Using PostGIS and H3 to analyze UFO Risk

Martin Davis

PostGIS Day 2024

November 2024

crunchy data

UFOs and Wind Turbines - a growing risk?

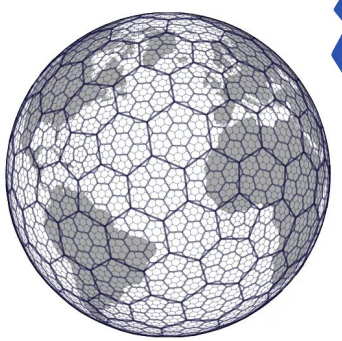
PostGIS + H3

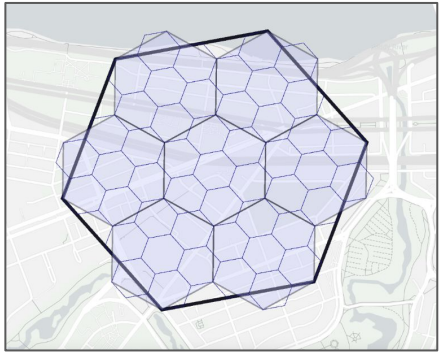
Outline

- What is H3, and why use it (or not)?
- H3 API and h3-pg
- Building a UFO Risk Dashboard with pg-tileserv and MapLibre
- Visualize UFO sightings with:
  - Heatmap
  - H3 Clustering
  - H3 Hexagonal tiling
- Analyzing UFO risk to Wind Turbines with H3

What is H3?

> *"Multiresolution hexagonal global grid system with hierarchical indexing"*



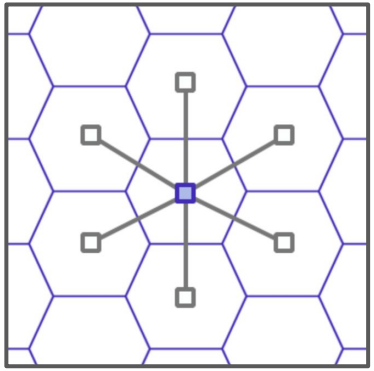


- Developed by Uber to enable global-scale analysis
- Grid of **hexagonal** *(mostly)* **cells** covering the globe
  - Also 12 pentagons
- 16 **grid resolutions**
  - Cells at a resolution have (*roughly*) same area
    - R0 = 4.3M km^2 R15 = 1 m^2
  - Cells refine (*roughly*) into 7 cells at next higher resolution
- Each cell has a unique **64-bit index**
  - E.g. `8426b47fffffffff`
- Extensive API to work with cells, ids, and geometries

Why use gridded data?

- **Analysis**
  - Common basis for analyzing data
    - Multivariate analysis
    - All spatial types: points, lines, areas, rasters
  - Inherently provides feature density
  - Faster to compute spatial joins, overlays
- **Display**
  - Reduce data size
    - Faster transfer and display
  - Uniform styling, visual appearance

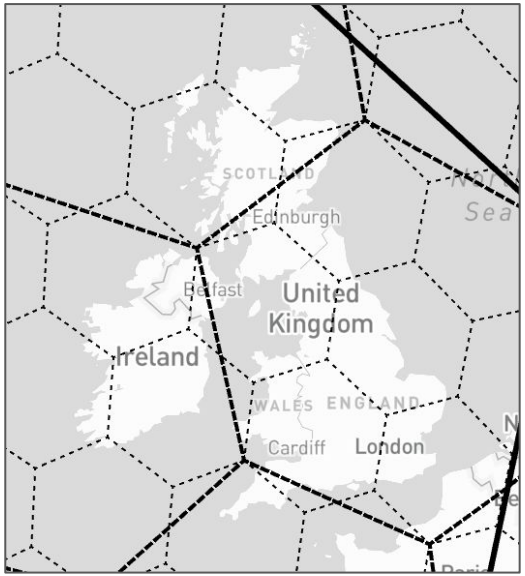
Why use H3?



- **Standard grid** for global datasets (sub-global too)
  - Easy to combine different datasets
- **Equal-area cells** good for data analysis / visualization
  - Avoids issues in high latitudes (*unlike square grids*)
- Cell neighbours are **equidistant** (*unlike square grids*)
  - Fewer edge/corner effects for representing non-rectilinear data
  - Better for representing paths
- **Independent** of human and natural area/boundary variation
- Cell ids provide **fast indexing, data joining**
- **Powerful API** allows easy use in a variety of ways
- **Wide support**

Why NOT use H3?

- Does not follow human or natural boundaries
- Geodetic only
  - not planar coordinate systems
- Data may require transformation, clipping or regridding into H3
- Cells at different resolutions do not nest exactly
- Hexagonal cells (somewhat) more difficult to store, process, display
- Need access to H3 API



H3 API and h3-pg

- **H3 API**
  - Algebra for working with H3 cells and geometries
  - Converts between cells, resolutions, points, and geometries
  - Many bindings: C, Python, Java, R, Go, Javascript etc
  - **https://h3geo.org/**

- **h3-pg**
  - PostgreSQL extension for H3 API
  - Integrates with PostGIS
  - 80+ functions
  - **https://github.com/zachasme/h3-pg**

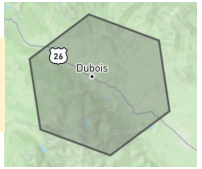
Some h3-pg functions

- **`h3_lat_lng_to_cell ( geometry, resolution integer ) ⇒ h3index`**
  - Determine cell for a geodetic point

```
POINT(-104, 47), 6 => 8426b47fffffffff
```

