A Conceptual Modelling Approach to Visualising Linked Data

P.J. McBrien and A. Poulovassilis

CoopIS 2019

Motivation

There are numerous Linked Open Datasets published on the Web

Many possible data visualisations may be appropriate for a user's given exploration or analysis task over a dataset

Users may find it difficult to identify specific visualisations that meet their needs

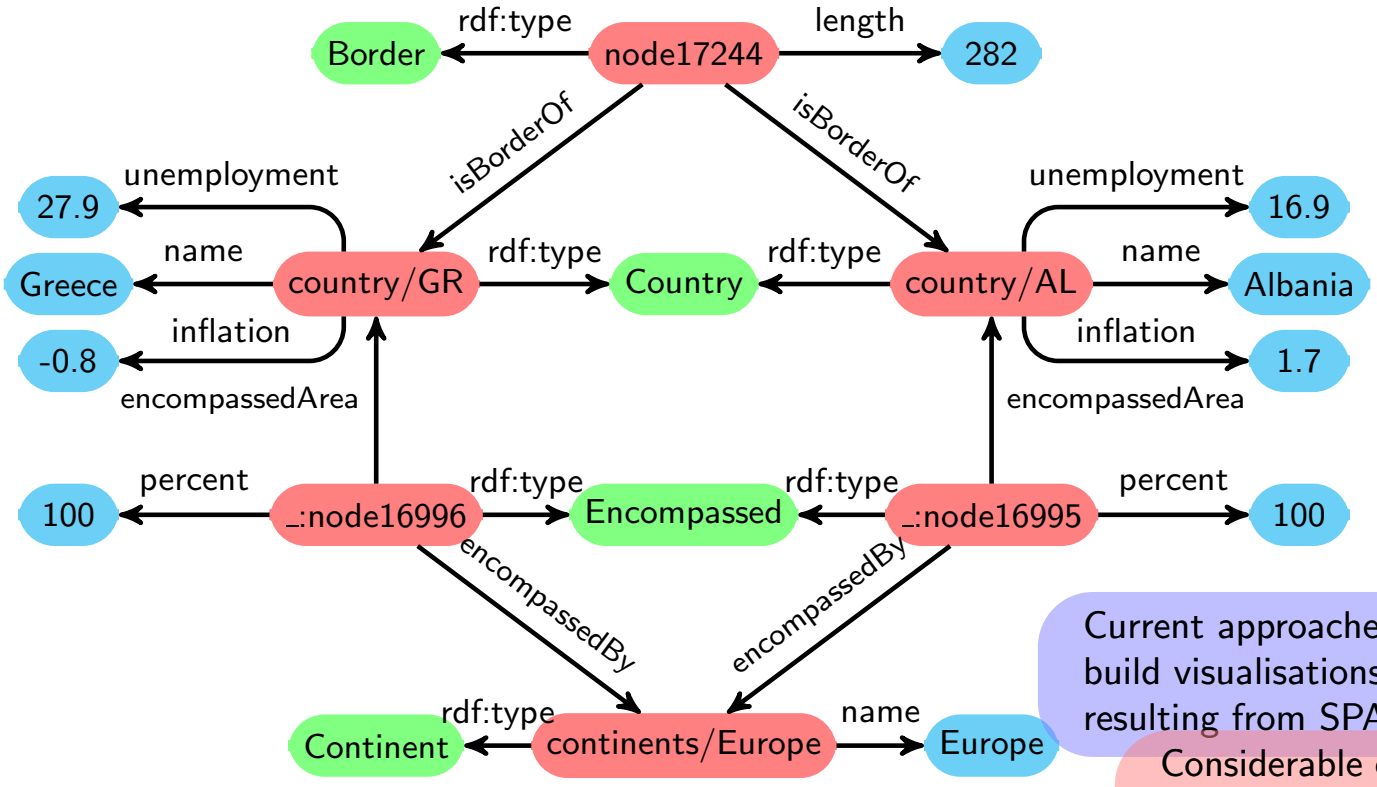
We address this by defining "visualisation patterns" expressed in OWL, semantically characterising groups of common data visualisations

We also define a SPARQL query template corresponding to each visualisation pattern

Users formulate SPARQL queries to extract the data to be visualised (directly, or through a visual query tool)

The set of OWL visualisation patterns and SPARQL query templates are used to analyse the data and the user's query, and to generate a focussed set of recommendations of possible visualisations

