# IEA (OECD) - Part 1

*Data visualisation designer - assignment*

In this notebook I check the data.

### Data loading

I load the provided dataset to the notebook to check the variables included in the dataset. There is one numeric variable: range, four different categorial variables and one time variable.

```
data = ▸ Table: 6 cols x 629 rows {_names: Array(6), _data: Object, _total: 6
```

```
data = aq.fromCSV(await FileAttachment("data@4.csv").text())
```

| technology | vehicle_type | category | year | region | range |
|---|---|---|---|---|---|
| Fuel_Cell | Transit_Bus | Bus | 2019 | Europe | 300 |
| Electric | Transit_Bus | Bus | 2019 | Europe | 250 |
| Electric | Cargo_Van | MFT | 2022 | U.S._Canada | 160.9344 |
| Electric | Cargo_Van | MFT | 2022 | Europe | 160 |
| Electric | School_Bus | Bus | 2019 | U.S._Canada | 193.12128 |

```
data.view(5)
```

### Data cleaning

Range and year are numeric variables, which appear as string variables, hence I change it back to numeric variables. Then I exclude missing values and clean out wrong values by using filter on range and year. Last but not least I order the dataset by the variable "range" to check extreme values.

| technology | vehicle_type | category | year | region | range |
|---|---|---|---|---|---|
| Electric | HD_Truck | HFT | 2024 | Europe | 1448.409600 |
| Fuel_Cell | HD_Truck | HFT | 2024 | U.S._Canada | 1448.409600 |
| Fuel_Cell | HD_Truck | HFT | 2023 | U.S._Canada | 1287.475200 |
| Fuel_Cell | HD_Truck | HFT | 2023 | Europe | 1287.000000 |
| Fuel_Cell | HD_Truck | HFT | 2021 | Europe | 804.672000 |

```
data
  .derive({ range: (d) => `${d["range"]} ` * 1 })
  .derive({ year: (d) => `${d["year"]} ` * 1 })
  .filter((d) => d.range > 0)
  .filter((d) => d.year > 0)
  .orderby(aq.desc("range"))
  .view(5)
```

The two highest values seem to be outliers and are the only two values available for the year 2024, therefore I decided to exclude them for further analysis.

| technology | vehicle_type | category | year | region | range |
|---|---|---|---|---|---|
| Fuel_Cell | HD_Truck | HFT | 2023 | U.S._Canada | 1287.475200 |
| Fuel_Cell | HD_Truck | HFT | 2023 | Europe | 1287.000000 |
| Fuel_Cell | HD_Truck | HFT | 2021 | Europe | 804.672000 |
| Fuel_Cell | HD_Truck | HFT | 2021 | U.S._Canada | 804.672000 |
| Fuel_Cell | HD_Truck | HFT | 2021 | China | 804.672000 |

```
data
  .derive({ range: (d) => `${d["range"]} ` * 1 })
  .derive({ year: (d) => `${d["year"]} ` * 1 })
```

```
  .filter((d) => d.range > 0)
  .filter((d) => (d.year > 0) & (d.year < 2024))
  .orderby(aq.desc("range"))
  .view(5)
```

Looking at the minimum values they seem at least from a statistical point of view ok.

| technology | vehicle_type | category | year | region | range | range2 |
|---|---|---|---|---|---|---|
| Electric | Shuttle_Bus | MiniBus | 2019 | U.S._Canada | 64.37376 | 64.373760 |
| Electric | Transit_Bus | MiniBus | 2019 | Europe | 69 | 69.000000 |
| Electric | Transit_Bus | Bus | 2020 | Europe | 70 | 70.000000 |
| Electric | HD_Truck | MFT | 2021 | China | 90 | 90.000000 |
| Electric | Other | HFT | 2019 | U.S._Canada | 90.123264 | 90.123264 |

```
data
  .derive({ range2: (d) => `${d["range"]} ` * 1 })
  .derive({ year: (d) => `${d["year"]} ` * 1 })
  .filter((d) => d.range2 > 0)
  .filter((d) => (d.year > 0) & (d.year < 2024))
  .orderby("range2")
  .view(5)
```

Saving the file as an array of objects.

```
data_final = ▸ Array(478) [Object, Object, Object, Object, Object, Object, Ob
```

```
data_final = data
  .derive({ range: (d) => `${d["range"]} ` * 1 })
  .filter((d) => d.range > 0)
  .derive({ year: (d) => `${d["year"]} ` * 1 })
  .filter((d) => (d.year > 0) & (d.year < 2024))
  .orderby(aq.desc("range"))
  .objects()
```

## Categorial variables

Now I check the sample size and average values by the different categories: technology, year, region, category and type.

