2021

Big Data Architecture & Governance

Projected sales of main products in 2013
Distribution of market share among the major industry players
Share of market activity
Projected sales of main products in 2013
Passive market share
25%
75%

Northeastern University

Assignment Name:

Group Assignment

Student Names:

Navaneeta Naik

Nikunj Doshi

Yu Ren

1. Contents

2. Assignment

2.1. Case

Each team should select a dataset to analyze and build an analytical dashboard as a Proof-of-concept to illustrate the value of data driven analytics. You need to present your dataset.

.

2.2. Assignment Goals

To work with datasets, Perform/Create:

- Group Assignment/Project on Velero with below mentioned activities:
  - Tasks/ Project Short/ Long Form/ Group Allocation/ Timesheet/ Issues & Risks.
- Data Profiling – Using Python profiling library, describe your understanding of the data.
- Data Wrangling and Cleansing - Pandas/Alteryx/XSV
  - Filtering and Aggregating if needed.
  - Missing value handling.
  - Deriving additional columns from existing datasets if needed.
  - Cleaning (removing blank spaces, formatting dates, Capitalizing etc.) .
- Database Installation: Install NEO4J database .
- Data Mapping and Integration to your Database for the Entire Dataset.
- Business and Technical Metadata – develop business term list describing all the data elements available in the file.
- Data Validation and Data Visualization using Python – Validate the data using python and provide a dashboard using python visualization libraries.
- System Integration and User Acceptance Testing - Test Cases – describe your validation & testing process.
- Risks/Issues of project.
- Describe challenges encountered and how you resolved them.
- End User Instructions (Steps to run your Dashboard) – provide a full description how to run your process:
  - Database Creation and load.
  - Visualization interpretation - describe information regarding your findings.

2.2.1. VISUALIZATION DELIVERABLES

Once you wrangle/clean/join/integrate the data, import the data into **NEO4J** and illustrate how to use the appropriate graph to illustrate various aspects of analysis.

Questions to consider:

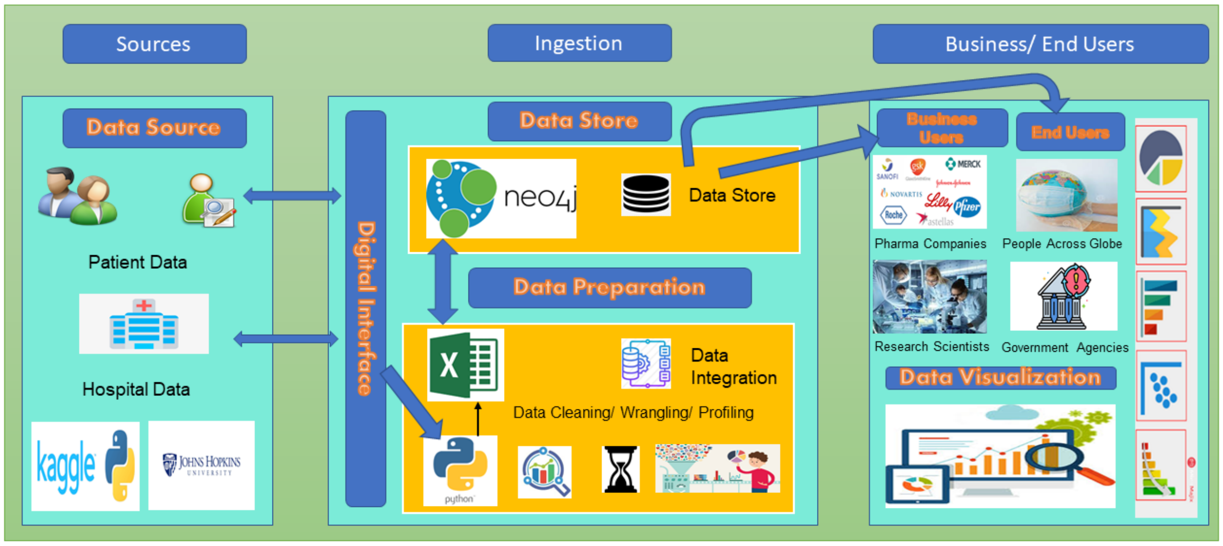
- Columns used for dimensions, and columns that are used for measurement.
- How would you generate new dimensions?
- Who would use this dashboard and how they benefit from your dashboard?
- What value would be generated using this dashboard?

2.2.2. OTHER DELIVERABLES

- Presentation of the entire work from the first step till the dashboards including the Velero screenshots.
- Business and technical metadata presentation – Identifying all available business terms and extracting related technical metadata.
- Complete explanation of the dashboard and usability.
- Complete instruction as how to implement and run the database load, technical meta data extraction, and dashboard.

3. Documentation

3.1. Vision Diagram



Data is being collected from data sources such as Patients and Hospitals and those have been compiled in the dataset on Kaggle through John Hopkins University

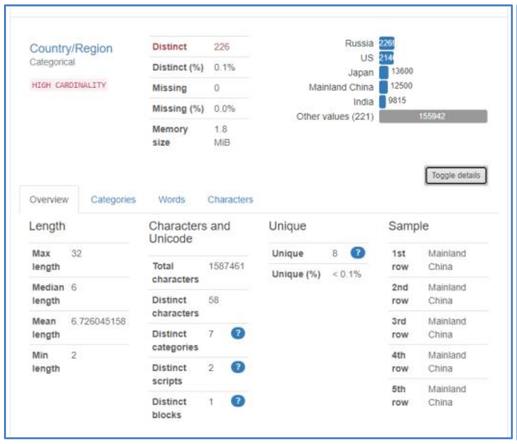
Before the data is being sent to the Data Store in Neo4j, it needs to clean, wrangled and profiled extensively which has been done using Python and Pandas Profiling Library.

Then, Data Modelling and Data Integration is done on the Neo4j database.

Visualizations we get after analyzing the data will be used by the Business Users such as Pharma Companies and the Research Scientists as well as End Users such as Government agencies and the people across the globe to track covid cases and implementation measures.