- **`h3_cell_to_boundary_geometry ( h3index ) ⇒ geometry`**
  - Generate hexagon geometry for cell

```
8426b47fffffffff =>
```



- **`h3_cell_area ( cell h3index ) ⇒ double precision`**
  - Get area of cell (in km^2)

```
8426b47fffffffff => 1803.8100025727072
```

- many others…

Data

- **UFO Sightings**
  - Corgis UFO Sightings CSV File
  - 60,632 sightings in US
  - `https://corgis-edu.github.io/corgis/csv/ufo_sightings/`

- **Wind Turbines**
  - Corgis Wind Turbines CSV File
  - 63,961 locations in US
  - `https://corgis-edu.github.io/corgis/csv/wind_turbines/`

UFO Sightings - pg_tileserv

pg_tileserv

Service Metadata

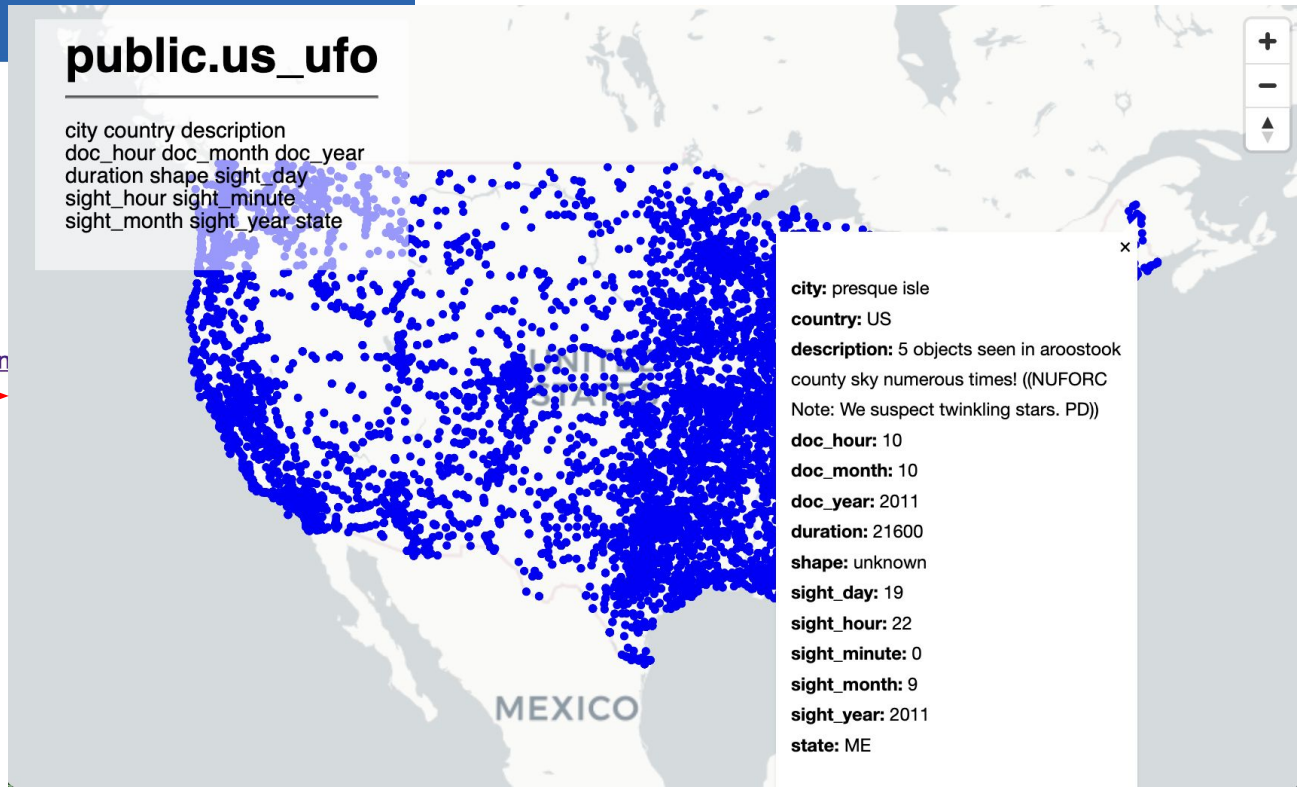
- index.json for layer list

Table Layers

- **public.geonames** (preview | json)
  US GNIS points
- **public.us_ufo** (preview | json)

Function Layers

- **public.geonames_h3** (preview | json)
- **public.h3hexes** (preview | json)
  H3 hexagonal grids.
- **public.us_ufo_density_r4** (preview | json)
- **public.us_ufo_h3** (preview | json)