### 1. Technology

The dataset includes far less data points with the technology fuel cell.

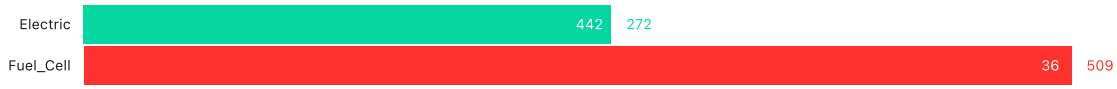
| technology | mean | median | max | count |
|---|---|---|---|---|
| Fuel_Cell | 509.235518 | 450.61632 | 1287.4752 | 36 |
| Electric | 271.670167 | 250.00000 | 804.6720 | 442 |

```
data
  .filter((d) => d.range > 0)
  .filter((d) => (d.year > 0) & (d.year < 2024))
  .derive({ range2: (d) => `${d["range"]} ` * 1 })
  .groupby("technology")
  .rollup({
    mean: (d) => op.mean(d.range2),
    median: (d) => op.median(d.range2),
    max: (d) => op.max(d.range2),
```

```
    count: op.count()
  })
  .orderby(aq.desc("mean"))
  .view()
```

```
data_tec = ▸ Array(2) [Object, Object]
```

Fuel has a range on average twice as big as electric.

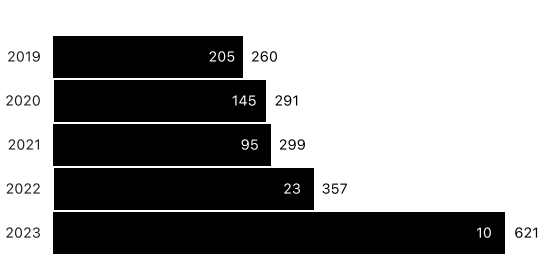



**2. Year**

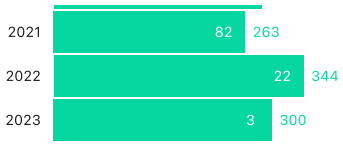
Already having removed the year 2024 all other category sample sizes are at least 10.

| year | mean | median | max | count |
|---|---|---|---|---|
| 2023 | 620.797830 | 441.764928 | 1287.4752 | 10 |
| 2022 | 356.956789 | 321.868800 | 804.6720 | 23 |
| 2021 | 298.754473 | 250.000000 | 804.6720 | 95 |
| 2020 | 291.098684 | 251.000000 | 760.0000 | 145 |
| 2019 | 260.496215 | 250.000000 | 666.0000 | 205 |

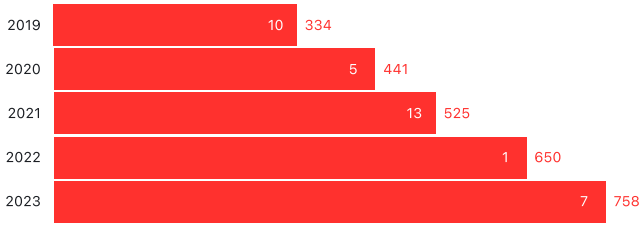
```
data
  .filter((d) => d.range > 0)
  .filter((d) => (d.year > 0) & (d.year < 2024))
  .derive({ range2: (d) => `${d["range"]} ` * 1 })
  .groupby("year")
  .rollup({
    mean: (d) => op.mean(d.range2),
    median: (d) => op.median(d.range2),
    max: (d) => op.max(d.range2),
    count: op.count()
  })
  .orderby(aq.desc("mean"))
  .view(6)
```




Electric does not increase much by year.




Fuel by year has generally very small sample size less than 10 but we see an increase year by year.