3.2. Data Wrangling and Cleansing

- Data Wrangling & cleaning is done to make use of the raw data for analytics
- Data Wrangling & Cleansing processes were performed using Pandas



Country/Region
Categorical
HIGH CARDINALITY

Distinct	226
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	1.8 MiB

Russia	226
US	214
Japan	13600
Mainland China	12500
India	9815
Other values (221)	155942

Toggle details

Overview Categories Words Characters

Length

Max length	32
Median length	6
Mean length	6.726045158
Min length	2

Characters and Unicode

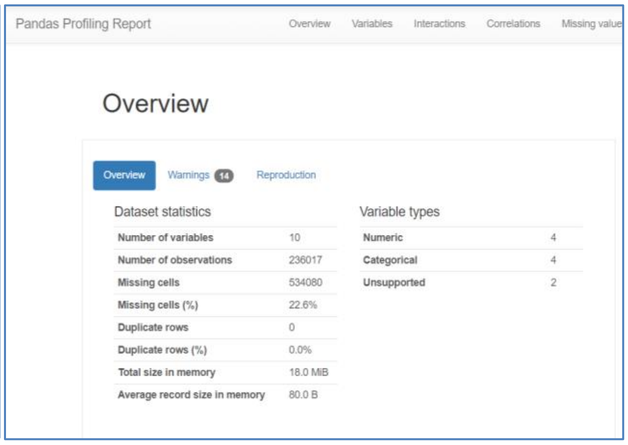
Total characters	1587461
Distinct characters	58
Distinct categories	7
Distinct scripts	2
Distinct blocks	1

Unique

Unique	8
Unique (%)	< 0.1%

Sample

1st row	Mainland China
2nd row	Mainland China
3rd row	Mainland China
4th row	Mainland China
5th row	Mainland China



Pandas Profiling Report

Overview Variables Interactions Correlations Missing value

Overview

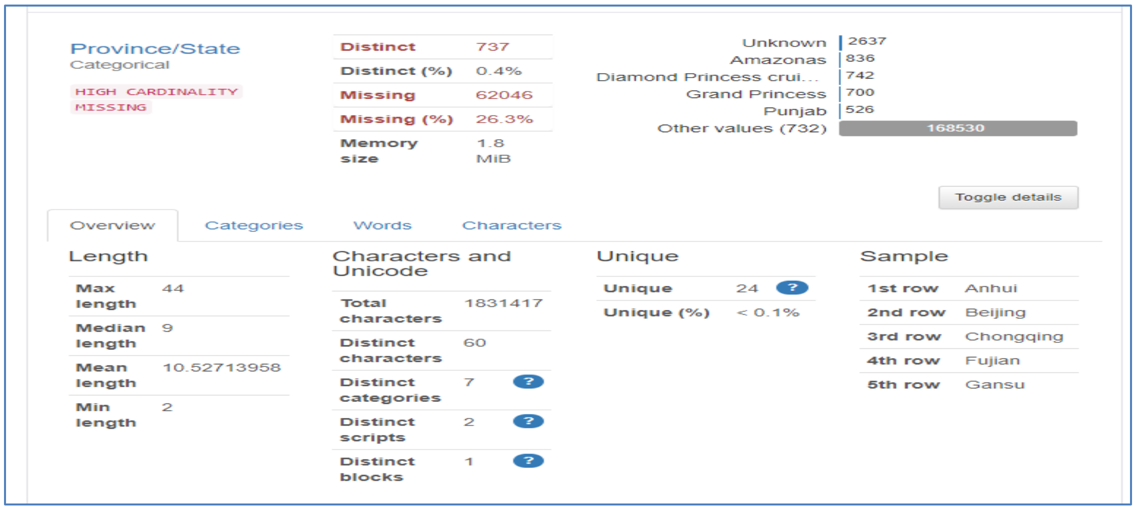
Overview Warnings 14 Reproduction

Dataset statistics

Number of variables	10
Number of observations	236017
Missing cells	534080
Missing cells (%)	22.6%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	18.0 MiB
Average record size in memory	80.0 B

Variable types

Numeric	4
Categorical	4
Unsupported	2



Province/State
Categorical
HIGH CARDINALITY
MISSING

Distinct	737
Distinct (%)	0.4%
Missing	62046
Missing (%)	26.3%
Memory size	1.8 MiB

Unknown	2637
Amazonas	836
Diamond Princess crui…	742
Grand Princess	700
Punjab	526
Other values (732)	168530

Toggle details

Overview Categories Words Characters

Length

Max length	44
Median length	9
Mean length	10.52713958
Min length	2

Characters and Unicode

Total characters	1831417
Distinct characters	60
Distinct categories	7
Distinct scripts	2
Distinct blocks	1

