image: Florian Thiery CC BY 4.0, created using OpenAI's DALL-E

Research Squirrel Engineers

How an independent RSE-driven network may help the NFDI

Florian Thiery M.Sc.

deRSE25 | 2025 | 25-27 Feb 2025 | Karlsruhe | Germany | 25/02/2025

Session: Nation-wide networks of RSEs

DOI 10.5281/zenodo.14914287

image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

This is **Squilly**, aiming for

Open Data with **Open Source** - **FAIR Data** with **FAIR4RS Tools**

image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

The collaborative creation and FAIRification of research data is becoming increasingly important in the Citizen Science community to become part of an interdisciplinary Knowledge Graph.

Only in this way can this data be linked to other data and actively integrated into international initiatives (e.g. NFDI) and community hubs (e.g. Wikidata, FactGrid or OpenStreetMap).

Unfortunately, open-source (**FLOSS**) research and FAIRification tools are **often unavailable**.

image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

However, these, in combination with Linked Open Data projects as demonstrators, can be created and curated by community and voluntary initiatives such as the **Research Squirrel Engineers Network**.

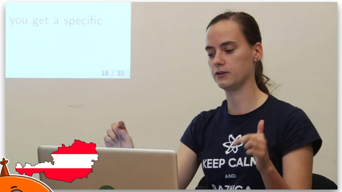
The Research Squirrel Engineers Network initiative may be helpful for the NFDI with FAIRification tools and LOD Knowledge Graphs.

image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

Goal: Open Data with Open Source: FAIR Data with FAIR4RS Tools

The Research Squirrel Engineers Network is a loose association of Linked Open Data/Wikidata enthusiasts, research software engineers and citizen scientists specialising in computational archaeology and geoinformatics.

Sophie
@SCSchmidt
0000-0003-4696-2101



Martina
@bellerophons-pegasus
0000-0003-0485-6861

Florian
@florianthiery
0000-0002-3246-3531

Timo
@situx
0000-0002-9499-5840

WIKIDATA

RDF

RESEARCH SQUIRREL ENGINEERS Network

+ B. Danthine, A.-K. Distel, Peter Thiery, Fiona Schenk et al.

The members develop and maintain research and FAIRification and FAIRification tools and implement them in specific projects.

Squilly can use …

image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

... the **SPARQL Unicorn Research Toolkit**.

SPARQL Unicorn Research Toolkit

by the Research Squirrel Engineers Network

QGIS

GitHub

A FAIRification tool for digital data management is the SPARQL Unicorn and its implementation for QGIS

It contains of

(i) the **SPARQLing Unicorn QGIS Plugin**

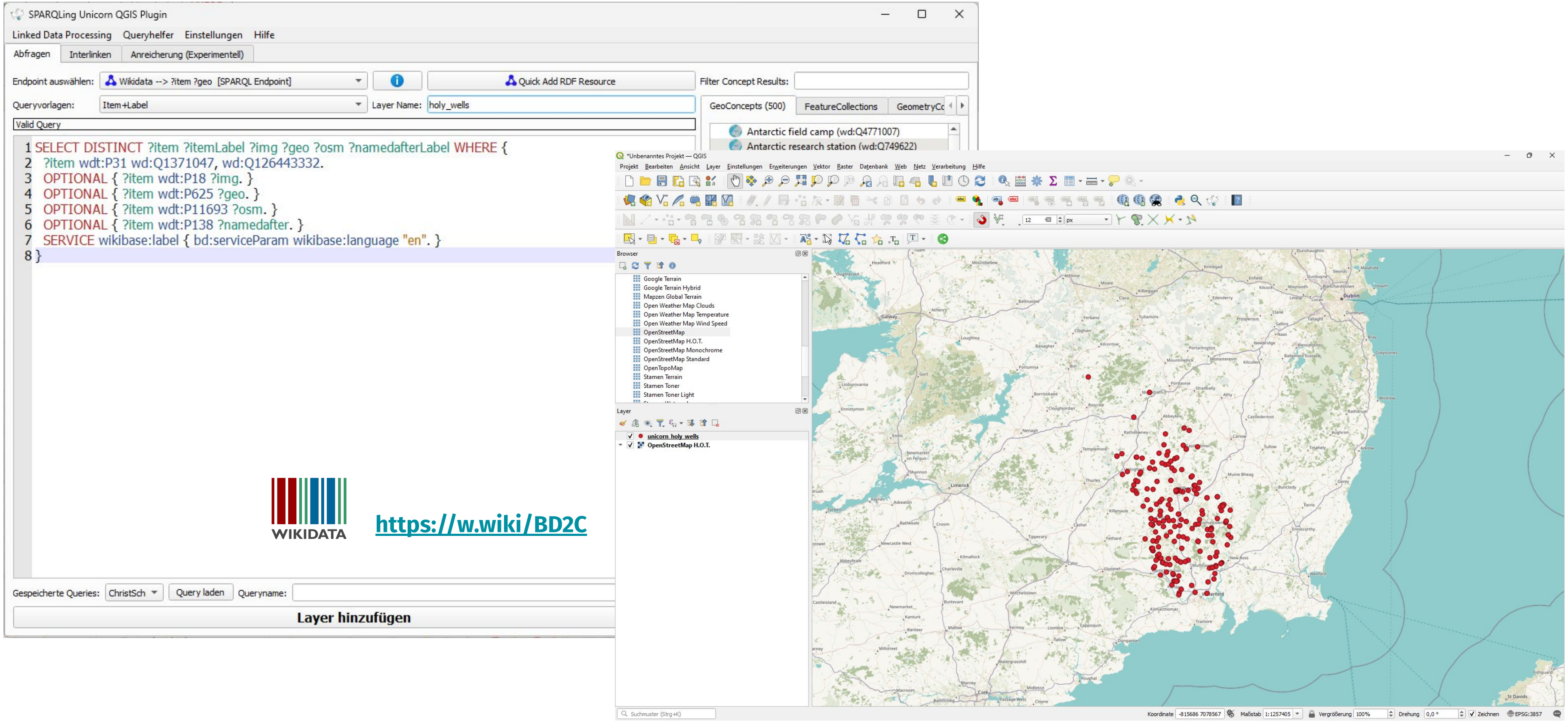
(ii) the **SPARQL Unicorn Ontology Documentation** tool

The plugin offers three functions:

(A) Simplified querying of Semantic Web data sources
(B) Transformation of QGIS vector layers to RDF
(C) RDF HTML documentation

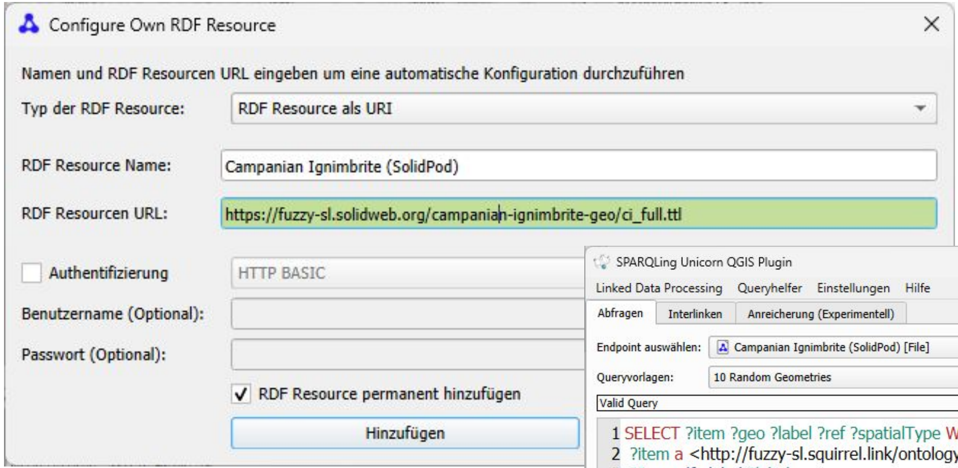
The SPARQLing Unicorn's Functions

Simplified querying of Semantic Web data sources

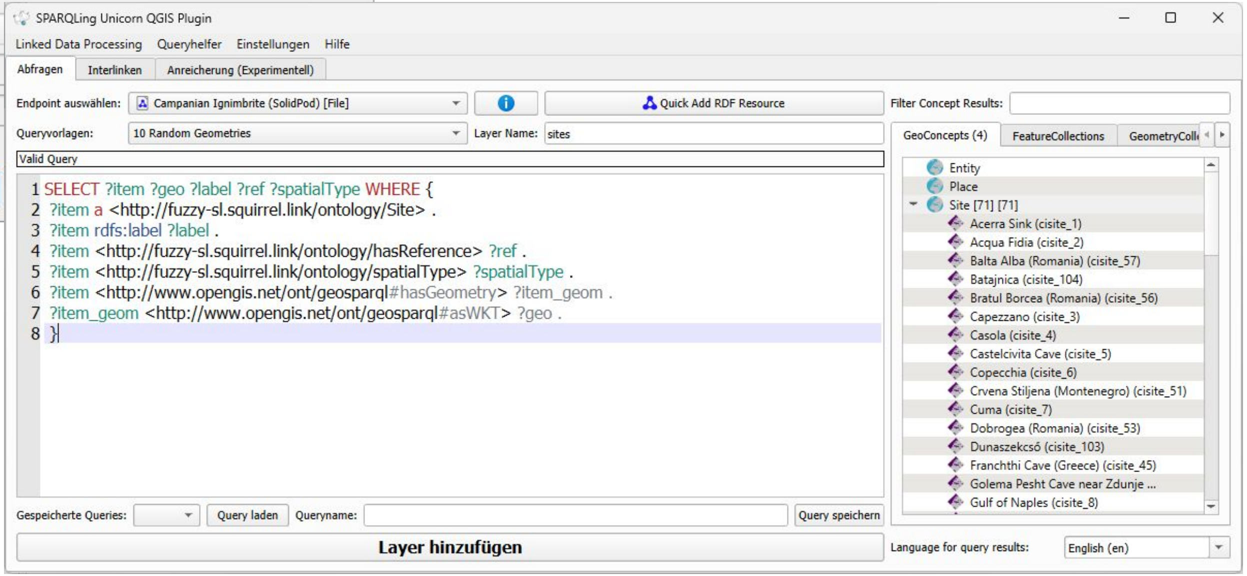


https://w.wiki/BD2C

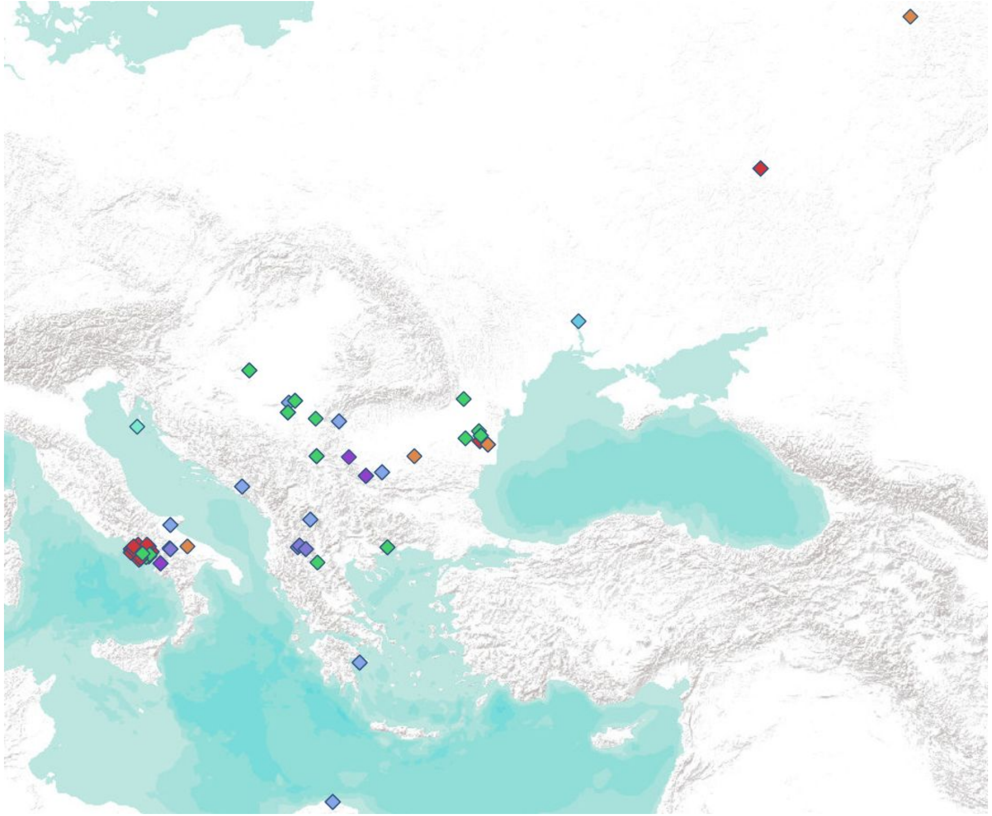
Wikidata Queries in QGIS - Holy Wells



https://fuzzy-sl.solidweb.org/campanian-ignimbrite-geo/ci_full.ttl



Campanian Ignimbrite Findspots via Solid Pod



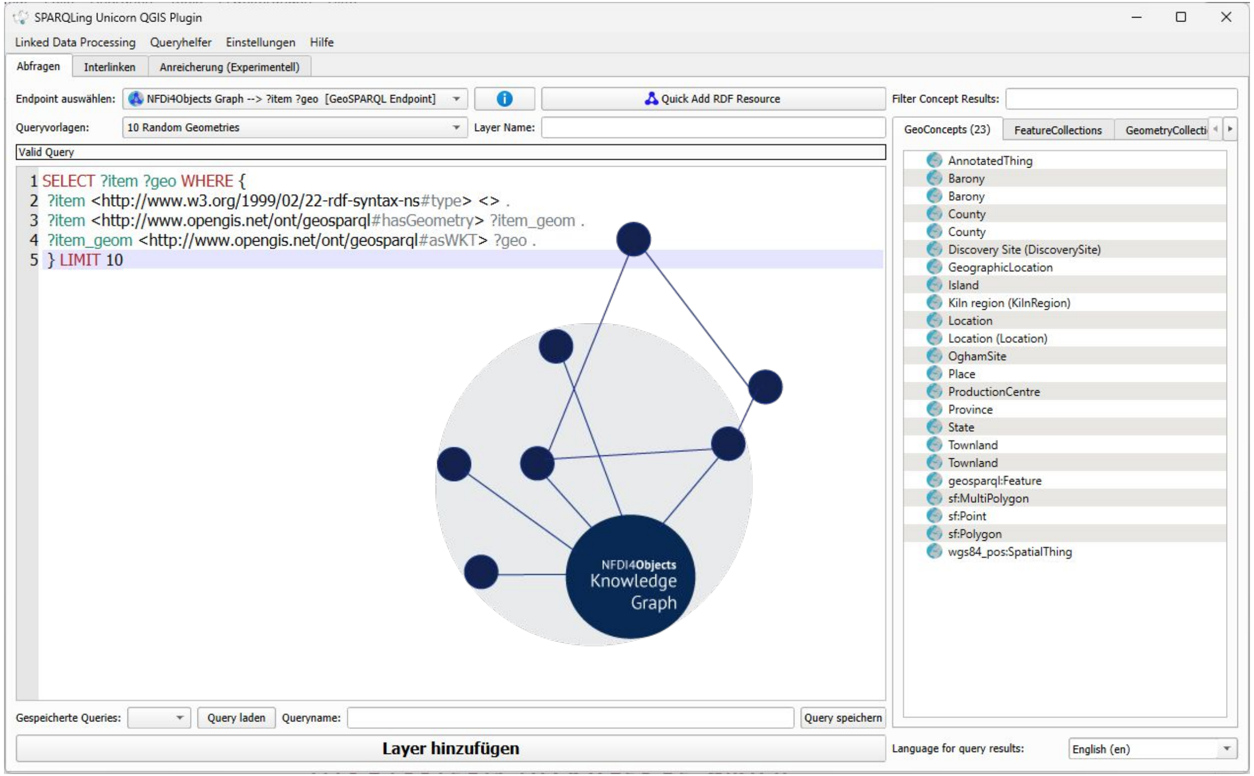
https://fuzzy-sl.solidweb.org/campanian-ignimbrite-geo/ci_full.ttl

