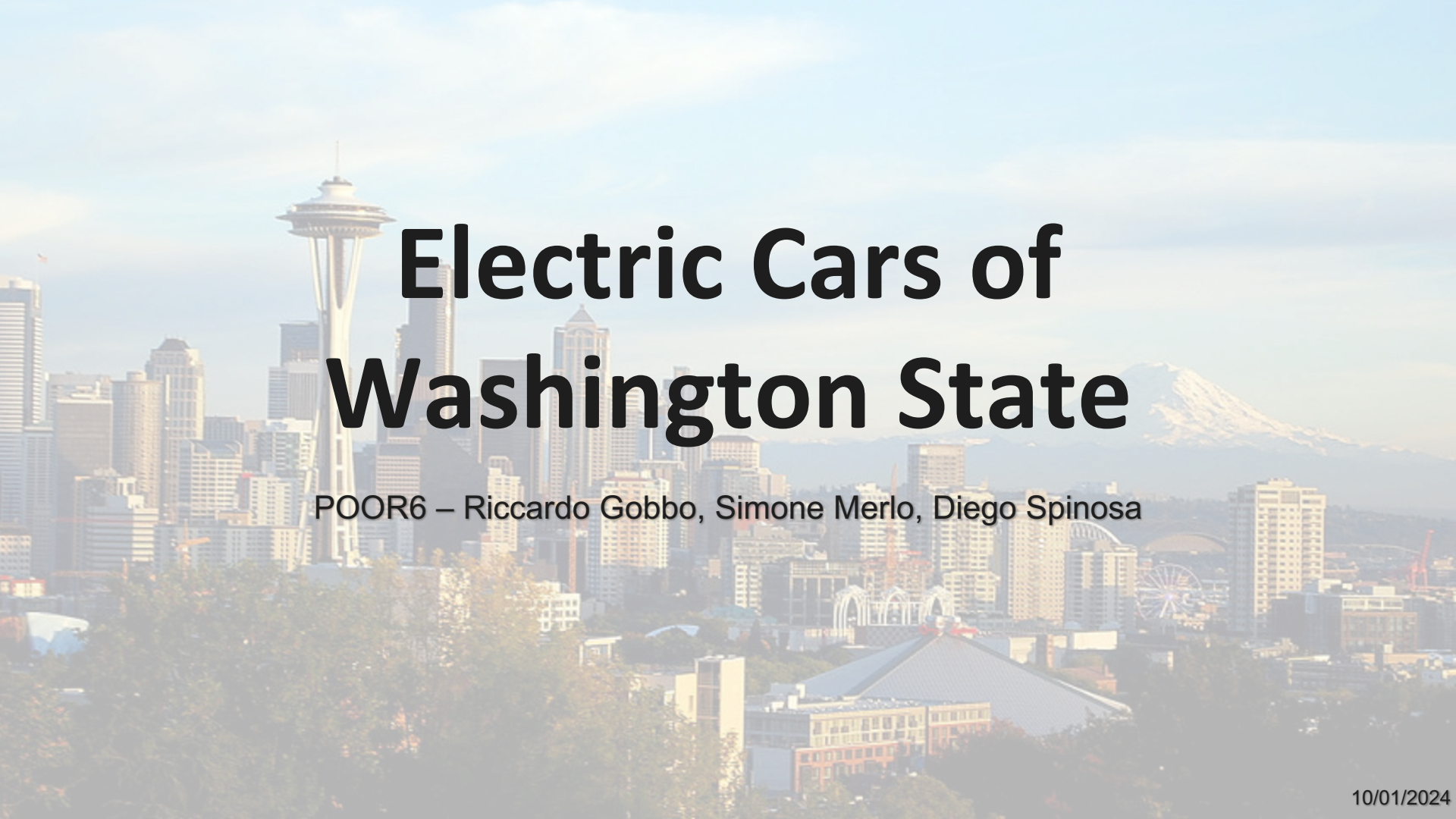


Electric Cars of Washington State

POOR6 – Riccardo Gobbo, Simone Merlo, Diego Spinosa

10/01/2024

Outline

01
Domain & Data Sources

Domain, goals and
data sources

02
Ontology

Domain model

03
Queries & Data Analysis

Queries, heatmaps
and graphs

04
Future Work

Future work &
possible expansion

01
Domain & Data Sources

Domain

Electric Cars

Plug-In and Battery cars registered in the state of Washington

Charging Stations

Private and Public Stations to recharge electric vehicles

Adjusted Gross Income

Adjusted Gross Income (AGI) for each ZIP Code

ZIP Codes

ZIP Codes and related cities and counties

Goal

The goal of our work is to provide a knowledge graph to evaluate the electric vehicles accessibility by highlighting the **correlation** between:

- The **electric cars**
- The **charging stations**
- The **average income**

of the people living in the state of Washington, focusing on the different territorial divisions.

Data Sources

Electric Cars

Plug-In and Electric (BEV) cars

2 possible sources:

- Kaggle
- data.wa.gov

AGI

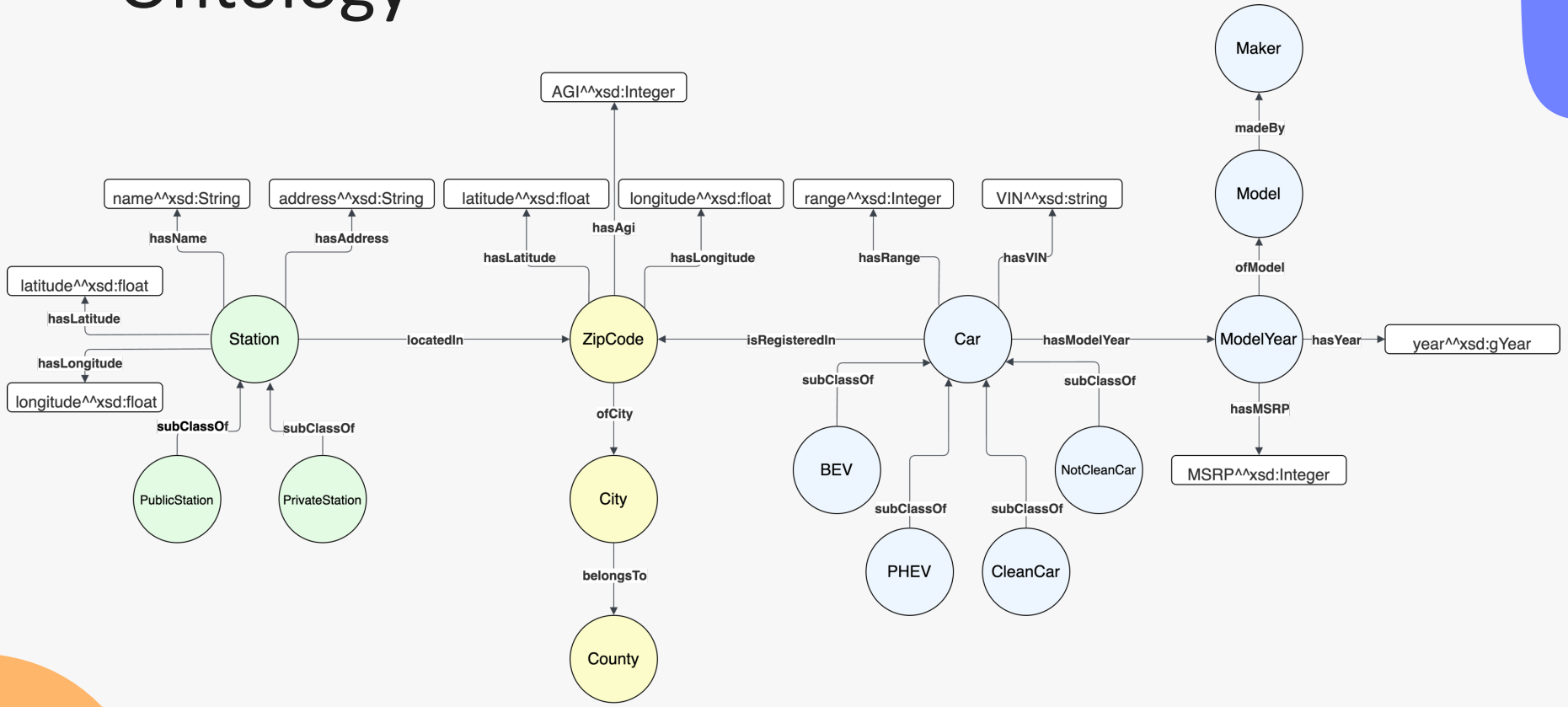
Adjusted Gross Income for each ZIP Code

Charging Station

Data about Public and Private station to charge electric vehicles

02
Ontology

Ontology

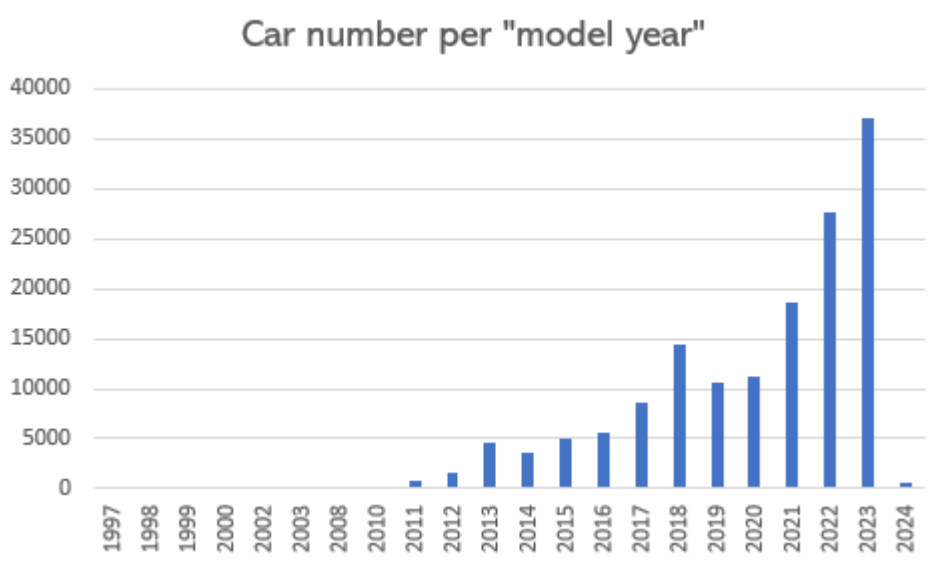

Maker
madeBy
Model
ofModel
ModelYear
hasYear
year^^xsd:gYear
hasMSRP
MSRP^^xsd:Integer
AGI^^xsd:Integer
name^^xsd:String
address^^xsd:String
latitude^^xsd:float
longitude^^xsd:float
range^^xsd:Integer
VIN^^xsd:string
hasName
hasAddress
hasAgi
hasLatitude
hasLongitude
hasRange
hasVIN
latitude^^xsd:float
hasLatitude
hasLongitude
longitude^^xsd:float
Station
locatedIn
ZipCode
isRegisteredIn
Car
hasModelYear
subClassOf
subClassOf
PublicStation
PrivateStation
ofCity
City
belongsTo
County
BEV
PHEV
CleanCar
NotCleanCar
subClassOf
subClassOf
subClassOf
subClassOf

03
Queries & Data Analysis

Cars Over Time

How many electric cars are registered for each model-year?

- Indicator of EV adoption through time.