crunchy data

Wind Turbines - pg_tileserv

pg_tileserv

Service Metadata

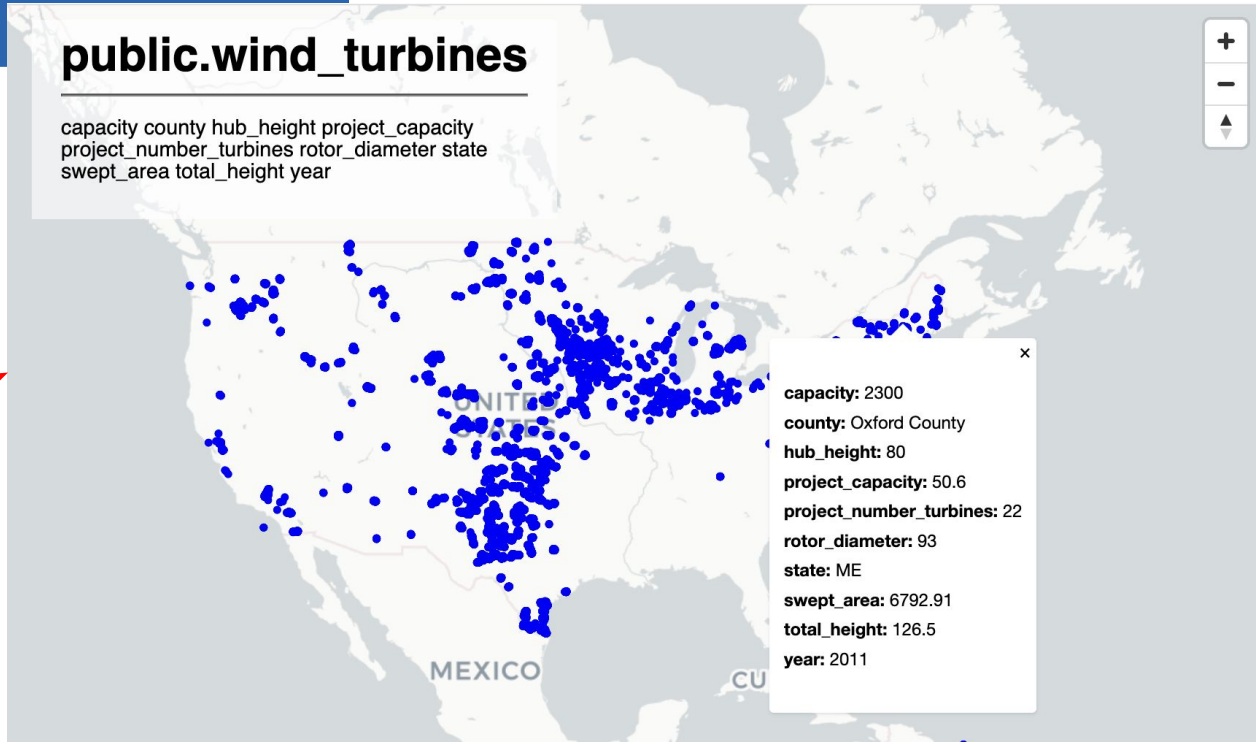
- index.json for layer list

Table Layers

- **public.geonames** (preview | json)
  US GNIS points
- **public.kontur_population_r4** (preview | json)
- **public.us_ufo** (preview | json)
- **public.wind_turbines** (preview | json)

Function Layers

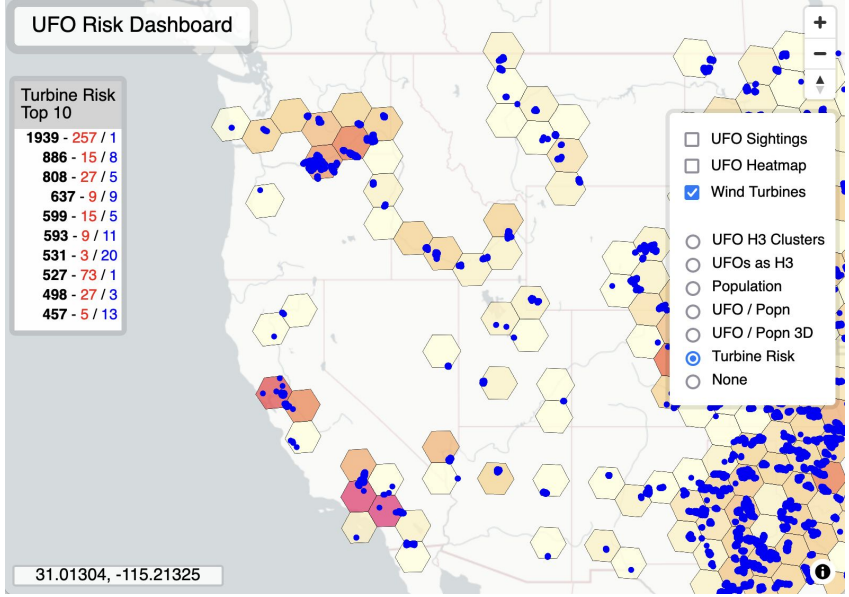
- **public.geonames_h3** (preview | json)
- **public.h3hexes** (preview | json)
  H3 hexagonal grids.
- **public.us_ufo_density_r4** (preview | json)
- **public.us_ufo_h3** (preview | json)

