Covid-19 Data Analysis

Introduction:

The Covid-19 pandemic has significantly impacted global health, economies, and societies. understanding the spread and impact of the virus is crucial for public health measures and prevent future outbreaks.so in this project we aim to analyse Covid-19 data from European Centre for Disease Prevention and Control (ECDC) website. Which consists of Europe countries data spanning from January 2020 to October 2020 using Azure cloud services and the main goal is to identify the trends of cases, deaths, hospital admissions (both daily and weekly), ICU admissions and number of Covid-19 tests conducted in each country, confirmed cases across the Europe to gain the valuable insights from the data.

Problem Description:

The challenge is the complexity of covid-19 data generated from various sources. Which include data from cases and deaths, hospital_admissions, testing and population csv files and they require data cleaning ,transformation to derive the insights. The main key issues are:

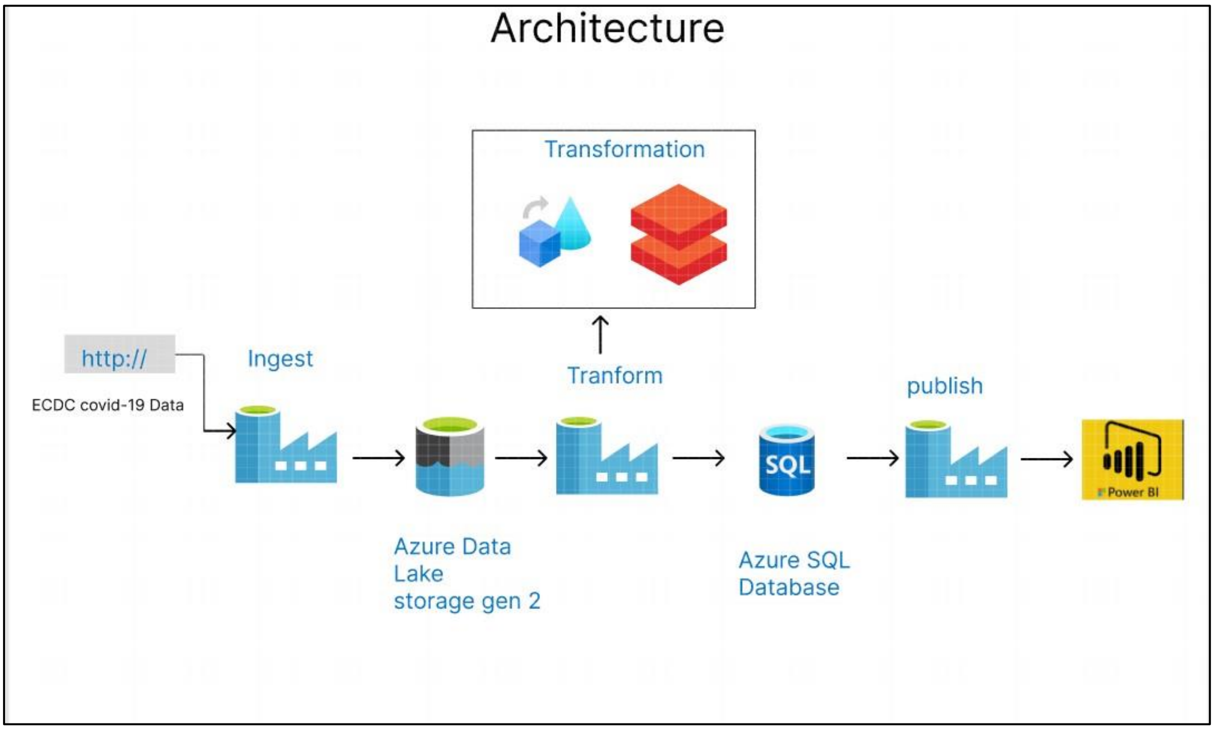
1) Data Extraction: Covid-19 data from ECDC website is uploaded to GitHub. The task is to develop a single pipeline in Azure Data Factory to extract the all data using HTTP connector and pipeline is parameterized using json configuration file which includes sourceBaseURL, sourceRelativeURL, sinkFileName as parameters finally the files are ingested to azure data lake.
2) Transformation: Azure dataflow is developed to process cases and deaths, hospital admissions and testing csv files which includes aggregations, lookup, pivot, filter, join transformation and python notebook is developed in azure databricks for population csv file.
3) Load: single pipeline is developed to load processed files from data lake to azure SQL database. This pipeline is parameterized using json configuration file which includes sourcefilepath, schema, table.
4) Reporting: Dashboards are developed in Power BI to visualize the trends of cases, deaths, hospital admissions , ICU admissions, testing , population of each country with percentage of each age group across the Europe provide detailed insights of the processed data.

