SMBUD 2021 - Project work 3

Aman Gabba - 10793117

Andrea Cerasani - 10680486

Giovanni Demasi - 10656704

Pasquale Dazzeo - 10562130

Vlad Marian Cimpeanu - 10606922

POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Problem Specification

The aim of this project was to design, store and query data on a NoSQL DB supporting a data analysis scenario over data about COVID-19 vaccination statistics. The purpose is that of building a comprehensive database of vaccinations.

A given vaccinations dataset has been assigned, with the purpose to pick a time interval of at least 3 months from it and, by using an ElasticSearch installation, import the data, apply the appropriate schema design choices, implement some queries aiming at exploring the data statistics and design a basic visualization dashboard of the results.

1.2 Hypothesis

The assumptions taken into account are the following:

- ...

2 Dataset

2.1 Vaccine administration dataset

The Dataset used for the project is named `"somministrazioni-vaccini-latest.csv"` and it has been downloaded from the `"dati"` folder from the official Italian Government Github repository at the following link : https://github.com/italia/covid19-opendata-vaccini.

2.1.1 Schema

This dataset contains information about administered vaccines in Italy and it is made by the following fields:

Field	**Data type**	**Description**
index	integer	Record identification code
area	string	Code of the delivery region
supplier	string	Complete name of the supplier of the vaccine
administration date	datetime	Administration date of the vaccines
age group	string	Age group to which the subjects to whom the vaccine were administered belong
male count	integer	Number of vaccinations administered to males per day, region and age group
female count	integer	Number of vaccinations administered to females per day, region and age group
first doses	integer	Number of people administered with the first dose
second doses	integer	Number of people administered with the second dose

post infection doses	integer	Number of administrations given to subjects with previous covid-19 infection in the 3-6 month period and who, therefore, conclude the vaccination cycle with a single dose
booster doses	integer	Number of people administered with an additional dose/recall
NUTS1 code	string	European classification of NUTS territorial units: NUTS level 1
NUTS2 code	string	European classification of NUTS territorial units: NUTS level 2
ISTAT region code	integer	ISTAT code of the Region
region name	string	Standard denomination of the area (where necessary bilingual denomination)

The data types written in the table are the 'original' ones, so the ones used by the dataset creator.

The same data types have been used to implement and use the dataset in ElasticSearch because they well represent the different parameters, so no changes were needed, except for the ISTAT region code field. It is better to keep all the ISTAT region codes as keywords instead of numbers for the following reason:

they are numbers but for compatibility reasons they should be considered as keywords. In Kibana, there is the possibility to associate data to regions according to ISTAT code convention, by the way, the format used by Kibana is the following "01, 02, 03, 04, ...", thus, if ISTAT codes are imported as numbers they will not be compatible with that convention as Kibana will find the following codes "1, 2, 3, 4, ...".

As the original dataset does not follow the Kibana convention, it has been adapted through the script `"dataset_cleaner.py"`.

2.2 Istat population dataset

As optional point of this project, analysis has been integrated with another dataset which contains information about the Italian population, like number of people per age range per region or number of people per gender per region.

2.2.1 Schema

...

3 Queries and Commands

In the following chapter all the queries and commands parameters (part of the code to substitute with desired values) will be highlighted with **magenta** bold text.

Some parameters information can be useful for different queries or commands so they are written here to avoid writing them multiple times:

- **Start date** and **End date** are, respectevely, the starting date of a period and the ending date of a period. Dates must be in the following format YYYY-MM-DD and, obviously, End date must be subsequent to Start date.

3.1 Queries

The first eight queries refer only on the vaccination dataset. Instead, the last two queries refer to both vaccination and Istat population dataset.