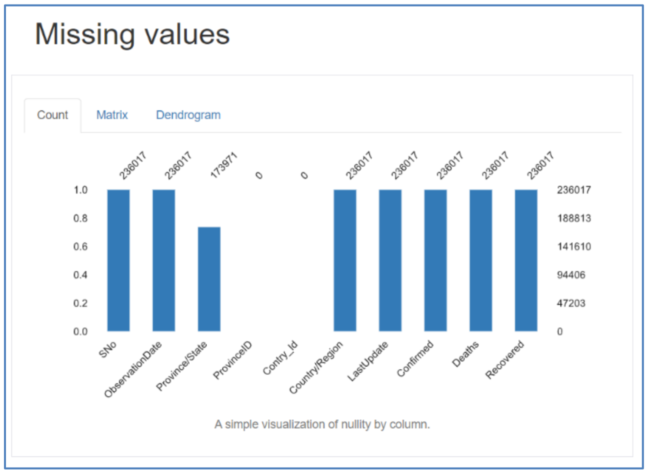
Unique

Unique	24
Unique (%)	< 0.1%

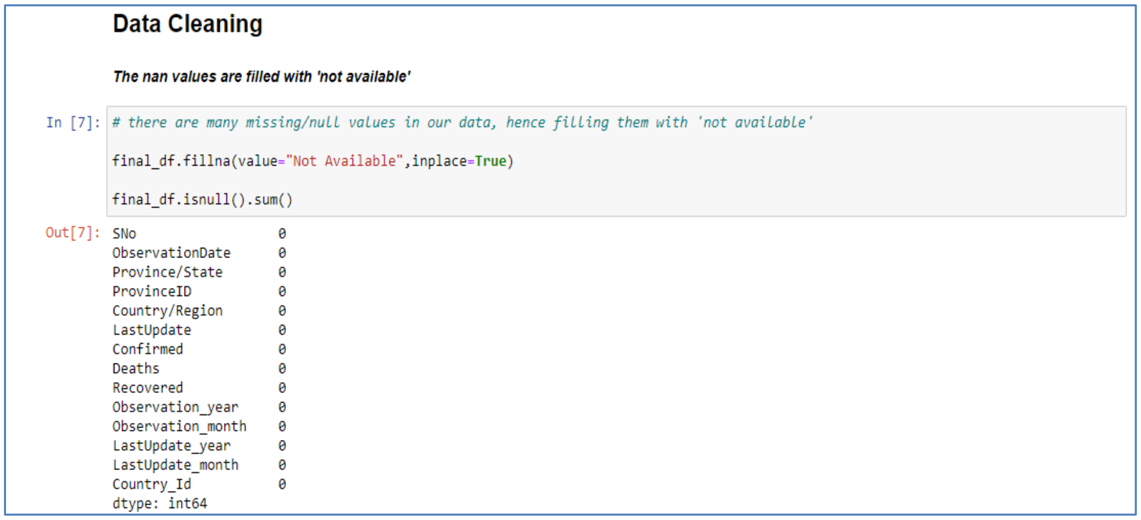
Sample

1st row	Anhui
2nd row	Beijing
3rd row	Chongqing
4th row	Fujian
5th row	Gansu

- Performing data profiling on our dataset gave us an insight on the missing values in our data and other details in our dataset which helped us understand our data



- Handling Missing Values:
    1. All the missing values in Province/State column were replaced with “not available”.
    2. All other columns did not have any missing values



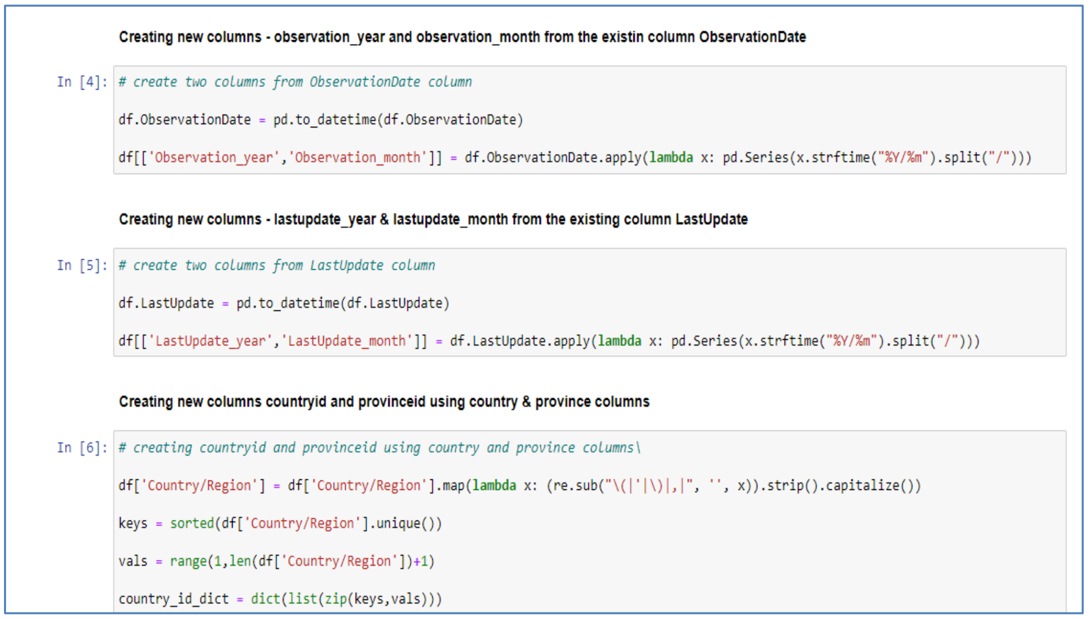
Data Cleaning

The nan values are filled with 'not available'

```
In [7]: # there are many missing/null values in our data, hence filling them with 'not available'

        final_df.fillna(value="Not Available",inplace=True)

        final_df.isnull().sum()

Out[7]: SNo                  0
        ObservationDate      0
        Province/State       0
        ProvinceID           0
        Country/Region       0
        LastUpdate           0
        Confirmed            0
        Deaths               0
        Recovered            0
        Observation_year     0
        Observation_month    0
        LastUpdate_year      0
        LastUpdate_month     0
        Country_Id           0
        dtype: int64
```

- New columns were generated using the existing columns.
    1. Date of Observation was used to derive year of observation and Month of observation columns
    2. Country & Province columns were used to derive unique id numbers country id and province id which would then identify unique countries or states

3. Last Update column was used to derive the last updated month and last updated year columns



Creating new columns - observation_year and observation_month from the existin column ObservationDate

```
In [4]: # create two columns from ObservationDate column

df.ObservationDate = pd.to_datetime(df.ObservationDate)

df[['Observation_year','Observation_month']] = df.ObservationDate.apply(lambda x: pd.Series(x.strftime("%Y/%m").split("/")))
```

Creating new columns - lastupdate_year & lastupdate_month from the existing column LastUpdate

```
In [5]: # create two columns from LastUpdate column

df.LastUpdate = pd.to_datetime(df.LastUpdate)

df[['LastUpdate_year','LastUpdate_month']] = df.LastUpdate.apply(lambda x: pd.Series(x.strftime("%Y/%m").split("/")))
```

Creating new columns countryid and provinceid using country & province columns

```
In [6]: # creating countryid and provinceid using country and province columns\

df['Country/Region'] = df['Country/Region'].map(lambda x: (re.sub("\(|'|\)|,|", '', x)).strip().capitalize())

keys = sorted(df['Country/Region'].unique())

vals = range(1,len(df['Country/Region'])+1)

country_id_dict = dict(list(zip(keys,vals)))
```

3.3. Database Installation

- Install Neo4j Desktop
- Create Database:

    1. Create a new project named “CSYE7250Final”.

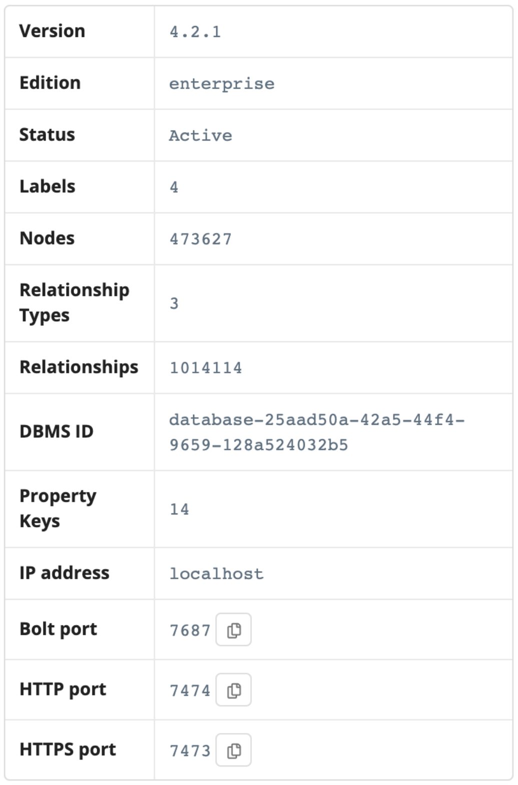
    2. Add a new local DBMS named “COVID19”. The default username is ‘neo4j’. Set the password ‘123456’.

- Import Data Source:

    1. Open the folder and enter import directory.

    2. Paste the CSV file to this directory and change it name ‘COVID19.csv’.

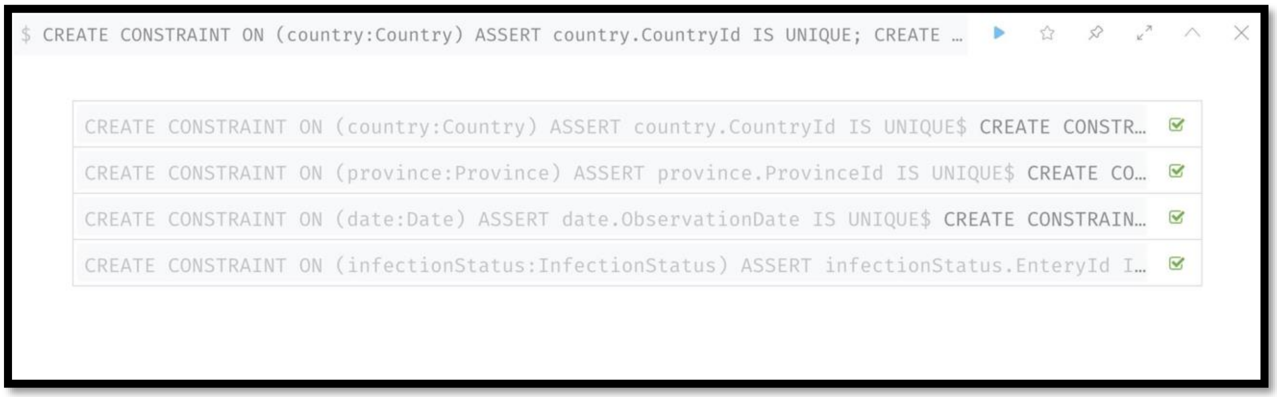
- Install APOC in plugins of the database for exporting the metadata.
- Start the database
- Database Information:



Version	4.2.1
Edition	enterprise
Status	Active
Labels	4
Nodes	473627
Relationship Types	3
Relationships	1014114
DBMS ID	database-25aad50a-42a5-44f4-9659-128a524032b5
Property Keys	14
IP address	localhost
Bolt port	7687
HTTP port	7474
HTTPS port	7473

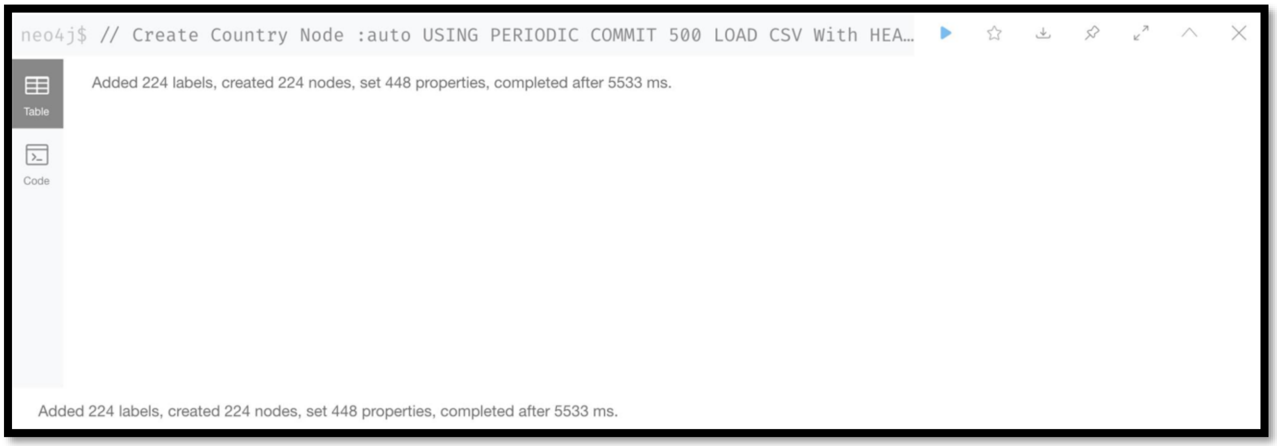
3.4. Data Mapping and Integration

- Open Neo4j Browser. It will connect with the database automatically.
- Create Constrains:
    1. country node: CountryId is unique
    2. province node: ProvinceId is unique
    3. date node: ObservationDate is unique
    4. infectionStatus node: EntryId is unique