Novelty:

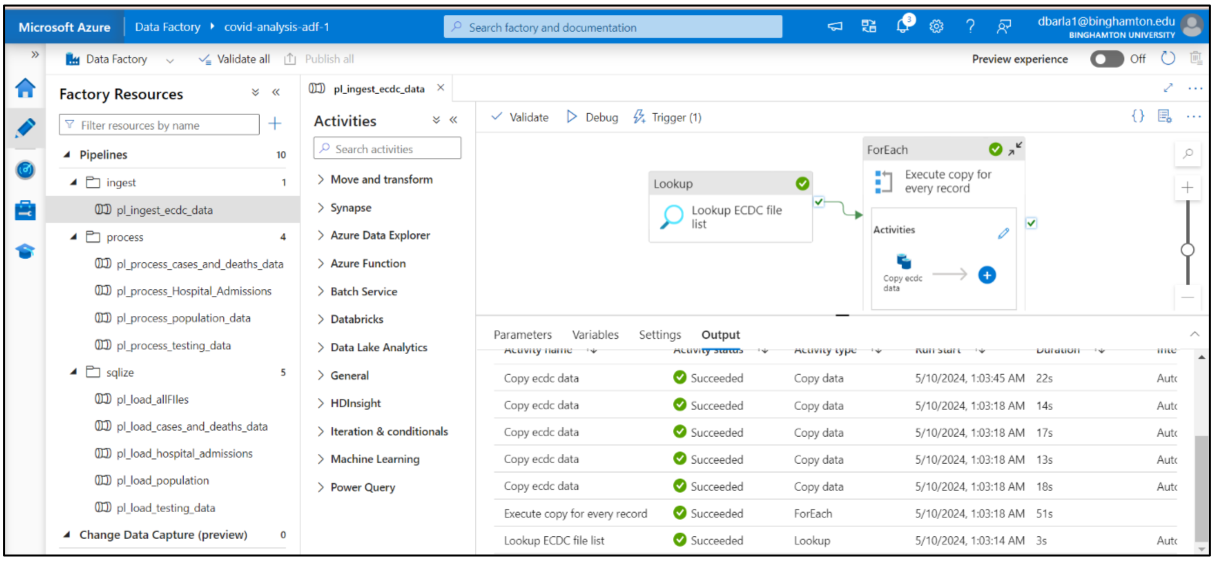
The novelty of this project is its comprehensive approach to Covid-19 data analysis and main key aspects are:

1) Leveraging Azure Cloud Services: The utilization of Azure Data Factory, Azure Data Lake, Azure Databricks and Azure SQL Database provides scalability, flexibility, and efficiency in processing large volumes of data.
2) Integration of multiple Data Sources & Pipelines: This project integrates data from multiple sources cases and deaths, hospital admissions, testing, population files provide deeper insights of data and parametrizing pipeline with json config file eliminates multiple pipelines to integrate into single pipeline for extracting and loading the data improves efficiency.
3) Interactive Visualization with Power BI: The dynamic dashboard helps to interact each visualization with other visualization which enhance user engagement and provide deep analysis.

Design and Implementation:

