

Pandemic Information System Model

SYSTEMS AND METHODS FOR BIG AND UNSTRUCTURED DATA

PROF. MARCO BRAMBILLA

THIRD DELIVERY
ELASTICSEARCH PROJECT

January 2022

Avci Oguzhan - 10557284
Gentile Nicole - 10594355
Rigamonti Davide - 10629791
Singh Raul - 10623232
Tagliaferri Mattia - 10572418



POLITECNICO
MILANO 1863

Contents

1	Introduction	3
1.1	Problem Specification	3
1.2	Hypoteses	3
2	Data	4
2.1	Schema description	4
2.2	Dataset description	7
2.3	Creating indexes	8
2.4	Queries	8
2.4.1	Number of daily vaccines for each dose	8
2.4.2	Number of doses per gender in each region	9
2.4.3	Number of vaccines per dose number in each region	9
2.4.4	Number of doses for each vaccine supplier	10
2.4.5	Number of doses for each NUTS1 zone	11
2.4.6	Number of people with previous infection by region	11
2.4.7	Number of doses for each age group	12
2.4.7.1	Grouping by age range and region	12
2.4.8	Number of average doses per region	13
2.4.9	Number of doses for each age group and vaccine supplier	13
2.4.10	Number of booster doses for age range	14
2.4.11	Top 10 dates with most vaccinations	15
2.4.12	Search by region name	15
2.4.13	Percentage of fully vaccinated people per region	15
2.4.14	Percentage of healed people per region	16
2.4.15	Daily number of total cases, hospitalizations and positives	18
2.4.16	Ratio between positive and hospitalized people	18
2.5	Commands	19
2.5.1	Add entries for a given day	19
2.5.2	Remove entries from a given day	20
3	Data Visualization with Kibana	21
3.1	Dashboard visualizations	21
3.1.1	Control panel	21
3.1.2	Vaccinations for each region	21
3.1.3	Pie charts	23
3.1.4	Metric Visualization	23
3.1.5	Number of vaccines by genre	24
3.1.6	Average number of vaccines by region	24
3.1.7	Number of 1st, 2nd and booster doses per week	25
3.1.8	Total number of vaccines by date	25
3.1.9	Number of vaccines per gender in each region	26
3.1.10	Number of vaccinations by age range	26
3.1.11	Number of people with previous infection in each region	27

4	Other features in HBase	28
4.1	Input Data Preparation	28
4.2	Storing data into HBase	28
4.3	Querying HBase	29
4.3.1	Number of daily vaccines for each dose	30
4.3.2	Number of doses for each vaccine supplier	30
4.3.3	Top 10 dates with most vaccinations	31
5	References and sources	32

1 Introduction

1.1 Problem Specification

The idea of the project is to use Elasticsearch to store and analyze data about COVID-19 vaccinations in Italy. The data we used can be found in the following GitHub repository: <https://raw.githubusercontent.com/italia/covid19-opendata-vaccini/master/dati/somministrazioni-vaccini-latest.csv> and contains data about daily vaccinations divided by region, age range and vaccine supplier.

Moreover, we used three additional datasets to integrate other information about the population of each region (*platea.csv*), the number of healed people in each region (*soggetti-guariti.csv*) and data about the number of daily covid cases (*dpc-covid19-ita-regioni.csv*).

The main goal is to analyze the data in order to build meaningful statistics about the vaccinations in Italy; examples can be the total number of doses for each vaccine brand, the top regions for vaccine administration and so on.

1.2 Hypotheses

We assumed that the general user doesn't know the Italian language, therefore all the fields' names contained in the dataset were renamed to an appropriate English translation.

For the file *dpc-covid19-ita-regioni.csv* we decided to not include some fields mainly because they are no longer updated or they are not entirely relevant for the scope of this project.

2 Data

2.1 Schema description

- Schema for **vaccines administrations** (somministrazioni-vaccini-latest.csv):
 - @timestamp: *date*
Timestamp assigned by elasticsearch.
 - administration_date (data_somministrazione): *date [iso8601]*
Date of all the doses' administrations using the international standard format.
 - supplier (fornitore): *keyword*
Brand of the administrated vaccine, mapped as keyword since entries are enumerated and aggregation is possible.
 - area (area): *keyword*
Abbreviation for a region's name, mapped as keyword since entries are enumerated and aggregation is possible.
 - age_range (fascia_anagrafica): *keyword*
Age range of the people administered with the vaccines, mapped as keyword since entries are enumerated and aggregation is possible.
 - male_gender (Sesso_maschile): *long*
Total number of vaccinations administered to males by day and region, mapped as a long since it represents a possibly large number.
 - female_gender (Sesso_femminile): *long*
Total number of vaccinations administered to females by day and region, mapped as a long since it represents a possibly large number.
 - first_dose (prima_dose): *long*
Total number of administered first doses, mapped as a long since it represents a possibly large number.
 - second_dose (seconda_dose): *long*
Total number of administered second doses, mapped as a long since it represents a possibly large number.
 - previous_infection (pregressa_infezione): *long*
Total number of administered doses to subjects with a previous Covid-19 infection within 3 to 6 months before the *administration date*, therefore needing only one dose, mapped as a long since it represents a possibly large number.
 - booster_dose (dose_addizionale_booster): *long*
Total number of administered booster doses, mapped as a long since it represents a possibly large number.
 - total_doses: *long*
Total number of administered doses considering first, second, booster and previous infection doses, mapped as a long since it represents a possibly large number.

- NUTS1_code (codice_NUTS1): *keyword*
European classifications for territorial units (Level: NUTS 1), mapped as keyword since entries are enumerated and aggregation is possible.
- NUTS2_code (codice_NUTS2): *keyword*
European classifications for territorial units (Level: NUTS 2), mapped as keyword since entries are enumerated and aggregation is possible.
- ISTAT_region_code (codice_regione_ISTAT): *keyword*
ISTAT code used for identifying regions, mapped as keyword since entries are enumerated and aggregation is possible. Additionally, in order to fully utilize the map representation capabilities of Kibana this value needs to be joined with Kibana's own ISTAT regional codes (stored as keywords), besides, it would be useless to execute arithmetical operations on this value due to its nature.
- area_name (nome_area): *text*
Full name of the region mapped as text in order to be better suited for queries that utilize an analyzer. Even though this field technically represents an enumeration, we found particularly useful the ability to search for a region's name utilizing a text analyzer since there are plenty other fields that can be used to aggregate regions.