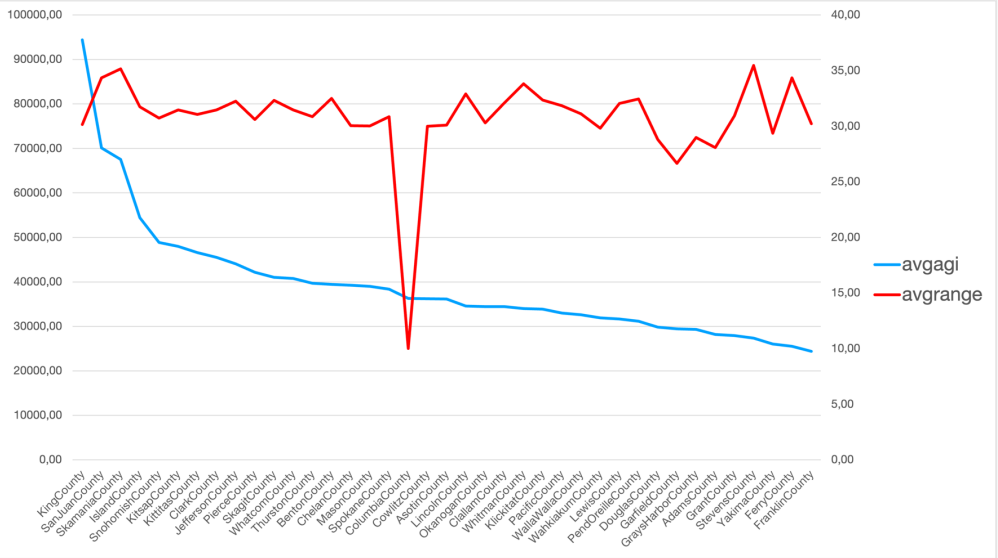

```
SELECT ?year (COUNT(?car) as ?amount) WHERE{
    ?car elec:hasModelYear ?moye .
    ?moye elec:hasYear ?year .
}
GROUP BY ?year
```

PHEV Cars Range and AGI

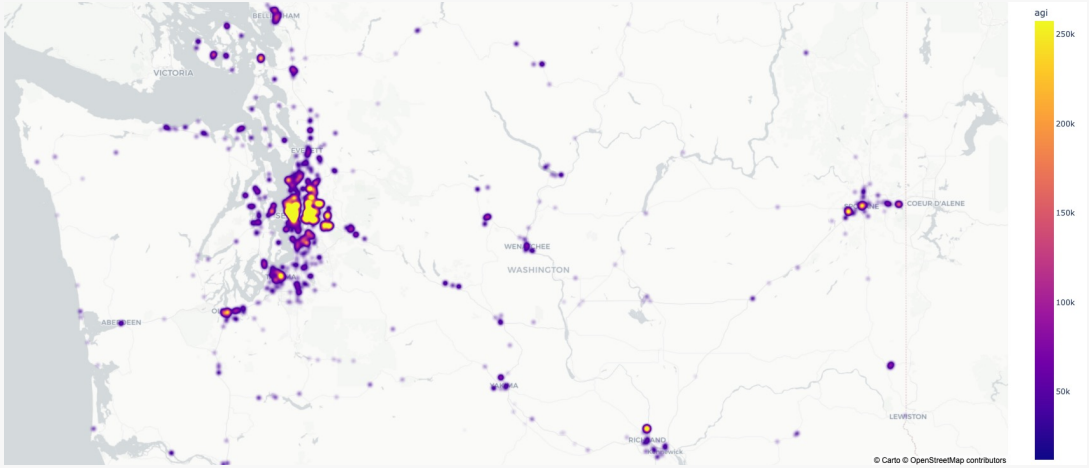
What is the average range of PHEV cars and average AGI for each county?

- Do, on average, richer regions buy longer-range hybrid vehicles?