3.1.1 Delta vaccination per area

For each region, this query returns the percentage of the difference between vaccinations of a given date and its precedent day, calculated with respect to the amount of vaccinations performed the day before. If the vaccinations have increased, the percentage will be positive, negative otherwise.

```
GET vaccinations/_search
{
  "size" : 0,
  "aggs": {
    "group_by_date": {
      "terms": {
        "field": "nome_area"
      },
      "aggs": {
        "today_vaccinations" :{
          "filter": {
            "term" : {
              "data_somministrazione": "now-31d/d"
            }
          },
          "aggs": {
            "amount": {
              "sum": {
                "script": {
                  "source": "doc['sesso_maschile'].value + doc['
    sesso_femminile'].value"
                }
              }
```

```
          }
        }
      },
      "yesterday_vaccinations" : {
        "filter": {
          "term" : {
            "data_somministrazione": "now-32d/d"
          }
        },
        "aggs" : {
          "amount": {
            "sum" :{
              "script": {
                "source": "doc['sesso_femminile'].value + doc['
    sesso_maschile'].value"
              }
            }
          }
        }
      },
      "delta_percentage" : {
        "bucket_script": {
          "buckets_path": {
            "today" : "today_vaccinations>amount",
            "yesterday" : "yesterday_vaccinations>amount"
          },
          "script": "(params.today - params.yesterday) / params.
    yesterday * 100"
        }
      }
    }
  }
}
```

3.1.2 Percentage full covered vaccinations

The following query calculates the percentage of people which has already completed the vaccination cycle. The percentage is calculated with respect to all the vaccinated people, so everyone that has already received at least one dose. The cycle is considered completed if a person:

- is vaccinated with Janssen vaccine.
- has already received the second dose.

```
GET vaccinations/_search
{
  "size" : 0,
```

```
"aggs":{
  "group_by": {
    "date_range": {
      "field": "data_somministrazione",
      "ranges": [
        {
          "from": "2020-12-27",
          "to": "now"
        }
      ]
    },
    "aggs": {
      "sum_first_dose": {
        "sum" :{
          "field" : "prima_dose"
        }
      },
      "sum_second_dose" : {
        "sum" : {
          "field" : "seconda_dose"
        }
      },
      "sum_Janssen": {
        "filter": {
          "term" : {
            "fornitore": "Janssen"
          }
        },
        "aggs": {
          "amount" : {
            "sum" : {
              "field": "prima_dose"
            }
          }
        }
      },
      "full_coverage_percentage" : {
        "bucket_script": {
          "buckets_path": {
            "first_dose": "sum_first_dose",
            "second_dose": "sum_second_dose",
            "janssen_vax": "sum_Janssen>amount"
          },
          "script": "(params.second_dose + params.janssen_vax)/ params.
  first_dose * 100"
        }
      }
    }
  }
}
```

```
}
```

3.1.3 Vaccination trend

This query returns, for each day, all the vaccinations done.

```
GET vaccinations/_search
{
  "size" : 0,
  "aggs": {
    "group_by_date": {
      "date_histogram": {
        "field": "data_somministrazione",
        "interval": "day"
      },
      "aggs": {
        "sum_vaccinations": {
          "sum": {
            "script": {
              "source": "doc['sesso_maschile'].value + doc['
    sesso_femminile'].value"
            }
          }
        }
      }
    }
  }
}
```