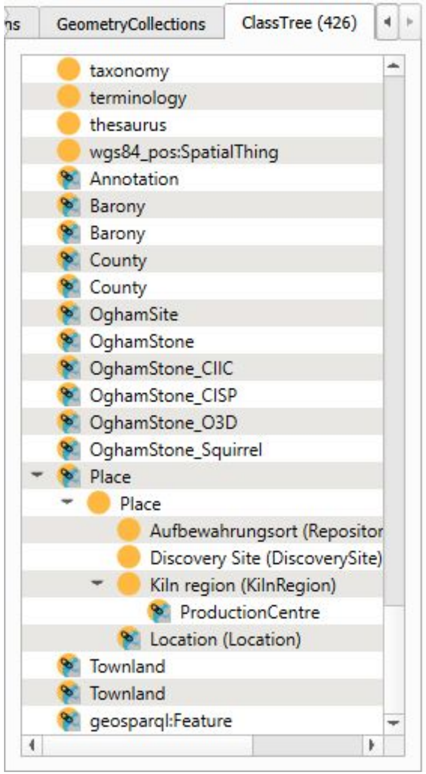
unicorn_sites — Objekte gesamt:104, gefiltert: 104, gewählt: 0

	id	label	ref	spatialType
1	http://fuzzy-sl.squirrel.link/data/cisite_1	Acerra Sink	Scandone et al., 1991	http://fuzzy-sl.squirrel.link/ontology/Sink
2	http://fuzzy-sl.squirrel.link/data/cisite_2	Acqua Fidia	Rosi et al., 1999	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
3	http://fuzzy-sl.squirrel.link/data/cisite_57	Balta Alba (Romania)	Pötter et al., 2021	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
4	http://fuzzy-sl.squirrel.link/data/cisite_104	Batajnica	Obreht, I. et al. (2017), Fig. 1	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
5	http://fuzzy-sl.squirrel.link/data/cisite_104	Batajnica	Buggle, B. et al. (2013)	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
6	http://fuzzy-sl.squirrel.link/data/cisite_104	Batajnica	Buggle, B. et al. (2014)	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
7	http://fuzzy-sl.squirrel.link/data/cisite_56	Bratul Borcea (Romania)	Pötter et al., 2021	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
8	http://fuzzy-sl.squirrel.link/data/cisite_3	Capezzano	Rosi et al., 1999	http://fuzzy-sl.squirrel.link/ontology/InhabitedPlace
9	http://fuzzy-sl.squirrel.link/data/cisite_4	Casola	Rosi et al., 1999	http://fuzzy-sl.squirrel.link/ontology/InhabitedPlace
10	http://fuzzy-sl.squirrel.link/data/cisite_5	Castelcivita Cave	Fedele et al., 2008	http://fuzzy-sl.squirrel.link/ontology/Cave
11	http://fuzzy-sl.squirrel.link/data/cisite_5	Castelcivita Cave	Giaccio et al., 2008	http://fuzzy-sl.squirrel.link/ontology/Cave
12	http://fuzzy-sl.squirrel.link/data/cisite_6	Copecchia	Rosi et al., 1999	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
13	http://fuzzy-sl.squirrel.link/data/cisite_51	Crvena Stiljena (Montenegro)	Morley & Woodward, 2011	http://fuzzy-sl.squirrel.link/ontology/Cave
14	http://fuzzy-sl.squirrel.link/data/cisite_51	Crvena Stiljena (Montenegro)	Morley & Woodward, 2011	http://fuzzy-sl.squirrel.link/ontology/ArchaeologicalSite
15	http://fuzzy-sl.squirrel.link/data/cisite_7	Cuma	Pappalardo et al., 1999	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
16	http://fuzzy-sl.squirrel.link/data/cisite_53	Dobrogea (Romania)	Fitzsimmons et al., 2014	http://fuzzy-sl.squirrel.link/ontology/Plateau
17	http://fuzzy-sl.squirrel.link/data/cisite_53	Dobrogea (Romania)	Fitzsimmons et al.,2013	http://fuzzy-sl.squirrel.link/ontology/Plateau
18	http://fuzzy-sl.squirrel.link/data/cisite_103	Dunaszekcső	Obreht, I. et al. (2017), Fig. 1	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
19	http://fuzzy-sl.squirrel.link/data/cisite_103	Dunaszekcső	Újvári, G. Et al. (2016)	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
20	http://fuzzy-sl.squirrel.link/data/cisite_45	Franchthi Cave (Greece)	Fedele et al., 2003	http://fuzzy-sl.squirrel.link/ontology/Cave
21	http://fuzzy-sl.squirrel.link/data/cisite_45	Franchthi Cave (Greece)	Fedele et al., 2003	http://fuzzy-sl.squirrel.link/ontology/ArchaeologicalSite
22	http://fuzzy-sl.squirrel.link/data/cisite_50	Golema Pesht Cave near Zdu...	Lowe et al., 2012	http://fuzzy-sl.squirrel.link/ontology/Cave
23	http://fuzzy-sl.squirrel.link/data/cisite_50	Golema Pesht Cave near Zdu...	Lowe et al., 2012	http://fuzzy-sl.squirrel.link/ontology/ArchaeologicalSite
24	http://fuzzy-sl.squirrel.link/data/cisite_8	Gulf of Naples	Arienzo et al., 2009	http://fuzzy-sl.squirrel.link/ontology/Bight
25	http://fuzzy-sl.squirrel.link/data/cisite_65	Haua-Fteah (Libya)	Lowe et al., 2012	http://fuzzy-sl.squirrel.link/ontology/Cave
26	http://fuzzy-sl.squirrel.link/data/cisite_65	Haua-Fteah (Libya)	Lowe et al., 2012	http://fuzzy-sl.squirrel.link/ontology/ArchaeologicalSite
27	http://fuzzy-sl.squirrel.link/data/cisite_61	Kisiljevo (Serbia)	Baykal et al., 2018	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
28	http://fuzzy-sl.squirrel.link/data/cisite_46	Klissoura (Greece)	Lowe et al., 2012	http://fuzzy-sl.squirrel.link/ontology/UnknownCategory
29	http://fuzzy-sl.squirrel.link/data/cisite_60	Kostenki (Russia)	Fedele et al., 2003	http://fuzzy-sl.squirrel.link/ontology/InhabitedPlace
30	http://fuzzy-sl.squirrel.link/data/cisite_59	Kostenki-Borshchevo (Russia)	Fedele et al., 2008	http://fuzzy-sl.squirrel.link/ontology/ArchaeologicalSite

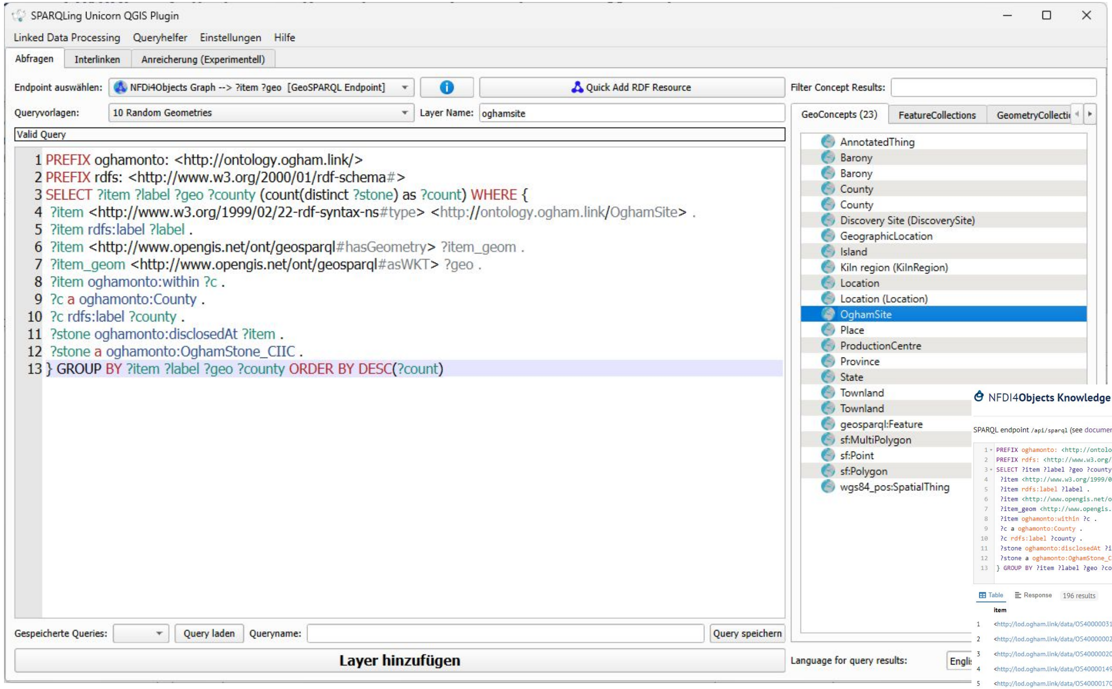
Alle Objekte anzeigen

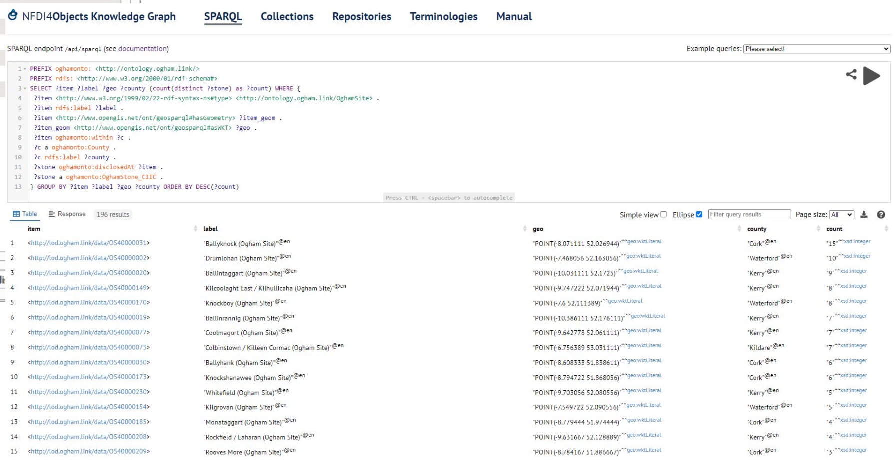
Campanian Ignimbrite Findspots via Solid Pod