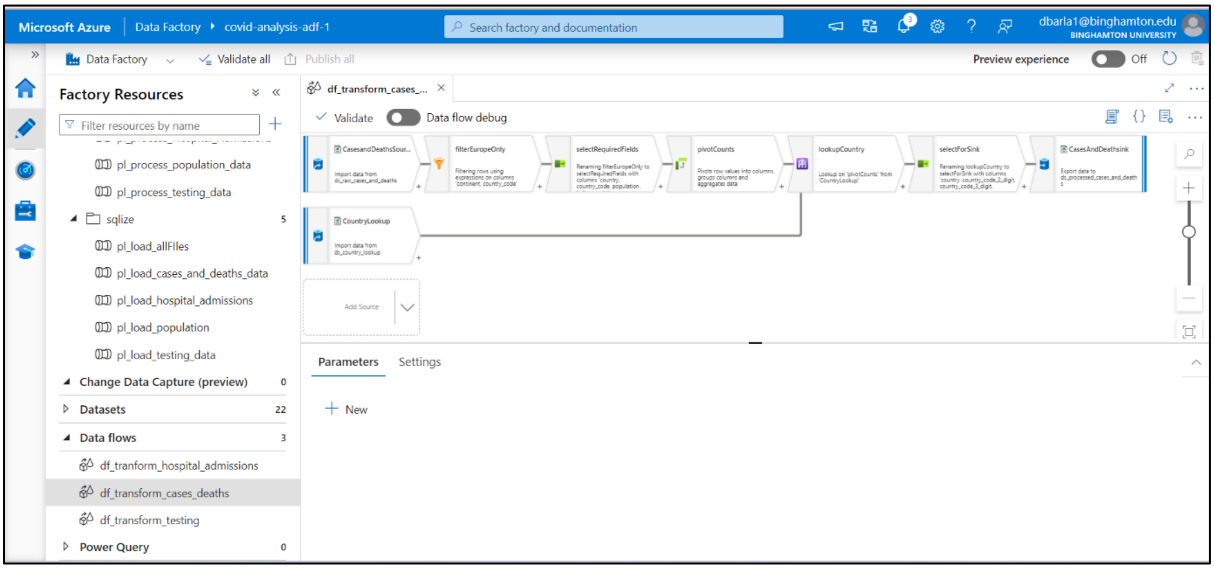


This is the end to end architecture of the project.

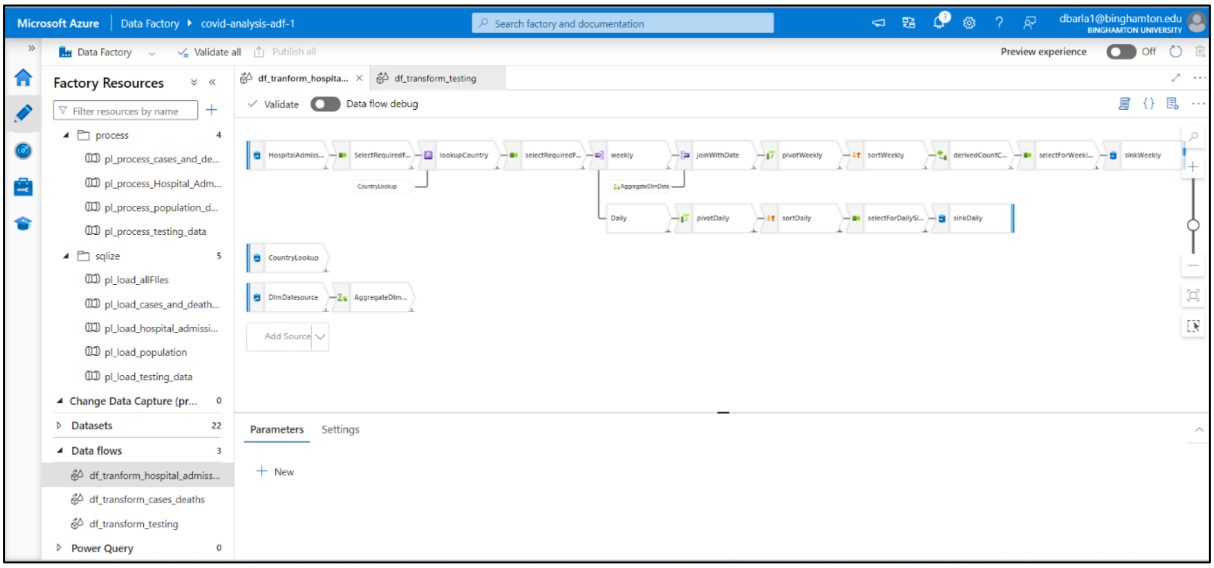


Activity name	Activity status	Activity type	Run start	Duration	Inte
Copy ecdc data	✓ Succeeded	Copy data	5/10/2024, 1:03:45 AM	22s	Autc
Copy ecdc data	✓ Succeeded	Copy data	5/10/2024, 1:03:18 AM	14s	Autc
Copy ecdc data	✓ Succeeded	Copy data	5/10/2024, 1:03:18 AM	17s	Autc
Copy ecdc data	✓ Succeeded	Copy data	5/10/2024, 1:03:18 AM	13s	Autc
Copy ecdc data	✓ Succeeded	Copy data	5/10/2024, 1:03:18 AM	18s	Autc
Execute copy for every record	✓ Succeeded	ForEach	5/10/2024, 1:03:18 AM	51s	
Lookup ECDC file list	✓ Succeeded	Lookup	5/10/2024, 1:03:14 AM	3s	Autc

The first step is to extract data from github to azure data lake and the pipeline uses lookup activity to get the json config file which provides sourceBaseURL, sourceRelativeURL, sinkFileName for each file and for loop execute for each file next copy activity gets the parameters to copy data to specified path in Azure data lake.