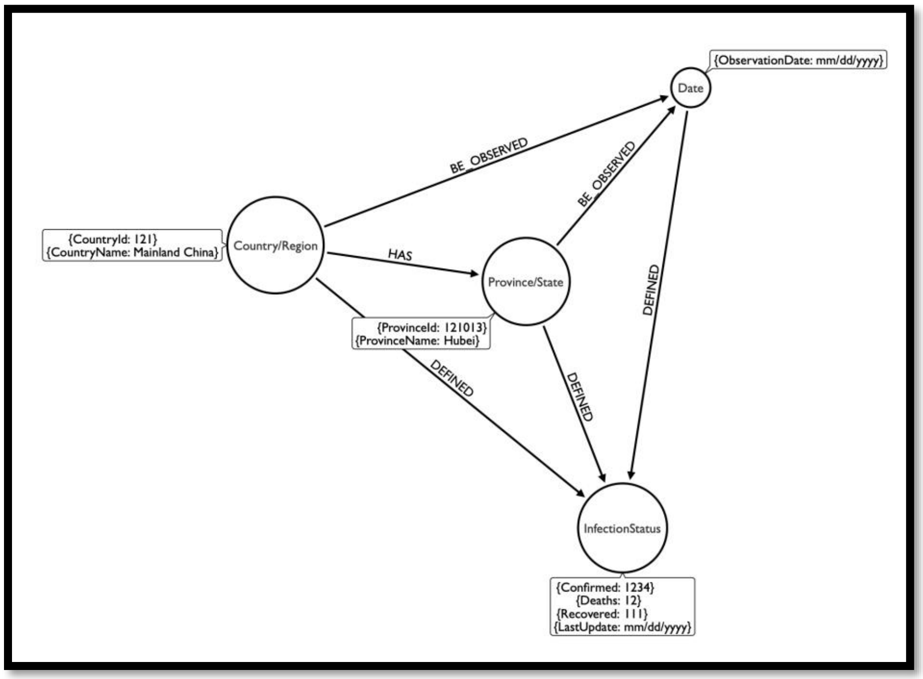
```
$ CREATE CONSTRAINT ON (country:Country) ASSERT country.CountryId IS UNIQUE; CREATE …

CREATE CONSTRAINT ON (country:Country) ASSERT country.CountryId IS UNIQUE$ CREATE CONSTR…
CREATE CONSTRAINT ON (province:Province) ASSERT province.ProvinceId IS UNIQUE$ CREATE CO…
CREATE CONSTRAINT ON (date:Date) ASSERT date.ObservationDate IS UNIQUE$ CREATE CONSTRAIN…
CREATE CONSTRAINT ON (infectionStatus:InfectionStatus) ASSERT infectionStatus.EnteryId I…
```

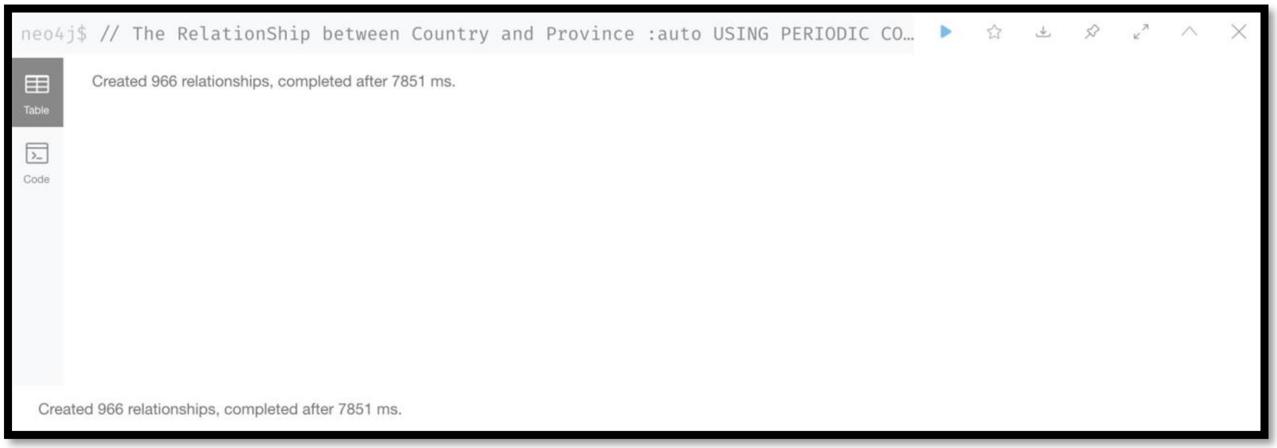
- Create Nodes and map data columns to the Node properties one by one



```
neo4j$ // Create Country Node :auto USING PERIODIC COMMIT 500 LOAD CSV With HEA…

Added 224 labels, created 224 nodes, set 448 properties, completed after 5533 ms.

Added 224 labels, created 224 nodes, set 448 properties, completed after 5533 ms.
```

- Establish the relationships according to the following draft:



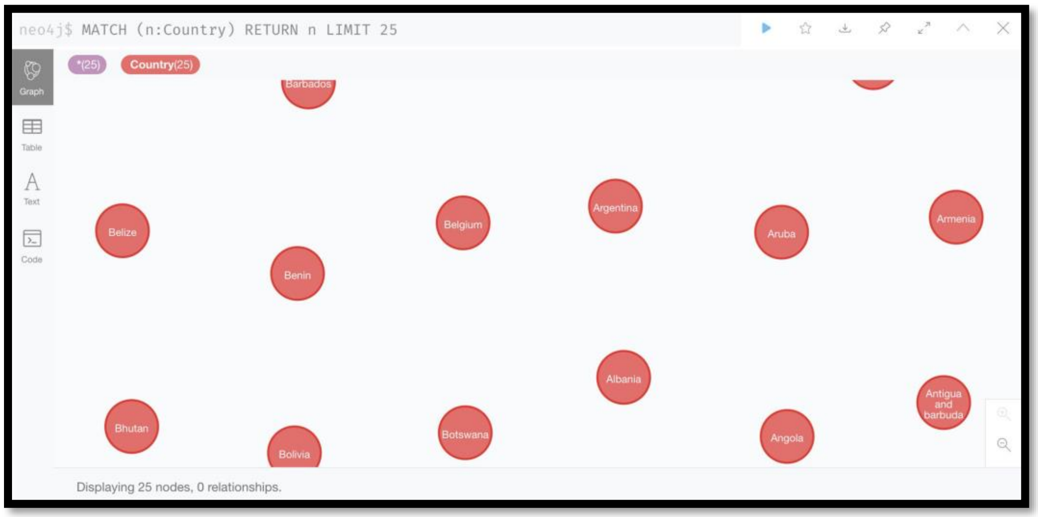


```
neo4j$ // The RelationShip between Country and Province :auto USING PERIODIC CO…
```

```
Created 966 relationships, completed after 7851 ms.
```

```
Created 966 relationships, completed after 7851 ms.
```