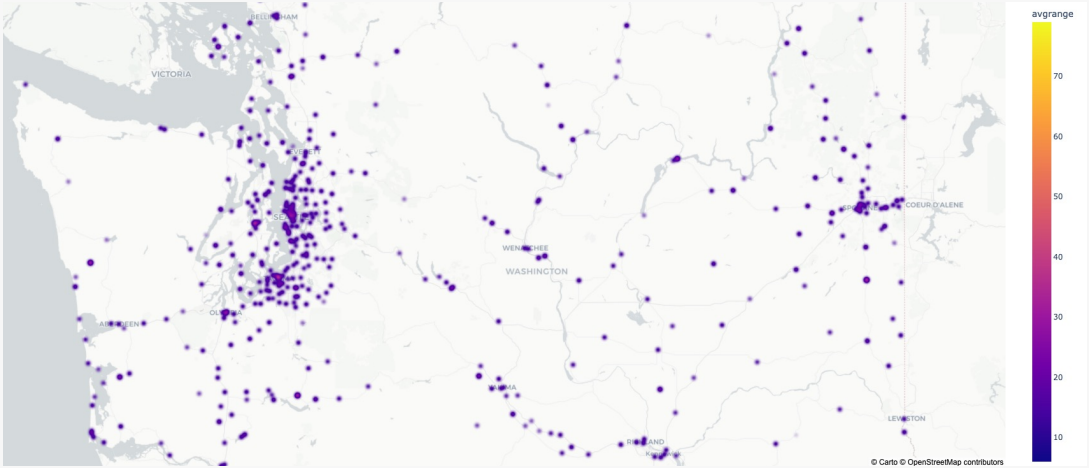
```
select ?county (AVG(?avgagi) AS ?avgagi) (AVG(?range) AS ?avgrange) where {
    ?county a elec:County.
    ?city elec:belongsTo ?county.
    ?zipcode elec:ofCity ?city.
    ?car a elec:PHEV;
         elec:isRegisteredIn ?zipcode;
         elec:hasRange ?range.

    FILTER(?county=?agicounty)

    {
        select ?agicounty (AVG(?agi) AS ?avgagi) where {
            ?agicounty a elec:County.
            ?city elec:belongsTo ?agicounty.
            ?zipcode elec:ofCity ?city;
                  elec:hasAgi ?agi.

        }
        GROUP BY ?agicounty
        ORDER BY asc(?agicounty)
    }

}
GROUP BY ?county
ORDER BY desc(?avgagi)
```



AGI distribution



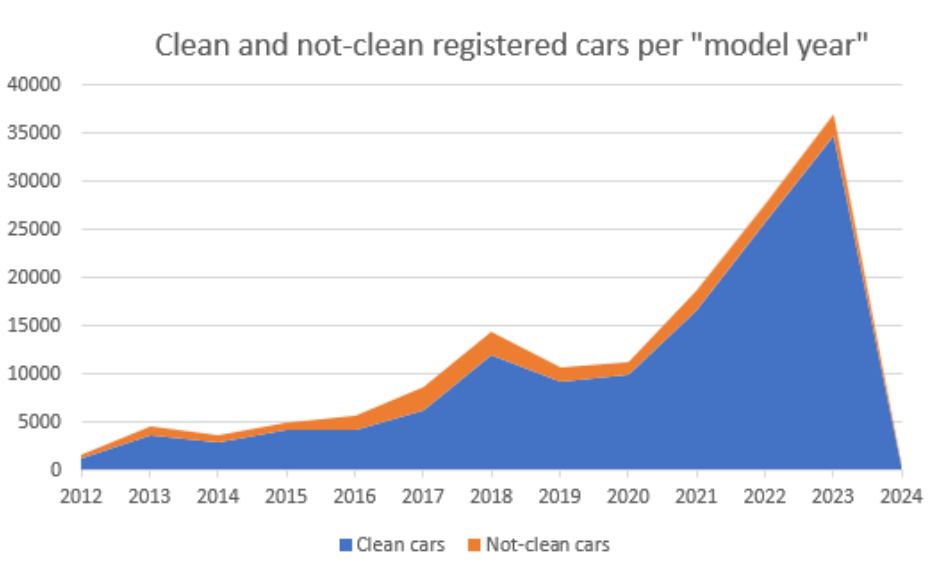
PHEV car range



Clean and Not Clean Cars

What is the clean and non-clean car count for each production year?

- Highlights how the amount of clear and non-clean registered vehicles varied over time and their proportion.