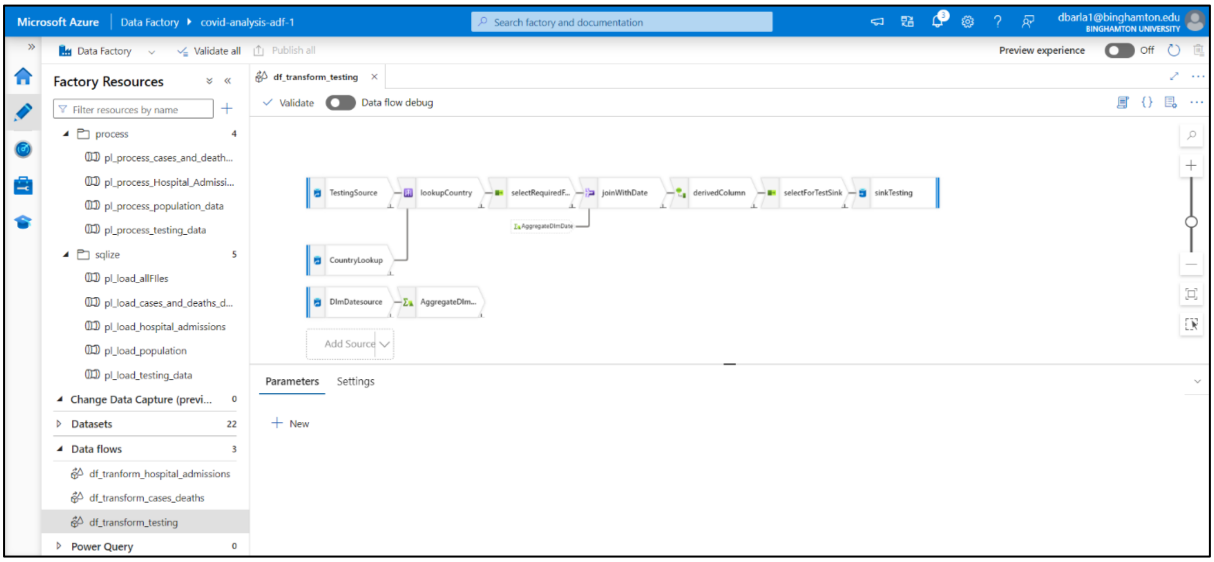

Microsoft Azure
Data Factory
covid-analysis-adf-1
Search factory and documentation
dbarla1@binghamton.edu
BINGHAMTON UNIVERSITY
Data Factory
Validate all
Publish all
Preview experience
Off
Factory Resources
Filter resources by name
pl_process_population_data
pl_process_testing_data
sqlize 5
pl_load_allFIles
pl_load_cases_and_deaths_data
pl_load_hospital_admissions
pl_load_population
pl_load_testing_data
Change Data Capture (preview) 0
Datasets 22
Data flows 3
df_tranform_hospital_admissions
df_transform_cases_deaths
df_transform_testing
Power Query 0
df_transform_cases_...
Validate
Data flow debug
CasesandDeathsSour...
Import data from ds_raw_cases_and_deaths
filterEuropeOnly
Filtering rows using expressions on columns 'continent, country_code'
selectRequiredFields
Renaming filterEuropeOnly to selectRequiredFields with columns 'country, country_code, population,
pivotCounts
Pivots row values into columns, groups columns and aggregates data
lookupCountry
Lookup on 'pivotCounts' from 'CountryLookup'
selectForSink
Renaming lookupCountry to selectForSink with columns 'country, country_code_2_digit, country_code_3_digit
CasesAndDeathsink
Export data to ds_processed_cases_and_deaths
CountryLookup
Import data from ds_country_lookup
Add Source
Parameters
Settings
New

This is the data flow for cases_deaths which include transformations of filter the data for Europe continent, select required fields , pivot the data on indicator column to get confirmed cases, deaths column and inner join with country file on country column to get the country_code_2_digit,country_code_3_digit columns and finally select required columns, dropping the duplicate columns to get ingested into processed container in azure data lake storage account.