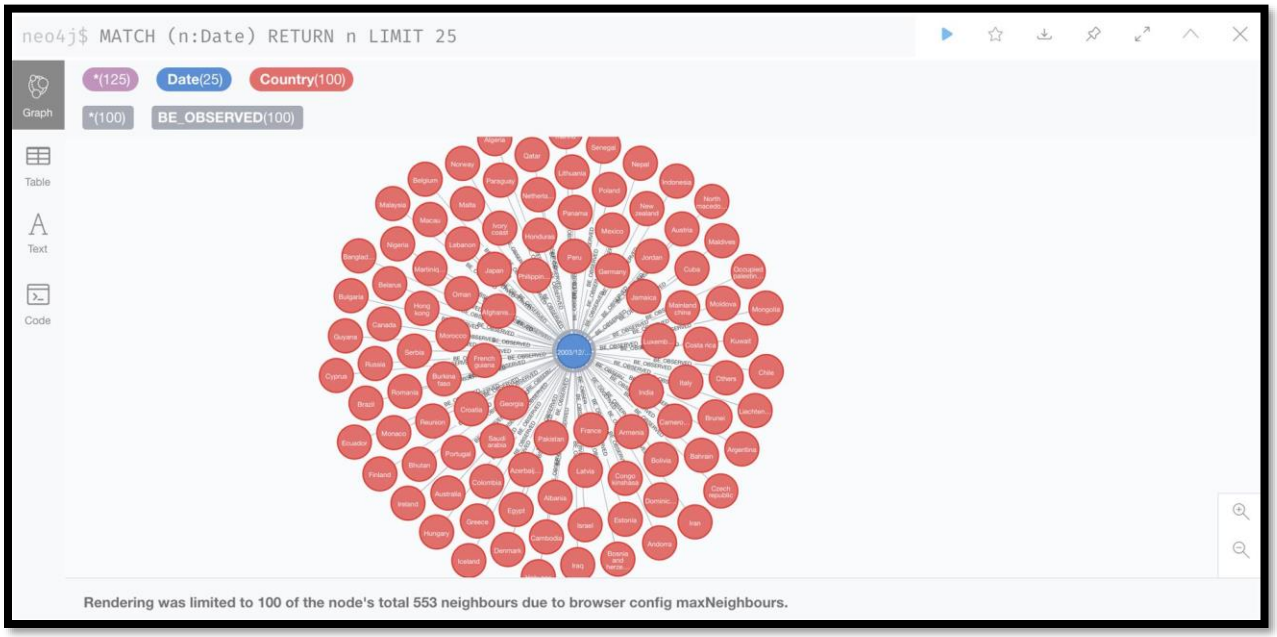
- Get the Nodes Information:



```
neo4j$ MATCH (n:Country) RETURN n LIMIT 25
```

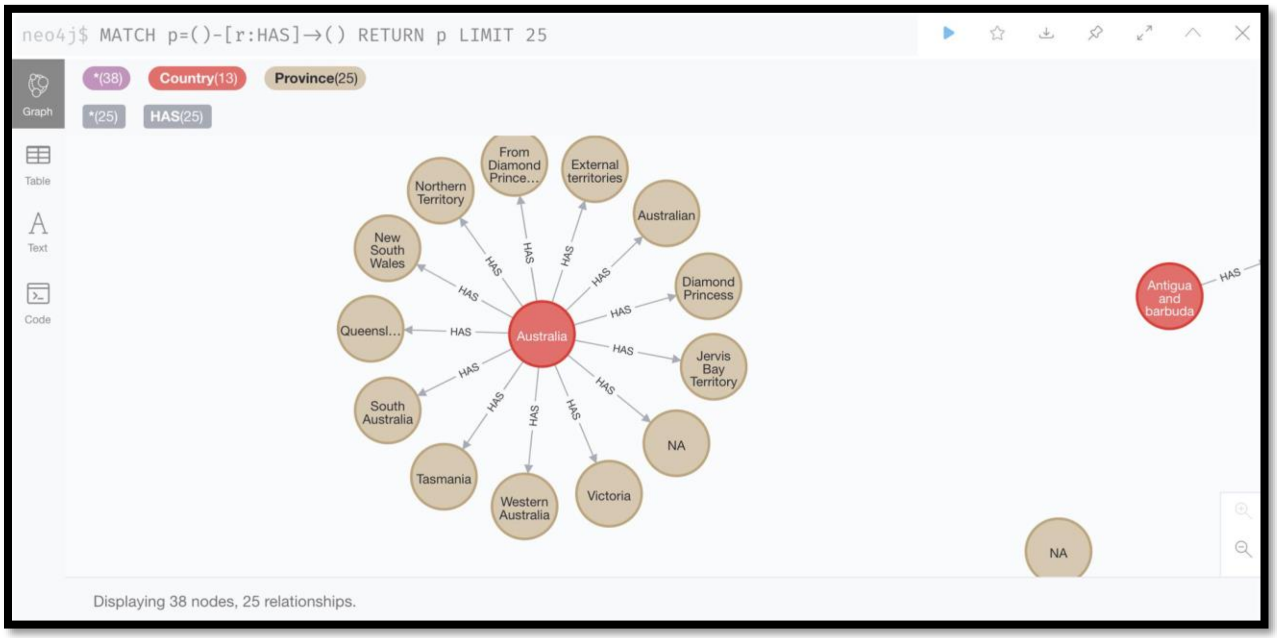
Displaying 25 nodes, 0 relationships.

- Get the Relationships Information:



```
neo4j$ MATCH (n:Date) RETURN n LIMIT 25
```

Rendering was limited to 100 of the node's total 553 neighbours due to browser config maxNeighbours.

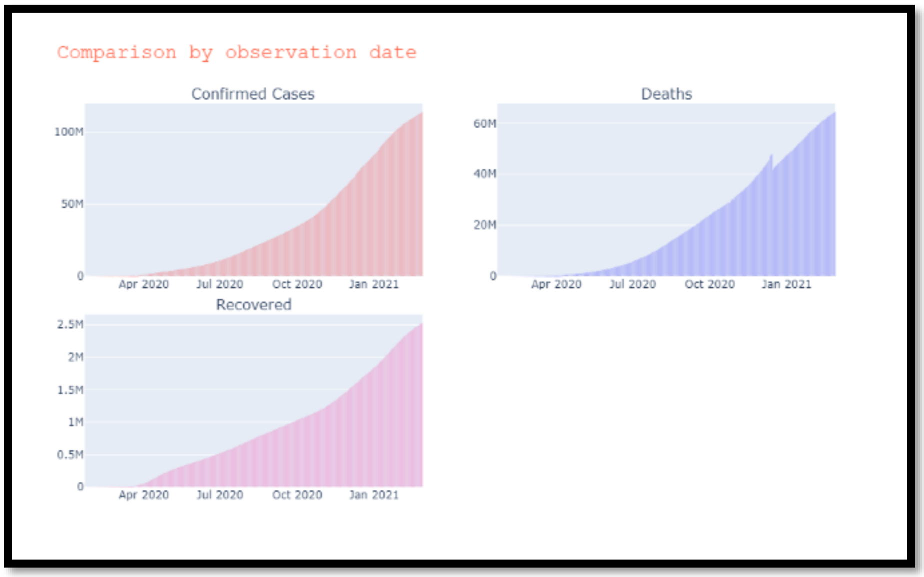


```
neo4j$ MATCH p=()-[r:HAS]→() RETURN p LIMIT 25
```