- Schema for **population** (platea.csv):

- area (area): *keyword*
Abbreviation for a region's name, mapped as keyword since entries are enumerated and aggregation is possible.
- area_name (nome_area): *text*
Full name of the region mapped as text in order to be better suited for queries that utilize an analyzer.
- age_range (fascia_anagrafica): *keyword*
Age range of the people administered with the vaccines, mapped as keyword since entries are enumerated and aggregation is possible.
- total_population (totale_popolazione): *long*
Total population for an associated area and age range, mapped as a long since it represents a large number.

- Schema for **healed** (soggetti-guariti.csv):

- area (area): *keyword*
Abbreviation for a region's name, mapped as keyword since entries are enumerated and aggregation is possible.
- area_name (nome_area): *text*
Full name of the region mapped as text in order to be better suited for queries that utilize an analyzer.
- age_range (fascia_anagrafica): *keyword*
Age range of the people administered with the vaccines, mapped as keyword since entries are enumerated and aggregation is possible.

- `total_healed (totale_healed)`: *long*
Total number of people that healed from an infection in a 6 months timespan for an associated area and age range, mapped as a long since it represents a large number.
- Schema for **covid-information** (dpc-covid19-ita-regioni.csv):
 - `@timestamp`: *date*
Timestamp assigned by elasticsearch.
 - `date`: *date [iso8601]*
Date of the government information.
 - `area_name (denominazione_regione)`: *text*
Full name of the region mapped as text in order to be better suited for queries that utilize an analyzer.
 - `ISTAT_region_code`: *keyword*
ISTAT code used for identifying regions, mapped as keyword since entries are enumerated and aggregation is possible.
 - `deceased`: *long*
Total number of people that are deceased due to Covid-19, mapped as a long since it represents a large number.
 - `recovered`: *long*
Total number of people that have been dismissed from the hospital because they have recovered from Covid-19, mapped as a long since it represents a large number.
 - `home_confinement`: *long*
Total number of people that are home confined due to Covid-19, mapped as a long since it represents a large number.
 - `lat`: *double*
Latitude of the region, mapped as a double since it's a float number.
 - `long`: *double*
Longitude of the region, mapped as a double since it's a float number.
 - `notes`: *text*
General notes from the government, mapped as text in order to be better suited for queries that utilize an analyzer.
 - `cases_notes`: *text*
Notes related to the tested cases from the government, mapped as text in order to be better suited for queries that utilize an analyzer.
 - `new_positives`: *long*
Daily new number of people that have been tested positive due to Covid-19, mapped as a long since it represents a large number.
 - `hospitalized_with_symptoms`: *long*
Total number of people that have been hospitalized with symptoms of Covid-19, mapped as a long since it represents a large number.

- **country:** *keyword*
Code indicating the country, mapped as keyword since entries are enumerated and aggregation is possible.
- **tests:** *long*
Total number of people that have been tested for Covid-19 (using molecular tests), mapped as a long since it represents a large number.
- **intensive_care:** *long*
Total number of people that are in intensive care due to Covid-19, mapped as a long since it represents a large number.
- **total_cases:** *long*
Total amount of positive cases, mapped as a long since it represents a large number.
- **total_hospitalized:** *long*
Total number of people that have been hospitalized due to Covid-19, mapped as a long since it represents a large number.
- **total_positives:** *long*
Total number of people that have been found positive to Covid-19 (either via test or home confinement), mapped as a long since it represents a large number.
- **total_positives_variation:** *long*
Daily variation of the total number of people that have been tested positive to Covid-19, mapped as a long since it represents a large number.
- **location:** *geo_point*
The combination of the latitude and longitude fields, mapped as geo_point since they represent a set of coordinates.

2.2 Dataset description

The dataset contains data about COVID-19 vaccinations in Italy.

For each day, region, age range and vaccine supplier we have information about: the number of vaccinated people (male and female), the number of 1st, 2nd and booster doses and information about the number of people who was previously infected.

We decided to integrate this data using the ‘platea.csv’ dataset that can be found in the specified repository; it contains data about the population for each region and each age range.

We also used information about the number of healed people in each region that can be found in ‘soggetti-guariti.csv’ present in the same repository.

In addition, we used the ‘dpc-covid19-ita-regioni.csv’ dataset to have more information about the pandemic, from the total number of positives to the number of hospitalized and deceased people.

2.3 Creating indexes

The four indexes utilized throughout the project were created following a specific naming convention.

We created one index for each CSV file (*somministrazioni-vaccini-latest.csv*, *platea.csv*, *soggetti-guariti.csv*, *dpc-covid19-ita-regioni.csv*), and named them respectively: "index-vaccines_administrations", "index-population", "index-healed" and "index-covid-information".

Then, we added in the "index-patterns" section an **aggregate index** named "index" as to create an overall "index-*" that combines all the indexes allowing for complex operations on multiple indexes.

In order to import data in the correct format, it is recommended to use the custom ingestion pipelines and mappings provided alongside this document.

2.4 Queries

2.4.1 Number of daily vaccines for each dose

Total number of first, second and booster doses administred in each date.

```
GET /index-vaccines_administrations/_search
{
  "size": 0,
  "aggs": {
    "Administration Date": {
      "terms": {
        "field": "administration_date",
        "size": 370
      },
    },
    "aggs": {
      "totalFirstDoses": {
        "sum": {
          "field": "first_dose"
        }
      },
      "totalSecondDoses": {
        "sum": {
          "field": "second_dose"
        }
      },
      "totalBoosters": {
        "sum": {
          "field": "booster_dose"
        }
      }
    }
  }
}
```

2.4.2 Number of doses per gender in each region

Total doses administred to men and to women in each region.

```
GET /index-vaccines_administrations/_search
{
  "size": 0,
  "aggs": {
    "Region": {
      "terms": {
        "field": "area",
        "size": 21
      },
      "aggs": {
        "totalWomenDoses": {
          "sum": {
            "field": "female_gender"
          }
        },
        "totalMenDoses": {
          "sum": {
            "field": "male_gender"
          }
        }
      }
    }
  }
}
```