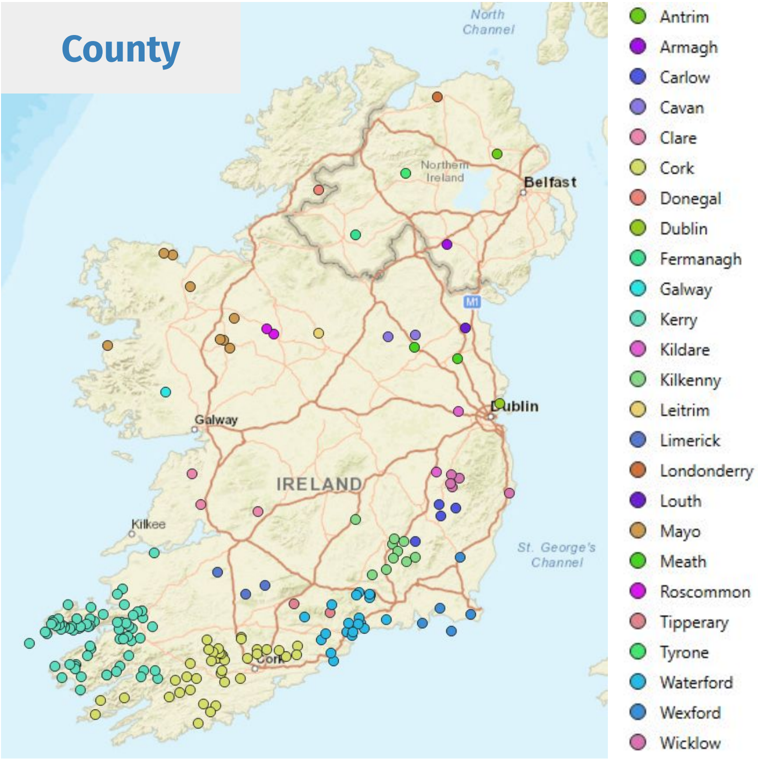
NFDI4Objects Knowledge Graph

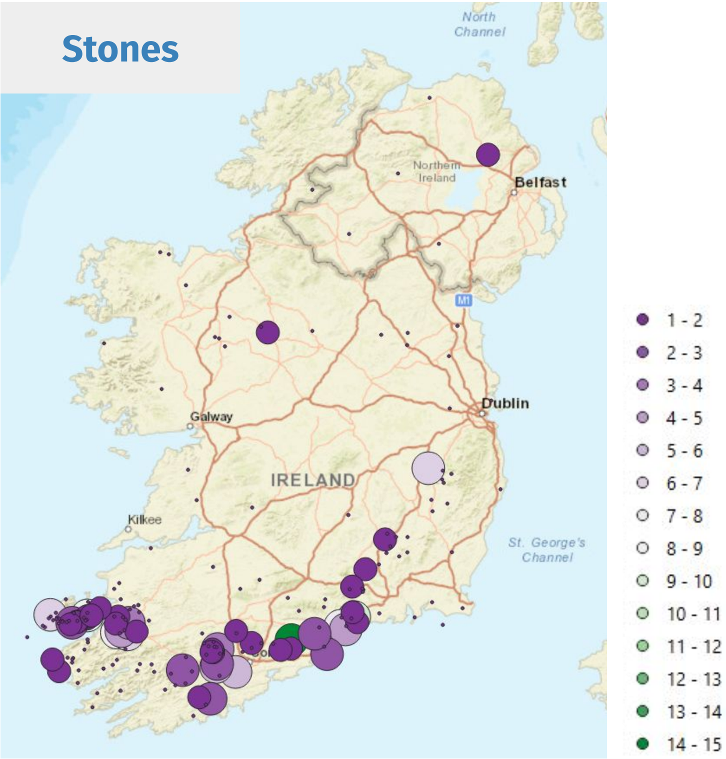




https://t1p.de/v0lg7

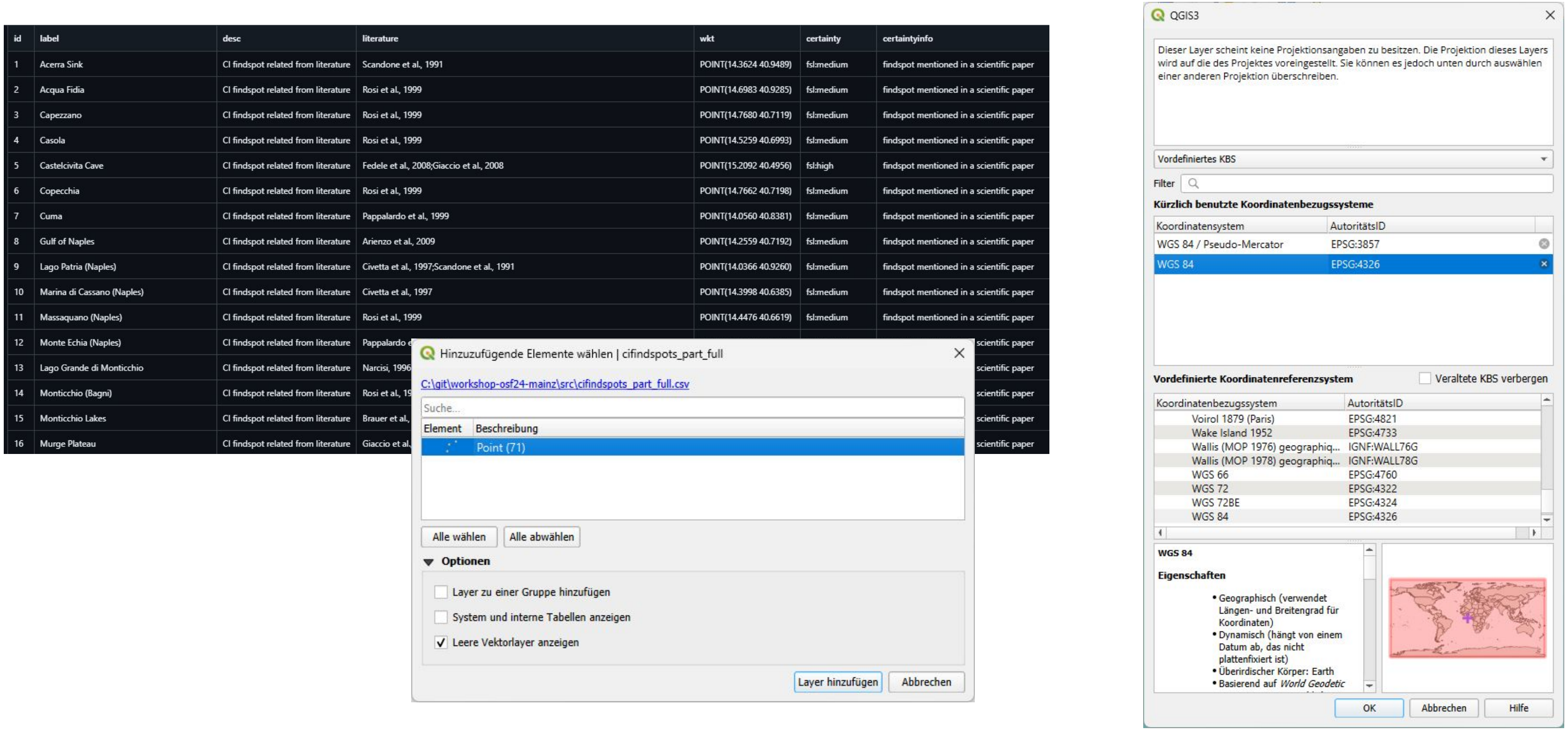
N4O KG - Ogham Stones / Sites / Counties



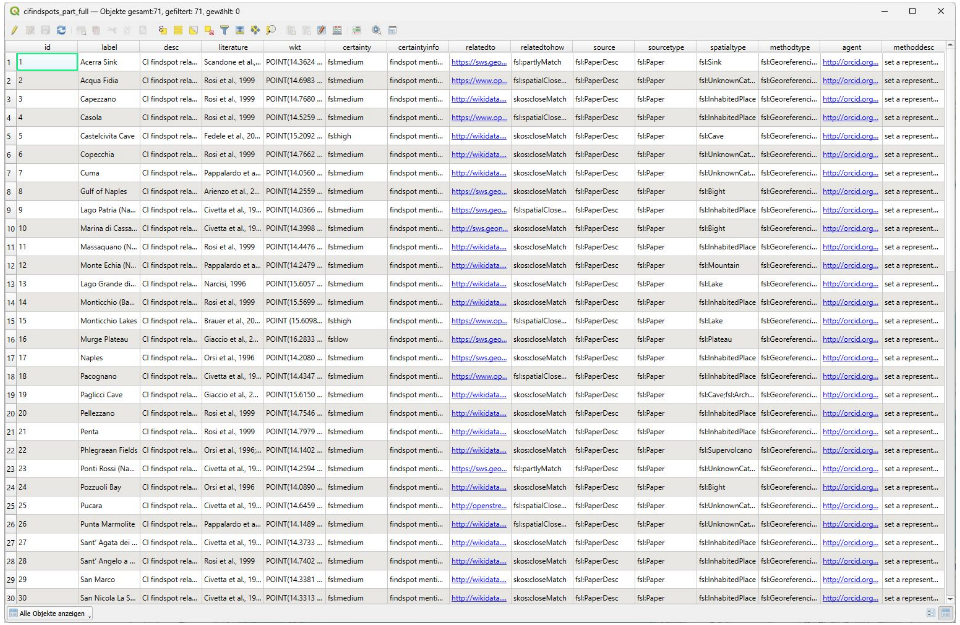


N4O KG - Ogham Stones / Sites / Counties

Transformation of QGIS vector layers to RDF



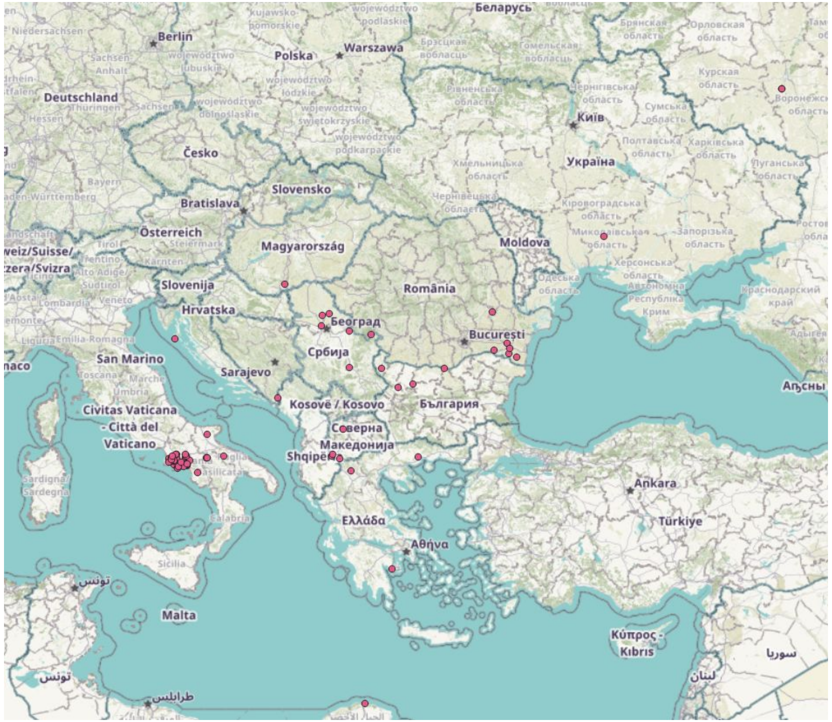
Campanian Ignimbrite Findspot CSV to RDF



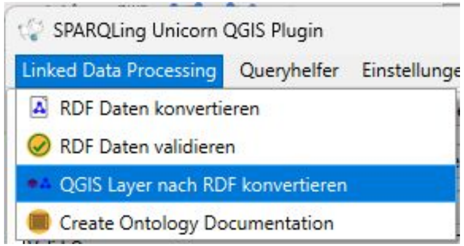
cifindspots_part_full — Objekte gesamt:71, gefiltert: 71, gewählt: 0

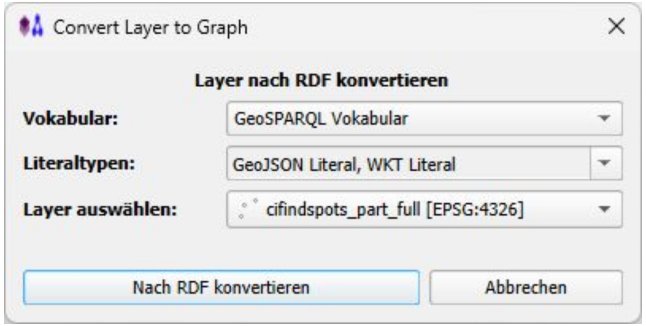
	id	label	desc	literature	wkt	certainty	certaintyinfo	relatedto	relatedtohow	source	sourcetype	spatialtype	methodtype	agent	methoddesc
1	1	Acerra Sink	CI findspot rela...	Scandone et al.,...	POINT(14.3624 ...	fsl:medium	findspot menti...	https://sws.geo...	fsl:partlyMatch	fsl:PaperDesc	fsl:Paper	fsl:Sink	fsl:Georeferenci...	http://orcid.org...	set a represent...
2	2	Acqua Fidia	CI findspot rela...	Rosi et al., 1999	POINT(14.6983 ...	fsl:medium	findspot menti...	https://www.op...	fsl:spatialClose...	fsl:PaperDesc	fsl:Paper	fsl:UnknownCat...	fsl:Georeferenci...	http://orcid.org...	set a represent...
3	3	Capezzano	CI findspot rela...	Rosi et al., 1999	POINT(14.7680 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
4	4	Casola	CI findspot rela...	Rosi et al., 1999	POINT(14.5259 ...	fsl:medium	findspot menti...	https://www.op...	fsl:spatialClose...	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
5	5	Castelcivita Cave	CI findspot rela...	Fedele et al., 20...	POINT(15.2092 ...	fsl:high	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Cave	fsl:Georeferenci...	http://orcid.org...	set a represent...
6	6	Copecchia	CI findspot rela...	Rosi et al., 1999	POINT(14.7662 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:UnknownCat...	fsl:Georeferenci...	http://orcid.org...	set a represent...
7	7	Cuma	CI findspot rela...	Pappalardo et a...	POINT(14.0560 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:UnknownCat...	fsl:Georeferenci...	http://orcid.org...	set a represent...
8	8	Gulf of Naples	CI findspot rela...	Arienzo et al., 2...	POINT(14.2559 ...	fsl:medium	findspot menti...	https://sws.geo...	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Bight	fsl:Georeferenci...	http://orcid.org...	set a represent...
9	9	Lago Patria (Na...	CI findspot rela...	Civetta et al., 19...	POINT(14.0366 ...	fsl:medium	findspot menti...	https://sws.geo...	fsl:spatialClose...	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
10	10	Marina di Cassa...	CI findspot rela...	Civetta et al., 19...	POINT(14.3998 ...	fsl:medium	findspot menti...	http://sws.geon...	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Bight	fsl:Georeferenci...	http://orcid.org...	set a represent...
11	11	Massaquano (N...	CI findspot rela...	Rosi et al., 1999	POINT(14.4476 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
12	12	Monte Echia (N...	CI findspot rela...	Pappalardo et a...	POINT(14.2479 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Mountain	fsl:Georeferenci...	http://orcid.org...	set a represent...
13	13	Lago Grande di...	CI findspot rela...	Narcisi, 1996	POINT(15.6057 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Lake	fsl:Georeferenci...	http://orcid.org...	set a represent...
14	14	Monticchio (Ba...	CI findspot rela...	Rosi et al., 1999	POINT(15.5699 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
15	15	Monticchio Lakes	CI findspot rela...	Brauer et al., 20...	POINT (15.6098...	fsl:high	findspot menti...	https://www.op...	fsl:spatialClose...	fsl:PaperDesc	fsl:Paper	fsl:Lake	fsl:Georeferenci...	http://orcid.org...	set a represent...
16	16	Murge Plateau	CI findspot rela...	Giaccio et al., 2...	POINT(16.2833 ...	fsl:low	findspot menti...	https://sws.geo...	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Plateau	fsl:Georeferenci...	http://orcid.org...	set a represent...
17	17	Naples	CI findspot rela...	Orsi et al., 1996	POINT(14.2080 ...	fsl:medium	findspot menti...	https://sws.geo...	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
18	18	Pacognano	CI findspot rela...	Civetta et al., 19...	POINT(14.4347 ...	fsl:medium	findspot menti...	https://www.op...	fsl:spatialClose...	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
19	19	Paglicci Cave	CI findspot rela...	Giaccio et al., 2...	POINT(15.6150 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Cave;fsl:Arch...	fsl:Georeferenci...	http://orcid.org...	set a represent...
20	20	Pellezzano	CI findspot rela...	Rosi et al., 1999	POINT(14.7546 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
21	21	Penta	CI findspot rela...	Rosi et al., 1999	POINT(14.7979 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
22	22	Phlegraean Fields	CI findspot rela...	Orsi et al., 1996;...	POINT(14.1402 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Supervolcano	fsl:Georeferenci...	http://orcid.org...	set a represent...
23	23	Ponti Rossi (Na...	CI findspot rela...	Civetta et al., 19...	POINT(14.2594 ...	fsl:medium	findspot menti...	https://sws.geo...	fsl:partlyMatch	fsl:PaperDesc	fsl:Paper	fsl:UnknownCat...	fsl:Georeferenci...	http://orcid.org...	set a represent...
24	24	Pozzuoli Bay	CI findspot rela...	Orsi et al., 1996	POINT(14.0890 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:Bight	fsl:Georeferenci...	http://orcid.org...	set a represent...
25	25	Pucara	CI findspot rela...	Civetta et al., 19...	POINT(14.6459 ...	fsl:medium	findspot menti...	http://openstre...	fsl:spatialClose...	fsl:PaperDesc	fsl:Paper	fsl:UnknownCat...	fsl:Georeferenci...	http://orcid.org...	set a represent...
26	26	Punta Marmolite	CI findspot rela...	Pappalardo et a...	POINT(14.1489 ...	fsl:medium	findspot menti...	http://wikidata....	fsl:spatialClose...	fsl:PaperDesc	fsl:Paper	fsl:UnknownCat...	fsl:Georeferenci...	http://orcid.org...	set a represent...
27	27	Sant' Agata dei ...	CI findspot rela...	Civetta et al., 19...	POINT(14.3733 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
28	28	Sant' Angelo a ...	CI findspot rela...	Rosi et al., 1999	POINT(14.7402 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
29	29	San Marco	CI findspot rela...	Civetta et al., 19...	POINT(14.3381 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...
30	30	San Nicola La S...	CI findspot rela...	Civetta et al., 19...	POINT(14.3313 ...	fsl:medium	findspot menti...	http://wikidata....	skos:closeMatch	fsl:PaperDesc	fsl:Paper	fsl:InhabitedPlace	fsl:Georeferenci...	http://orcid.org...	set a represent...