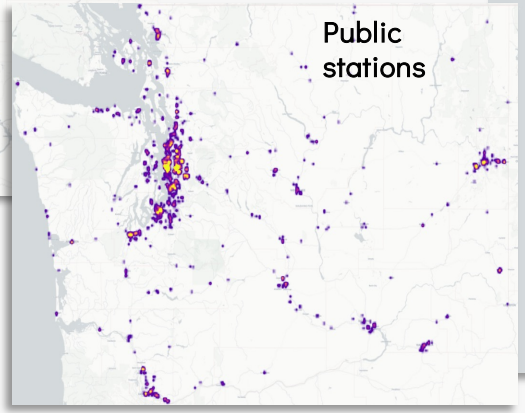


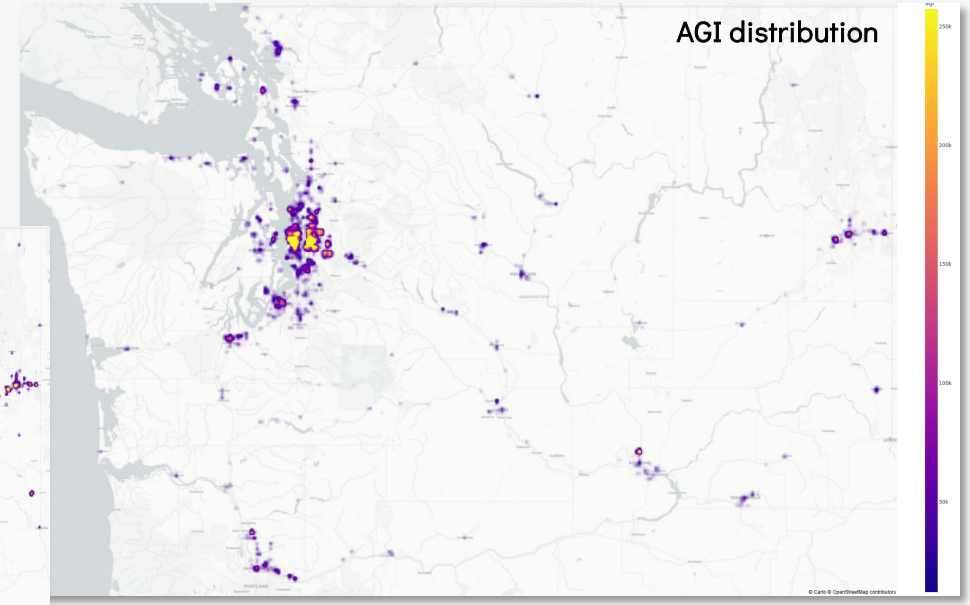

```
SELECT ?year ?clc ?nclc WHERE{
    # Below, the amount of registered clean "model years" is queried.
    {
        SELECT ?year (COUNT(?car) AS ?clc) WHERE{
            ?moye elec:hasYear ?year .
            ?car elec:hasModelYear ?moye .
            ?car rdf:type elec:CleanCar .
        }
        GROUP BY ?year
    }
    # Below, the amount of registered non-clean "model years" is queried.
    {
        SELECT ?year (COUNT(?car) AS ?nclc) WHERE{
            ?moye elec:hasYear ?year .
            ?car elec:hasModelYear ?moye .
            ?car rdf:type elec:NotCleanCar .
        }
        GROUP BY ?year
    }
}
ORDER BY DESC(?year)
```

AGI and Station densities

What are the mean AGI and charging station amount/type distributions?

- Highlights the eventual correlation between wealth and charging station density/type