3.1.4 Brand administrated vaccines percentage for a given period

The following query, given a period of time, returns the percentage of the administrated vaccines per brand.

```
GET vaccinations/_search
{
  "size" : 0,
  "aggs":{
    "group_by": {
      "date_range": {
        "field": "data_somministrazione",
        "ranges": [
          {
            "from": "Start date",
            "to": "End date"
          }
        ]
      },
```

```
 "aggs": {
   "Janssen_vaccinations": {
     "filter": {
       "term" : {
         "fornitore": "Janssen"
       }
     },
     "aggs": {
       "amount" : {
         "sum" : {
           "script": {
             "source": "doc['prima_dose'].value + doc['seconda_dose
'].value + doc['dose_addizionale_booster'].value"
           }
         }
       }
     }
   },
   "Pfizer_vaccinations": {
     "filter": {
       "term" : {
         "fornitore": "Pfizer/BioNTech"
       }
     },
     "aggs": {
       "amount" : {
         "sum" : {
           "script": {
             "source": "doc['prima_dose'].value + doc['seconda_dose
'].value + doc['dose_addizionale_booster'].value"
           }
         }
       }
     }
   },
   "Moderna_vaccinations": {
     "filter": {
       "term" : {
         "fornitore": "Moderna"
       }
     },
     "aggs": {
       "amount" : {
         "sum" : {
           "script": {
             "source": "doc['prima_dose'].value + doc['seconda_dose
'].value + doc['dose_addizionale_booster'].value"
           }
         }
       }
```

```
      }
    },
    "Astrazeneca_vaccinations": {
      "filter": {
        "term" : {
          "fornitore": "Vaxzevria (AstraZeneca)"
        }
      },
      "aggs": {
        "amount" : {
          "sum" : {
            "script": {
              "source": "doc['prima_dose'].value + doc['seconda_dose
'].value + doc['dose_addizionale_booster'].value"
            }
          }
        }
      }
    },
    "Prifez_Percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Pfizer": "Pfizer_vaccinations>amount",
          "Moderna": "Moderna_vaccinations>amount",
          "Janssen": "Janssen_vaccinations>amount",
          "Astrazeneca": "Astrazeneca_vaccinations>amount"
        },
        "script": "(params.Pfizer)/ (params.Pfizer + params.Moderna +
 params.Janssen + params.Astrazeneca) * 100"
      }
    },
    "Moderna_Percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Pfizer": "Pfizer_vaccinations>amount",
          "Moderna": "Moderna_vaccinations>amount",
          "Janssen": "Janssen_vaccinations>amount",
          "Astrazeneca": "Astrazeneca_vaccinations>amount"
        },
        "script": "(params.Moderna)/ (params.Pfizer + params.Moderna
+ params.Janssen + params.Astrazeneca) * 100"
      }
    },
    "Astrazeneca_Percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Pfizer": "Pfizer_vaccinations>amount",
          "Moderna": "Moderna_vaccinations>amount",
          "Janssen": "Janssen_vaccinations>amount",
          "Astrazeneca": "Astrazeneca_vaccinations>amount"
```

```
            },
            "script": "(params.Astrazeneca)/ (params.Pfizer + params.
    Moderna + params.Janssen + params.Astrazeneca) * 100"
          }
        },
        "Janssen_Percentage" : {
          "bucket_script": {
            "buckets_path": {
              "Pfizer": "Pfizer_vaccinations>amount",
              "Moderna": "Moderna_vaccinations>amount",
              "Janssen": "Janssen_vaccinations>amount",
              "Astrazeneca": "Astrazeneca_vaccinations>amount"
            },
            "script": "(params.Janssen)/ (params.Pfizer + params.Moderna
    + params.Janssen + params.Astrazeneca) * 100"
          }
        }
      }
    }
  }
}
```

3.1.5 Percentages of administrated doses, by dose number, for a given period

This query returns the percentage of first doses, second doses and boosters administrated during the given period.

```
GET vaccinations/_search
{
  "size" : 0,
  "aggs": {
    "group_by_date": {
      "date_range": {
        "field": "data_somministrazione",
        "ranges": [
          {
            "from": "Start date",
            "to": "End date"
          }
        ]
      },
      "aggs": {
        "first_doses": {
          "sum": {
            "field" : "prima_dose"
          }
        },
        "second_doses": {
```

```
        "sum": {
          "field" : "seconda_dose"
        }
      },
      "boosters": {
        "sum": {
          "field" : "dose_addizionale_booster"
        }
      },
      "First_dose_Percentage" : {
        "bucket_script": {
          "buckets_path": {
            "First": "first_doses",
            "Second": "second_doses",
            "Booster": "boosters"
          },
          "script": "(params.First)/ (params.First + params.Second +
    params.Booster) * 100"
        }
      },
      "Second_dose_Percentage" : {
        "bucket_script": {
          "buckets_path": {
            "First": "first_doses",
            "Second": "second_doses",
            "Booster": "boosters"
          },
          "script": "(params.Second)/ (params.First + params.Second +
    params.Booster) * 100"
        }
      },
      "Booster_Percentage" : {
        "bucket_script": {
          "buckets_path": {
            "First": "first_doses",
            "Second": "second_doses",
            "Booster": "boosters"
          },
          "script": "(params.Booster)/ (params.First + params.Second +
    params.Booster) * 100"
        }
      }
    }
  }
}
```