Alle Objekte anzeigen



Campanian Ignimbrite Findspot CSV to RDF






```
suni:c434e049-c2ab-4308-8d6b-b3247b5ffba0 a suni:d1f06db3-dd8c-4b96-a360-f28909f76c65 ;
    suni:agent <http://orcid.org/0009-0008-2877-3204> ;
    suni:certainty "fsl:medium"^^xsd:string ;
    suni:certaintyinfo "Fitzsimmons et al. (2014), p.76: “In this paper we investigate [...] the site of Urluia Quarry on the Dobrogea loess plateau [...], some 15
    km south of the Danube River. The site immediately overlies the Quaternary-uplifted Cretaceous-Tertiary-age limestone basement rocks (Munteanu et al., 2008),
    which were the target of earlier quarrying activities.”; Pötter et al. (2021), p.5: “The Urluia (URL) LPS is located in an abandoned limestone quarry on the
    limestone plateau of the Dobrogea (Fitzsimmons et al., 2013; Fitzsimmons and Hambach, 2014; Obreht et al., 2017).”"^^xsd:string ;
    suni:desc "CI findspot related from literature"^^xsd:string ;
    suni:label "Urluia (Romania)"^^xsd:string ;
    suni:literature "Fitzsimmons et al., 2014;Fitzsimmons et al.,, 2013;Obreht et al., 2017;Pötter et al., 2021"^^xsd:string ;
    suni:methoddesc "set a representative point based on scientific papers using Google Maps"^^xsd:string ;
    suni:methodtype "fsl:Georeferencing"^^xsd:string ;
    suni:relatedto <http://sws.geonames.org/664132;http://openstreetmap.org/way/84975654> ;
    suni:relatedtohow "skos:closeMatch"^^xsd:string ;
    suni:source "fsl:PaperDesc"^^xsd:string ;
    suni:sourcetype "fsl:Paper"^^xsd:string ;
    suni:spatialtype "fsl:InhabitedPlace"^^xsd:string ;
    suni:wkt "POINT(27.9021 44.0947)"^^xsd:string ;
    geo:hasGeometry suni:c434e049-c2ab-4308-8d6b-b3247b5ffba0_geom .
```

```
suni:c434e049-c2ab-4308-8d6b-b3247b5ffba0_geom a geo:Point ;
    geo:asWKT "Point (27.902100000000079 44.094700000000312)"^^geo:wktLiteral .
```

Campanian Ignimbrite Findspot CSV to RDF

RDF HTML documentation

SPARQL Unicorn Ontology Documentation

DOI 10.5281/zenodo.8190763

This repository hosts a standalone version of the HTML documentation feature included in the SPARQLing Unicorn QGIS Plugin.

Rather than initiating the documentation generation within the SPARQLing Unicorn QGIS Plugin, this python script allows the generation of the documentation standalone or as a Github Action.

The standalone script does not rely on QGIS classes and does not provide the full functionality available in the SPARQLUnicorn QGIS Plugin.

Deviations from the SPARQLing Unicorn Plugin are listed as follows:

- Support for less geometry literals: Only WKT and GeoJSON literals are supported for rendering

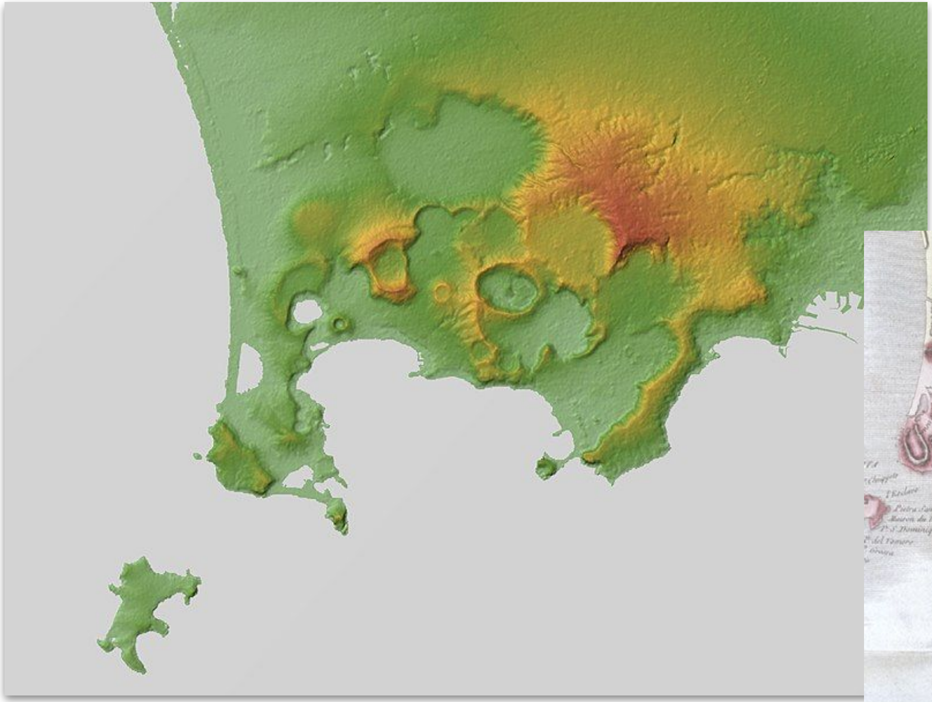
Usage Example as Github Action

For a usage example please refer to this repository: https://github.com/sparqlunicorn/sparqlunicornGoesGIS_testdata

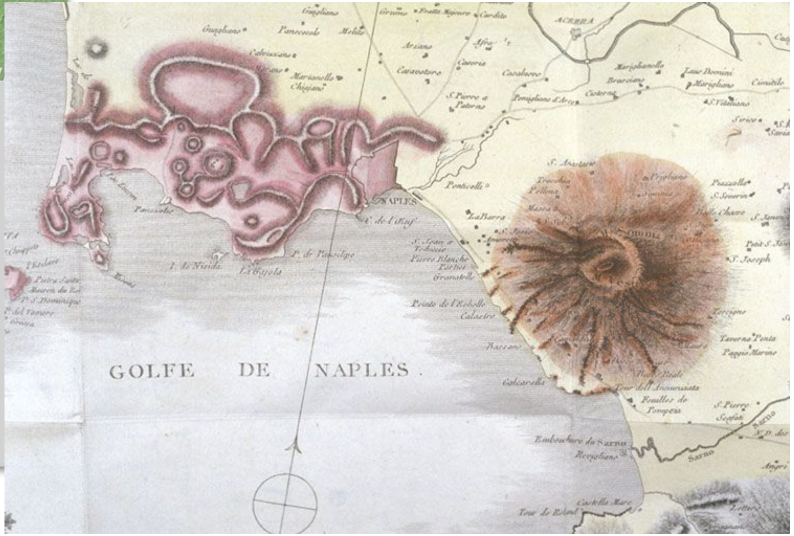
https://github.com/sparqlunicorn/sparqlunicornGoesGIS-ontdoc
via 10.5281/zenodo.8190763

HTML creation with the help of Unicorn and GitHub Actions

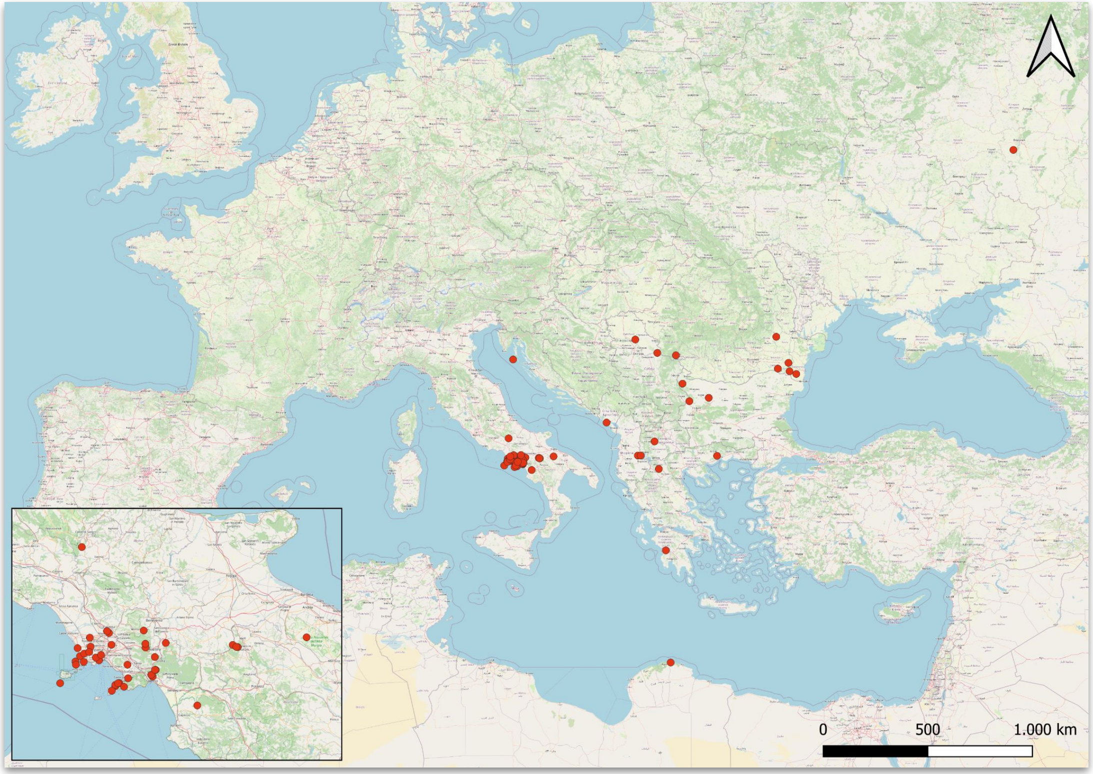
Batholith, Public domain, via Wikimedia Commons



Breislak, Scipione (1748-1826). Voyages physiques et lythologiques dans la Campanie. Paris: Dentu, Imprimeur-libraire, 1801, via https://vulcan.lindahall.org/12_large.shtml

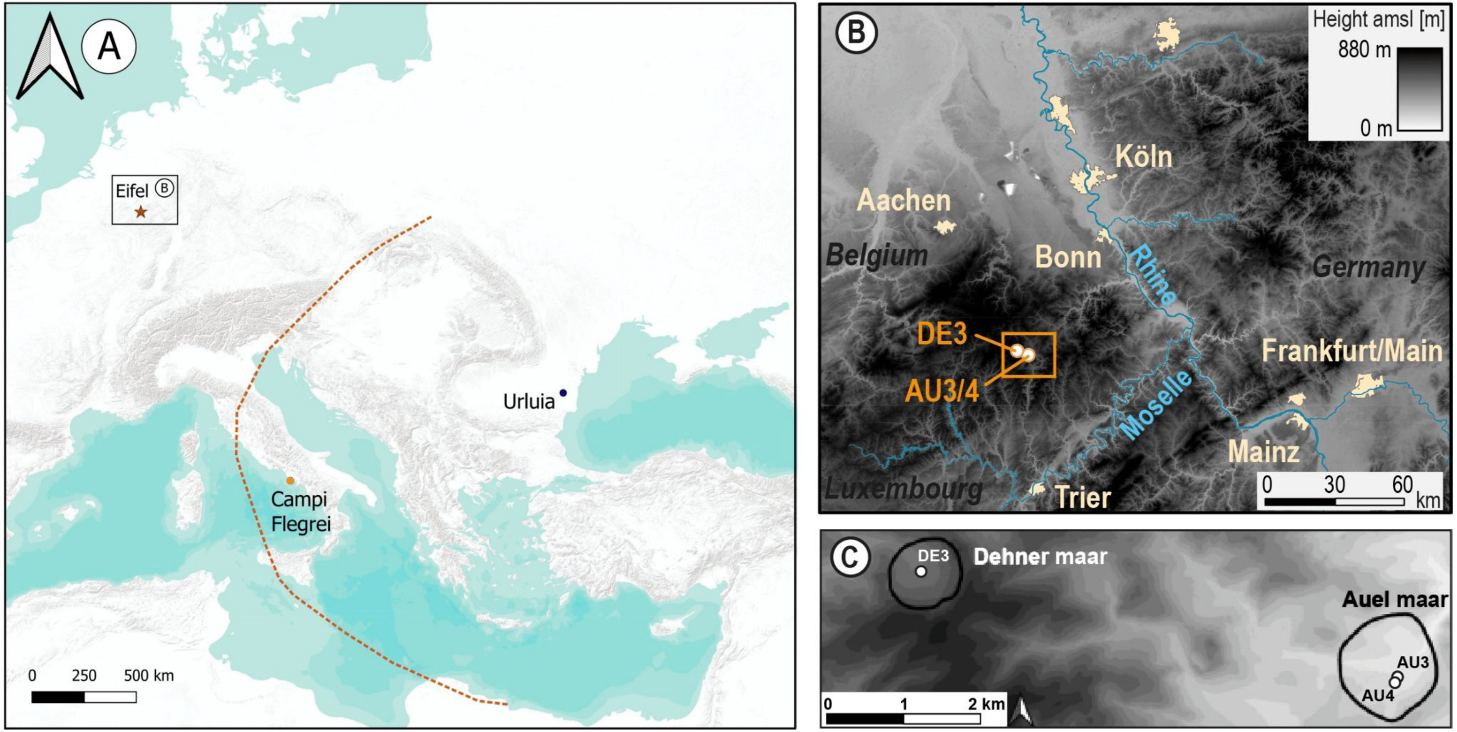


About 40,000 yr b2k ago, the largest eruption of the Campanian Ignimbrite (CI) took place in the Phlegraean Fields.