Motivating Example

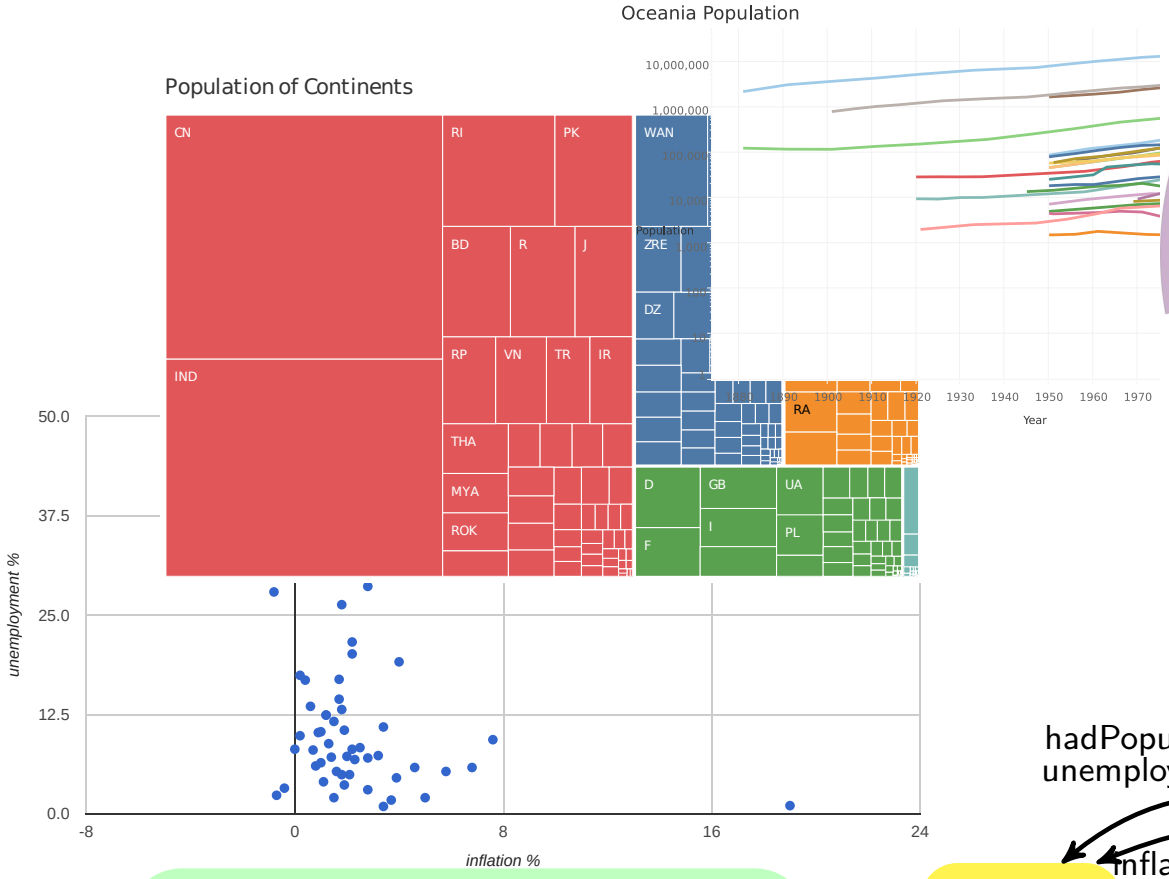


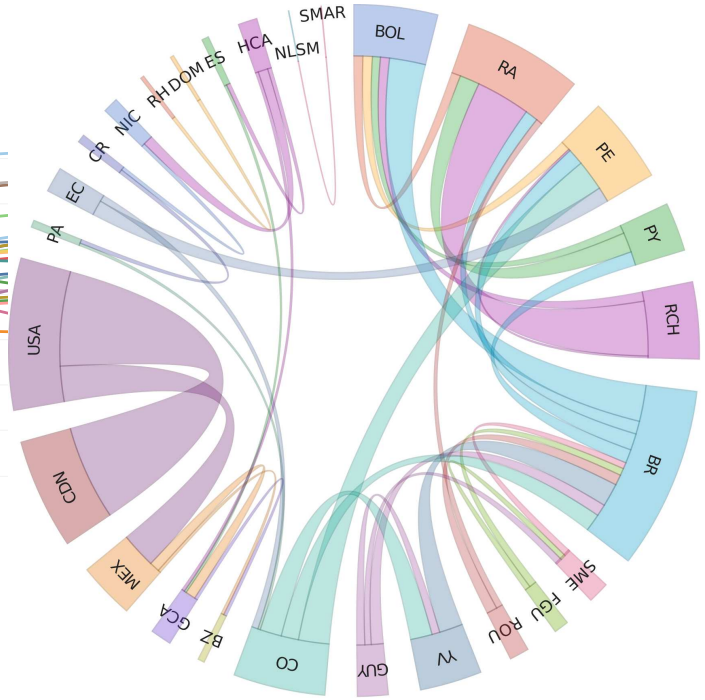
Current approaches require users to build visualisations from tabular data resulting from SPARQL queries.

Considerable effort is needed to understand the data and decide on which visualisation is used.

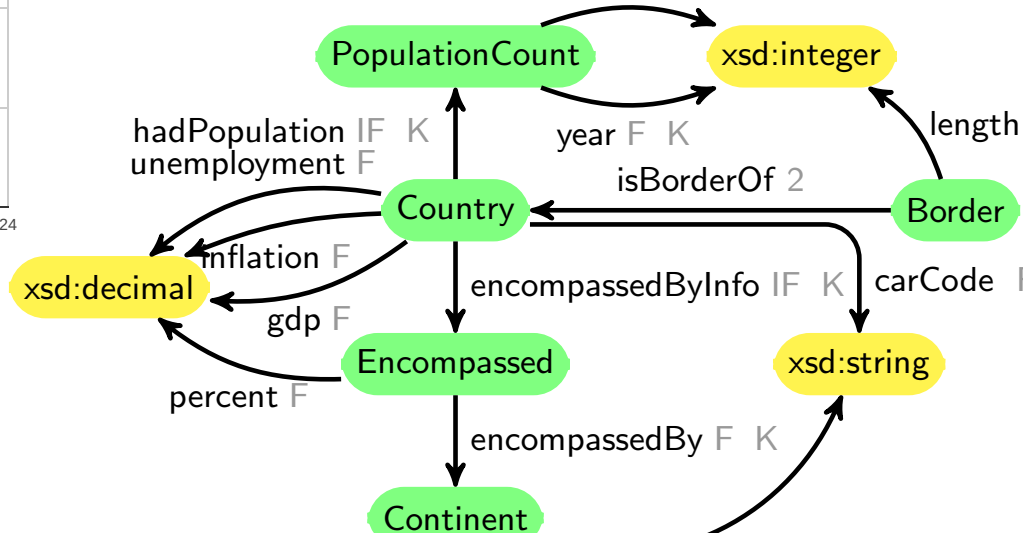
Our approach uses the RDF/OWL schema of the data to semi-automatically aid the user's choice of visualisations.

Motivating Example





Our approach uses the RDF/OWL schema of the data to semi-automatically aid the user's choice of visualisations.



Dimensions of a Visualisation

The elements of a visualisation are often classified in the Visualisation literature as:

- **marks** *e.g.* points, lines, areas
- **channels** *e.g.* colour, length, shape, coordinate, texture

In our approach, each functional data or object property is associated with a dimension of the visualisation.
Dimensions are of two types:

- **discrete dimensions** which map to a mark or a channel
- **scalar dimensions** which map to a channel

Discrete Dimension Example

continent.name={'Europe','Asia','Australia/Oceania','Africa','America'}

Dimensions of a Visualisation

The elements of a visualisation are often classified in the Visualisation literature as:

- **marks** *e.g.* points, lines, areas
- **channels** *e.g.* colour, length, shape, coordinate, texture