3.1.6 Vaccination percentage per age range for a given period

The following query returns the percentage of vaccinated people per age range during the given period.

```
GET vaccinations/_search
{
  "size" : 0,
  "aggs": {
    "group_by_date": {
      "date_range": {
        "field": "data_somministrazione",
        "ranges": [
          {
            "from": "Start date",
            "to": "End date"
          }
        ]
      },
      "aggs": {
        "teens_range": {
          "filter": {
            "term" : {
              "fascia_anagrafica": "12-19"
            }
          },
          "aggs": {
            "amount" : {
              "sum" : {
                "script": {
                  "source": "doc['sesso_maschile'].value + doc['
    sesso_femminile'].value"
                }
              }
            }
          }
        },
        "20s_range": {
          "filter": {
            "term" : {
              "fascia_anagrafica": "20-29"
            }
          },
          "aggs": {
            "amount" : {
              "sum" : {
                "script": {
                  "source": "doc['sesso_maschile'].value + doc['
    sesso_femminile'].value"
                }
```

```
          }
        }
      }
    },
    "30s_range": {
      "filter": {
        "term" : {
          "fascia_anagrafica": "30-39"
        }
      },
      "aggs": {
        "amount" : {
          "sum" : {
            "script": {
              "source": "doc['sesso_maschile'].value + doc['
sesso_femminile'].value"
            }
          }
        }
      }
    },
    "40s_range": {
      "filter": {
        "term" : {
          "fascia_anagrafica": "40-49"
        }
      },
      "aggs": {
        "amount" : {
          "sum" : {
            "script": {
              "source": "doc['sesso_maschile'].value + doc['
sesso_femminile'].value"
            }
          }
        }
      }
    },
    "50s_range": {
      "filter": {
        "term" : {
          "fascia_anagrafica": "50-59"
        }
      },
      "aggs": {
        "amount" : {
          "sum" : {
            "script": {
              "source": "doc['sesso_maschile'].value + doc['
sesso_femminile'].value"
```

```
          }
        }
      }
    },
    "60s_range": {
      "filter": {
        "term" : {
          "fascia_anagrafica": "60-69"
        }
      },
      "aggs": {
        "amount" : {
          "sum" : {
            "script": {
              "source": "doc['sesso_maschile'].value + doc['
sesso_femminile'].value"
            }
          }
        }
      }
    },
    "70s_range": {
      "filter": {
        "term" : {
          "fascia_anagrafica": "70-79"
        }
      },
      "aggs": {
        "amount" : {
          "sum" : {
            "script": {
              "source": "doc['sesso_maschile'].value + doc['
sesso_femminile'].value"
            }
          }
        }
      }
    },
    "80s_range": {
      "filter": {
        "term" : {
          "fascia_anagrafica": "80-89"
        }
      },
      "aggs": {
        "amount" : {
          "sum" : {
            "script": {
```

```
            "source": "doc['sesso_maschile'].value + doc['
sesso_femminile'].value"
          }
        }
      }
    }
  },
  "90s_range": {
    "filter": {
      "term" : {
        "fascia_anagrafica": "90+"
      }
    },
    "aggs": {
      "amount" : {
        "sum" : {
          "script": {
            "source": "doc['sesso_maschile'].value + doc['
sesso_femminile'].value"
          }
        }
      }
    }
  },
  "teen_percentage" : {
    "bucket_script": {
      "buckets_path": {
        "Teen": "teens_range>amount",
        "20": "20s_range>amount",
        "30": "30s_range>amount",
        "40": "40s_range>amount",
        "50": "50s_range>amount",
        "60": "60s_range>amount",
        "70": "70s_range>amount",
        "80": "80s_range>amount",
        "90": "90s_range>amount"
      },
      "script": "(params.Teen) / (params.Teen + params.20 + params
.30 + params.40 +params.50 + params.60 + params.70 + params.80 +
params.90) * 100"
    }
  },
  "20s_percentage" : {
    "bucket_script": {
      "buckets_path": {
        "Teen": "teens_range>amount",
        "20": "20s_range>amount",
        "30": "30s_range>amount",
        "40": "40s_range>amount",
        "50": "50s_range>amount",
```

```
          "60": "60s_range>amount",
          "70": "70s_range>amount",
