刻まれた標示地名は現在の地名に比定してCSVアドレスマッチングサービスで緯度経度に変換しました。

CARTOにアップロードしたデータセットは[こちら](#)です。

緯度経度のうち、loc_latとloc_lonは道標の所在地を、fy、fxは道標に刻まれた標示地を表します。

所在地と標示地の組み合わせ毎に1レコードとしています。今回は複数の標示地のある道標を対象にしているため、1つの道標について必ず複数のレコードがあることになります。

直線でつなぐ

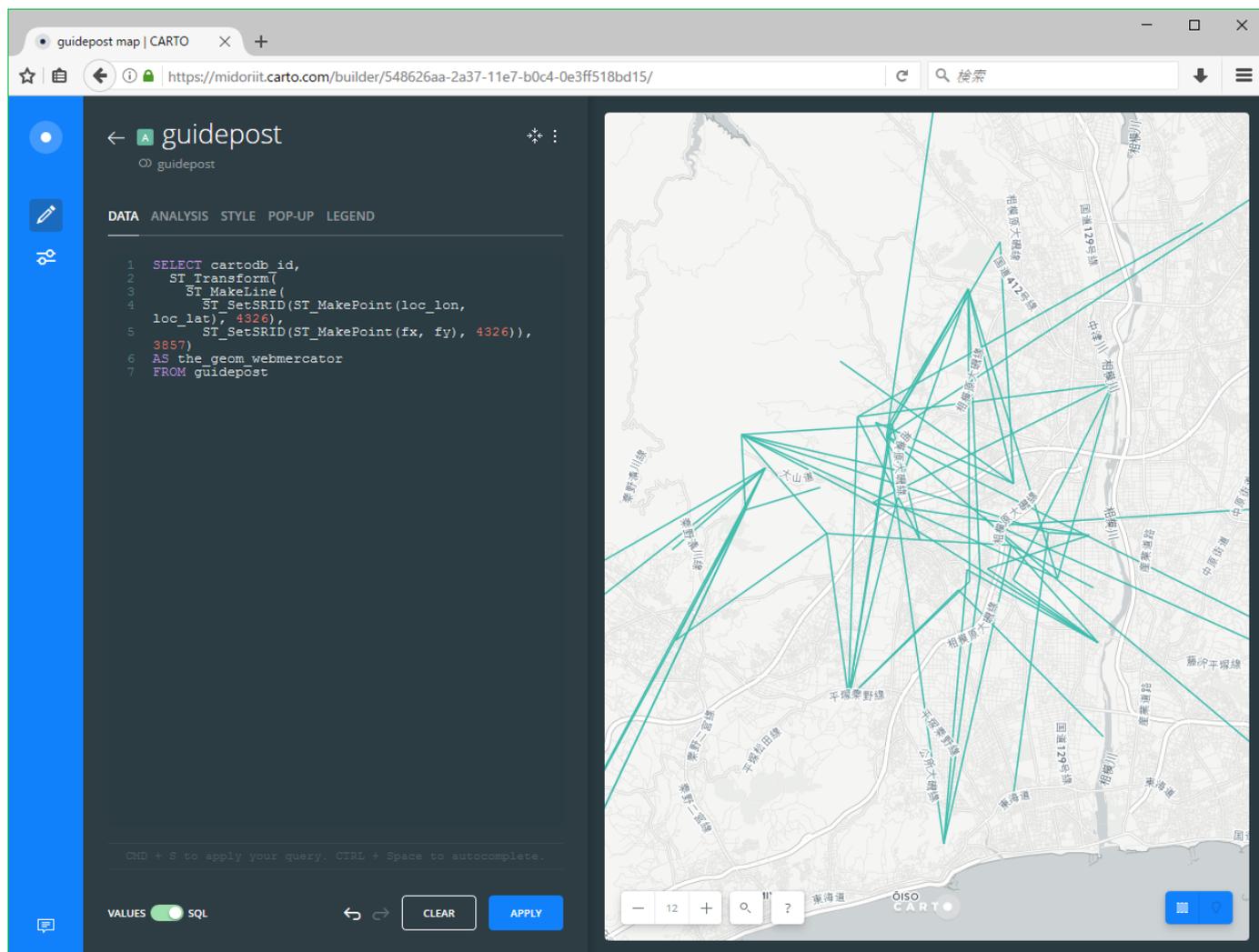
CARTOにアップロードしたデータセットには、各レコードに2組の緯度経度が数値型として入っていますが、geometry型のデータは持っていません。そこで、SQL文で線分を作成します。