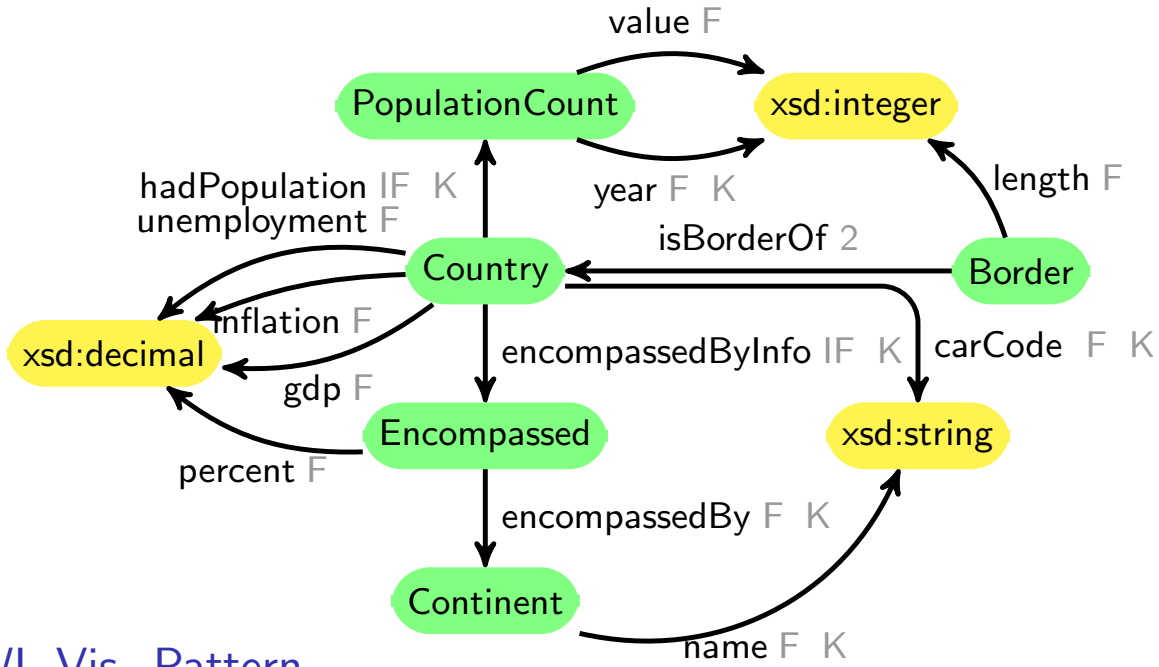
In our approach, each functional data or object property is associated with a dimension of the visualisation.
Dimensions are of two types:

- **discrete dimensions** which map to a mark or a channel
- **scalar dimensions** which map to a channel

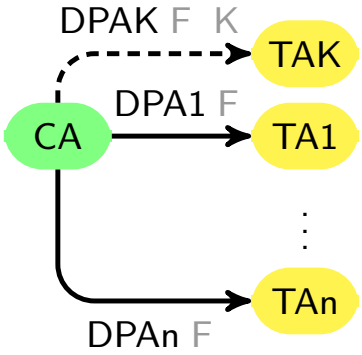
Scalar Dimension Example

borders.length=$\{0.34,1.20,3.2,4.40,6.30.9.00,9.60,\ldots,5150,6846,8893\}$

Pattern 1: Class with Key Data Property



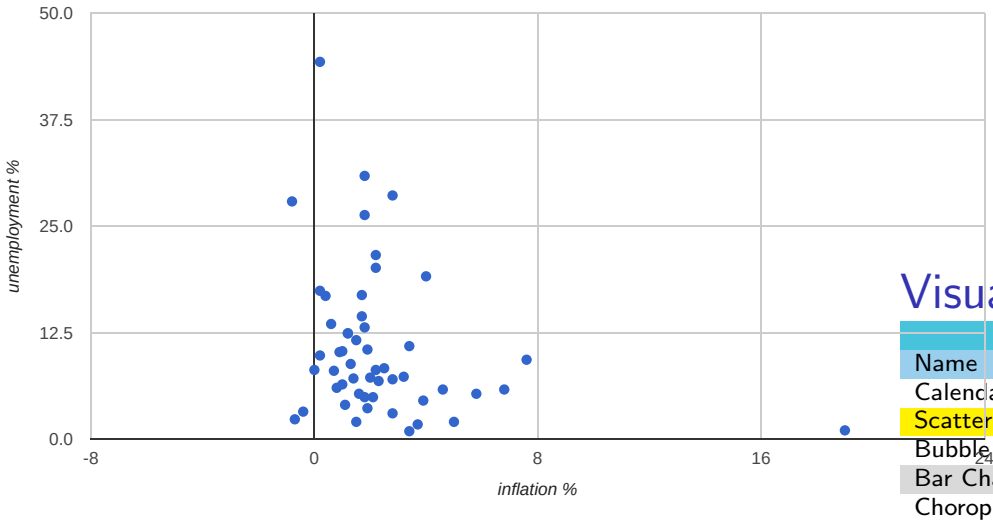
OWL Vis. Pattern



Instantiation

```
SELECT ?inflation ?unemployment WHERE {
    ?c  rdf:type :Country ;
        :inflation ?inflation ;
        :unemployment ?unemployment ;
}
```

Pattern 1: Class with Key Data Property



Visualisation Types

Visualisations			
Name	|CA|	mandatory	optional
Calendar Chart	1..*	TA1 temporal scalar	TAK,TA2 colour
Scatter Diagrams	1..*	TA1,TA2 scalar	TAK,TA3 colour
Bubble Charts	1..*	TA1,TA2,TA3 scalar	TAK,TA4 colour
Bar Chart	1..100	TAK, TA1 scalar	-
Choropleth Maps	1..*	TAK geographical, TA1 colour	TA2 texture
Word Clouds	1..*	TAK lexical, TA1 scalar	TA2 colour

SPARQL Template

```
SELECT ?TAK ?TA1 ?TAn WHERE {
    ?CA rdf:type :CA ;
        :DPK ?TAK ;
        :DPA1 ?TA1 ;
        ...
        :DPAn ?TAn .
}
```

Instantiation

```
SELECT ?inflation ?unemployment WHERE {
    ?c rdf:type :Country ;
       :inflation ?inflation ;
       :unemployment ?unemployment ;
}
```

Overall Matching Process

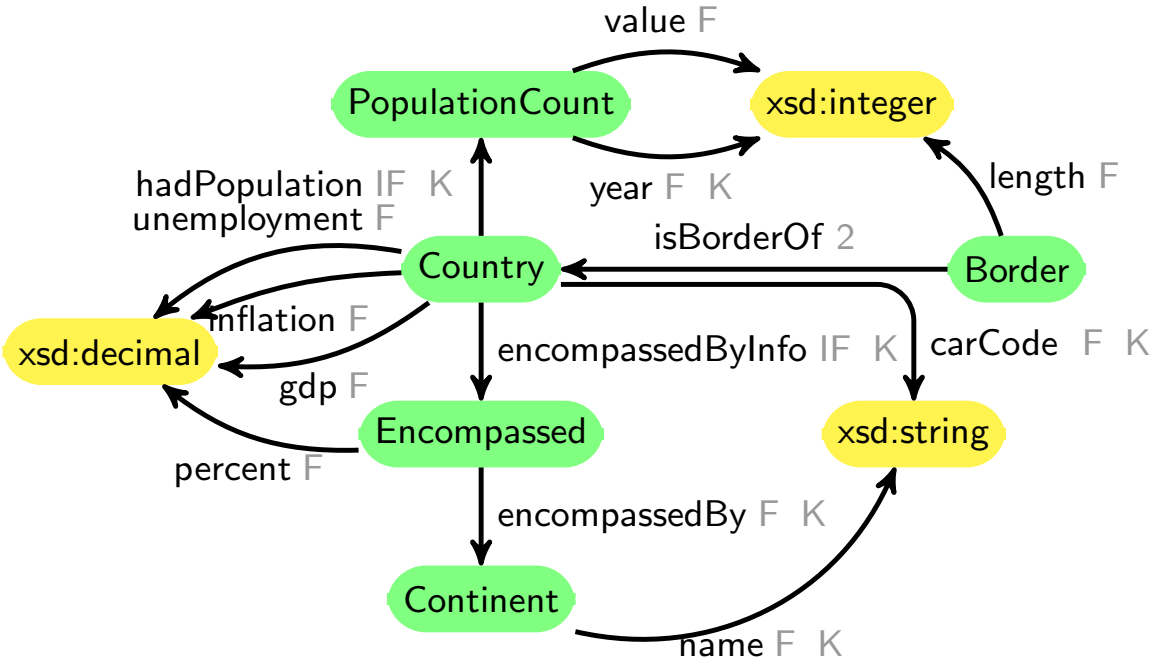
A system implementing our approach would match the user's query against the SPARQL query pattern corresponding to each group of visualisations