2.4.3 Number of vaccines per dose number in each region

Total number of first, second, booster and total doses administred in each region.

```

GET /index-vaccines_administrations/_search
{
  "size": 0,
  "aggs": {
    "Region": {
      "terms": {
        "field": "area",
        "size": 21
      },
      "aggs": {
        "totalFirstDoses": {
          "sum": {
            "field": "first_dose"
          }
        },
        "totalSecondDoses": {
          "sum": {
            "field": "second_dose"
          }
        },
        "totalBoosters": {
          "sum": {
            "field": "booster_dose"
          }
        },
        "totalDoses": {
          "sum": {
            "field": "total_doses"
          }
        }
      }
    }
  }
}

```

2.4.4 Number of doses for each vaccine supplier

Total doses for each vaccine supplier (Pfizer/BioNTech, Moderna, Vaxzevria (AstraZeneca), Janssen, Pfizer Pediatrico).

```

GET /index-vaccines_administrations/_search
{"size":0,
  "aggs": {
    "supplier": {
      "terms": {
        "field": "supplier"
      },
      "aggs": {
        "total_doses" : { "sum" : {
          "field": "total_doses"
        }
      }
    }
  }
}

```

2.4.5 Number of doses for each NUTS1 zone

Total number of doses administered in each NUTS1 zone (ITC= NorthWest, ITH = NorthEast, ITI = Center, ITF = South, ITG = Isles)

```
GET /index-vaccines_administrations/_search
{"size":0,
 "aggs": {
   "by_NUTS1": {
     "terms": {
       "field": "NUTS1_code"
     },
     "aggs": {
       "total_doses" : { "sum" : {
         "field": "total_doses"
       }
     }
   }
 }
}
```

2.4.6 Number of people with previous infection by region

Total number of people who were infected by Covid-19 within 3 to 6 months before the date of the vaccination for each region.

```
GET /index-vaccines_administrations/_search
{
  "size":0,
  "aggs": {
    "Region": {
      "terms": {
        "field": "area"
      },
      "aggs": {
        "People with previous infection": {
          "sum": {
            "field": "previous_infection"
          }
        }
      }
    }
  }
}
```

2.4.7 Number of doses for each age group

Total number of doses administered to each age range ('05-11', '12-19', '20-29', '30-39', '40-49', '50-59', '60-69', '70-79', '80-89', '90+').

```
GET /index-vaccines_administrations/_search
{"size":0,
 "aggs": {
   "by_age": {
     "terms": {
       "field": "age_range"
     },
     "aggs": {
       "total_doses" : { "sum" : {
         "field": "total_doses"
       }
     }
   }
 }
}
```

2.4.7.1 Grouping by age range and region

Total number of doses administered to each age range in each region.

```
GET /index-vaccines_administrations/_search
{"size":0,
 "aggs": {
   "genres_and_products": {
     "multi_terms": {
       "terms": [{
         "field": "age_range"
       }, {
         "field": "area"
       }],
       "size": 1000
     },
     "aggs": {
       "total_doses" : { "sum" : {
         "field": "total_doses"
       }
     }
   }
 }
}
```

2.4.8 Number of average doses per region

Average number of administred doses in each region.

```
GET /index-vaccines_administrations/_search
{"size":0,
  "aggs": {
    "Region": {
      "terms": {
        "field": "area"
      },
      "aggs": {
        "avg_doses" : { "avg": {
          "field": "total_doses"
        } }
      }
    }
  }
}
```

2.4.9 Number of doses for each age group and vaccine supplier

Total number of doses by age group and vaccine supplier.

```

GET /index-vaccines_administrations/_search
{
  "size": 0,
  "track_total_hits": true,
  "aggs": {
    "age_groups_types": {
      "composite": {
        "sources": [{
          "fascia_anagrafica": {
            "terms": {
              "field": "age_range"
            }
          }
        ]
      }, {
        "supplier": {
          "terms": {
            "field": "supplier"
          }
        }
      }
    },
    "size": 10000
  },
  "aggs": {
    "total_vaccines": {
      "sum": {
        "field": "total_doses"
      }
    }
  }
}

```

2.4.10 Number of booster doses for age range

Total number of booster doses for each age range.

```

GET /index-vaccines_administrations/_search
{
  "size": 0,
  "aggs": {
    "age_groups": {
      "terms": {
        "field": "age_range"
      }
    },
    "aggs": {
      "boosters_per_group": {
        "sum": {
          "field": "booster_dose"
        }
      }
    }
  }
}

```

2.4.11 Top 10 dates with most vaccinations

Top 10 dates having the highest number of administered doses.

```
GET index-vaccines_administrations/_search
{
  "size": 0,
  "aggs": {
    "dates": {
      "terms": {
        "field": "administration_date"
      },
      "aggs": {
        "sum_vaccinations": {
          "sum": {
            "field": "total_doses"
          }
        },
        "dates_sort": {
          "bucket_sort": {
            "sort": [
              { "sum_vaccinations": { "order" : "desc" } }
            ],
            "size": 10
          }
        }
      }
    }
  }
}
```

2.4.12 Search by region name

Returns information about vaccines in the specified region.

```
GET /index-vaccines_administrations/_search
{
  "query": {
    "match": {
      "area_name": "Val d'aosta"
    }
  }
}
```

2.4.13 Percentage of fully vaccinated people per region

Returns the percentage of fully vaccinated people (i.e. having 1st and 2nd dose) for each region.

In order to obtain this result, we used additional data about the total population of each region contained into *platea.csv*.

```
GET /index-vaccines_administrations,index-population/_search
{
  "size": 0,
  "fields": [
    {
      "field": "@timestamp",
      "format": "date_time"
    },
    {
      "field": "administration_date",
      "format": "date_time"
    }
  ],
  "aggs": {
    "area_percentage": {
      "terms": {
        "field": "area",
        "size": 26
      },
      "aggs": {
        "hits": {
          "top_hits": {
            "size": 1,
            "_source": {
              "includes": "area_name"
            }
          }
        },
        "total_population": {
          "sum": {
            "field": "total_population"
          }
        },
        "second_dose": {
          "sum": {
            "field": "second_dose"
          }
        },
        "percentage_second_doses": {
          "bucket_script": {
            "buckets_path": {
              "total_pop": "total_population",
              "second_dose": "second_dose"
            },
            "script": "100 * params.second_dose / params.total_pop"
          }
        }
      }
    }
  }
}
```

2.4.14 Percentage of healed people per region

Returns the number of healed people over the total population for each region.
We used:

- Data about the total population in each region from *platea.csv*;
- The number of healed people for each region that can be found in *soggetti-guariti.csv*.

```
GET /index-healed,index-population/_search
{
  "size": 0,
  "fields": [
    {
      "field": "@timestamp",
      "format": "date_time"
    },
    {
      "field": "administration_date",
      "format": "date_time"
    }
  ],
  "aggs": {
    "area_percentage": {
      "terms": {
        "field": "area",
        "size": 26
      },
      "aggs": {
        "hits": {
          "top_hits": {
            "size": 1,
            "_source": {
              "includes": "area_name"
            }
          }
        },
        "total_population": {
          "sum": {
            "field": "total_population"
          }
        },
        "healed": {
          "sum": {
            "field": "total_healed"
          }
        },
        "percentage_healed": {
          "bucket_script": {
            "buckets_path": {
              "total_pop": "total_population",
              "healed": "healed"
            },
            "script": "100 * params.healed / params.total_pop"
          }
        }
      }
    }
  }
}
```

2.4.15 Daily number of total cases, hospitalizations and positives

Returns, the total number of cases, hospitalizations and positives for each day.

Given that this is not the total for a single day, but the overall total that is increased day by day.

We used additional data from *dpc-covid-19-ita-regioni.csv*.

```
GET /index-covid-information/_search
{
  "size": 0,
  "aggs": {
    "date": {
      "terms": {
        "field": "data",
        "size": 365,
        "order": {
          "_key": "asc"
        }
      },
    },
    "aggs": {
      "total_cases": {
        "sum": {
          "field": "total_cases"
        }
      },
      "total_positives": {
        "sum": {
          "field": "total_positives"
        }
      },
      "total_hospitalized": {
        "sum": {
          "field": "total_hospitalized"
        }
      }
    }
  }
}
```

2.4.16 Ratio between positive and hospitalized people

Returns the ratio between positive and hospitalized people for each region. It is calculated using the data of *dpc-covid19-ita-regioni.csv*.

```

GET /index-vaccines_administrations/_search
{
  "size": 0,
  "fields": [
    {
      "field": "data",
      "format": "date"
    }
  ],
  "sort": [
    {
      "data": {
        "order": "desc"
      }
    }
  ],
  "aggs": {
    "ratio_positive_hospitalized": {
      "terms": {
        "field": "data",
        "size": 365
      },
      "aggs": {
        "region": {
          "terms": {
            "field": "ISTAT_region_code"
          },
          "aggs": {
            "total_positives": {
              "sum": {
                "field": "total_positives"
              }
            },
            "total_hospitalized": {
              "sum": {
                "field": "total_hospitalized"
              }
            },
            "percentage_positive_hospitalized": {
              "bucket_script": {
                "buckets_path": {
                  "total_hospitalized": "total_hospitalized",
                  "total_positives": "total_positives"
                },
                "script": "100*params.total_hospitalized/params.total_positives"
              }
            }
          }
        }
      }
    }
  }
}

```

2.5 Commands

2.5.1 Add entries for a given day

Adds a new document in the dataset.

```
POST /index-vaccines_administrations/_doc/
{
  "area" : "FVG",
  "area_name" : "Friuli-Venezia Giulia",
  "second_dose" : 0,
  "previous_infection" : 1,
  "administration_date" : "2022-01-03",
  "NUTS1_code" : "ITH",
  "ISTAT_region_code" : 6,
  "male_gender" : 3,
  "first_dose" : 2,
  "age_range" : "30-39",
  "supplier" : "Moderna",
  "total_doses" : 3,
  "booster_dose" : 0,
  "NUTS2_code" : "ITH4",
  "female_gender" : 0,
  "total_doses" : 3
}
```

2.5.2 Remove entries from a given day

Removes all documents from the index for the specified date.

```
POST /index-vaccines_administrations/_delete_by_query
{
  "query": {
    "match": {
      "administration_date": "2022-01-03"
    }
  }
}
```

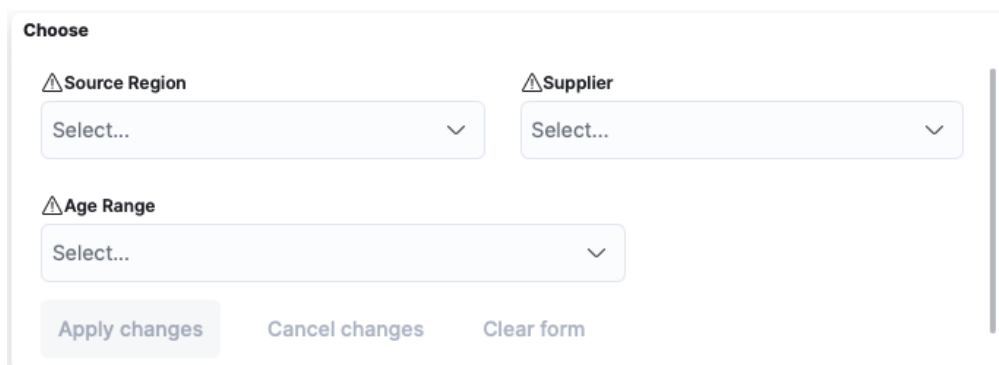
3 Data Visualization with Kibana

We used the dashboard to give a visual representation of the most meaningful queries. In order to use properly the dashboard, all imported indexes must follow the naming convention specified in the [Creating indexes](#) section.

3.1 Dashboard visualizations

3.1.1 Control panel

It allows to filter the results of the other elements of the dashboard based on region, vaccine supplier and age range.



The screenshot shows a control panel titled "Choose" with three filter sections. Each section has a warning icon and a label: "Source Region", "Supplier", and "Age Range". Below each label is a dropdown menu with "Select..." and a downward arrow. At the bottom of the panel are three buttons: "Apply changes" (highlighted in light blue), "Cancel changes", and "Clear form".