まず、各々の緯度経度からST_MakePoint()で点を作成し、それらを引数としてST_MakeLine()で線分を作成します。元データの緯度経度は世界測地系（WGS84）を用いているため、ST_SetSRID()でEPSG:4326を設定しています。

CARTOはWebメルカトル（EPSG:3857）のデータをthe_geom_webmercatorという名前で受け取って地図に描画します。そこで、ST_Transform()で空間参照系を変換した結果にthe_geom_webmercatorという名前を付けて返します。

```
SELECT cartodb_id,
       ST_Transform(
         ST_MakeLine(
           ST_SetSRID(ST_MakePoint(loc_lon, loc_lat), 4326),
           ST_SetSRID(ST_MakePoint(fx, fy), 4326)), 3857)
AS the_geom_webmercator
FROM guidepost
```

実行結果は以下のようになります。



曲線でつなぐ

次に、この直線をカーブさせます。副問合せで線分を作成し、主問い合わせでST_CurveToLine()を用いて曲線を作成します。ST_CurveToLine()には、始点と中間点、終点の座標を渡します。始点と終点は副問合せで作成した線分の始点と終点をそのまま用います。中間点は、副問合せで作成した線分をST_OffsetCurve()で平行にずらして得た線分からst_Centroid()で取得した中心点を用います。

```
SELECT gp.cartodb_id,
       ST_CurveToLine(
         'CIRCULARSTRING(' ||
         st_x(st_startpoint(gp.the_geom)) || ' ' ||
         st_y(st_startpoint(gp.the_geom)) || ', ' ||
         st_x(
           st_Centroid(
             ST_OffsetCurve(
               gp.the_geom,
               st_length(gp.the_geom)/10,
               'quad_segs=4 join=bevel')) || ' ' ||
         st_y(
           st_Centroid(
             ST_OffsetCurve(
               gp.the_geom,
               st_length(gp.the_geom)/10,
               'quad_segs=4 join=bevel')) || ', ' ||
         st_x(st_endpoint(gp.the_geom)) || ' ' ||
         st_y(st_endpoint(gp.the_geom)) || ')') AS the_geom_webmercator
FROM (
  SELECT cartodb_id,
         ST_Transform(
           ST_MakeLine(
             ST_SetSRID(ST_MakePoint(loc_lon, loc_lat), 4326),
             ST_SetSRID(ST_MakePoint(fx, fy), 4326)), 3857)
         AS the_geom
  FROM guidepost) AS gp
```

実行結果は以下のようになります。

The screenshot shows the CARTO Builder interface. On the left, a SQL query is displayed in a dark-themed editor. The query uses `ST_CurveToLine` and `ST_OffsetCurve` to create circular arcs at the ends of road segments. The map on the right shows the result: a network of teal lines with circular arcs at their ends, overlaid on a street map. The interface includes a top navigation bar, a left sidebar with tool icons, and a bottom control bar with zoom and search tools.

曲線でつなぐ 2

上の図で横方向に伸びる曲線を見ると、上にカーブする場合と下にカーブする場合があることに気がきます。終点が始点の左（西）にある場合に下にカーブしています。そこで、RESASの〇〇花火図に倣って上方向のカーブに統一されるように修正します。