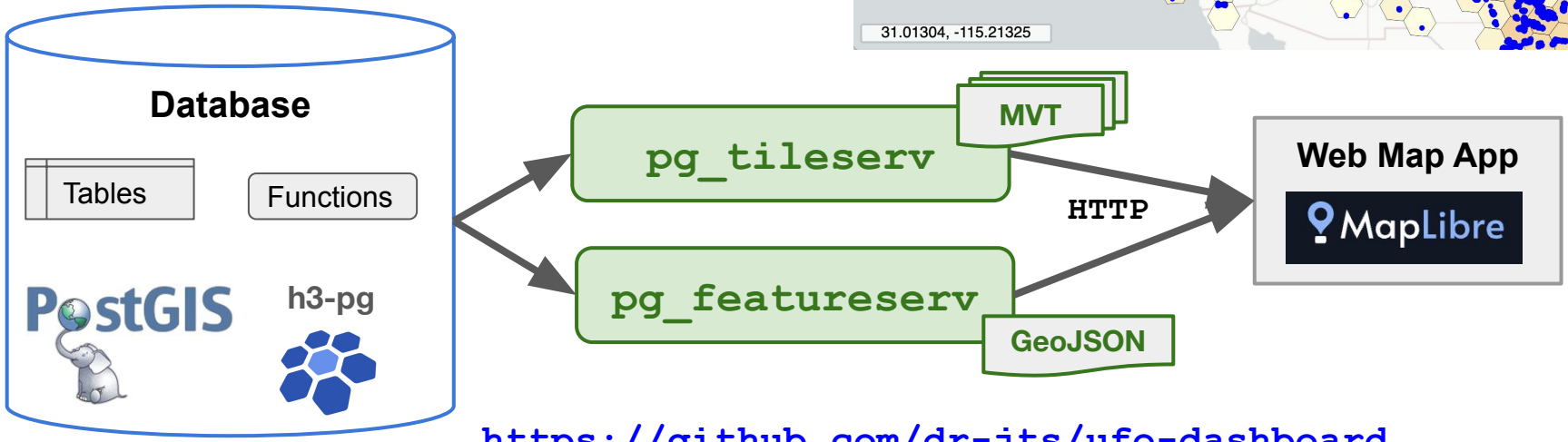


crunchy data

UFO Risk Dashboard

- **Data tier:** PostGIS, h3-pg
- **App tier:** pg_tileserv, pg_featureserv
- **Client:** Web app with HTML, JavaScript, MapLibre