Evidence of the ash fall from this late Pleistocene volcanic event can be found throughout Central Europe.

via https://doi.org/10.3390/quat7020017, Fig. 1



These sites are recorded in several publications, e.g. with coordinates or references to cities, regions, caves and archaeological sites.

via https://doi.org/10.1029/139GM20 © Fedele et al.

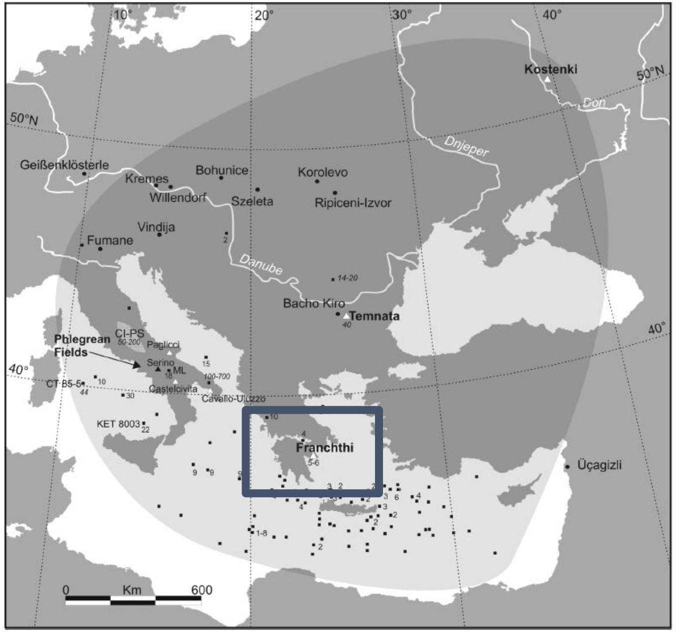


Figure 1. Geographic distribution of the Campanian Ignimbrite deposits, including archaeological and sampling sites mentioned in the paper. Solid squares: CI tephra occurrences and related thickness in cm (in italics if reworked) [modified from *Cornell et al.*, 1983; *Amirkhanov et al.*, 1993; *Cini Castagnoli et al.*, 1995; *Narcisi and Vezzoli*, 1999; *Fedele et al.*, 2002; *Upton et al.*, 2002]. Dotted area in central Italy: distribution of the CI-derived paleosol (CI-PS [*Frezzotti and Narcisi*, 1996]). Solid triangles: archaeological sites with CI ash layer. Blank triangles: archaeological sites with ash layer attributed to the CI on the basis of cultural-stratigraphic position and ^{14}C dating (*Fedele et al.*, 2002; *Giaccio and Isaia*, on file). Solid circles: selected European Palaeolithic sites within the area potentially affected by the CI air-fall.

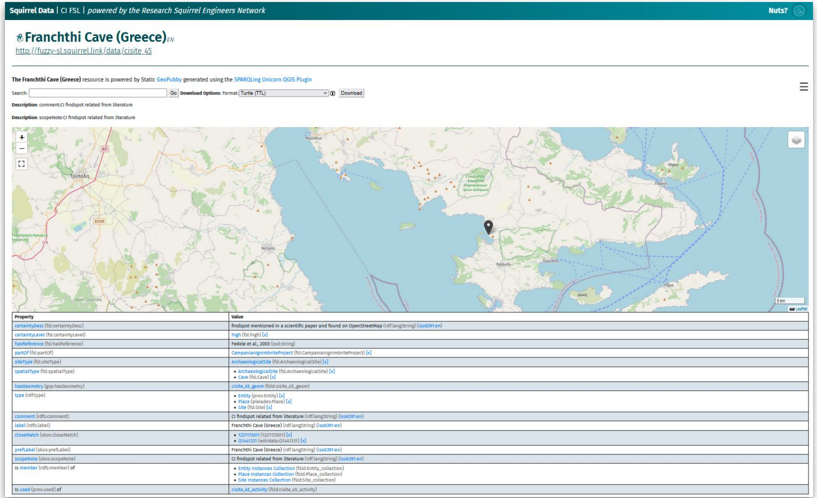
Franchthi Cave (Greece) in Fedele et al. (2003)

Zde, CC BY-SA 4.0, via Wikimedia Commons

Zde, CC BY-SA 4.0, via Wikimedia Commons

Xalaros, Attribution, via Wikimedia Commons

https://research-squirrel-engineers.github.io/campanian-ignimbrite-geo/cisite_45/index.html



Franchthi Cave as Archaeological Site in the LOD Cloud

https://research-squirrel-engineers.github.io/campanian-ignimbrite-geo/Site_collection/index.html

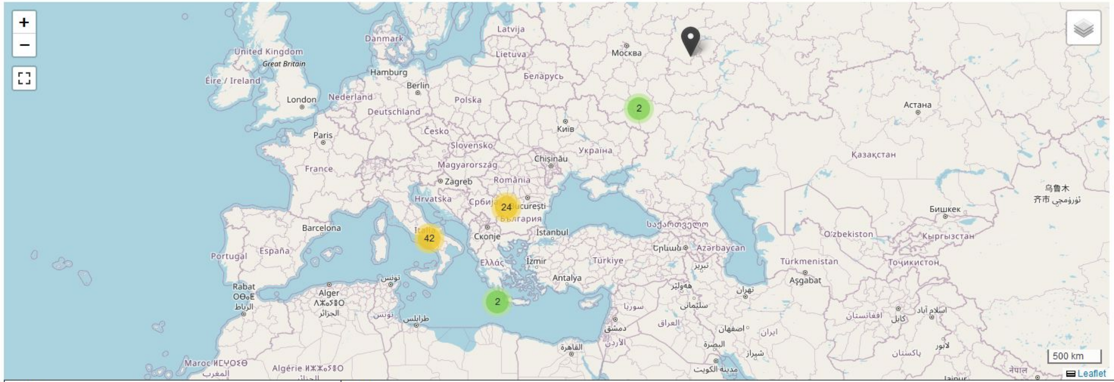
Squirrel Data | CI FSL | *powered by the Research Squirrel Engineers Network* **Nuts?**

Site Instances Collection *EN*

http://fuzzy-sl.squirrel.link/data/Site_collection

The Site Instances Collection resource is powered by Static GeoPubby generated using the SPARQLing Unicorn QGIS Plugin

Search: Go **Download Options:** Format: Turtle (TTL) Download

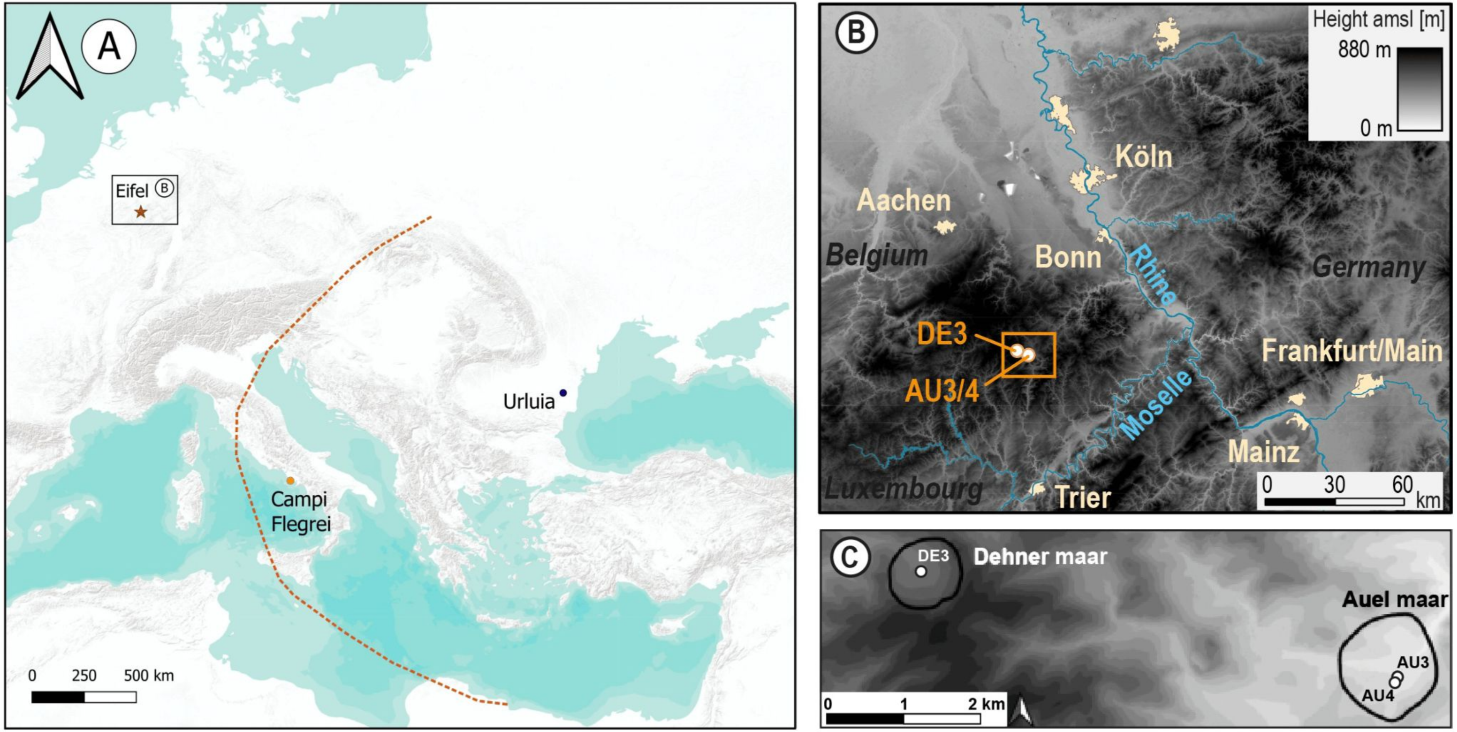


Property	Value
type (rdf:type)	FeatureCollection (gsp:FeatureCollection) [x]
label (rdfs:label)	Site Instances Collection (rdf:langString) (iso6391:en)
member (rdfs:member)	▾ 71 values ▪ Acerra Sink (fsld:cisite_1) ▪ Acqua Fidia (fsld:cisite_2) ▪ Balta Alba (Romania) (fsld:cisite_57) ▪ Batajnica (fsld:cisite_104) ▪ Bratul Borcea (Romania) (fsld:cisite_56) ▪ Capezzano (fsld:cisite_3) ▪ Casola (fsld:cisite_4) ▪ Castelcivita Cave (fsld:cisite_5) ▪ Copecchia (fsld:cisite_6)

All these sites and information can be visualised via the Unicorn

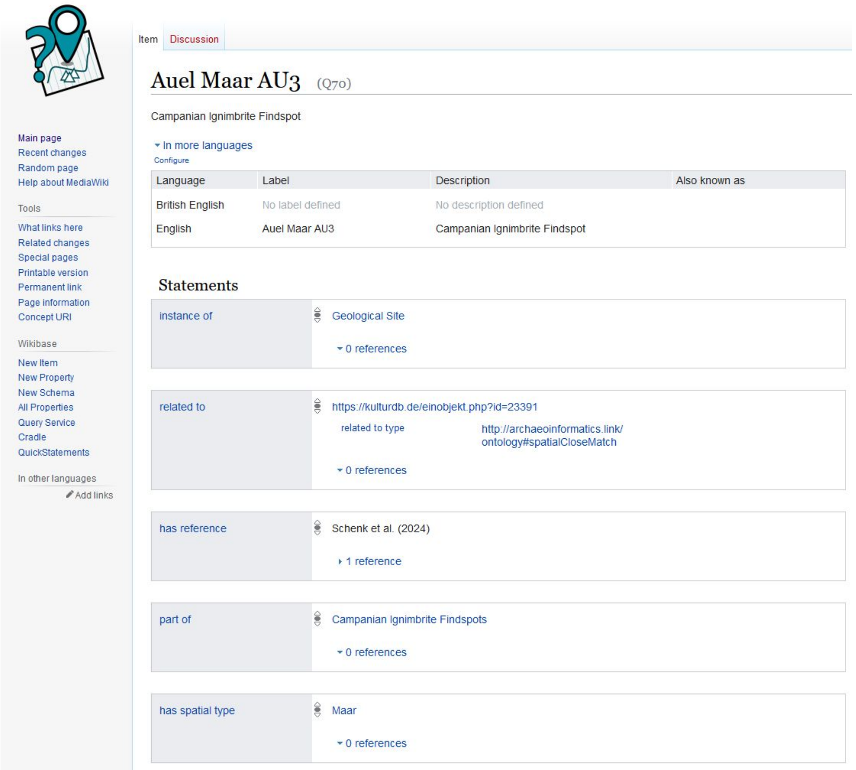
image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

Using **Python Minions** these data could be also transformed into **QuickStatements** within the **fuzzy-sl Wikibase** via https://doi.org/10.3390/quat7020017, Fig. 1



Auel Maar (AU3/4) & Dehner Maar (DE3) sites in Schenk et al. (2024)

via https://fuzzy-sl.wikibase.cloud/wiki/Item:Q70



Item | Discussion

Auel Maar AU3 (Q70)

Campanian Ignimbrite Findspot

In more languages
Configure

Language	Label	Description	Also known as
British English	No label defined	No description defined	
English	Auel Maar AU3	Campanian Ignimbrite Findspot	