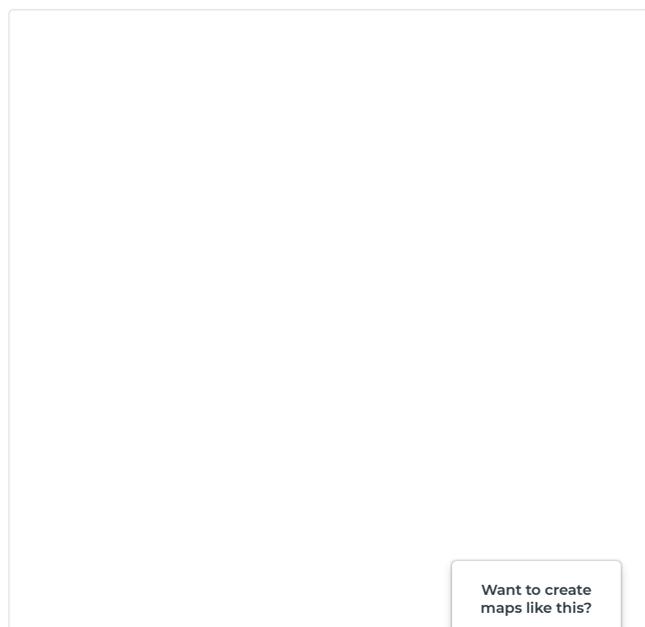
`ST_OffsetCurve()`の第2引数の符号で操作することもできますが、主問い合わせは既に複雑になっているため、今回は副問合せの中で、始点と終点の左右（東西）を統一することにします。そのために、副問合せの中でCASE式を用いて、始点の経度が終点の経度よりも大きければ（東があれば）始点と終点を入れ替えた線分を返すようにします。

```
SELECT gp.cartodb_id,
ST_CurveToLine(
  'CIRCULARSTRING(' ||
  st_x(st_startpoint(gp.the_geom)) || ' ' ||
  st_y(st_startpoint(gp.the_geom)) || ', ' ||
  st_x(
    st_Centroid(
      ST_OffsetCurve(
        gp.the_geom,
        st_length(gp.the_geom)/10,
        'quad_segs=4 join=bevel')) || ' ' ||
  st_y(
    st_Centroid(
      ST_OffsetCurve(
        gp.the_geom,
        st_length(gp.the_geom)/10,
        'quad_segs=4 join=bevel')) || ' ' ||
  st_x(st_endpoint(gp.the_geom)) || ' ' ||
  st_y(st_endpoint(gp.the_geom)) || ')') AS the_geom_webmercator
FROM (
  SELECT cartodb_id,
  CASE WHEN (loc_lon > fx) THEN
    ST_Transform(
      ST_MakeLine(
        ST_SetSRID(ST_MakePoint(fx, fy), 4326),
        ST_SetSRID(ST_MakePoint(loc_lon, loc_lat), 4326)), 3857)
  ELSE
    ST_Transform(
      ST_MakeLine(
        ST_SetSRID(ST_MakePoint(loc_lon, loc_lat), 4326),
        ST_SetSRID(ST_MakePoint(fx, fy), 4326)), 3857)
  END
  AS the_geom
FROM guidepost) AS gp
```

実行結果は以下のようになります。

The screenshot shows the CARTO Builder interface. On the left, a SQL query is applied to a layer named 'guidepost'. The query uses the `ST_CurveToLine` function to convert road geometries into lines, and the `ST_OffsetCurve` function to create circular markers at the start and end of each road segment. The markers are positioned at the centroid of the road segment, offset by 10 units. The map on the right displays the result: a network of roads with green circular markers at various points. The interface includes a toolbar on the left with icons for layers, data, analysis, style, pop-up, and legend. The bottom of the interface shows a status bar indicating 'SQL query applied'.

あとは、道標の位置をアイコンで示したり、標示地名を表示するなどの細かな追加をすれば完成です。



道標の下には造立年を表示し、曲線の色も造立年によって変化させています。

カテゴリー: ビジュアライゼーション, 地理空間情報 | タグ: CARTO | 投稿日: 2017年4月26日

[<https://midoriit.com/2017/04/cartoe3%81%ab%e3%82%88%e3%82%8b%e8%8a%b1%e7%81%ab%e5%9b%b3%e3%81%ae%e4%bd%9c%e6%88%90.html>] | 投稿者: 小池隆