The system would then validate the corresponding OWL visualisation pattern(s) against the RDFS/OWL statements in the data relating to classes/properties mentioned in the user's query

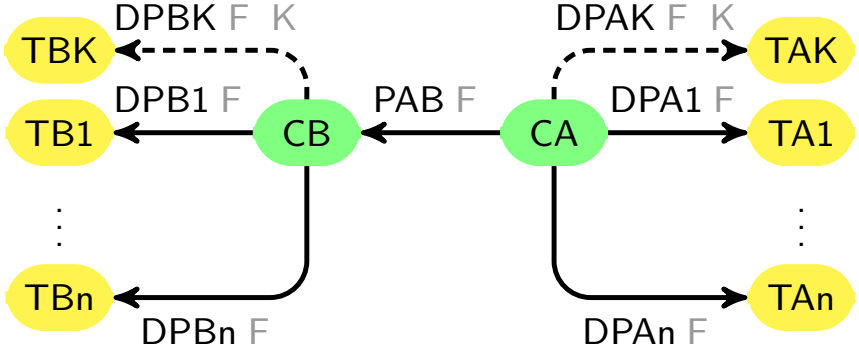
The matching group(s) of visualisations would be checked against the data for the additional constraints relating to individual visualisations

The matching visualisations would be offered to the user as possible alternatives for generating their visualisation

Pattern 2: Two Classes & Functional Property



OWL Vis. Pattern

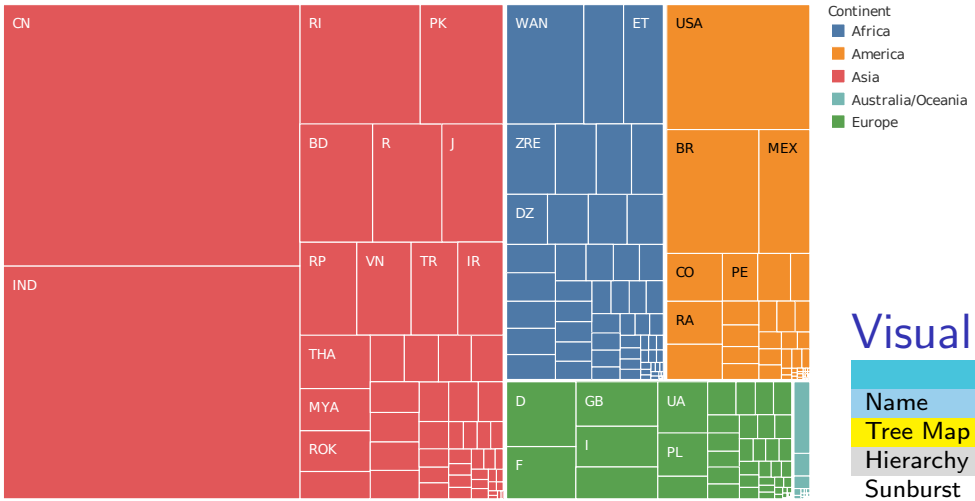


Instantiation

```
SELECT ?continent ?carcode ?population
WHERE {
    ?c  rdf:type :Country ;
        :carCode ?carcode ;
        :population ?population ;
        :encompassedByInfo ?en .
    ?en :encompassedBy ?con;
        :percent ?percent.
    ?con rdf:type :Continent ;
         :name ?continent .
    FILTER ( ?percent>50 )
}
```

Pattern 2: Two Classes & Functional Property

Population of Continents



Visualisation Types

Visualisations				
Name	|CB|	|CA PAB CB|	mandatory	optional
Tree Map	1..20	1..100	TAK,TBK,TA1 scalar	TB1 colour,TA2 colour
Hierarchy Tree	1..100	1..100	TAK,TBK	TA1 colour,TB1 colour
Sunburst	1..20	1..20	TAK,TBK,TA1 scalar	TA1 colour,TB1 colour
Circle Packing	1..20	1..20	TAK,TBK,TA1 scalar	TA1 colour,TB1 colour

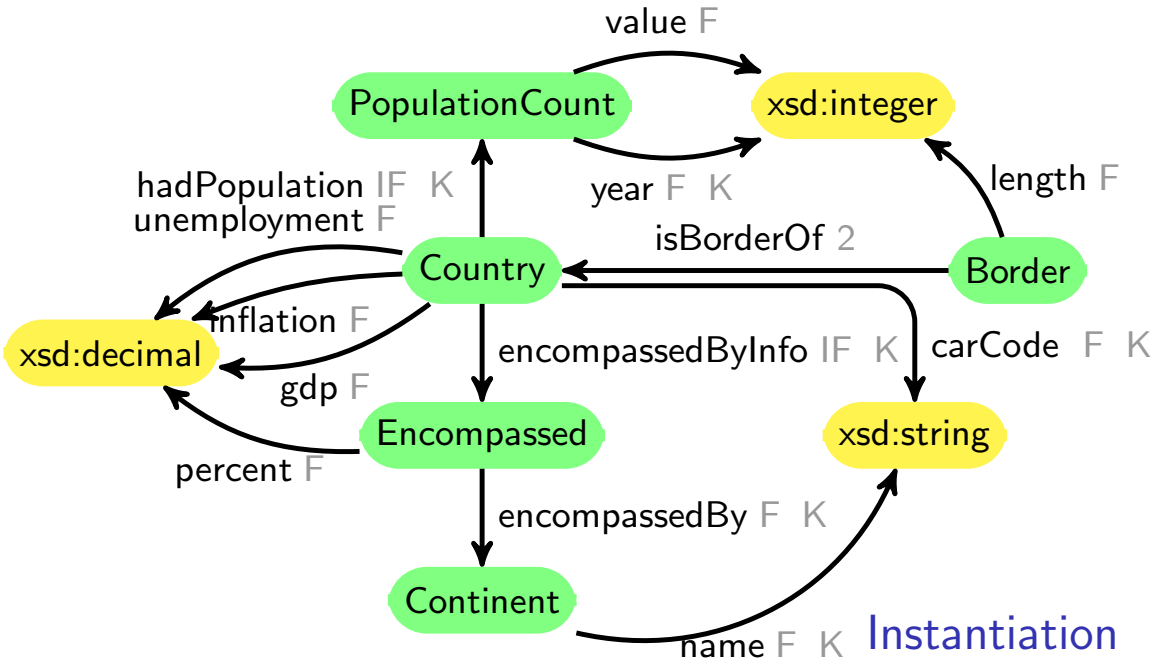
SPARQL Template

```
SELECT ?TBK ?TAK ?TA1 ?TAn ?TB1 ?TBn WHERE {
    ?CA rdf:type :CA ;
        :DPAK ?TAK ;
        :DPA1 ?TA1 ;
        ...
        :DPAn ?TAn .
    ?CB rdf:type :CB ;
        :DPBK ?TBK ;
        :DPB1 ?TB1 ;
        ...
        :DPBn ?TBn .
    ?CA :PAB ?CB .
}
```