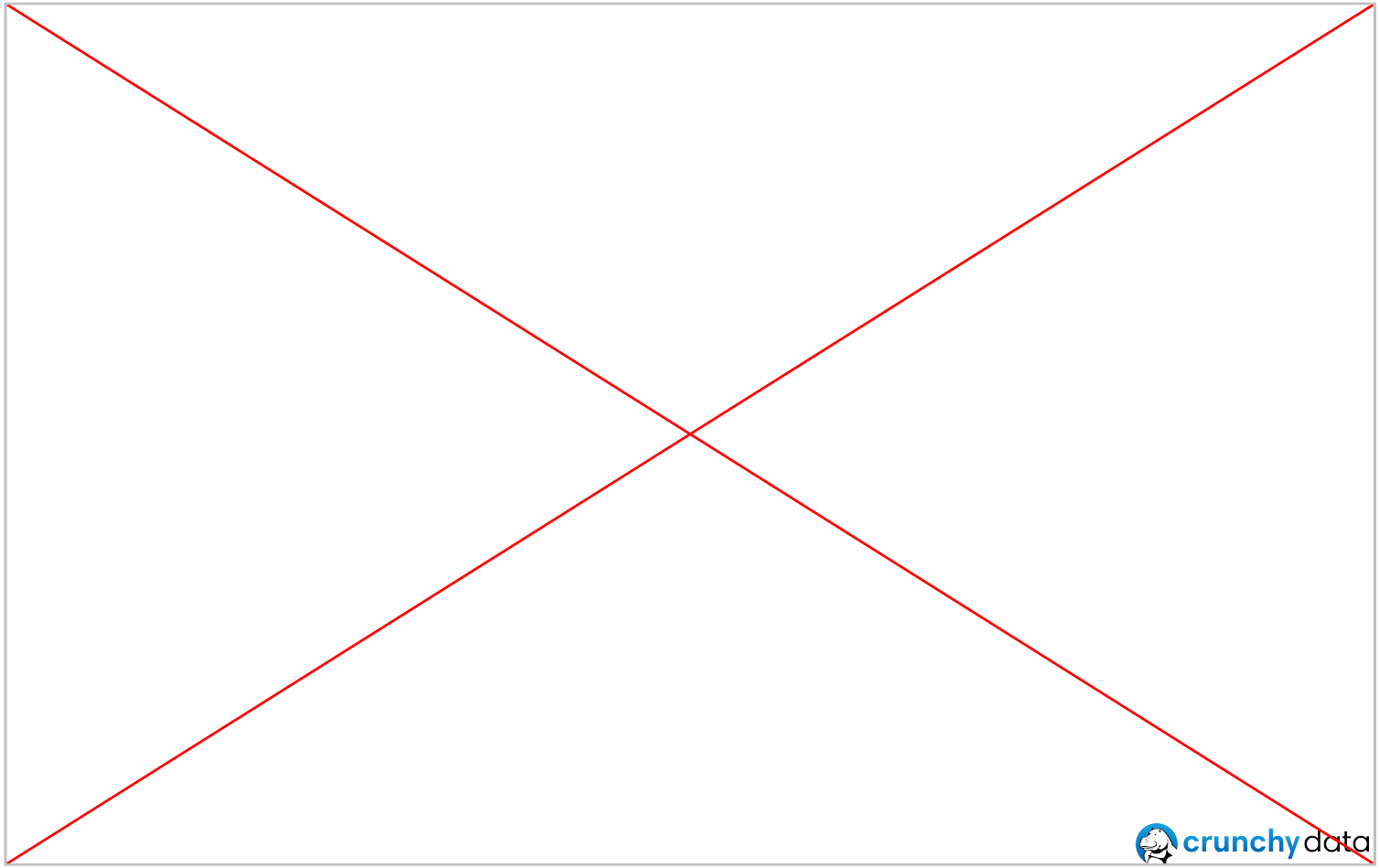


https://github.com/dr-jts/ufo-dashboard

crunchy data

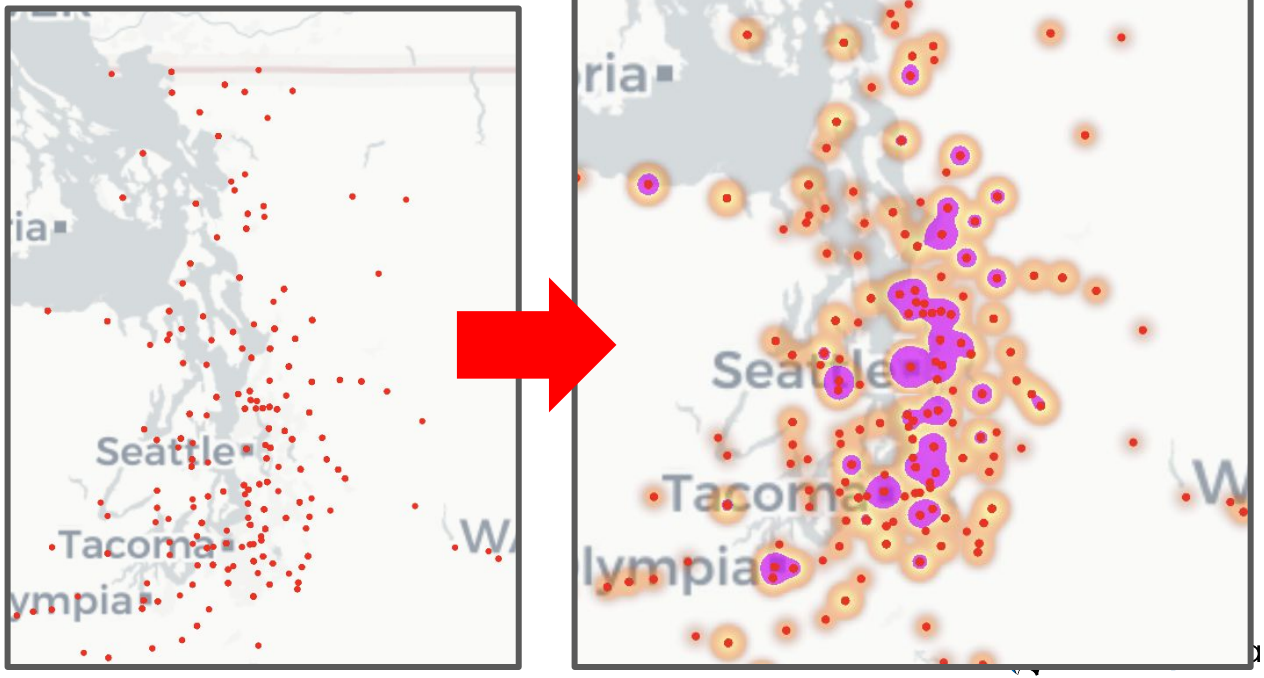
UFO and Wind Turbine points in Dashboard

- MVT layers from pg_tileserv



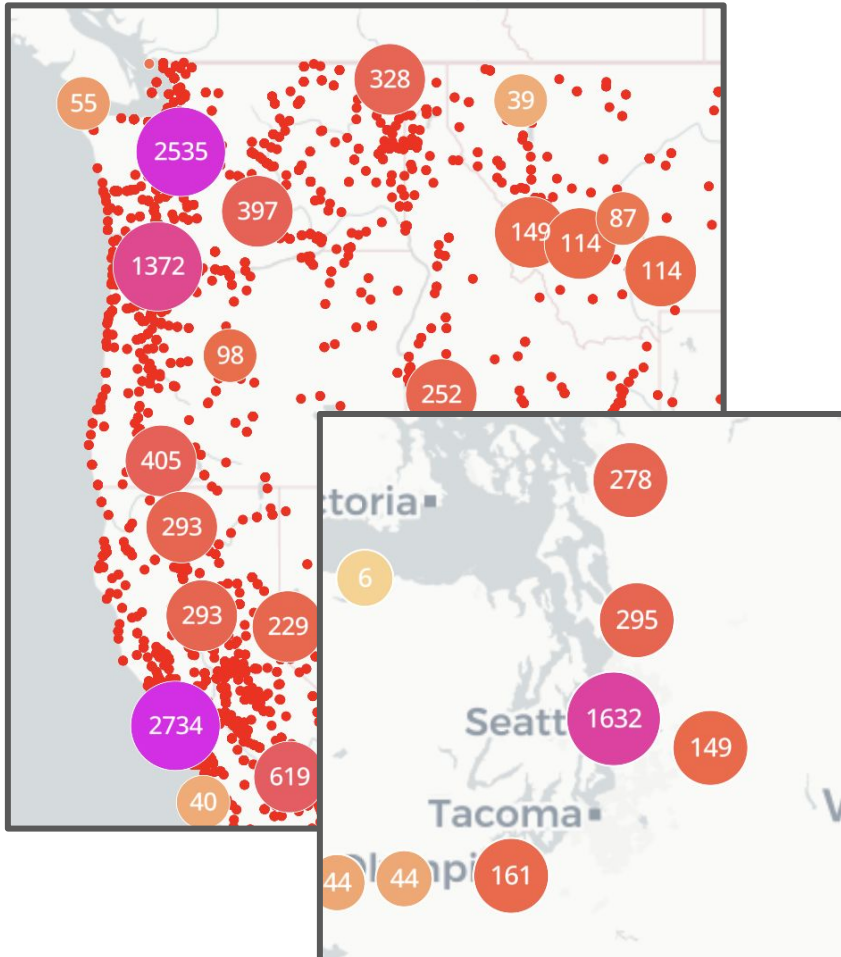
Visualize UFO Sightings - Client Heatmap