3.1.2 Vaccinations for each region

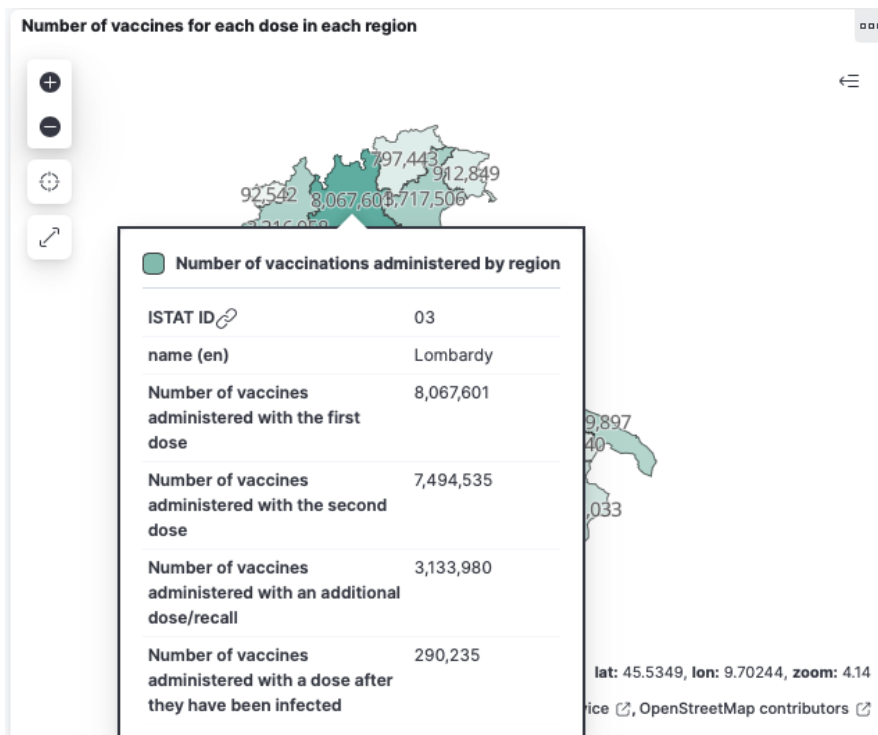
It allows to represent a dynamic map of Italy divided in regions. By default it shows the number of administered first doses in each region.

By hovering over a region, it shows a description of the number of vaccinations for each type (first, second, booster doses and so on) administered by that region.

Number of vaccines for each dose in each region



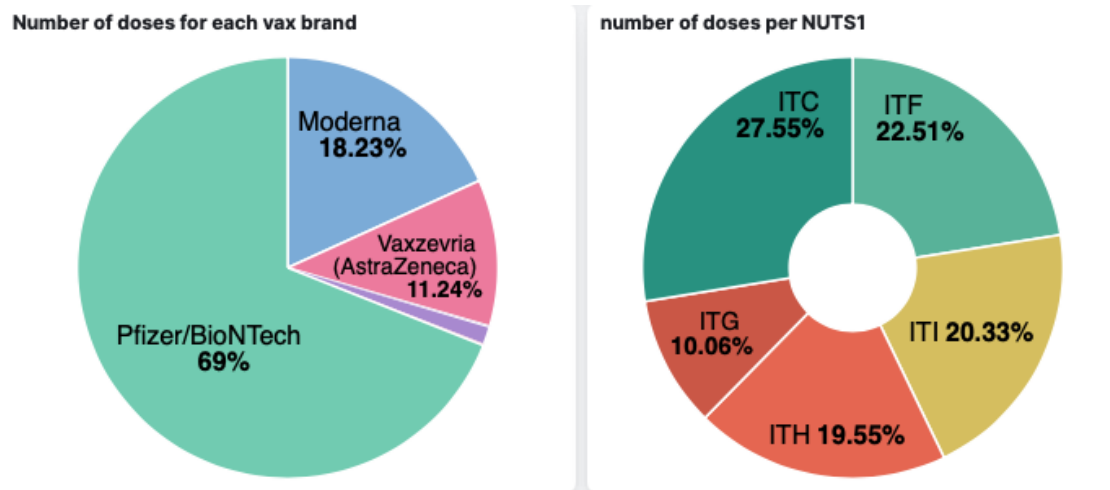
Map of Italy



Result of hovering

3.1.3 Pie charts

The two pie charts represent the percentage of doses for each vaccine supplier and for each NUTS1 over the total administered vaccines.



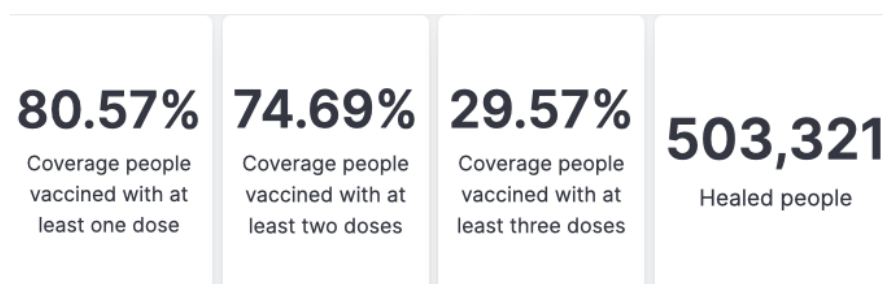
Percentage of doses for each vaccine brand (left) and for each NUTS1 (right)

3.1.4 Metric Visualization

It represents the percentage of people with at least one, two or three doses together with the number of people who healed from Covid-19 in the selected region (or the whole Italy if none is selected).

These numbers are obtained by integrating two different datasets:

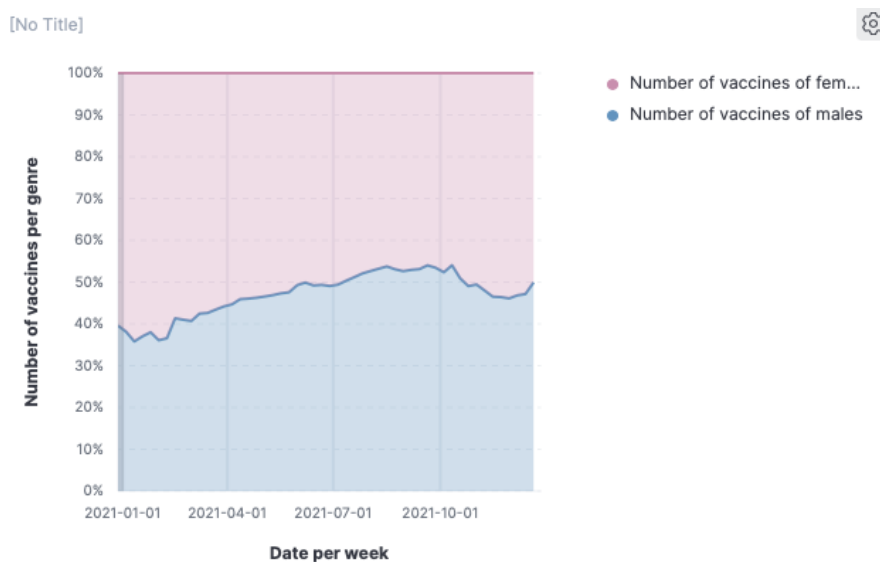
- The percentages of vaccinated people are calculated using data about the total population in each region from *platea.csv*;
- The number of healed people for each region can be found in *soggetti-guariti.csv*.