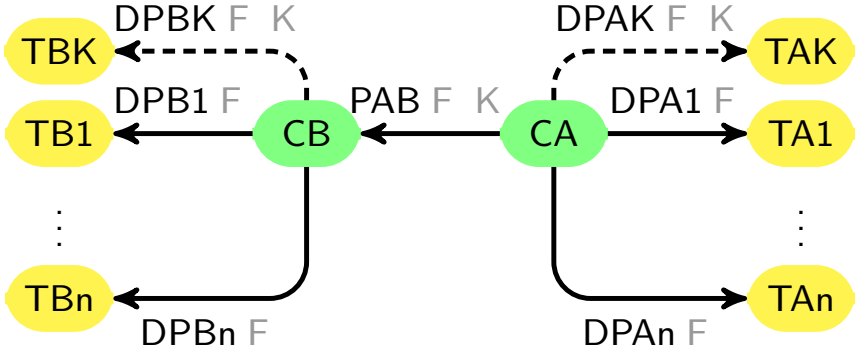
Instantiation

```
SELECT ?continent ?carcode ?population
WHERE {
    ?c  rdf:type :Country ;
        :carCode ?carcode ;
        :population ?population ;
        :encompassedByInfo ?en .
    ?en :encompassedBy ?con;
        :percent ?percent.
    ?con rdf:type :Continent ;
         :name ?continent .
    FILTER ( ?percent>50 )
}
```

Pattern 3: Two Classes & Key Functional Property



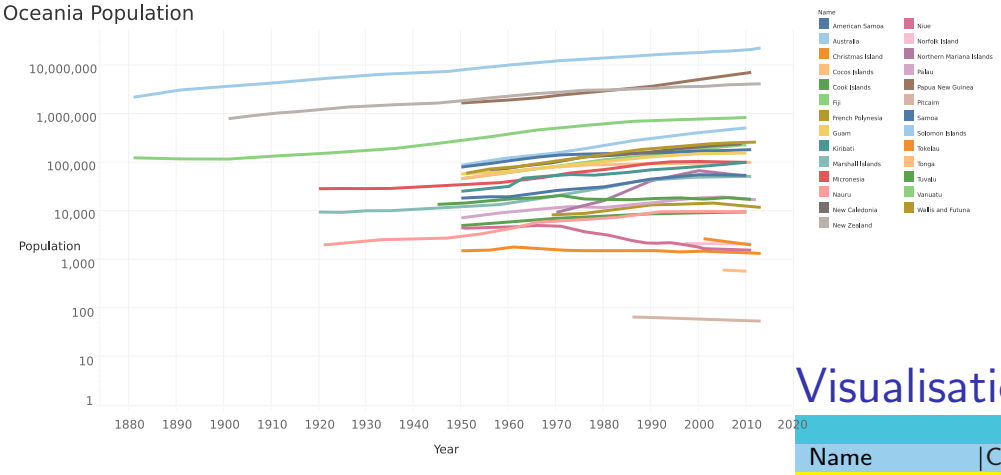
OWL Vis. Pattern



Instantiation

```
SELECT ?country ?year ?population
WHERE {
      ?c   rdf:type :Country ;
           :name ?country ;
           :encompassedByInfo ?en .
      ?py  rdf:type :PopulationCount ;
           :year ?year;
           :value ?population .
      ?c   :hadPopulation ?py .
      # Filter conditions
      ?en :encompassedBy ?con .
      ?con rdf:type :Continent ;
           :name "Australia/Oceania" .
}
```

Pattern 3: Two Classes & Key Functional Property



Visualisation Types

Visualisations of key functional properties					
Name	|CB|	|CA|	complete	mandatory	optional
Line	1..20	1..*	no	TAK scalar,TBK,TA1 scalar	TA2 scalar,TA3/TB1 colour
Spider	3..10	1..20	yes	TAK,TBK,TA1 scalar	TA2/TB1 colour
Stacked Bar	1..100	1..20	yes	TAK colour,TBK,TA1 scalar	TA2 texture
Grouped Bar	1..20	1..20	no	TAK colour,TBK,TA1 scalar	TA2 texture