- MapLibre Heatmap
- Disadvantages:
  - Client-side - all data transferred
  - No statistics
  - Single variable



Visualize UFO Sightings - H3 Clustering

- Group data points by H3 cell id
  - H3 resolution based on zoom level
- For each cell compute:
  - count of points in cell
  - A representative point (geometry and data)
- Advantages:
  - Server-side
  - Fast cluster computation
  - Low # of points per MVT tile
  - Cluster distribution looks “natural”



H3 Clustering function for pg_tileserv

```
CREATE OR REPLACE
FUNCTION public.us_ufo_h3_cluster(z integer, x integer, y integer)
RETURNS bytea
AS $$
WITH
-- Compute the tile envelope to avoid repeated calculations
tile_env AS (
    SELECT ST_Transform(
               ST_TileEnvelopeClip(z, x, y, margin => 0.125), 4326)
           AS env_geom
),
-- Compute the H3 resolution based on zoom level
resolution AS (
    SELECT CASE
        WHEN z <= 2 THEN 2
        WHEN z <= 4 THEN 3
        WHEN z <= 6 THEN 4
        WHEN z <= 8 THEN 5
                    ELSE 6
    END AS h3_res
),
```

MVT tile function

Data query extent = tile env with margin, in 4326 (see https://github.com/dr-jts/ufo-dashboard/blob/main/ST_TileEnvelopeClip.sql

H3 grid resolution based on tile zoom level

H3 Clustering function for pg_tileserv (2)

```
cell AS (
  SELECT count(*) AS ufo_count,
         ST_Transform( ANY_VALUE(geom), 3857) AS geom,
         ANY_VALUE(state) AS state,
         h3_lat_lng_to_cell(geom,
                    (SELECT h3_res FROM resolution)) AS cellid
  FROM us_ufo
  WHERE ST_Intersects(geom,
                ST_Transform((SELECT env_geom FROM tile_env), 4326))
  GROUP BY cellid
),
```

Create H3 cell cluster record with:
 # data points per cell
representative location
attribute value(s)

H3 cell ids for data points
H3 resolution for zoom

Only query points in tile

One result record per cell

H3 Clustering function for pg_tileserv (3)

```
-- Tile bounds in Web Mercator (3857)
bounds AS ( SELECT ST_TileEnvelope(z, x, y) AS geom ),
mvtgeom AS (
    -- Generate MVT-compatible geometry (quantize and clip to tile)
    SELECT ST_AsMVTGeom(cell.geom, bounds.geom) AS geom,
           cellid, ufo_count, state
    FROM cell, bounds
)
-- Generate MVT encoding of MVT features
SELECT ST_AsMVT(mvtgeom, 'default') FROM mvtgeom
$$
LANGUAGE 'sql' STABLE STRICT PARALLEL SAFE;
```

Create MVT tile feature

Create MVT tile

https://github.com/dr-jts/ufo-dashboard/blob/main/ufo_cluster_h3_fn.sql

UFO Sighting H3 Clusters - pg_tileserv

pg_tileserv

Service Metadata

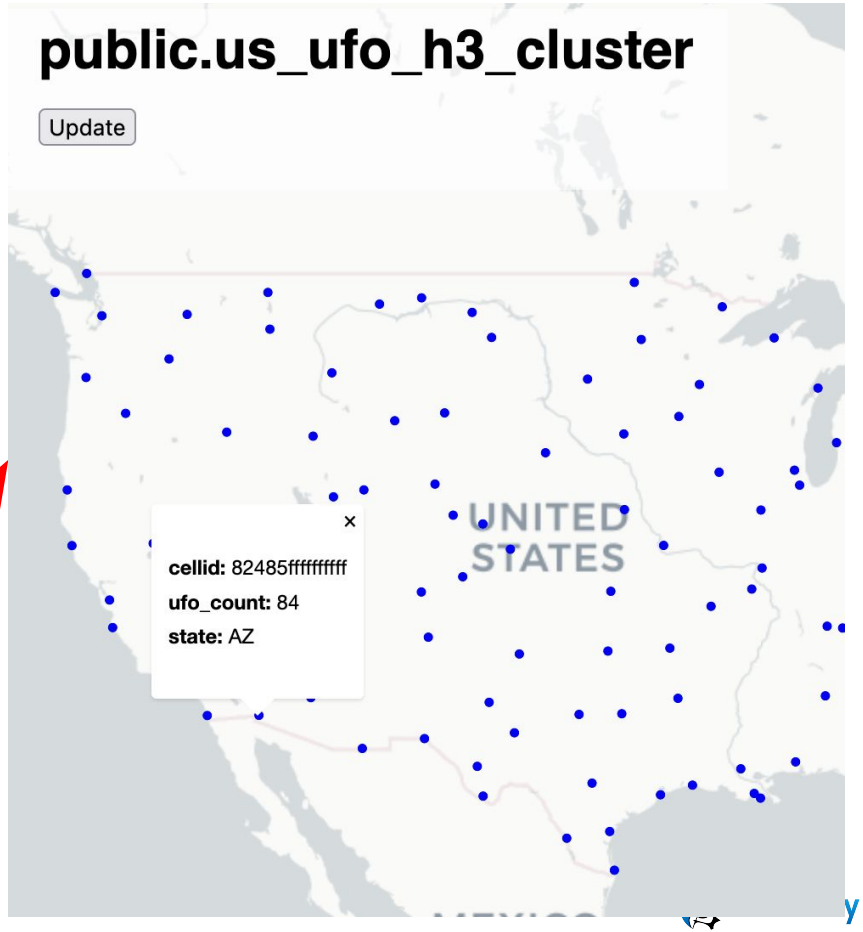
- index.json for layer list

Table Layers

- **public.geonames** (preview | json)
  US GNIS points
- **public.geonames_cluster_min** (preview | json)
- **public.kontur_population_r4** (preview | json)
- **public.ufo_turbine_risk_h3_r6** (preview | json)
- **public.us_ufo** (preview | json)
- **public.wind_turbines** (preview | json)