Metric visualization

3.1.5 Number of vaccines by genre

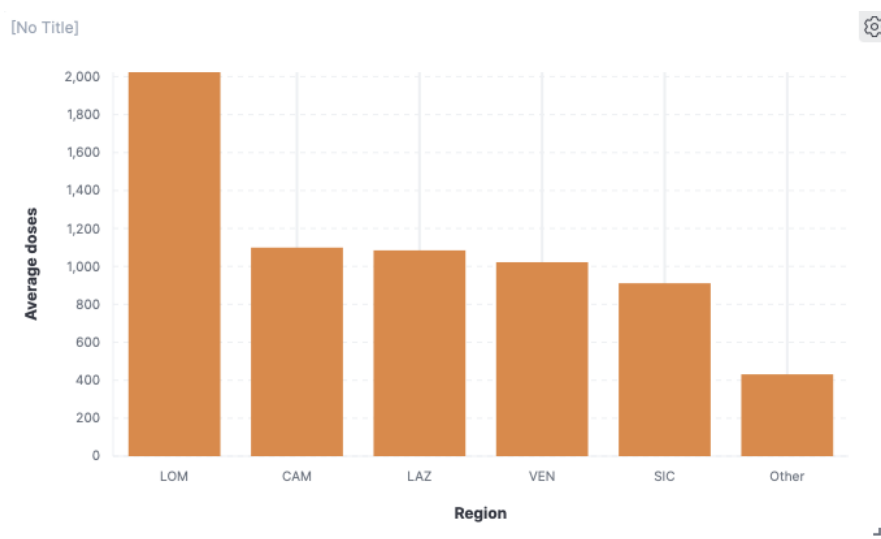
Represents a comparison between the total number of vaccines done by males and females, in each week from the starting date selected. By hovering with the mouse, a more detailed percentage is shown.



Number of vaccines by genre

3.1.6 Average number of vaccines by region

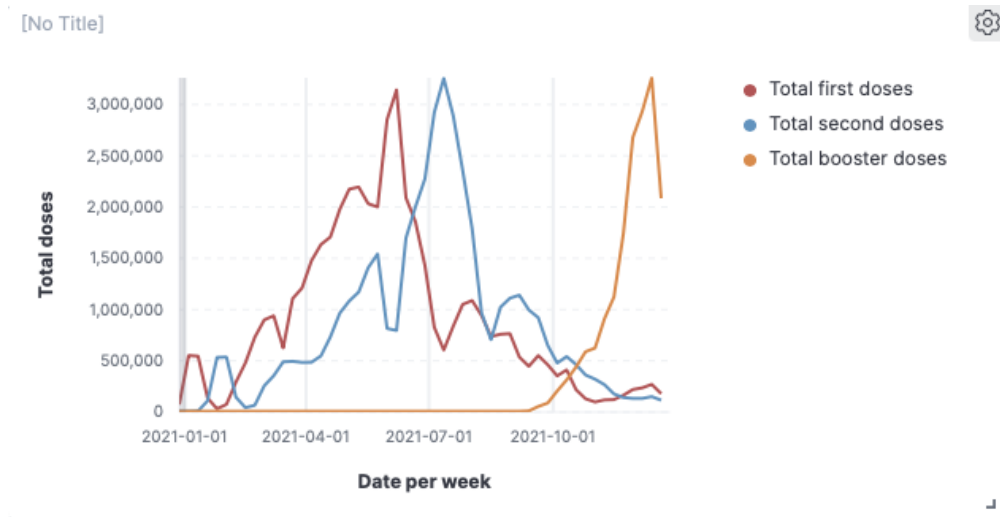
Represents the average number of vaccines administered in each region.



Average number of vaccines by region

3.1.7 Number of 1st, 2nd and booster doses per week

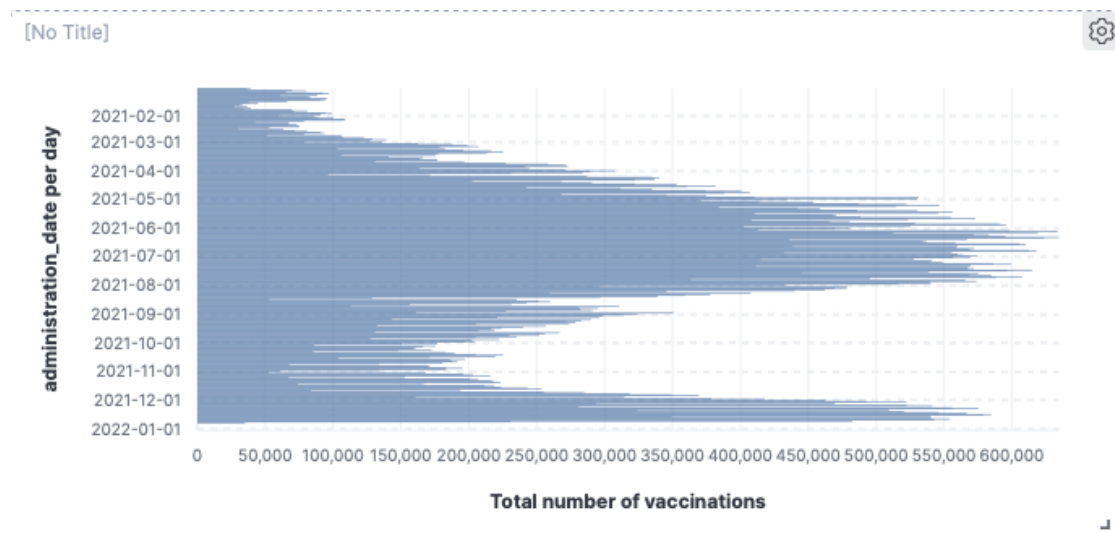
It shows a graph that, for each week from the starting date selected, represents the trend of the total doses administered by type (1st, 2nd or booster doses).