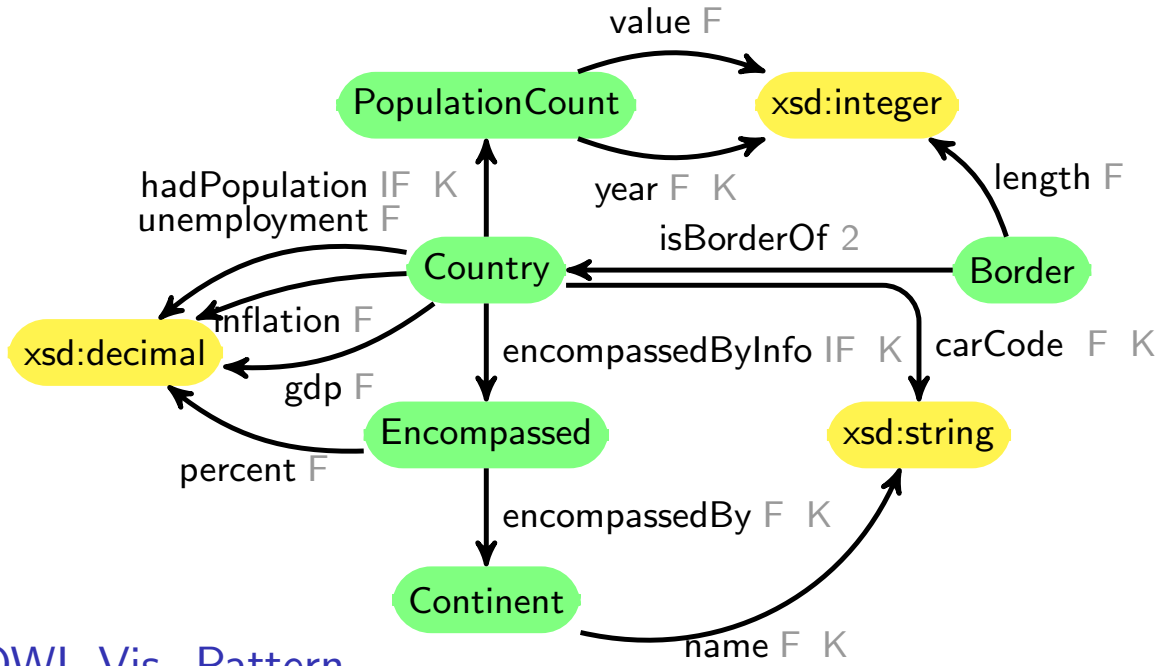
SPARQL Template

```
SELECT ?TBK ?TAK ?TA1 ?TAn ?TB1 ?TBn WHERE {
    ?CA rdf:type :CA ;
        :DPAK ?TAK ;
        :DPA1 ?TA1 ;
        ...
        :DPAn ?TAn .
    ?CB rdf:type :CB ;
        :DPBK ?TBK ;
        :DPB1 ?TB1 ;
        ...
        :DPBn ?TBn .
    ?CA :PAB ?CB .
}
```

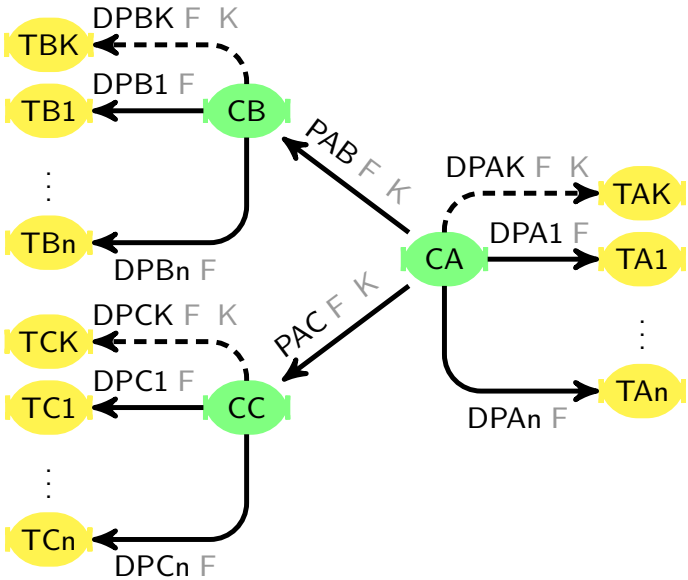
Instantiation

```
SELECT ?country ?year ?population
WHERE {
    ?c  rdf:type :Country ;
        :name ?country ;
        :encompassedByInfo ?en .
    ?py rdf:type :PopulationCount ;
        :year ?year;
        :value ?population .
    ?c  :hadPopulation ?py .
    # Filter conditions
    ?en :encompassedBy ?con .
    ?con rdf:type :Continent ;
        :name "Australia/Oceania" .
}
```

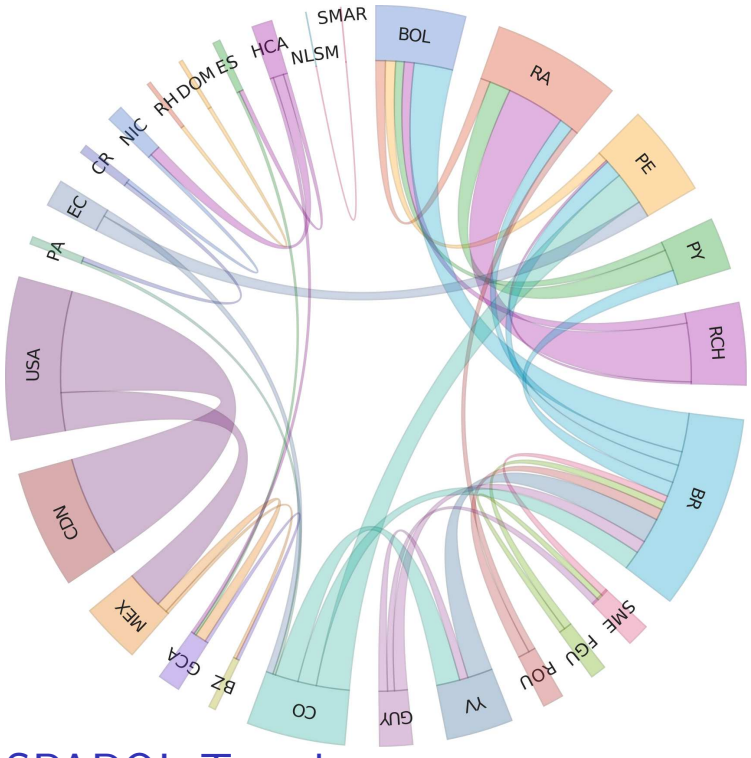
Pattern 4: Three Classes & Functional Properties



OWL Vis. Pattern



Pattern 4: Three Classes & Functional Properties



SPARQL Template

```
SELECT ?TBK ?TCK ?TA1 ?TAn WHERE {
    ?CA rdf:type :CA ;
        :PAB ?CB ;
        :PAC ?CC ;
        :DPA1 ?TA1 ;
        ...
        :DPAn ?TAn .
    ?CB rdf:type :CB ;
        :DPBK ?TBK .
    ?CC rdf:type :CC ;
        :DPCK ?TCK .
}
```