Function Layers

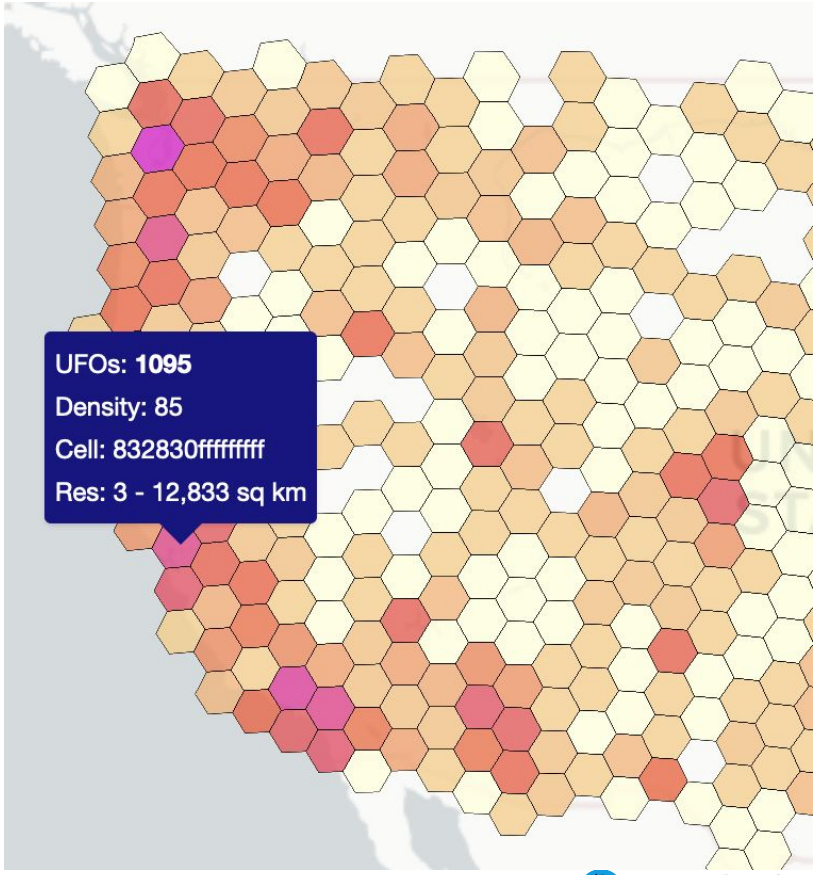
- **public.us_ufo_density_r4** (preview | json)
- **public.us_ufo_h3** (preview | json)
- **public.us_ufo_h3_cluster** (preview | json)
  US UFO sightings clustered by H3 cells.



UFO Sighting H3 Clustering in Dashboard

Visualize UFO Sightings - H3 Tiling

- Aggregate data points by H3 cell
- Display H3 hexagons
  - H3 resolution based on zoom level
- For each cell compute:
  - count of points in cell



UFO H3 Tiling - Function for pg_tileserv

```
cell AS (
  SELECT count(*) AS ufo_count,

        h3_lat_lng_to_cell(geom,
                (SELECT h3_res FROM resolution) ) AS cellid
    FROM us_ufo
    WHERE ST_Intersects(geom, (SELECT env_geom FROM tile_env))
   GROUP BY cellid
),

feature AS (
   SELECT cellid, ufo_count,
     round(1000 * ufo_count / h3_cell_area( cellid)) AS density,
     h3_cell_area( cellid) AS area,
     ST_Transform( h3_cell_to_boundary_geometry( cellid ), 3857) AS geom
   FROM cell
),
```