Number of 1st, 2nd and booster doses

3.1.8 Total number of vaccines by date

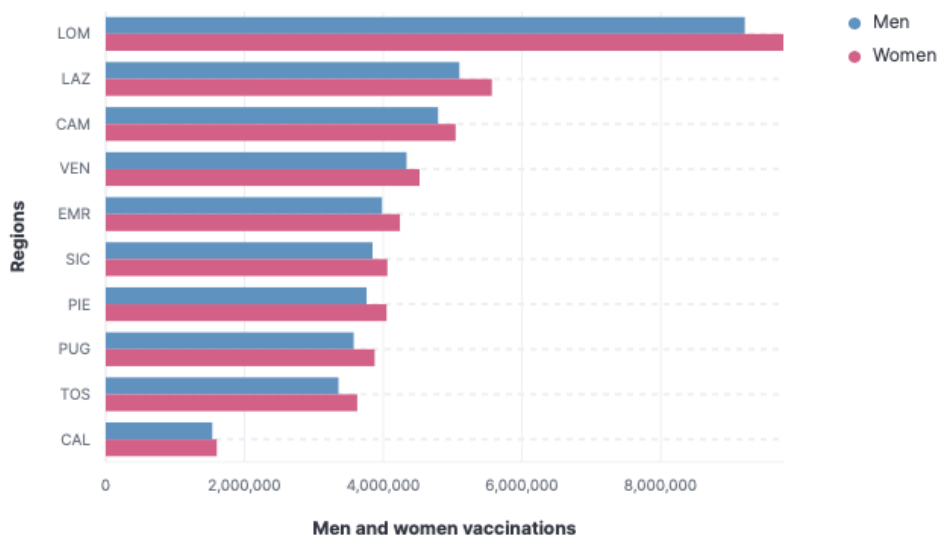
It shows the total number of administered doses in each day.



Total number of vaccines by date

3.1.9 Number of vaccines per gender in each region

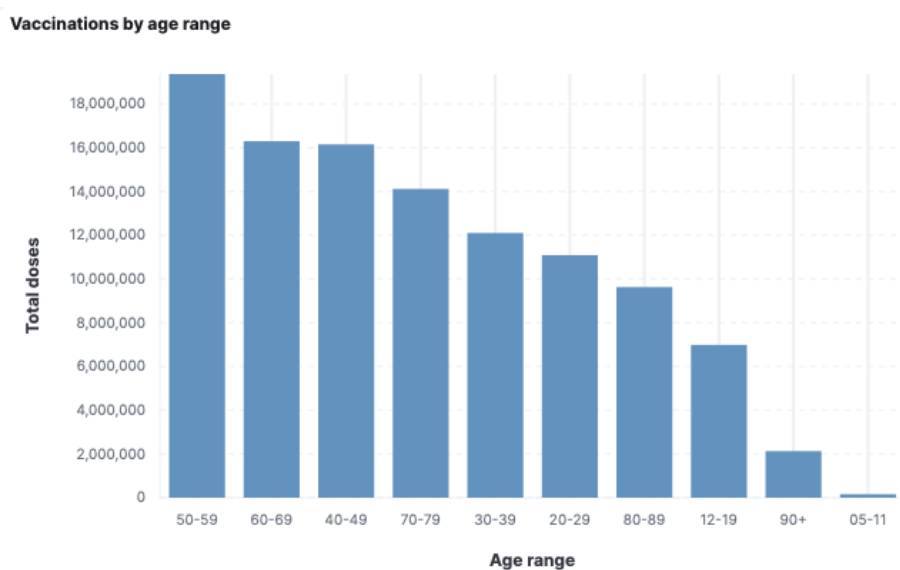
It shows the top 10 regions for number of administred doses divided by gender.



Number of vaccines per gender in each region

3.1.10 Number of vaccinations by age range

It shows the number of administred doses for each age range, sorted by number of doses.

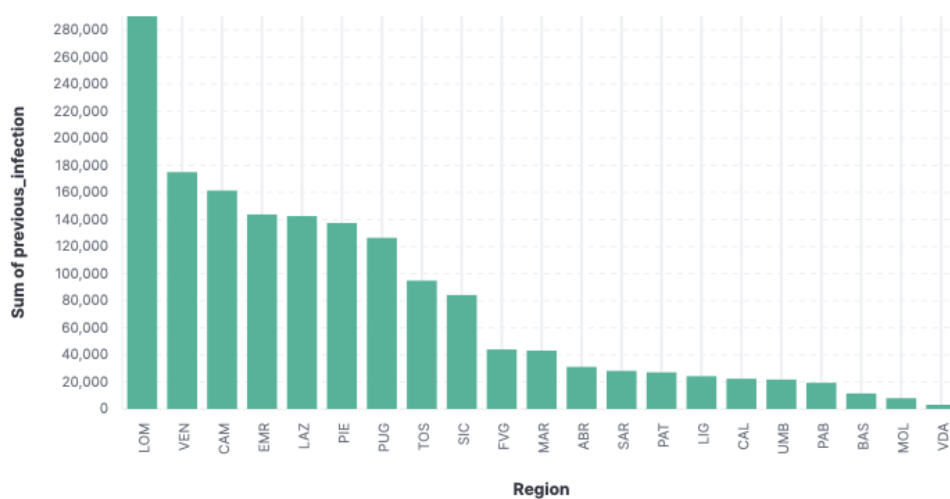


Number of vaccinations by age range

3.1.11 Number of people with previous infection in each region

It shows, for each region, the number of people who were infected by Covid-19 before doing the vaccination.

Number of people with previous infection by region



Number of people with previous infection in each region

4 Other features in HBase

We decided to use *HBase* as an alternative NoSQL DBMS in order to implement some additional features.

HBase is a column-oriented DBMS usually utilized in the context of the *Apache Hadoop* framework, it can be run locally using *Apache Zookeeper* or by installing *Hadoop* and running *HBase* on top of *HDFS*.

4.1 Input Data Preparation

In order to import data into *HBase*, the first step is to eliminate the labels from the csv file containing raw data.

In our case the file has been renamed as *vaccines_administrations_latest.csv* and a unique ID generated using *Python* `uuid.uuid4()` function (from the `uuid` library) has been added as the first item for each row since *HBase* expects a row key in that particular position.