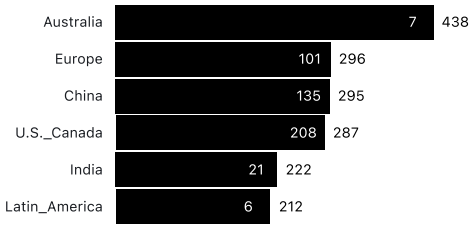



### Region

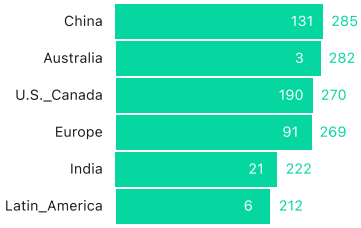
Australia and Latin America have very small sample sizes.

| region | mean | median | max | count |
|---|---|---|---|---|
| Australia | 437.563200 | 305.0000 | 804.6720 | 7 |
| Europe | 295.989126 | 250.0000 | 1287.0000 | 101 |
| China | 294.546411 | 270.0000 | 804.6720 | 135 |
| U.S._Canada | 287.271617 | 241.4016 | 1287.4752 | 208 |
| India | 222.238095 | 250.0000 | 300.0000 | 21 |
| Latin_America | 211.597749 | 200.0000 | 260.0000 | 6 |

All




Electric

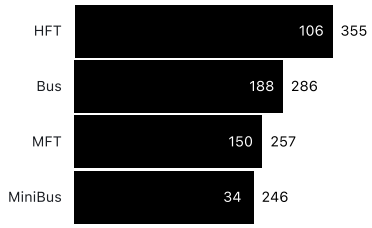



### Category

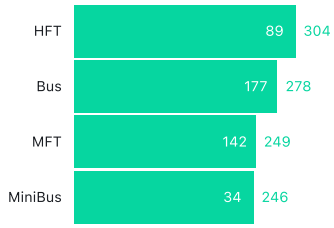
Looks good.

| category | mean | median | max | count |
|---|---|---|---|---|
| HFT | 354.510829 | 271.794240 | 1287.4752 | 106 |
| Bus | 286.446807 | 250.500000 | 675.0000 | 188 |
| MFT | 257.431458 | 241.401600 | 563.2704 | 150 |
| MiniBus | 246.053711 | 233.354880 | 480.0000 | 34 |

All




Electric

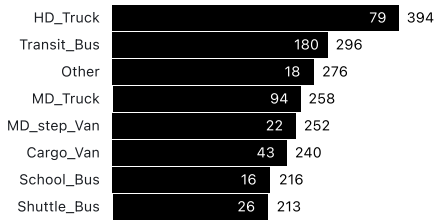



**Vehicle type**

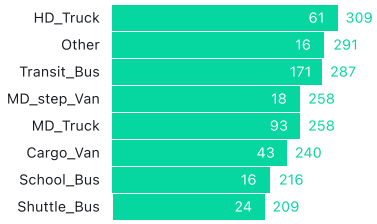
Looks good.

| vehicle_type | mean | median | max | count |
|---|---|---|---|---|
| HD_Truck | 393.738552 | 300.000000 | 1287.4752 | 79 |
| Transit_Bus | 295.697479 | 258.747520 | 675.0000 | 180 |
| Other | 276.341124 | 221.284800 | 760.0000 | 18 |
| MD_Truck | 258.237053 | 241.401600 | 430.0000 | 94 |
| MD_step_Van | 251.589789 | 241.401600 | 321.8688 | 22 |
| Cargo_Van | 239.630754 | 241.401600 | 402.3360 | 43 |
| School_Bus | 216.155016 | 208.410048 | 321.8688 | 16 |
| Shuttle_Bus | 212.838437 | 213.238080 | 321.8688 | 26 |

All




Electric



## Appendix

```
data_year = ▶ Array(5) [Object, Object, Object, Object, Object]

data_year_electric = ▶ Array(5) [Object, Object, Object, Object, Object]

data_year_fuel = ▶ Array(5) [Object, Object, Object, Object, Object]

data_region = ▶ Array(6) [Object, Object, Object, Object, Object, Object]

data_region_electric = ▶ Array(6) [Object, Object, Object, Object, Object, Ob

data_category = ▶ Array(4) [Object, Object, Object, Object]

data_category_electric = ▶ Array(4) [Object, Object, Object, Object]

data_vehicle_type = ▶ Array(8) [Object, Object, Object, Object, Object, Objec

data_vehicle_type_electric = ▶ Array(8) [Object, Object, Object, Object, Obje

width = 900

color_iea = "#0000FF"

d3 = ▶ Object {format: ƒ(t), formatPrefix: ƒ(t, n), timeFormat: ƒ(t), timePa

import {aq, op, table} from "@uwdata/arquero"

<style>
```