Create H3 cell record with # data points per cell

H3 cell ids for data points
H3 resolution for zoom

Only query points in tile

One result record per cell

Create H3 hexagon feature

Hex area
Hex geometry

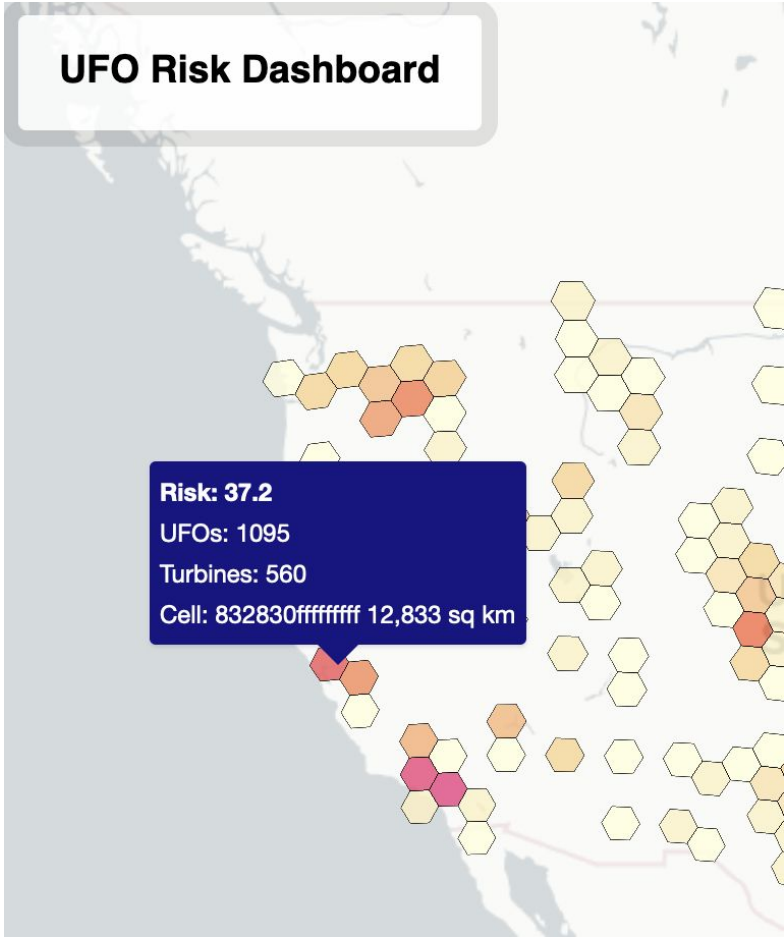
https://github.com/dr-jts/ufo-dashboard/blob/main/ufo_h3_fn.sql

crunchy data

UFO Sighting H3 Tiling in Dashboard

UFO/Turbine Risk Analysis with H3 Hexagons

- **Bivariate analysis**
- **UFO / Wind Turbine Risk**
  - Risk $\propto$ UFO Density
  - Risk $\propto$ Turbine Density
- Risk = `count(UFO) x count(Turbines) / area^2`



UFO Risk in H3 - Function for pg_tileserv

Count UFO sightings by H3 cells

```
ufo_cell AS (
  SELECT count(*) AS cnt,
        h3_lat_lng_to_cell(geom, r.h3_res ) AS cellid,
        ANY_VALUE(r.h3_res) As h3_res
     FROM us_ufo
    CROSS JOIN resolution r
     WHERE ST_Intersects(geom, (SELECT env_geom FROM tile_env))
    GROUP BY cellid
),
```

Count Turbine locations by H3 cells

```
turbine_cell AS (
  SELECT count(*) AS cnt,
        h3_lat_lng_to_cell(geom, r.h3_res ) AS cellid
     FROM wind_turbines
    CROSS JOIN resolution r
     WHERE ST_Intersects(geom, (SELECT env_geom FROM tile_env))
    GROUP BY cellid
),
```

UFO Risk in H3 - Function for pg_tileserv (2)

```
cell AS (
    SELECT
        u.cnt AS ufo_count,
        t.cnt AS turbine_count,
        h3_cell_area( u.cellid) AS area,
        u.cellid,
        u.h3_res
    FROM ufo_cell u
         INNER JOIN turbine_cell t ON u.cellid = t.cellid
),
feature AS (
    SELECT ufo_count, turbine_count,
        10000 * (ufo_count * turbine_count) / (area * area) AS risk,
        cellid,
        h3_res, area,
        ST_Transform( h3_cell_to_boundary_geometry( cellid ), 3857) AS geom
    FROM cell
),
```

Create H3 cell record with

 # data points per cell

H3 cell area for density
H3 cell id
H3 resolution for zoom

Merge H3 cells

Create H3 hexagon feature

Risk formula
Hex id,
resolution, area
Hex geometry

https://github.com/dr-jts/ufo-dashboard/blob/main/ufo_turbine_risk_h3_fn.sql

UFO Risk in H3 - SQL View

```
CREATE VIEW ufo_turbine_risk_h3_r6 AS
WITH resolution AS (
   SELECT 6 AS h3_res
),
ufo_cell AS (...),
turbine_cell AS (...),

cell AS ( SELECT
       u.cnt AS ufo_count,
       t.cnt AS turbine_count,
       h3_cell_area( u.cellid) AS area,
       u.cellid,
       u.h3_res
   FROM ufo_cell u INNER JOIN turbine_cell t ON u.cellid = t.cellid
)
SELECT ufo_count, turbine_count,
       10000 * (ufo_count * turbine_count) / (area * area) AS risk,
       cellid, h3_res, area,
       h3_cell_to_geometry( cellid )::geometry(Point, 4326) AS geom
FROM cell;
```

Fixed resolution = 6

Merge H3 cells

Create H3 hexagon as point
Risk formula
Hex stats,
Hex centroid

https://github.com/dr-jts/ufo-dashboard/blob/main/ufo_turbine_risk_h3_r6_vw.sql

UFO Risk in H3 - SQL View in pg_featureserv

Demo - Bridge *pg_featureserv - Version 1.3.1*

Home / Collections JSON

Feature Collections

public.geonames JSON View
US GNIS points

public.geonames_cluster_min JSON View
Data for table public.geonames_cluster_min

public.kontur_population_r4 JSON View
Data for table public.kontur_population_r4

public.ufo_turbine_risk_h3_r6 JSON View
Data for table public.ufo_turbine_risk_h3_r6

public.us_ufo JSON View
Data for table public.us_ufo

public.wind_turbines JSON View
Data for table public.wind_turbines

```
http://localhost:9000
/collections/public.ufo_risk_h3_r6
/items.json
?sortBy=-risk
&limit=10
```

UFO Risk Analysis with H3 in Dashboard

crunchy data

Wrap-up

https://github.com/dr-jts/ufo-dashboard

Questions?

Comments?

Ideas?

martin.davis@crunchydata.com
postgis-users@lists.osgeo.org