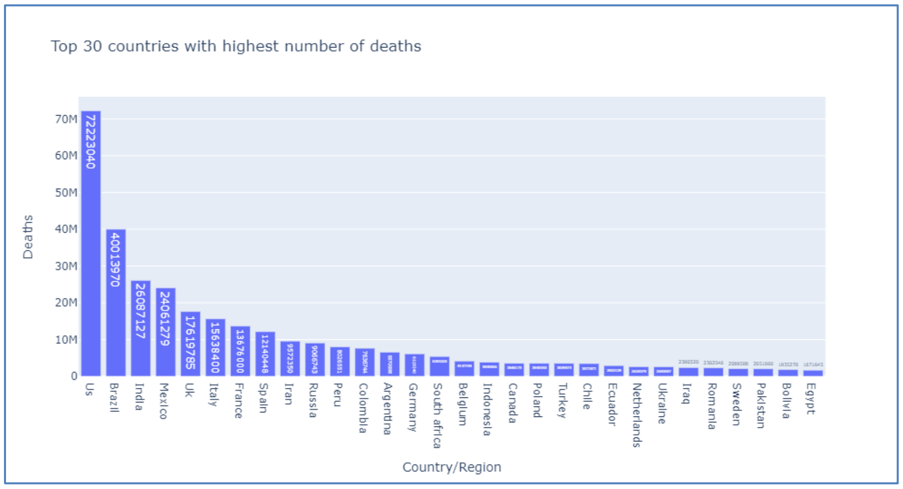
Displaying 38 nodes, 25 relationships.

3.5. Data Validation and Data Visualization

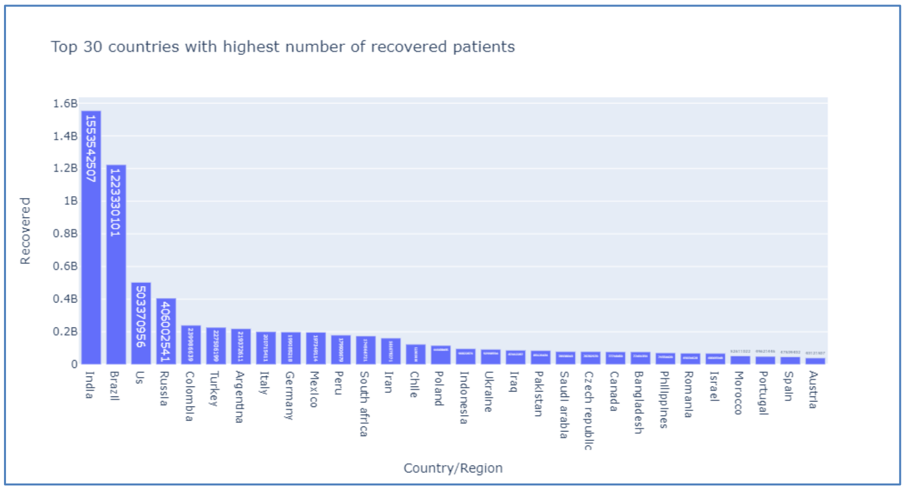
- Data Visualization is the graphic representation of data that helps understand the data without requiring technical knowledge
- Data validation & Data visualization were performed using Python
- These visualizations could answer questions like
- What was the difference between the confirmed, recovered and death numbers for different month & year?



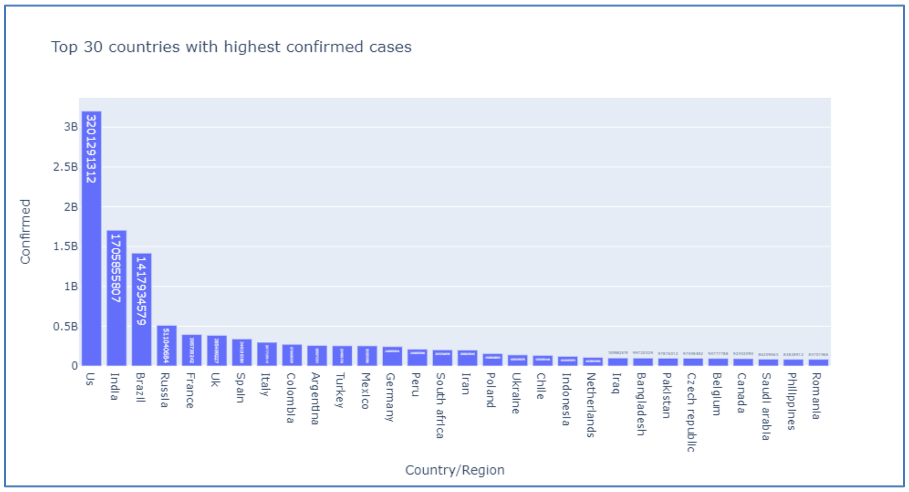
- What was the country that had highest number of deaths?