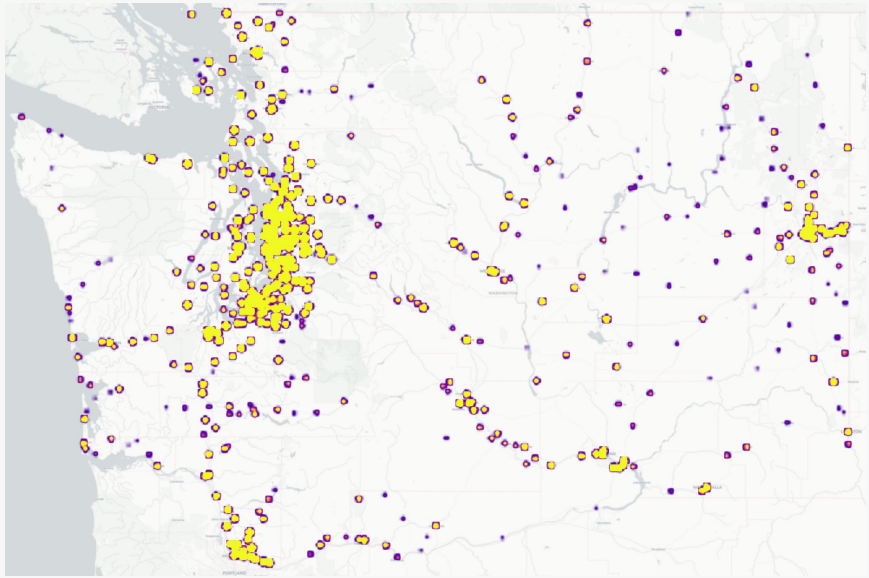


Car and Station densities

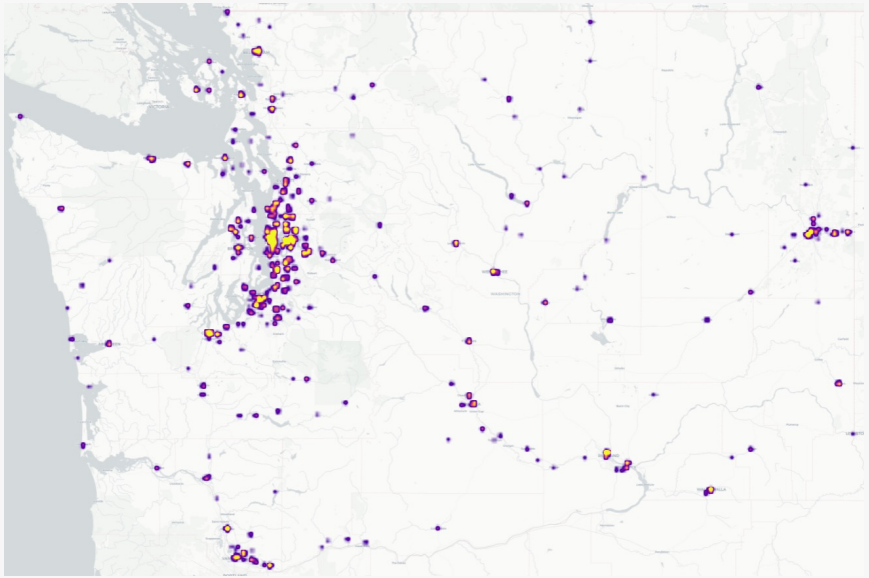
What are the car and the electric station distributions?

- Highlights the eventual correlation between car density and charging station density