Main page
Recent changes
Random page
Help about MediaWiki

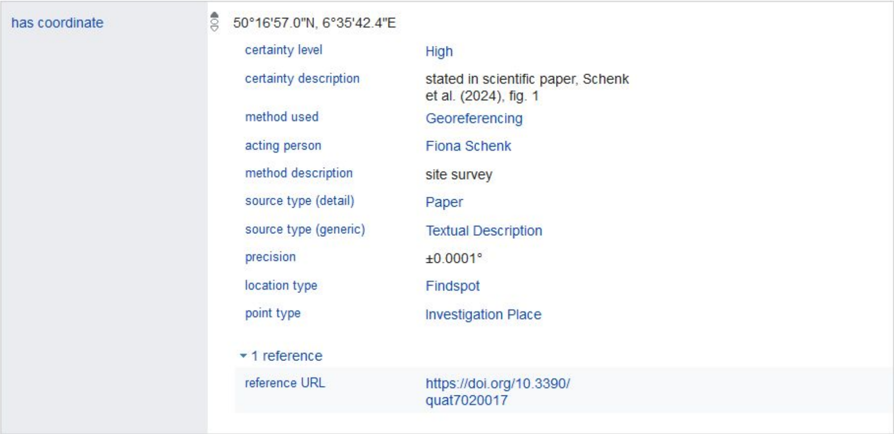
Tools
What links here
Related changes
Special pages
Printable version
Permanent link
Page information
Concept URI

Wikibase
New Item
New Property
New Schema
All Properties
Query Service
Cradle
QuickStatements

In other languages
Add links

Statements

instance of	Geological Site 0 references
related to	https://kulturdb.de/einobjekt.php?id=23391 related to type: http://archaeoinformatics.link/ontology#spatialCloseMatch 0 references
has reference	Schenk et al. (2024) 1 reference
part of	Campanian Ignimbrite Findspots 0 references
has spatial type	Maar 0 references

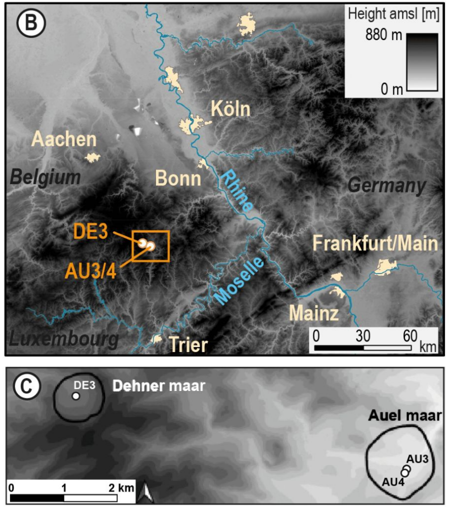


has coordinate	50°16'57.0"N, 6°35'42.4"E
certainty level	High
certainty description	stated in scientific paper, Schenk et al. (2024), fig. 1
method used	Georeferencing
acting person	Fiona Schenk
method description	site survey
source type (detail)	Paper
source type (generic)	Textual Description
precision	±0.0001°
location type	Findspot
point type	Investigation Place

1 reference

reference URL	https://doi.org/10.3390/quat7020017

via https://doi.org/10.3390/quat7020017, Fig. 1



Auel AU3 [Q70] in the fuzzy-sl Wikibase

Squilly can use …

image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

… **Jupyter Python Minions** as **Jupyter Notebooks**.

Jupyter Python Minions as Jupyter Notebooks

by the Research Squirrel Engineers Network

Ogham Sites

... in Wikidata and the NFDI4Objects Knowledge Graph to create diagrams and maps.

Define SPARQL query service

```python
import os
from SPARQLWrapper import SPARQLWrapper, JSON
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
from shapely.geometry import Point
import contextily as ctx  # For adding OpenStreetMap basemaps
from matplotlib.patches import Patch
from scipy.stats import gaussian_kde
import numpy as np

def querySparql(query):
    sparql = SPARQLWrapper("https://query.wikidata.org/sparql")
    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    results = sparql.queryAndConvert()
    return results['results']['bindings']

# Define the GeoJSON file path
geojson_file = os.path.join(os.getcwd(), "gs_ireland_island.geojson")  # Adjusted for Jupyter Notebook
```

Define the SPARQL Query

```python
# SPARQL Query
oghamQuery = """
SELECT ?item ?itemLabel ?geo ?site ?siteLabel ?county ?countyLabel WHERE {
  ?item wdt:P31 wd:Q2016147.
  ?item wdt:P189 ?site.
  ?site wdt:P31 wd:Q72617071.
  ?item wdt:P189 ?county.
  ?county wdt:P31 wd:Q179872.
  ?item wdt:P625 ?geo.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
}
"""
```

Fetch Data and Convert to DataFrame

```python
# Fetch data using the SPARQL query
sparql_results = querySparql(oghamQuery)

# Convert SPARQL JSON results into a DataFrame
data = []
for result in sparql_results:
    geo = result['geo']['value'] if 'geo' in result else None
    lat, lon = (None, None)
    if geo:
        lon, lat = map(float, geo.replace("Point(", "").replace(")", "").split())
    data.append({
        "item": result['item']['value'],
        "itemLabel": result['itemLabel']['value'],
        "site": result['site']['value'],
        "siteLabel": result['siteLabel']['value'],
        "county": result['county']['value'],
        "countyLabel": result['countyLabel']['value'],
        "latitude": lat,
        "longitude": lon,
    })

df = pd.DataFrame(data)
df
```

WIKIDATA

https://research-squirrel-engineers.github.io/jupyter-nb-lod/wikidata-ogham-sites-map

	item	itemLabel	site	siteLabel	county	countyLabel	latitude	longitude
0	http://www.wikidata.org/entity/Q70892459	CIIC 71 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85394002	Ahalisky (Ogham Site)	http://www.wikidata.org/entity/Q162475	County Cork	51.678889	-8.846944
1	http://www.wikidata.org/entity/Q70892460	CIIC 72 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85394004	Aultagh (Ogham Site)	http://www.wikidata.org/entity/Q162475	County Cork	51.770833	-9.088056
2	http://www.wikidata.org/entity/Q70892463	CIIC 73 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393926	Carhoovauler (Ogham Site)	http://www.wikidata.org/entity/Q162475	County Cork	51.688333	-8.956111
3	http://www.wikidata.org/entity/Q70892466	CIIC 74 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393926	Carhoovauler (Ogham Site)	http://www.wikidata.org/entity/Q162475	County Cork	51.688333	-8.956111
4	http://www.wikidata.org/entity/Q70892468	CIIC 75 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393928	Keenrath (Ogham Site)	http://www.wikidata.org/entity/Q162475	County Cork	51.759444	-9.185833
...	...	...	...	...	...	...	...	...
328	http://www.wikidata.org/entity/Q70892620	CIIC 156 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393964	Ballintaggart (Ogham Site)	http://www.wikidata.org/entity/Q184469	County Kerry	52.172500	-10.031111
329	http://www.wikidata.org/entity/Q70892623	CIIC 157 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393964	Ballintaggart (Ogham Site)	http://www.wikidata.org/entity/Q184469	County Kerry	52.172500	-10.031111
330	http://www.wikidata.org/entity/Q70892626	CIIC 158 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393964	Ballintaggart (Ogham Site)	http://www.wikidata.org/entity/Q184469	County Kerry	52.172500	-10.031111
331	http://www.wikidata.org/entity/Q70892629	CIIC 159 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393964	Ballintaggart (Ogham Site)	http://www.wikidata.org/entity/Q184469	County Kerry	52.172500	-10.031111
332	http://www.wikidata.org/entity/Q70892630	CIIC 160 (Ogham Stone Concept by RAS Macalister)	http://www.wikidata.org/entity/Q85393964	Ballintaggart (Ogham Site)	http://www.wikidata.org/entity/Q184469	County Kerry	52.172500	-10.031111

333 rows × 8 columns

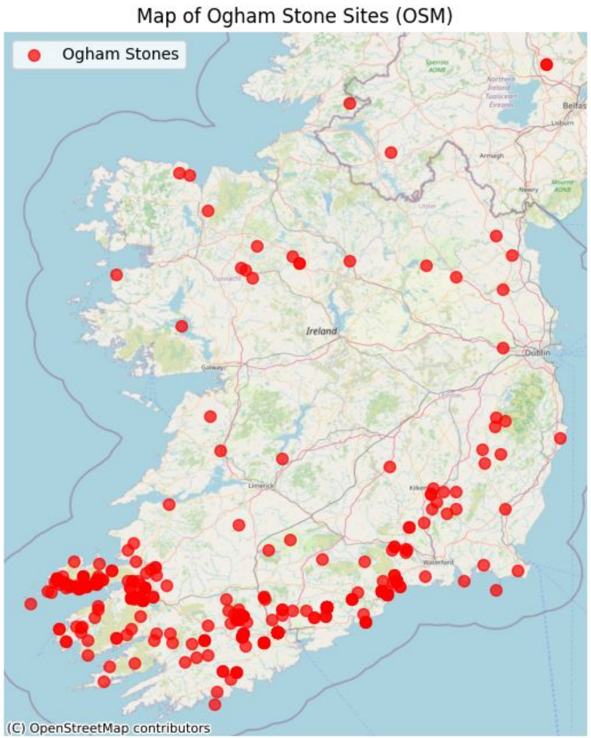
SPARQL query to create DataFrame

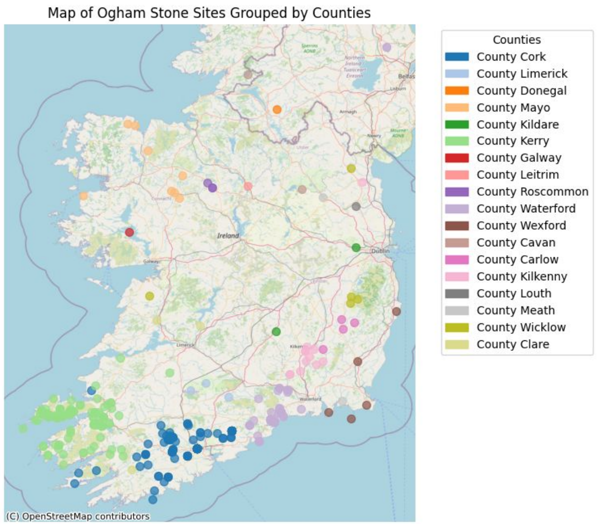
Visualise the Data as Maps

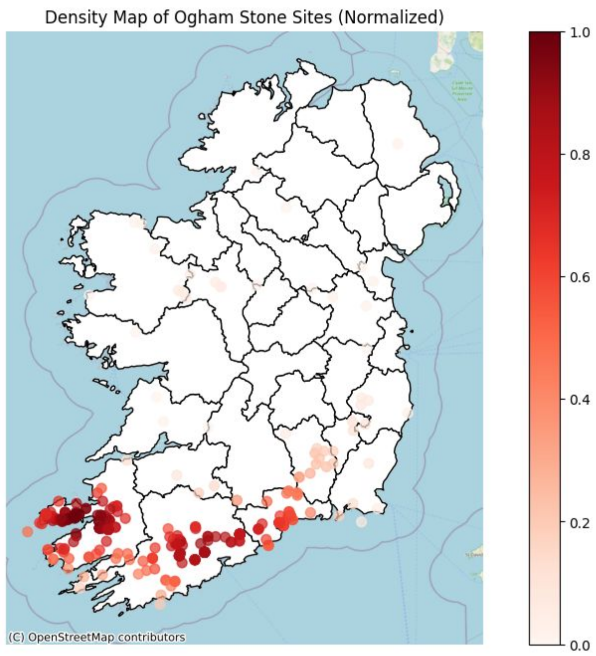
```
# Check if DataFrame is populated
if df.empty:
    print("No data retrieved from the query.")
else:
    # Filter rows with valid coordinates
    df_with_coords = df.dropna(subset=['latitude', 'longitude'])

    # Create a GeoDataFrame
    gdf = gpd.GeoDataFrame(
        df_with_coords,
        geometry=[Point(xy) for xy in zip(df_with_coords['longitude'], df_with_coords['latitude'])],
        crs="EPSG:4326"
    )

    # Convert to Web Mercator for OSM basemap
    gdf_mercator = gdf.to_crs(epsg=3857)

    # Load Ireland boundary from GeoJSON
    ireland_boundary = gpd.read_file(geojson_file)
    ireland_boundary = ireland_boundary.to_crs(epsg=3857)

    # Map 1: Plot points without text decorations
    fig, ax = plt.subplots(figsize=(12, 8))
    gdf_mercator.plot(ax=ax, color='red', markersize=50, alpha=0.7, label="Ogham Stones")
    ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik, zoom=8)
    ax.set_axis_off()
    plt.title("Map of Ogham Stone Sites (OSM)")
    plt.legend()
    plt.show()
```

```
# Map 2: Plot with points colored by county and fix legend
fig, ax = plt.subplots(figsize=(12, 8))
unique_counties = gdf['countyLabel'].unique()
colors = plt.cm.tab20.colors[:len(unique_counties)]  # Generate unique colors
county_colors = {county: colors[idx] for idx, county in enumerate(unique_counties)}

patches = []
for county, color in county_colors.items():
    county_data = gdf_mercator[gdf_mercator['countyLabel'] == county]
    county_data.plot(ax=ax, color=color, markersize=50, alpha=0.7)
    patches.append(Patch(color=color, label=county))  # Add patch for legend

ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik, zoom=8)
ax.set_axis_off()
plt.title("Map of Ogham Stone Sites Grouped by Counties")
plt.legend(handles=patches, title="Counties", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

# Map 3: Density map with normalized values
x = gdf_mercator.geometry.x
y = gdf_mercator.geometry.y

# Calculate kernel density
xy = np.vstack([x, y])
kde = gaussian_kde(xy)
density = kde(xy)

# Normalize density to range [0, 1]
density_normalized = (density - density.min()) / (density.max() - density.min())
gdf_mercator['density'] = density_normalized

fig, ax = plt.subplots(figsize=(12, 8))
ireland_boundary.plot(ax=ax, color="white", edgecolor="black")
gdf_mercator.plot(ax=ax, column='density', cmap="Reds", markersize=50, alpha=0.7, legend=True)
ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik, zoom=8)
ax.set_axis_off()
plt.title("Density Map of Ogham Stone Sites (Normalized)")
plt.show()
```

Create Maps


Map of Ogham Stone Sites (OSM)
Ogham Stones
Ireland
Dublin
(C) OpenStreetMap contributors


Map of Ogham Stone Sites Grouped by Counties
Ireland
(C) OpenStreetMap contributors
Counties
County Cork
County Limerick
County Donegal
County Mayo
County Kildare
County Kerry
County Galway
County Leitrim
County Roscommon
County Waterford
County Wexford
County Cavan
County Carlow
County Kilkenny
County Louth
County Meath
County Wicklow
County Clare


Density Map of Ogham Stone Sites (Normalized)
(C) OpenStreetMap contributors
1.0
0.8
0.6
0.4
0.2
0.0