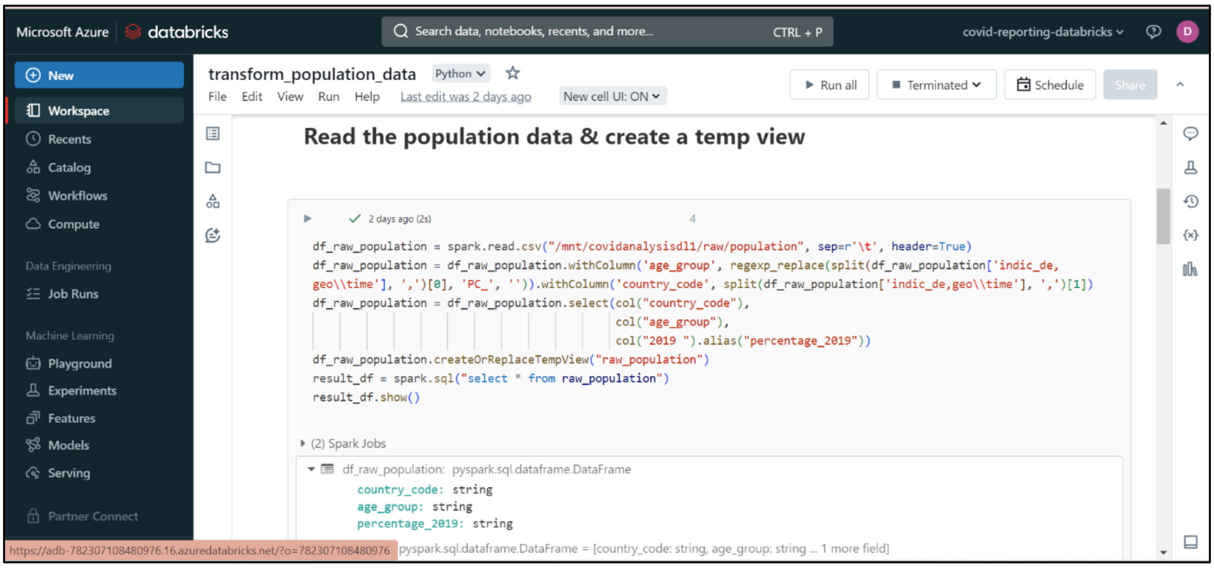


This is dataflow for hospital admissions which include select required columns, joining with country file on country column to get counry_2_digit,3_digit code, population columns next is conditional split on indicator column to filter weekly,daily hospital, icu admissions followed by joining with dimdate source to get week start, end dates and weekly data gets pivoted on inidicator get weekly hospital ,icu admissions followed by sorting in descending order on reported date to get latest data , derived column is round the pivoted columns to decimal (20,2) and sink transformation ingests file to hospital admissions weekly folder in data lake.

Daily hospital admissions dataflow has similar to weekly hospital admissions but the pivoted columns are on daily hospital, icu admissions next sorting data on reporting date in descending order finally sink transformation ingests file to hospital admissions daily folder in data lake.



This is testing data flow which include lookup country to get country 2digit,3digit codes , followed by selecting required columns and joining with Dimdate to get week start, end dates , rounding off the testing rate,positivity rate columns to two digit precision, removing duplicate columns and ingesting data to testing folder in data lake.



Read the population data & create a temp view

```
df_raw_population = spark.read.csv("/mnt/covidanalysisdl1/raw/population", sep=r'\t', header=True)
df_raw_population = df_raw_population.withColumn('age_group', regexp_replace(split(df_raw_population['indic_de,
geo\\time'], ',')[0], 'PC_', '')).withColumn('country_code', split(df_raw_population['indic_de,geo\\time'], ',')[1])
df_raw_population = df_raw_population.select(col("country_code"),
                                             col("age_group"),
                                             col("2019 ").alias("percentage_2019"))
df_raw_population.createOrReplaceTempView("raw_population")
result_df = spark.sql("select * from raw_population")
result_df.show()
```