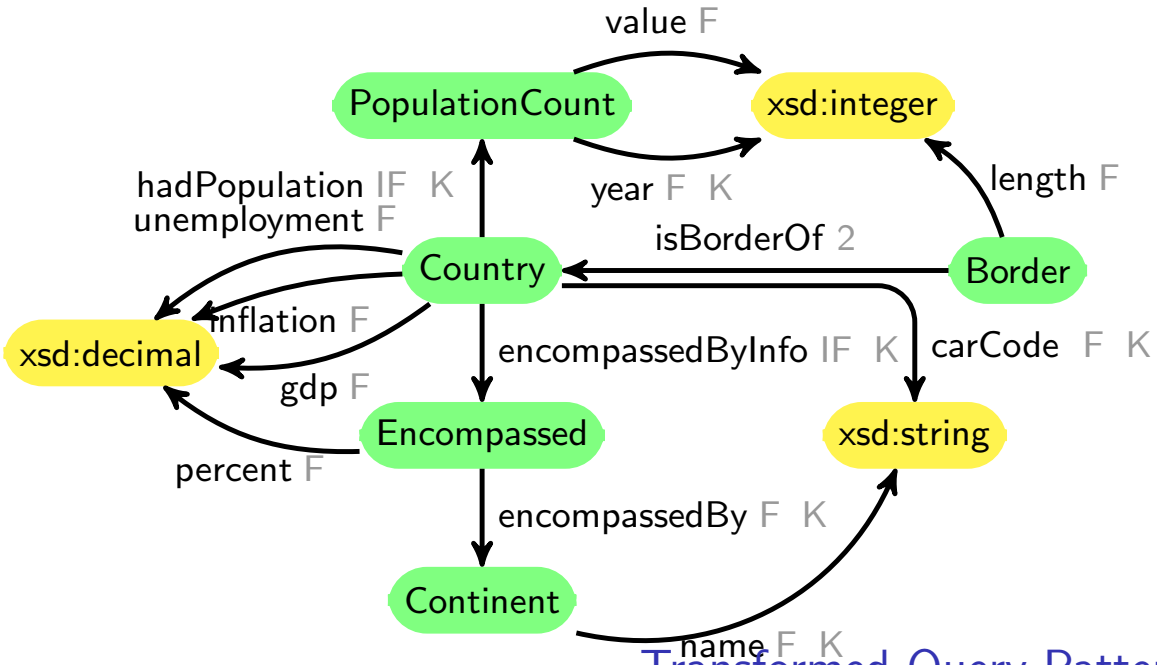
Visualisation Types

Visualisations					
Name	|CB|	|CC|	reflexive	mandatory	optional
Sankey	1..20	1..20	no	TA1 scalar	TA2 colour
Network Chart	1..1000	1..1000	yes	-	TAK,TA1 colour
Chord	1..100	1..100	yes	-	TA1 size,TA2 colour
Heatmap	1..100	1..100	yes	TA1 colour	TA2 texture

Instantiation

```
SELECT ?country1 ?country2 ?length WHERE {
    ?b rdf:type :Border ;
       :isBorderOf ?c1 ;
       :isBorderOf ?c2 ;
       :length ?length .
    ?c1 rdf:type :Country ;
        :carCode ?country1 .
    ?c2 rdf:type :Country ;
        :carCode ?country2 .
    # Filter conditions
    FILTER (?country1<?country2)
}
```

Transformations: Functional Subqueries



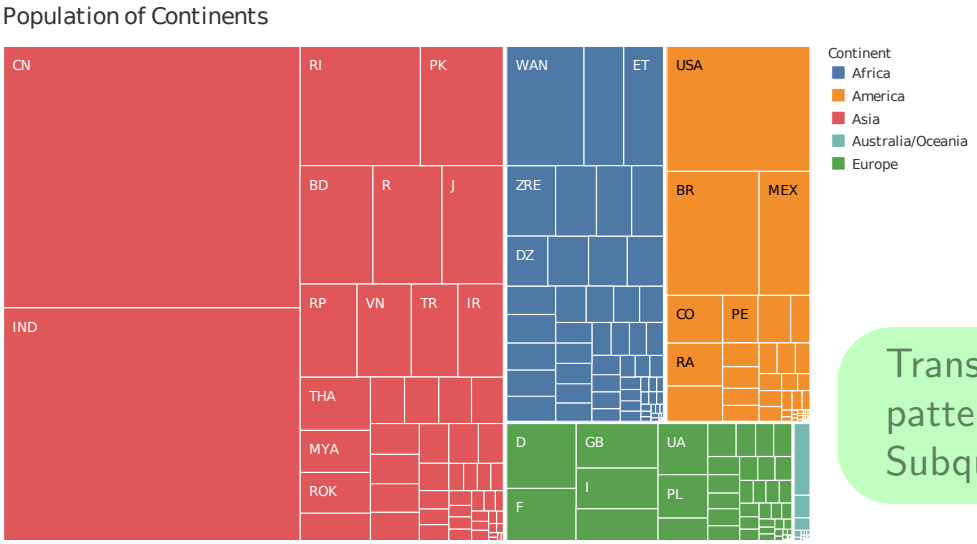
User Query Pattern

```
SELECT ?A ?TA1 ?TA2 ... ?B ?TB1 ?TB2 ...
WHERE {
    ?A  rdf:type :CA ;
        :DPA1 ?TA1 ;
        :DPA2 ?TA2 ;
        ... .
    ?B  rdf:type :CB ;
        :DPB1 ?TB1 ;
        :DPB2 ?TB2 ;
        ... .
    ?A  ...
    ?B  ...
}
```

Transformed Query Pattern

```
SELECT ?A ?TA1 ?TA2 ... ?B ?TB1 ?TB2 ...
WHERE {
    ?A  rdf:type :CA ;
        :DPA1 ?TA1 ;
        :DPA2 ?TA2 ;
        ... .
    ?B  rdf:type :CB ;
        :DPB1 ?TB1 ;
        :DPB2 ?TB2 ;
        ... .
    SELECT ?A ?B
    WHERE {
        ?A  ...
        ?B  ...
    }
}
```

Transformations: Functional Subqueries



Transformed query matches query visualisation pattern 2
Subquery represents a functional object property