Output

https://research-squirrel-engineers.github.io/jupyter-nb-lod/n4okg-ogham-sites-county

Define SPARQL query service

```
from SPARQLWrapper import SPARQLWrapper, JSON
import pandas as pd
import matplotlib.pyplot as plt

def querySparql(query):
    sparql = SPARQLWrapper("https://graph.nfdi4objects.net/api/sparql")
    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    results = sparql.queryAndConvert()
    return results['results']['bindings']
```

Define the SPARQL Query

```
# SPARQL Query for Samian Ware Kiln Sites
oghamQuery = """
PREFIX oghamonto: <http://ontology.ogham.link/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?county (count(distinct ?stone) as ?count) WHERE {
 ?item <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://ontology.ogham.link/OghamSite> .
 ?item oghamonto:within ?c .
 ?c a oghamonto:County .
 ?c rdfs:label ?county .
 ?stone oghamonto:disclosedAt ?item .
 ?stone a oghamonto:OghamStone_CIIC .
} GROUP BY ?county ORDER BY DESC(?count)
"""
```

Fetch Data and Convert to DataFrame

```
# Fetch data using the SPARQL query
sparql_results = querySparql(oghamQuery)

# Convert SPARQL JSON results into a DataFrame
data = []
for result in sparql_results:
    data.append({
        "county": result.get("county", {}).get("value", None),
        "count": int(result.get("count", {}).get("value", 0))
    })

df = pd.DataFrame(data)
df
```

	county	count
0	Kerry	127
1	Cork	81
2	Waterford	47
3	Kilkenny	12
4	Kildare	8
5	Mayo	8
6	Wexford	5
7	Wicklow	5
8	Carlow	4
9	Clare	3
10	Limerick	3
11	Roscommon	3
12	Antrim	2
13	Cavan	2
14	Meath	2
15	Tipperary	2

SPARQL query to create DataFrame

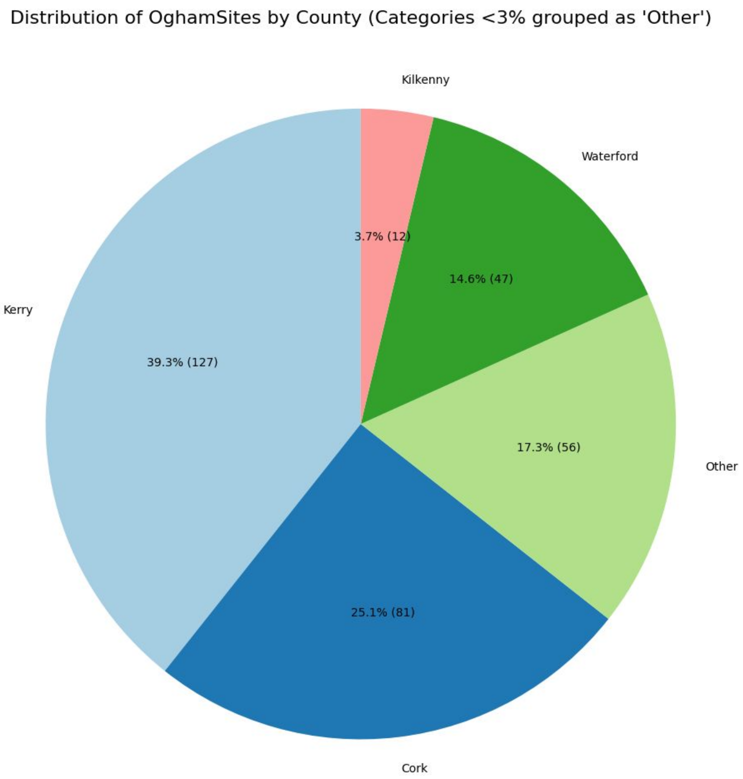
Pie Chart

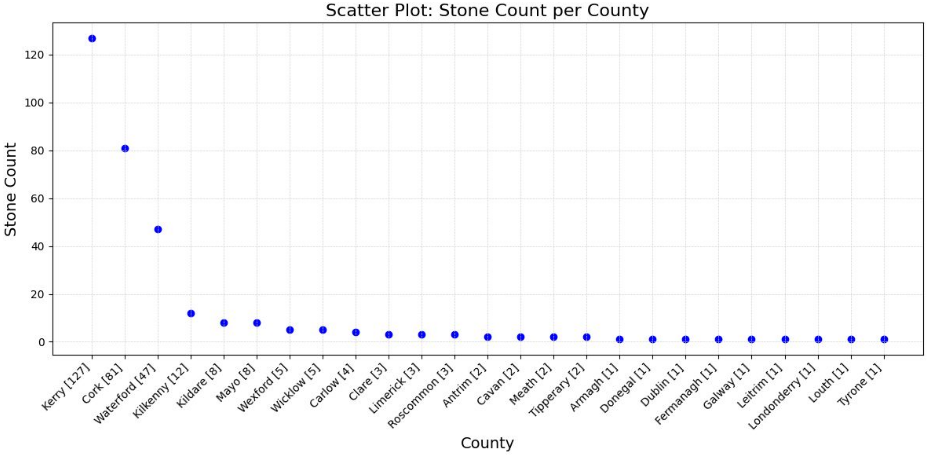
```
# Calculate the total number of stones.
total_count = df["count"].sum()

# Compute the percentage for each county.
df["percentage"] = df["count"] / total_count * 100

# Separate counties with a percentage >= 3% from those with less than 3%.
major_df = df[df["percentage"] >= 3].copy()
minor_df = df[df["percentage"] < 3].copy()

# If there are counties with less than 3%, combine them into an 'Other' category.
if not minor_df.empty:
    other_count = minor_df["count"].sum()
    major_df = pd.concat([major_df, pd.DataFrame([{"county": "Other", "count": other_count}])], ignore_index=True)

# Optional: Sort the categories by count in descending order.
major_df = major_df.sort_values("count", ascending=False)

# Define a function to format the autopct text to include both the percentage and the stone count.
def make_autopct(values):
    def my_autopct(pct):
        total = sum(values)
        count = int(round(pct * total / 100.0))
        return '{:.1f}% ({:d})'.format(pct, count)
    return my_autopct

# Create a pie chart displaying the distribution of OghamSites by county.
plt.figure(figsize=(10, 10))
major_df.set_index("county")["count"].plot(
    kind="pie",
    autopct=make_autopct(major_df["count"]),
    startangle=90,
    colors=plt.cm.Paired.colors
)
plt.title("Distribution of OghamSites by County (Categories <3% grouped as 'Other')", fontsize=16)
plt.ylabel("")  # Remove the default y-label.
plt.tight_layout()
plt.show()
```

Scatter Plot

```
# Create a scatter plot with county on the x-axis and stone count on the y-axis.
plt.figure(figsize=(12, 6))
x_positions = range(len(df))
plt.scatter(x_positions, df["count"], color="blue")
plt.title("Scatter Plot: Stone Count per County", fontsize=16)
plt.xlabel("County", fontsize=14)
plt.ylabel("Stone Count", fontsize=14)

# Create customised x-axis labels with county and count in square brackets.
xtick_labels = [f"{county} [{count}]" for county, count in zip(df["county"], df["count"])]
plt.xticks(x_positions, xtick_labels, rotation=45, ha="right")

# Add a light grey grid.
plt.grid(color="lightgrey", linestyle="--", linewidth=0.5)

plt.tight_layout()
plt.show()
```

Create Charts

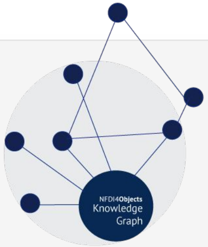

Distribution of OghamSites by County (Categories <3% grouped as 'Other')
Kilkenny
3.7% (12)
Waterford
14.6% (47)
Kerry
39.3% (127)
17.3% (56)
Other
25.1% (81)
Cork


Scatter Plot: Stone Count per County
Stone Count
0
20
40
60
80
100
120
Kerry [127]
Cork [81]
Waterford [47]
Kilkenny [12]
Kildare [8]
Mayo [8]
Wexford [5]
Wicklow [5]
Carlow [4]
Clare [3]
Limerick [3]
Roscommon [3]
Antrim [2]
Cavan [2]
Meath [2]
Tipperary [2]
Armagh [1]
Donegal [1]
Dublin [1]
Fermanagh [1]
Galway [1]
Leitrim [1]
Londonderry [1]
Louth [1]
Tyrone [1]
County

Output

https://research-squirrel-engineers.github.io/jupyter-nb-lod/n4okg-ogham-sites-maps

Define SPARQL query service




```python
import os
from SPARQLWrapper import SPARQLWrapper, JSON
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
from shapely.geometry import Point
import contextily as ctx  # For adding OpenStreetMap basemaps
from matplotlib.patches import Patch
from scipy.stats import gaussian_kde
import numpy as np

def querySparql(query):
    sparql = SPARQLWrapper("https://graph.nfdi4objects.net/api/sparql")
    sparql.setQuery(query)
    sparql.setReturnFormat(JSON)
    results = sparql.queryAndConvert()
    return results['results']['bindings']

# Define the GeoJSON file path
geojson_file = os.path.join(os.getcwd(), "gs_ireland_island.geojson")  # Adjusted for Jupyter Notebook
```

Define the SPARQL Query

```python
# SPARQL Query
oghamQuery = """
PREFIX oghamonto: <http://ontology.ogham.link/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?item ?label ?geo ?county (count(distinct ?stone) as ?count) WHERE {
 ?item <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://ontology.ogham.link/OghamSite> .
 ?item rdfs:label ?label .
 ?item <http://www.opengis.net/ont/geosparql#hasGeometry> ?item_geom .
 ?item_geom <http://www.opengis.net/ont/geosparql#asWKT> ?geo .
 ?item oghamonto:within ?c .
 ?c a oghamonto:County .
 ?c rdfs:label ?county .
 ?stone oghamonto:disclosedAt ?item .
 ?stone a oghamonto:OghamStone_CIIC .
} GROUP BY ?item ?label ?geo ?county ORDER BY DESC(?count)
"""
```

Fetch Data and Convert to DataFrame

```python
# Fetch data using the SPARQL query
sparql_results = querySparql(oghamQuery)

# Convert SPARQL JSON results into a DataFrame
data = []
for result in sparql_results:
    geo = result['geo']['value'] if 'geo' in result else None
    lat, lon = (None, None)
    if geo:
        lon, lat = map(float, geo.replace("POINT(", "").replace(")", "").split())
    data.append({
        "item": result['item']['value'],
        "label": result['label']['value'],
        "county": result['county']['value'],
        "count": int(result.get("count", {}).get("value", 0)),
        "latitude": lat,
        "longitude": lon,
    })

df = pd.DataFrame(data)
df
```

	item	label	county	count	latitude	longitude
0	http://lod.ogham.link/data/OS40000031	Ballyknock (Ogham Site)	Cork	15	52.026944	-8.071111
1	http://lod.ogham.link/data/OS40000002	Drumlohan (Ogham Site)	Waterford	10	52.163056	-7.468056
2	http://lod.ogham.link/data/OS40000020	Ballintaggart (Ogham Site)	Kerry	9	52.172500	-10.031111
3	http://lod.ogham.link/data/OS40000149	Kilcoolaght East / Kilhullicaha (Ogham Site)	Kerry	8	52.071944	-9.747222
4	http://lod.ogham.link/data/OS40000170	Knockboy (Ogham Site)	Waterford	8	52.111389	-7.600000
...	...	...	...	...	...	...
191	http://lod.ogham.link/data/OS40000062	Castletimon (Ogham Site)	Wicklow	1	52.909167	-6.063611
192	http://lod.ogham.link/data/OS40000168	Knickeen (Ogham Site)	Wicklow	1	52.998056	-6.535000
193	http://lod.ogham.link/data/OS40000045	Boleycarrigeen (Ogham Site)	Wicklow	1	52.947012	-6.608898
194	http://lod.ogham.link/data/OS40000098	Donard (Ogham Site)	Wicklow	1	53.016944	-6.617778
195	http://lod.ogham.link/data/OS40000040	Baltinglass (Ogham Site)	Wicklow	1	52.966389	-6.624444

196 rows × 6 columns

SPARQL query to create DataFrame

Visualise the Data as Maps

```
# Filter rows with valid coordinates
df_with_coords = df.dropna(subset=['latitude', 'longitude'])

# Create a GeoDataFrame
gdf = gpd.GeoDataFrame(
    df_with_coords,
    geometry=[Point(xy) for xy in zip(df_with_coords['longitude'], df_with_coords['latitude'])],
    crs="EPSG:4326"
)

# Convert to Web Mercator for OSM basemap
gdf_mercator = gdf.to_crs(epsg=3857)

# Load Ireland boundary from GeoJSON
ireland_boundary = gpd.read_file(geojson_file)
ireland_boundary = ireland_boundary.to_crs(epsg=3857)
```