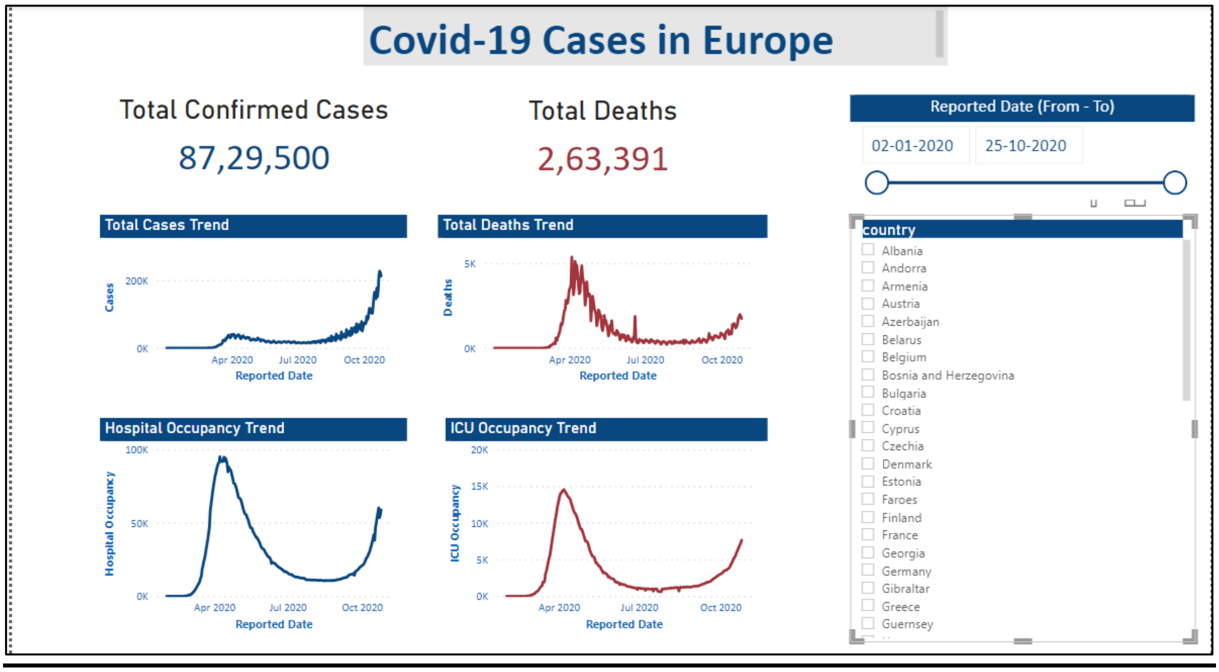
```
df_raw_population:  pyspark.sql.dataframe.DataFrame
    country_code: string
    age_group: string
    percentage_2019: string
pyspark.sql.dataframe.DataFrame = [country_code: string, age_group: string ... 1 more field]
```

This is Azure databricks for transforming population data initially cluster is created based on our requirements and mount the storage containers raw, processed, lookup using python notebook to read, write data on the data lake.

Python notebook is developed to read population data from data lake to create temporary view ‘raw_population’ and then split ‘indic_de,geo\time’ to get country code, age group, percentage_2019 to get data in 2019 year and pivot the age group to get distribution of population in each age group followed by inner join with country lookup to get population of each country, 3 digit country code and finally population data is mounted to processed container in azure data lake.

Pipeline is developed in azure data factory to run the notebook and all the necessary linked service , dataset are created for notebook activity.

Reporting:



Covid-19 Cases in Europe

Total Confirmed Cases

87,29,500

Total Deaths

2,63,391

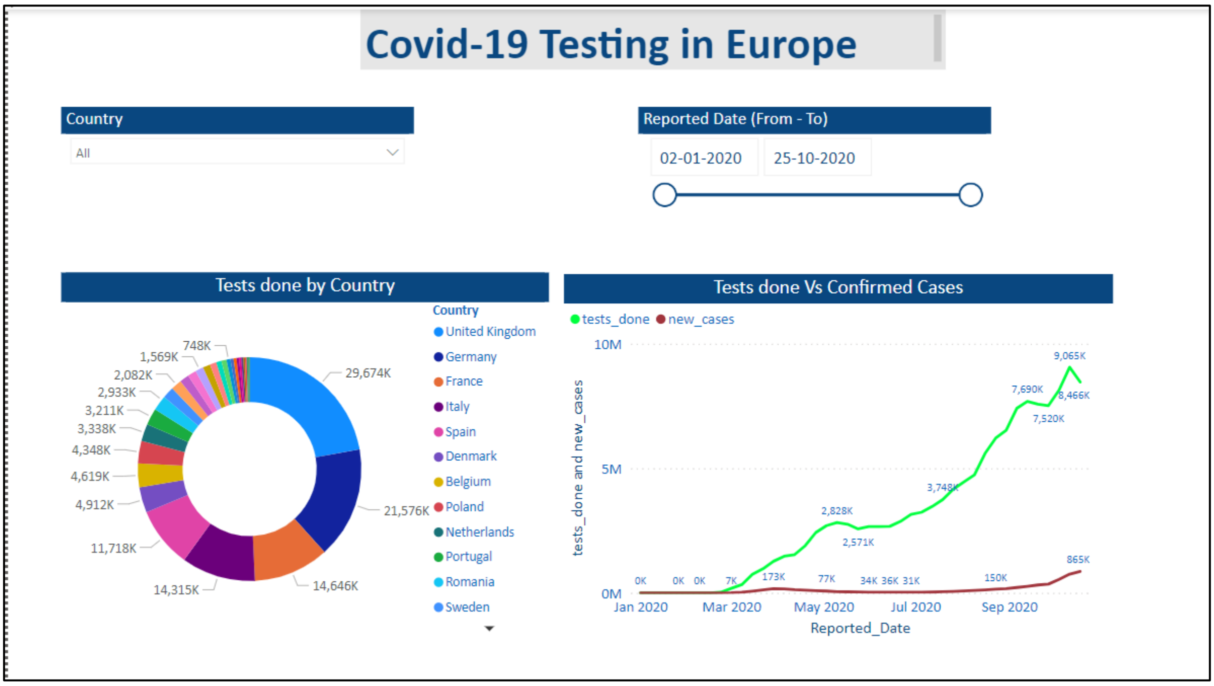
Reported Date (From - To)

02-01-2020 25-10-2020

country

□ Albania
□ Andorra
□ Armenia
□ Austria
□ Azerbaijan
□ Belarus
□ Belgium
□ Bosnia and Herzegovina
□ Bulgaria
□ Croatia
□ Cyprus
□ Czechia
□ Denmark
□ Estonia
□ Faroes
□ Finland
□ France
□ Georgia
□ Germany
□ Gibraltar
□ Greece
□ Guernsey

This dashboard represent the trend of cases, deaths , hospital , ICU occupancy across the Europe and we can filter data for each country and for a particular time period using slicer which makes an interactive dashboard.