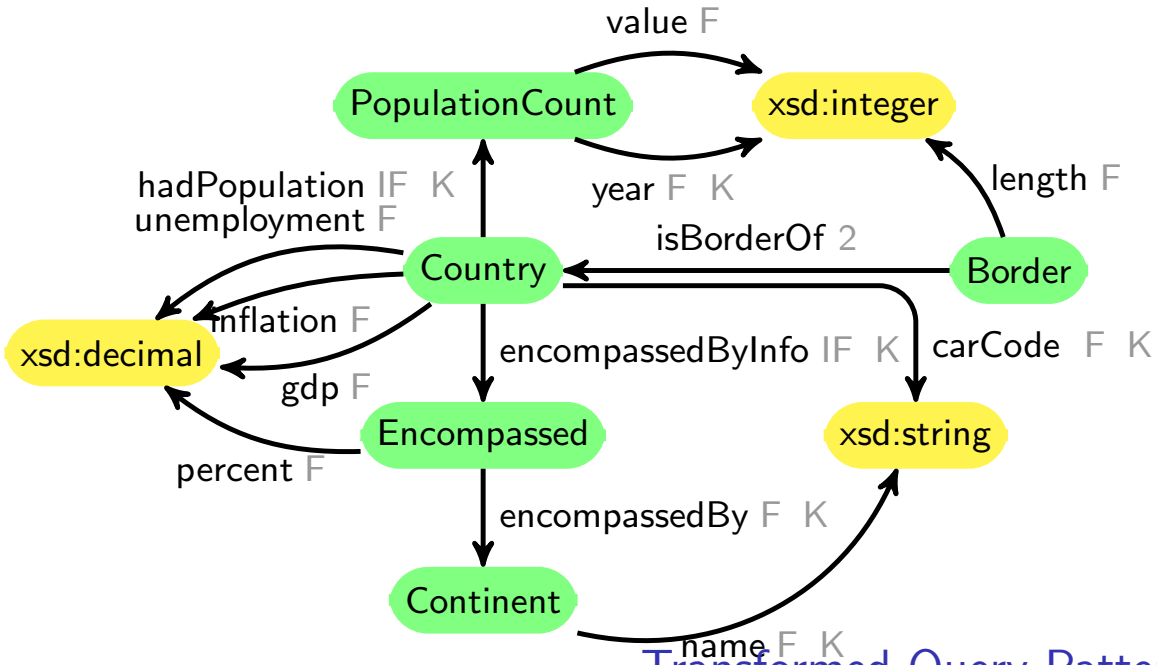
User Query

```
SELECT ?continent ?carcode ?population
WHERE {
      ?c   rdf:type :Country ;
           :carCode ?carcode ;
           :population ?population ;
           :encompassedByInfo ?en .
      ?en :encompassedBy ?con;
           :percent ?percent.
      ?con rdf:type :Continent ;
            :name ?continent .
      FILTER ( ?percent>50 )
}
```

Transformed Query

```
SELECT ?continent ?carcode ?population
WHERE {
      ?c   rdf:type :Country ;
           :carCode ?carcode ;
           :population ?population .
      ?con rdf:type :Continent ;
            :name ?continent .
      SELECT ?c ?con
      WHERE {
          ?c   :encompassedByInfo ?en .
          ?en :encompassedBy ?con;
               :percent ?percent.
          FILTER ( ?percent>50 )
      }
}
```

Transformations: Denormalisation



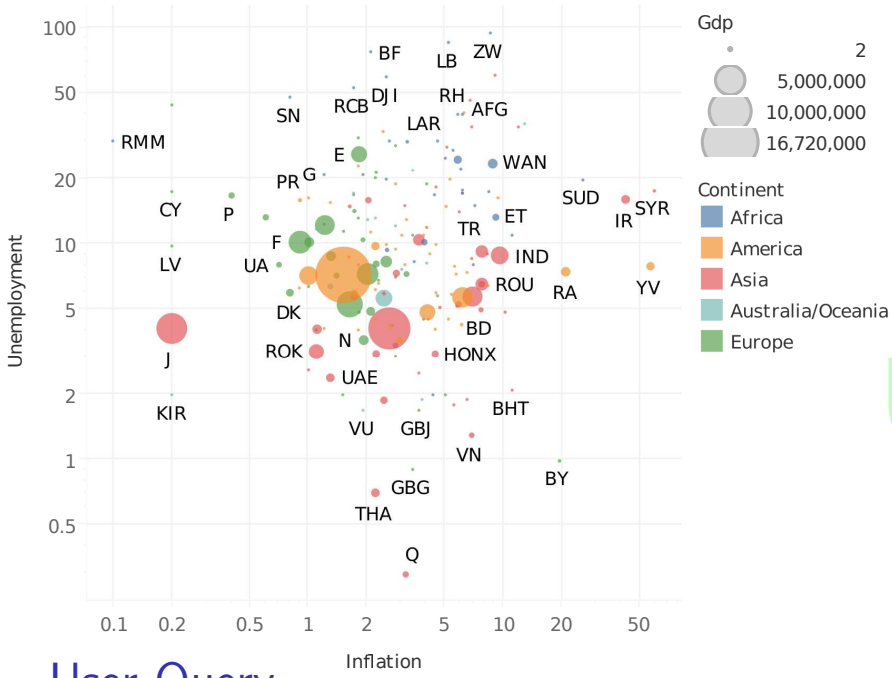
User Query Pattern

```
SELECT ?A ?TA1 ?TA2 ?TB1
WHERE {
    ?A  rdf:type :CA ;
        :DPA1 ?TA1 ;
        :DPA2 ?TA2 ;
        ... .
    ?B  rdf:type :CB ;
        :DPB1 ?TB1 ;
        :DPB2 ?TB2 ;
        ... .
    ?A  ...
    ?B  ...
}
```

Transformed Query Pattern

```
SELECT ?A ?TA1 ?TA2 ... ?B ?TB1 ?TB2
WHERE {
    ?A  rdf:type :CA ;
        :DPA1 ?TA1 ;
        :DPA2 ?TA2 ;
        ... .
    SELECT ?A ?TB1 ?TB2
    WHERE {
        ?B  rdf:type :CB ;
            :DPB1 ?TB1 ;
            :DPB2 ?TB2 ;
            ... .
        ?A  ...
        ?B  ...
    }
}
```

Transformations: Denormalisation



Transformed query matches query visualisation pattern 1
Subquery represents a functional data property

User Query

```
SELECT ?carcode ?inf ?unemployment
       ?continent ?gdp
WHERE {
    ?c  rdf:type :Country ;
        :carCode ?carcode ;
        :inflation ?inf;
        :unemployment ?unemployment;
        :gdp ?gdp ;
        :encompassedByInfo ?en .
    ?en :encompassedBy ?con;
        :percent ?percent.
    ?con :name ?continent .
    FILTER ( ?percent>50 )
}
```

Transformed Query

```
SELECT ?carcode ?inf ?unemployment
       ?continent ?gdp
WHERE {
    ?c  rdf:type :Country ;
        :carCode ?carcode ;
        :inflation ?inf;
        :unemployment ?unemployment;
        :gdp ?gdp .
    SELECT ?c ?continent
    WHERE {
        ?c  :encompassedByInfo ?en .
        ?en :encompassedBy ?con;
            :percent ?percent.
        ?con :name ?continent .
        FILTER ( ?percent>50 )
    }
}
```

Conclusions

We have proposed a conceptual modelling approach to matching linked data and visualisations, based on
(i) OWL visualisation patterns
(ii) corresponding SPARQL query templates

These underpin a process of automatic recommendation of a more focussed set of possible visualisations to the user

No previous work provides conceptual abstraction of groups of visualisations using OWL, nor a recommendation process to visualise Linked Data as we have described here

Future work includes prototype implementation and empirical evaluation, scalability investigation, and user customisation features

We view our approach as being part of a broader set of solutions to aid users in formulating queries and exploring Linked Data, e.g. combined with browsing/exploration, faceted search, structural summaries