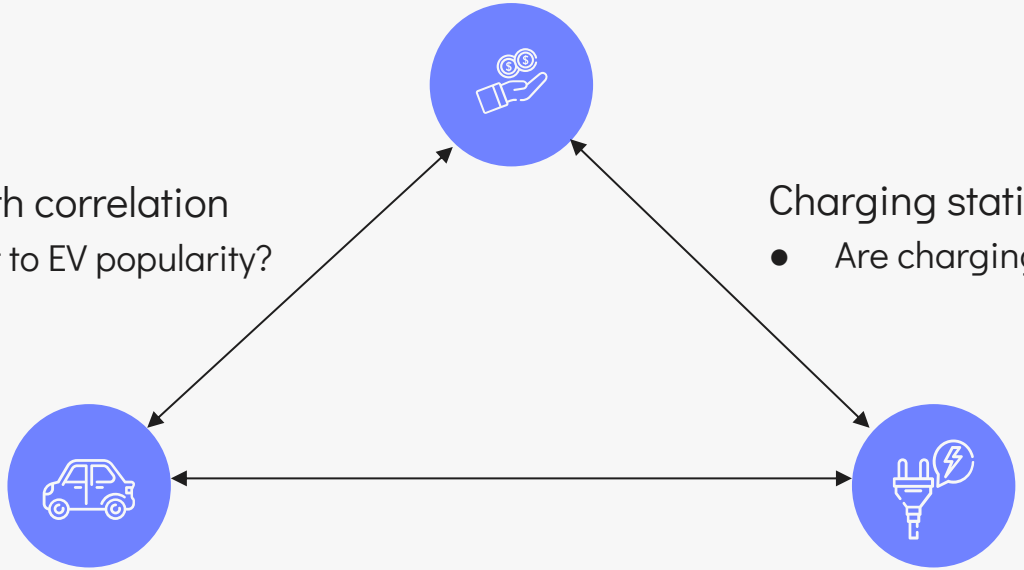
Car density



Station density



KG Applications



Car number-wealth correlation

- Is income a limit to EV popularity?

Charging station-wealth correlation

- Are charging stations evenly installed?

Charging station-Car number correlation

- Does charging station density influence EV diffusion, both by limiting or supporting it?

04
Future Work

Future Development & Expansion

Add more US States

Expand the kg with data from other US states

Expand to Europe/World

Challenge due to the ZIP Codes

Range of electric cars

Range of electric cars is not maintained since they are considered all clean

Add power suppliers

Need a data source to merge cars and stations power suppliers

Thanks for your attention!

Serialization (1)

URI processing:

```
def urify_string(s: str):
    s=unidecode(s)
    pattern = "[^0-9a-zA-Z\s]+"
    s = re.sub(pattern, " ", s).title().replace(" ", "")
    return s
```

Stations File pre-processing:

```
file = open(SOURCE_FOLDER + STATIONS_FILE, "r", encoding="utf-8")                  # Input file
wa_fuel_stations = wa_e_stations = open(DATA_FOLDER + STATIONS_FILE, "w", encoding="utf-8")     # Output file

# Write CSV headers
wa_fuel_stations.write(file.readline())

row = file.readline()               # Read first line
while (row2 := file.readline()):
    row2_error = False

    # If the row is interrupted, recover it (there can be multiple interruption)
    while("ELEC" not in row2):
        row2_error = True
        index = row2.find('",')                                          # Find the end of last interrupted string, if ex
        row = row.strip() + row2[index if index != -1 else 0 : ]        # Concatenate the row begin with the second part
        row2 = file.readline()

    if ",WA," in row: wa_e_stations.write(row)
    row = row2                                                           # Check on next cycle

file.close()
wa_e_stations.close()
```

Serialization (2)

Car processing:

Car classification:

```
#Adding car type (BEV/PHEV or CleanCar/NotCleanCar)
if(re.search(".*BEV.*",row['Electric Vehicle Type'])):
    graph.add((Car, RDF.type, ECO.CleanCar))
    graph.add((Car, RDF.type, ECO.BEV))
else:
    graph.add((Car, RDF.type, ECO.PHEV))
    if ((row['Electric Range'] >0) and (row['Electric Range'] <30)):
        graph.add((Car, RDF.type, ECO.NotCleanCar))
    elif ((row['Electric Range'] >=30)):
        graph.add((Car, RDF.type, ECO.CleanCar))



#Adding range only if maintained (grater than zero)
if (row['Electric Range'] > 0):
    graph.add((Car, ECO['hasRange'], Literal(row['Electric Range'], datatype=XSD.integer) ))
```

Car location:

```
# If there are valid coordinates
if (row['Vehicle Location'] is not None) and (point := re.findall("(?<=\().*(?=\))", str(row['Vehicle Location']
    coordinates = point.pop().split()
    latitude = coordinates.pop()
    longitude = coordinates.pop()
    # Cars' coordinates represent the ZipCode centers, hence when these information is retrieved, it's added to
    graph.add((ZipCode, ECO['hasLongitude'], Literal(longitude, datatype=XSD.float)))
    graph.add((ZipCode, ECO['hasLatitude'], Literal(latitude, datatype=XSD.float)))
```