          "80": "80s_range>amount",
          "90": "90s_range>amount"
        },
        "script": "(params.20)/ (params.Teen + params.20 + params.30
+ params.40 +params.50 + params.60 + params.70 + params.80 + params
.90) * 100"
      }
    },
    "30s_percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Teen": "teens_range>amount",
          "20": "20s_range>amount",
          "30": "30s_range>amount",
          "40": "40s_range>amount",
          "50": "50s_range>amount",
          "60": "60s_range>amount",
          "70": "70s_range>amount",
          "80": "80s_range>amount",
          "90": "90s_range>amount"
        },
        "script": "(params.30)/ (params.Teen + params.20 + params.30
+ params.40 +params.50 + params.60 + params.70 + params.80 + params
.90) * 100"
      }
    },
    "40s_percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Teen": "teens_range>amount",
          "20": "20s_range>amount",
          "30": "30s_range>amount",
          "40": "40s_range>amount",
          "50": "50s_range>amount",
          "60": "60s_range>amount",
          "70": "70s_range>amount",
          "80": "80s_range>amount",
          "90": "90s_range>amount"
        },
        "script": "(params.40)/ (params.Teen + params.20 + params.30
+ params.40 +params.50 + params.60 + params.70 + params.80 + params
.90) * 100"
      }
    },
    "50s_percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Teen": "teens_range>amount",
```

```
          "20": "20s_range>amount",
          "30": "30s_range>amount",
          "40": "40s_range>amount",
          "50": "50s_range>amount",
          "60": "60s_range>amount",
          "70": "70s_range>amount",
          "80": "80s_range>amount",
          "90": "90s_range>amount"
        },
        "script": "(params.50)/ (params.Teen + params.20 + params.30
+ params.40 +params.50 + params.60 + params.70 + params.80 + params
.90) * 100"
      }
    },
    "60s_percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Teen": "teens_range>amount",
          "20": "20s_range>amount",
          "30": "30s_range>amount",
          "40": "40s_range>amount",
          "50": "50s_range>amount",
          "60": "60s_range>amount",
          "70": "70s_range>amount",
          "80": "80s_range>amount",
          "90": "90s_range>amount"
        },
        "script": "(params.60)/ (params.Teen + params.20 + params.30
+ params.40 +params.50 + params.60 + params.70 + params.80 + params
.90) * 100"
      }
    },
    "70s_percentage" : {
      "bucket_script": {
        "buckets_path": {
          "Teen": "teens_range>amount",
          "20": "20s_range>amount",
          "30": "30s_range>amount",
          "40": "40s_range>amount",
          "50": "50s_range>amount",
          "60": "60s_range>amount",
          "70": "70s_range>amount",
          "80": "80s_range>amount",
          "90": "90s_range>amount"
        },
        "script": "(params.70)/ (params.Teen + params.20 + params.30
+ params.40 +params.50 + params.60 + params.70 + params.80 + params
.90) * 100"
      }
    },
```

```
        "80s_percentage" : {
          "bucket_script": {
            "buckets_path": {
              "Teen": "teens_range>amount",
              "20": "20s_range>amount",
              "30": "30s_range>amount",
              "40": "40s_range>amount",
              "50": "50s_range>amount",
              "60": "60s_range>amount",
              "70": "70s_range>amount",
              "80": "80s_range>amount",
              "90": "90s_range>amount"
            },
            "script": "(params.80)/ (params.Teen + params.20 + params.30
    + params.40 +params.50 + params.60 + params.70 + params.80 + params
    .90) * 100"
          }
        },
        "90+_percentage" : {
          "bucket_script": {
            "buckets_path": {
              "Teen": "teens_range>amount",
              "20": "20s_range>amount",
              "30": "30s_range>amount",
              "40": "40s_range>amount",
              "50": "50s_range>amount",
              "60": "60s_range>amount",
              "70": "70s_range>amount",
              "80": "80s_range>amount",
              "90": "90s_range>amount"
            },
            "script": "(params.90)/ (params.Teen + params.20 + params.30
    + params.40 +params.50 + params.60 + params.70 + params.80 + params
    .90) * 100"
          }
        }
      }
    }
  }
}
```