The unique ID was generated in this way since an incremental sequential number as a row key is not recommended according to the *HBase* documentation.

Once we have the data in CSV format, we have to store it in a path from where *HBase* can access it.

If we want to load the file using *Hadoop*, we will have to copy the file to the *HDFS* location, this can be done using the command:

```
hadoop fs -copyFromLocal <LOCAL_PATH> <HDFS_PATH>
```

4.2 Storing data into HBase

```
create 'vaccines_administrations', {NAME => '_date'},  
{NAME => '_type'}, {NAME => 'region'},  
{NAME => 'stats'}, {NAME => 'doses'}, {NAME => 'nuts'}
```

This command will create an *HBase* table in order to store the data.

Notice that we are creating five column families for some of the columns of the CSV file, this is done for accessing and querying the data in a more easier and flexible way using *Apache Drill*.

```

./hbase org.apache.hadoop.hbase.mapreduce.ImportTsv
-Dimporttsv.separator=', '
-Dimporttsv.columns=HBASE_ROW_KEY, _date:administration_date,
_type:supplier, region:area, stats:age_range, stats:male_gender,
stats:female_gender, doses:first_dose, doses:second_dose,
doses:previous_infection, doses:booster_dose, nuts:NUTS1_code,
nuts:NUTS2_code, region:ISTAT.region_code, region:area_name
vaccines_administrations
<PATH>

```

Once we submit this command a *MapReduce* that will import the CSV file into the HBase table that we created will start.

Each column of the CSV file is mapped into the respective column family.

<PATH> needs be changed to the HDFS path if we are using *Hadoop* or to the local path of the CSV file if we are using *Zookeeper*.

The output should resemble the following:

```

2022-01-02 11:46:09,146 INFO [main] mapreduce.Job: map 100% reduce 0%
2022-01-02 11:46:09,147 INFO [main] mapreduce.Job: Job job_local1490239975_0001 completed successfully
2022-01-02 11:46:09,157 INFO [main] mapreduce.Job: Counters: 16
  File System Counters
    FILE: Number of bytes read=49849936
    FILE: Number of bytes written=31811946
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
  Map-Reduce Framework
    Map input records=166273
    Map output records=166273
    Input split bytes=133
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=80
    Total committed heap usage (bytes)=304611328
  ImportTsv
    Bad Lines=0
  File Input Format Counters
    Bytes Read=18748380
  File Output Format Counters
    Bytes Written=0

```

4.3 Querying HBase

For querying *HBase* we decided to use *Apache Drill*, this means that our queries will be written in SQL and can benefit from all the extensions that the framework provides.

To be able to query an *HBase* database, after running *Apache Drill* in embedded mode, we will have to go to the storage section and enable *HBase*.

The translation of some Elasticsearch queries with sample screenshots of possible results will be provided here, a more complete selection of queries will be provided separately.

4.3.1 Number of daily vaccines for each dose

```
SELECT CONVERT_FROM(vaccines_administrations._date.administration_date, 'UTF8')
      AS date_time,
      SUM(CAST(vaccines_administrations.doses.first_dose AS INT))
      AS number_of_daily_vaccines_first_dose,
      SUM(CAST(vaccines_administrations.doses.second_dose AS INT))
      AS number_of_daily_vaccines_second_dose,
      SUM(CAST(vaccines_administrations.doses.booster_dose AS INT))
      AS number_of_daily_vaccines_booster_dose
FROM vaccines_administrations
GROUP BY date_time
```

Possible result:

date_time	number_of_daily_vaccines_first_dose	number_of_daily_vaccines_second_dose	number_of_daily_vaccines_booster_dose
2021-08-12	175183	153335	0
2021-05-16	267611	137130	0
2021-10-20	35488	71043	77925
2021-01-03	35809	0	0
2021-09-21	82431	158470	8714
2021-07-08	110206	469998	0
2021-11-18	20622	28908	192862
2021-10-11	55483	82102	40970
2021-06-10	494184	128531	0
2021-04-16	275398	99764	0

Showing 1 to 10 of 10 entries

Previous 1 Next

4.3.2 Number of doses for each vaccine supplier

```
SELECT CONVERT_FROM(vaccines_administrations._type.supplier, 'UTF8')
      AS supplier,
      SUM(CAST(vaccines_administrations.doses.first_dose AS INT) +
          CAST(vaccines_administrations.doses.second_dose AS INT) +
          CAST(vaccines_administrations.doses.booster_dose AS INT) +
          CAST(vaccines_administrations.doses.previous_infection AS INT))
      AS total_doses
FROM vaccines_administrations
GROUP BY supplier
```

Possible result:

supplier	total_doses
Moderna	19718391
Vaxzevria (AstraZeneca)	12161093
Pfizer/BioNTech	74688059
Pfizer Pediatrico	158884
Janssen	1500055

4.3.3 Top 10 dates with most vaccinations

```
SELECT CONVERT_FROM(vaccines_administrations._date.administration_date, 'UTF8')
      AS administration_date,
      SUM(CAST(vaccines_administrations.doses.first_dose AS INT) +
          CAST(vaccines_administrations.doses.second_dose AS INT) +
          CAST(vaccines_administrations.doses.booster_dose AS INT) +
          CAST(vaccines_administrations.doses.previous_infection AS INT))
      AS total_vaccinations
FROM vaccines_administrations
GROUP BY administration_date
ORDER BY total_vaccinations ASC
LIMIT 10;
```

Possible result:

Top 10 dates with most vaccinations	
Number of vaccinations	Date per day
44,212	2021-06-03
38,135	2021-07-08
37,704	2021-06-04
37,542	2021-06-02
33,275	2021-07-09
33,101	2021-06-17
32,815	2021-07-07

5 References and sources

In order to develop this project, the following tools were used:

- Elasticsearch and Kibana in order to store, query and visualize data;
- HBase, HDFS, Hadoop, Apache Drill and Zookeeper as an alternative NoSQL framework for some features;
- Python as a mean to easily modify and prepare files for import;
- \LaTeX to write the report;
- Github as a versioning and collaboration mean;
- <https://github.com/italia/covid19-opendata-vaccini> as a source of updated and real data about vaccinations in Italy.
- <https://github.com/pcm-dpc/COVID-19> as an alternative source for additional data about Covid-19 in Italy.