Map 1: Plot with points coloured by county

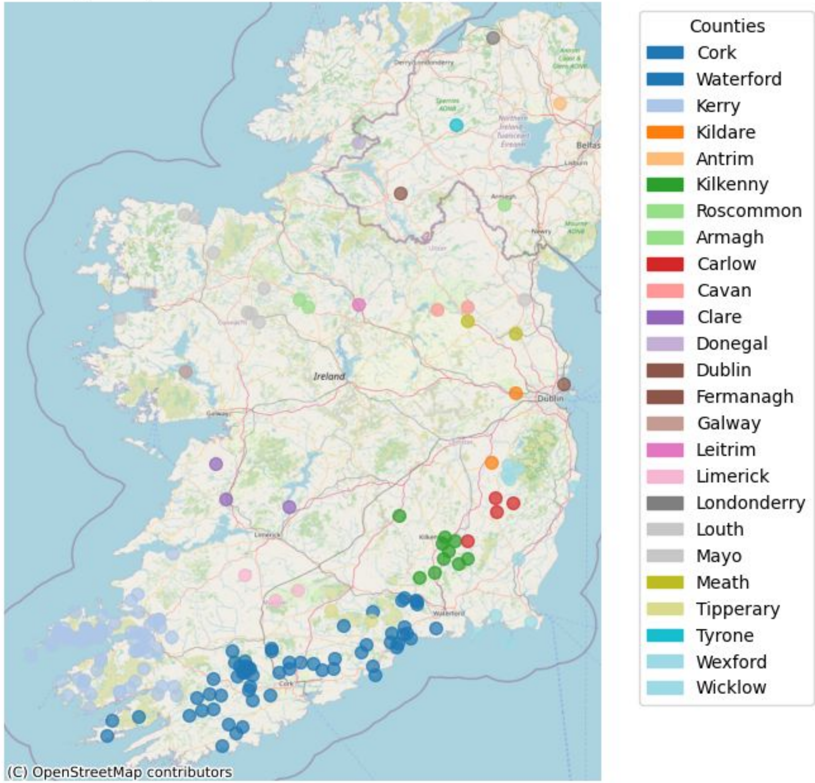
```
# Map 1: Plot with points coloured by county
fig, ax = plt.subplots(figsize=(12, 8))
unique_counties = gdf['county'].unique()
# Create a colormap with as many colours as unique counties
cmap = plt.get_cmap('tab20', len(unique_counties))
county_colors = {county: cmap(idx) for idx, county in enumerate(unique_counties)}

patches = []
for county, color in county_colors.items():
    county_data = gdf_mercator[gdf_mercator['county'] == county]
    county_data.plot(ax=ax, color=color, markersize=50, alpha=0.7)
    patches.append(Patch(color=color, label=county))  # Add patch for legend

ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik, zoom=8)
ax.set_axis_off()
plt.title("Map of Ogham Stone Sites Grouped by Counties")
plt.legend(handles=patches, title="Counties", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```

Map 2: Plot with point colours and sizes grouped/styled by stone count

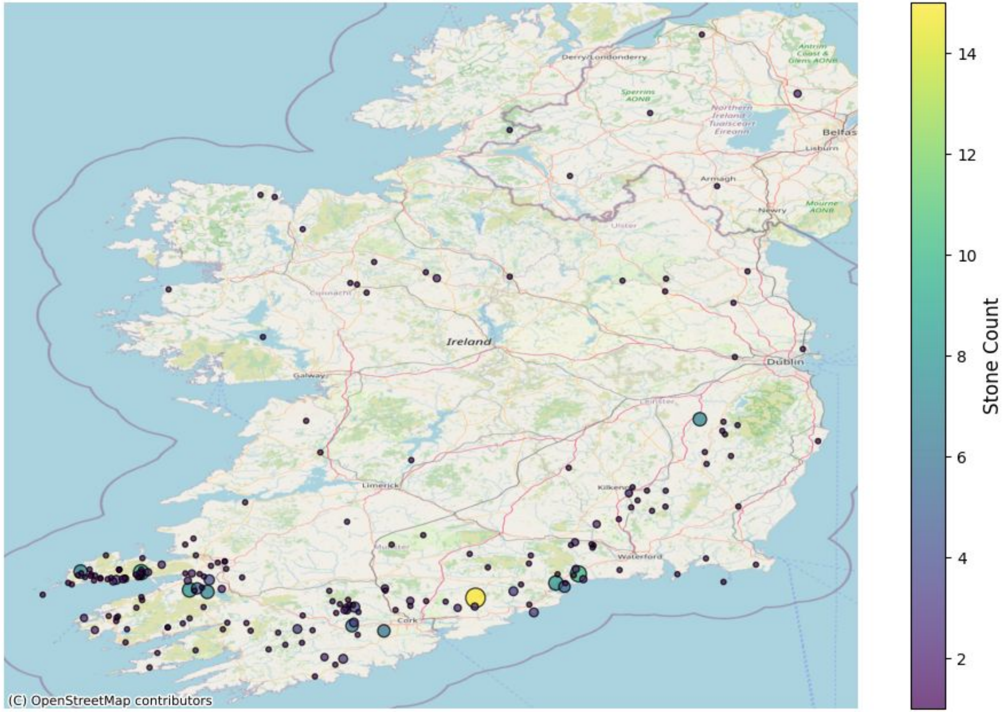
```
# Map 2: Plot with point colours and sizes grouped/styled by stone count.
fig, ax = plt.subplots(figsize=(12, 8))
# Extract x and y coordinates from the GeoDataFrame.
x = gdf_mercator.geometry.x
y = gdf_mercator.geometry.y

# Define a scaling factor for point sizes.
size_factor = 10
sizes = gdf_mercator["count"] * size_factor

# Plot the points using a continuous colormap (e.g. 'viridis') based on the stone count.
sc = ax.scatter(x, y, s=sizes, c=gdf_mercator["count"], cmap="viridis", alpha=0.7, edgecolor="k")
ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik, zoom=8)
ax.set_axis_off()
plt.title("Map of Ogham Stone Sites: Styled by Stone Count", fontsize=16)

# Add a colour bar with label.
cbar = plt.colorbar(sc, ax=ax)
cbar.set_label("Stone Count", fontsize=12)
plt.show()
```

Create Maps

Map of Ogham Stone Sites Grouped by Counties



Map of Ogham Stone Sites: Styled by Stone Count



Output

Campanian Ignimbrite

… sites in a Solid Pod to create diagrams / maps.

Define SPARQL query service

```python
from SPARQLWrapper import SPARQLWrapper, TURTLE
from rdflib import Graph, Namespace
from rdflib.namespace import RDF, RDFS
import os
from SPARQLWrapper import SPARQLWrapper, JSON
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
from shapely.geometry import Point
import contextily as ctx  # For adding OpenStreetMap basemaps
from matplotlib.patches import Patch
from scipy.stats import gaussian_kde
import numpy as np

# Function to query Solid Pod and retrieve data in TURTLE format
def querySolidPod(sparql_endpoint, query):
    sparql = SPARQLWrapper(sparql_endpoint)
    sparql.setQuery(query)
    sparql.setReturnFormat(TURTLE)  # Request Turtle format
    results = sparql.query().convert()
    return results
```

Define the SPARQL Query

```python
# SPARQL Query
solid_pod_query = """
SELECT ?item ?geo ?label ?ref ?spatialType {
 ?item a <http://fuzzy-sl.squirrel.link/ontology/Site> .
 ?item rdfs:label ?label.
 ?item <http://fuzzy-sl.squirrel.link/ontology/hasReference> ?ref .
 ?item <http://fuzzy-sl.squirrel.link/ontology/spatialType> ?spatialType .
 ?item <http://www.opengis.net/ont/geosparql#hasGeometry> ?item_geom .
 ?item_geom <http://www.opengis.net/ont/geosparql#asWKT> ?geo .
}
"""
```

Fetch Data and Convert to DataFrame

```python
sparql_endpoint = "https://fuzzy-sl.solidweb.org/campanian-ignimbrite-geo/ci_full.ttl"

# Query the endpoint
turtle_data = querySolidPod(sparql_endpoint, solid_pod_query)

# Parse Turtle data with rdflib
g = Graph()
g.parse(data=turtle_data, format="turtle")

# Extract namespaces
site_ns = Namespace("http://fuzzy-sl.squirrel.link/ontology/")
gs = Namespace("http://www.opengis.net/ont/geosparql#")

# Run the SPARQL query against the loaded Turtle data
results = g.query(solid_pod_query)

# Process the Results into a DataFrame
data = []
for row in results:
    spatial_type = str(row.spatialType).replace("http://fuzzy-sl.squirrel.link/ontology/", "")
    geo_str = str(row.geo)
    if geo_str.find("POINT(") != -1:
        geo_str = geo_str.replace("<http://www.opengis.net/def/crs/EPSG/0/4326>", "").replace("POINT(", "").replace(")", "")
    else:
        geo_str = geo_str.replace("<http://www.opengis.net/def/crs/EPSG/0/4326>", "")
        geo_str = geo_str.replace(")", "")
        geo_str = geo_str.replace(" POINT (", "")

    lon, lat = map(float, geo_str.split())

    data.append({
        "spatialType": spatial_type,
        "latitude": lat,
        "longitude": lon,
    })

# Convert the data to a pandas DataFrame
df = pd.DataFrame(data)
df
```

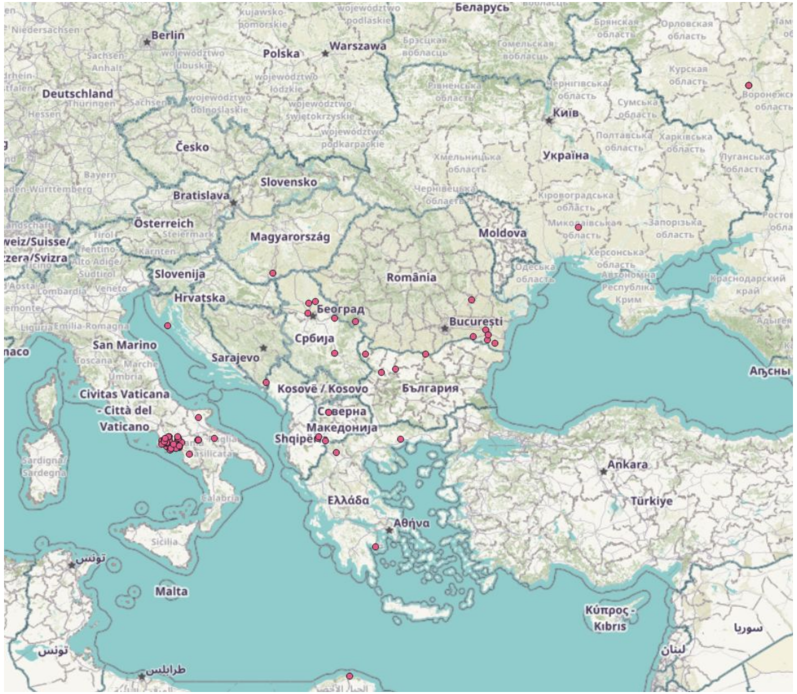
image: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

https://research-squirrel-engineers.github.io/jupyter-nb-lod/solidpod-ci

SPARQL query to create DataFrame

	spatialType	latitude	longitude
0	Sink	40.9489	14.3624
1	UnknownCategory	40.9285	14.6983
2	InhabitedPlace	40.7119	14.7680
3	InhabitedPlace	40.6993	14.5259
4	Cave	40.4956	15.2092
...	...	...	...
98	UnknownCategory	44.9032	20.2802
99	UnknownCategory	44.2462	27.9347
100	UnknownCategory	44.2462	27.9347
101	Lake	40.9167	21.0000
102	Lake	40.9167	21.0000

103 rows × 3 columns



SPARQL query to create DataFrame

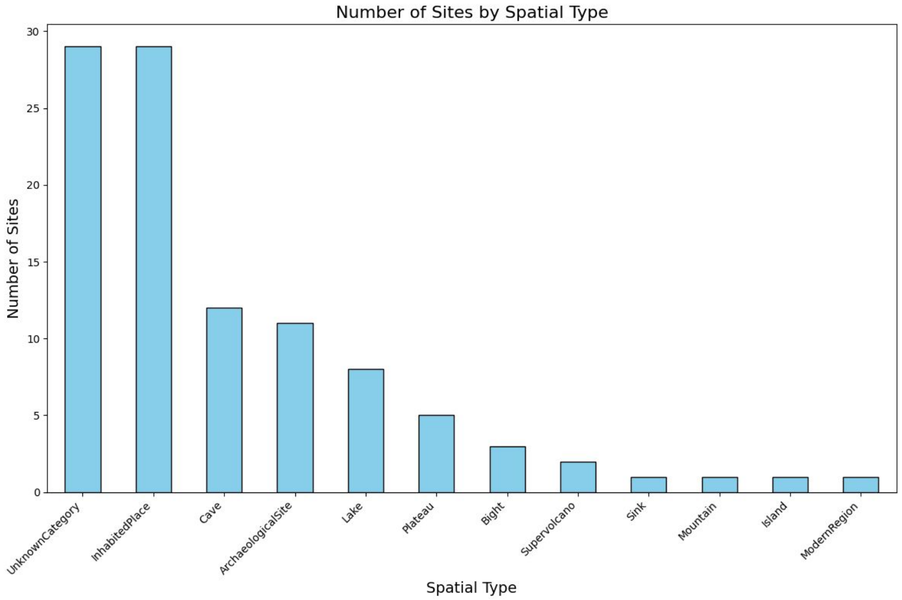
Visualise the Data in a bar chart

```python
# Check if DataFrame is populated
if not df.empty and 'spatialType' in df:
    # Plot 1: Number of Sites by Spatial Type
    plt.figure(figsize=(12, 8))
    df['spatialType'].value_counts().plot(kind='bar', color='skyblue', edgecolor='black')
    plt.title("Number of Sites by Spatial Type", fontsize=16)
    plt.xlabel("Spatial Type", fontsize=14)
    plt.ylabel("Number of Sites", fontsize=14)
    plt.xticks(rotation=45, ha="right")
    plt.tight_layout()
    plt.show()
else:
    print("No data retrieved or 'spatialType' column is missing.")
```

Visualise the Data in a map

```python
# Check if DataFrame is populated
if df.empty:
    print("No data retrieved from the query.")
else:
    # Filter rows with valid coordinates
    df_with_coords = df.dropna(subset=['latitude', 'longitude'])

    # Create a GeoDataFrame
    gdf = gpd.GeoDataFrame(
        df_with_coords,
        geometry=[Point(xy) for xy in zip(df_with_coords['longitude'], df_with_coords['latitude'])],
        crs="EPSG:4326"
    )

    # Convert to Web Mercator for OSM basemap
    gdf_mercator = gdf.to_crs(epsg=3857)

    # Plot points on the map
    fig, ax = plt.subplots(figsize=(12, 8))
    gdf_mercator.plot(ax=ax, color='red', markersize=50, alpha=0.7, label="CI sites")

    # Add OSM basemap
    ctx.add_basemap(ax, source=ctx.providers.OpenStreetMap.Mapnik)

    ax.set_axis_off()
    plt.title("Map of CI sites in Europe")
    plt.legend()
    plt.show()
```

Create Chart & Map


Number of Sites by Spatial Type
Number of Sites
0
5
10
15
20
25
30
UnknownCategory
InhabitedPlace
Cave
ArchaeologicalSite
Lake
Plateau
Bight
Supervolcano
Sink
Mountain
Island
ModernRegion
Spatial Type


Map of CI sites in Europe
CI sites
(C) OpenStreetMap contributors

Output

Conclusio

Squirrels as RSE network within the NFDI?

We as the **Research Squirrel Engineers Network** think,
we **can contribute** with our expertise and skills **to NFDI** ...

images: Florian Thiery CC BY 4.0, created using OpenAI's DALL·E

... with the SPARQL Unicorn Research Toolkit and Jupyter Python Minions!

Thanks to JuRSE and FZJ

JÜLICH
Forschungszentrum

BETTER SOFTWARE BETTER RESEARCH

DOI 10.5281/zenodo.14914287

Finis!

Thx!

Questions?

Florian Thiery M.Sc.
Research Squirrel Engineers Network
mail@fthiery.de
ORCID: 0000-0002-3246-3531

DOI 10.5281/zenodo.14914287