3.1.7 Query 7

The following query...

```
CODE
```

3.1.8 Query 8

The following query...

```
CODE
```

3.1.9 Query 9

The following query...

```
CODE
```

3.1.10 Query 10

The following query...

```
CODE
```

3.2 Commands

3.2.1 Command 1

The following query...

```
CODE
```

3.2.2 Command 2

The following query...

```
CODE
```

4 Dashboard description

The Kibana Dashboard is made by different section, each focusing on a sepxific analysis. Here there is a brief description of each part is given.

4.1 Percentage of Vaccines brand per age range

In this section a pie chart has been used to show two different things: in the inner layer there is the percentage of people vaccinated per age range, in the outer layer there is the percentage of brand vaccines used per age range.

4.2 First doses administrated

This histogram shows the trend of the total amount of first doses administrated and it division per vaccine brand.

4.3 Vaccines administrated per gender

This section shows the trend of the amount of vaccines administrated to men and to women

4.4 Vaccines administrated per region

Here a map has been used to show the percentage of first doses, second doses and boosters administrated from the beginning of the pandemic per region, with respect to its total population.

4.5 Description

This section has been used to briefly describe all the different graphs of the dashboard.

5 User guide

5.1 Import data

CHECH - SEZIONE SCRITTA VELOCEMENTE, da rifinire

5.1.1 Import vaccinations dataset

After having opened Kibana, it is necessary to click on the `"upload file"` button present in the home page. Now the file named `"cleaned_data.csv"` must be dragged and dropped in the opened page.

After that, the `"import"` button must be clicked. In the new page, in the advanced setting section, it is necessary to use the following index name `"vaccinations"`. In the mapping section, at line number 16, the word long must be replaced with keyword.

In the Ingest pipeline section, lines from number 36 to number 42 (both included) must be deleted. Then, by clicking the import button, vaccinations data will be imported succesfully.

5.1.2 Import Istat population dataset

Qui piu o meno è come quello sopra, anche i suoi codici istat vanno adattati e andrà messo keyword nel mapping

5.2 Import dashboard

After having opened Kibana it is necessary to go in the `"Stack management"` section. From this page, click the `"Saved objects"` button present in the left bar, in the kibana section. In the new page click the `"Import data"` button and then upload the `"dashboard.ndjson"` file.

It is important to select as index the one referred to (?????? probabilmente ci andra quello misto tra istat e vaccini ?????). After that just click the `"Import"` button; the dashboard will now be visible in `"Dashboard"` section.

6 Conclusion

Some interesting conclusions can be drawn from the development of this project:

Elasticsearch and Kibana are a perfect match to make different type of analysis about trends even by using a big amount of data.

Kibana makes Elasticsearch queries output really simple to understand and visualize, and it can be really helpful because in this way the analysis results can be understand even by those who doesn't have any knowledge about computer science.

7 References and Sources

- Elastic Guide: https://www.elastic.co/guide/index.html
- Italian Government repository: https://github.com/italia/covid19-opendata-vaccini