Covid-19 Testing in Europe

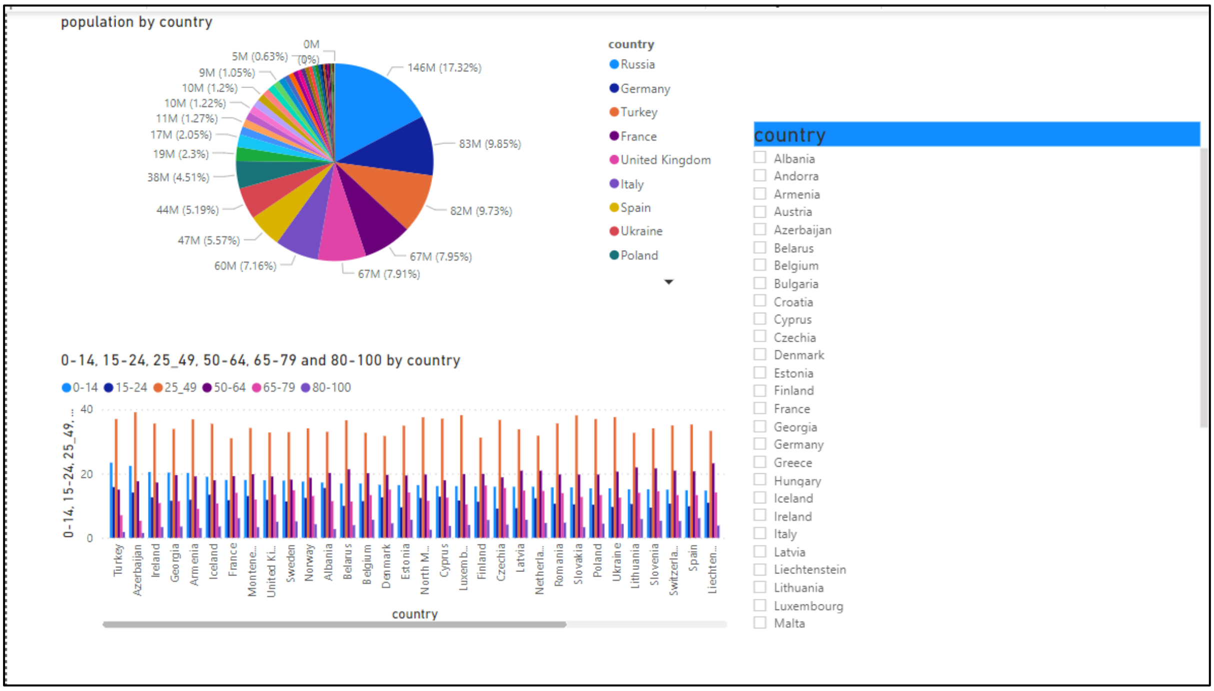
Country

All

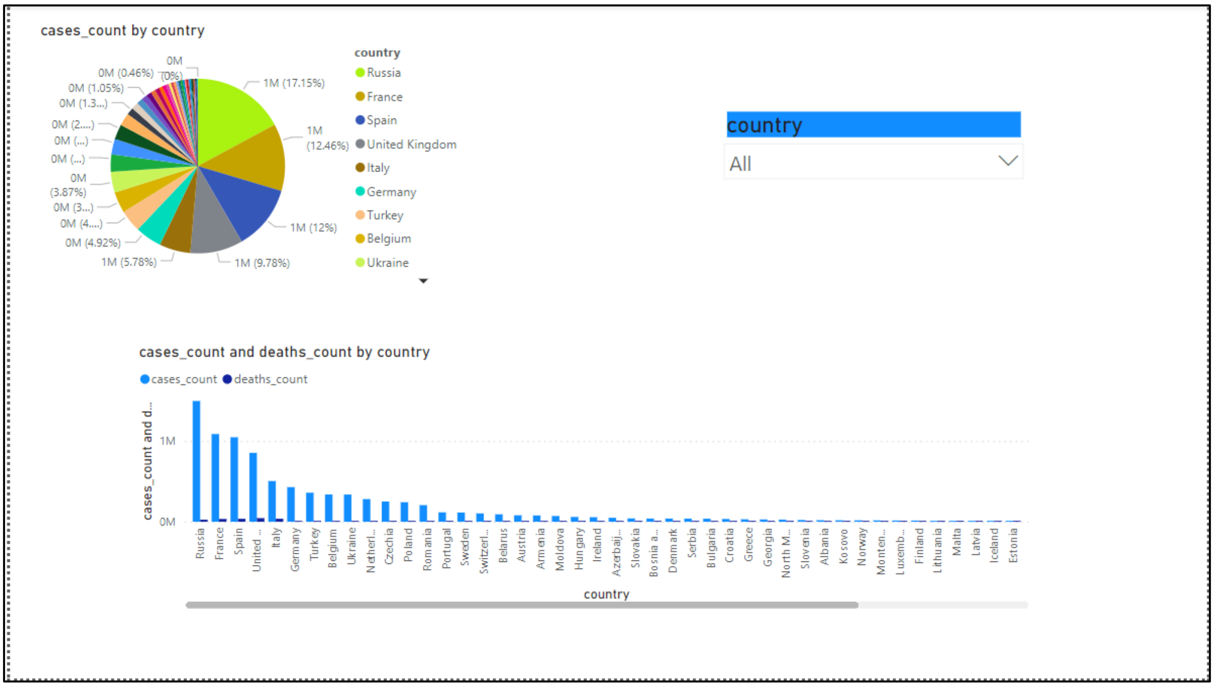
Reported Date (From - To)

02-01-2020 25-10-2020

This dashboard represents test being done in each country, number of confirmed cases in it and we can filter data based on country and reported date using slicer.



This dashboard represents population across each country and percentage of each age group in the country. We can filter the data based on the country and also we can compare data between two or more countries using the slicer.



This dashboard represents the cases count, deaths count in all the countries and we can filter the data by each country using slicer also compare the data between two countries and analyse the data.

Evaluation:

By analyzing the data the covid cases are at peak in April 2020 and then gradually decreased followed by rise in October 2020 across the Europe. The Russia has highest number of cases, deaths count because the population is also highest across Europe as we can see in population dashboard.