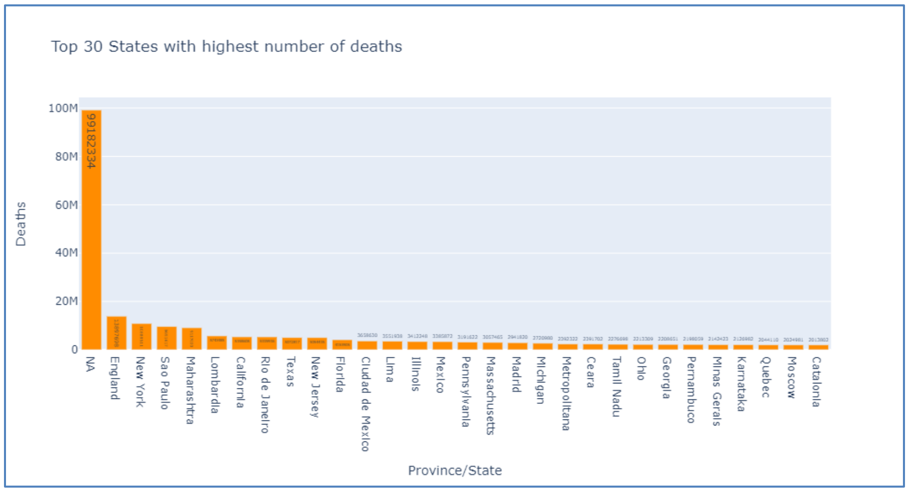
- What was the country that had highest number of recovered patients?



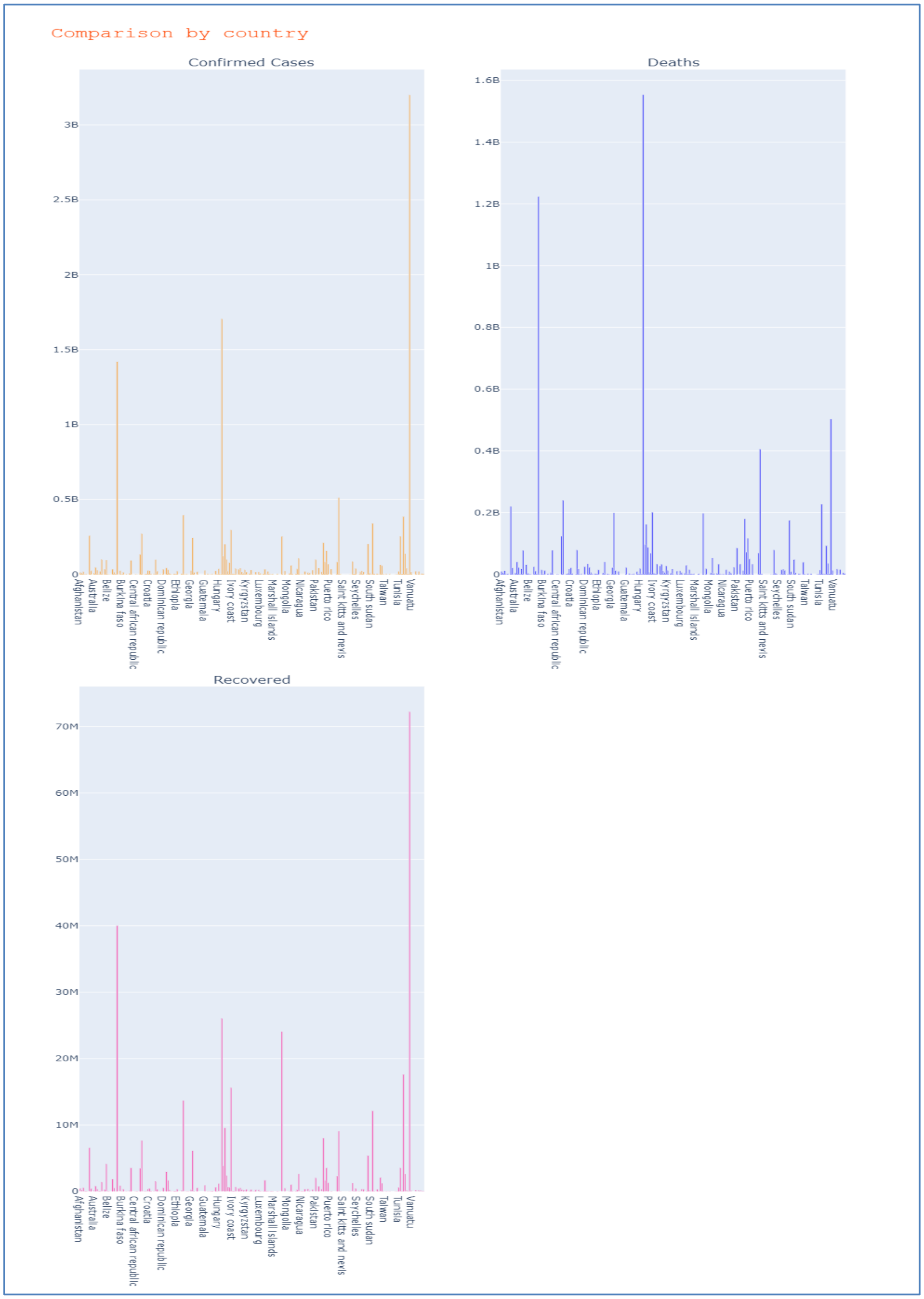
- What was the country that had highest number of confirmed cases?



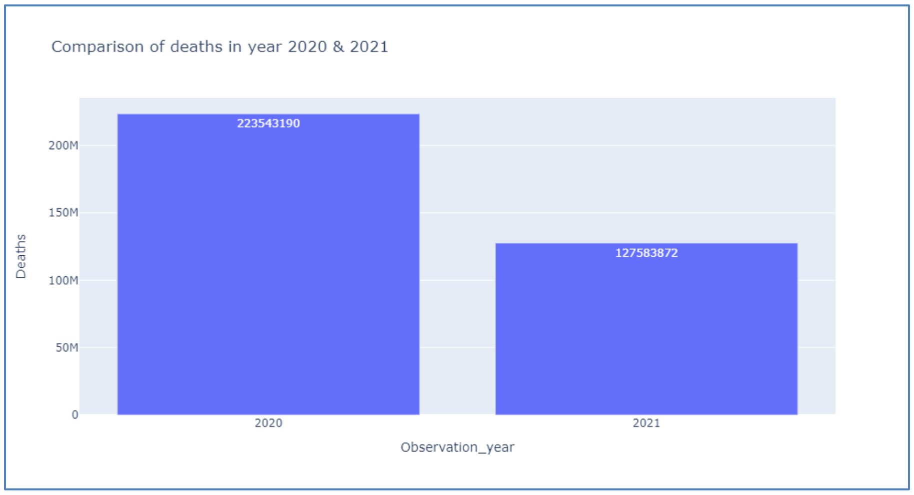
- Which State had highest number of deaths?



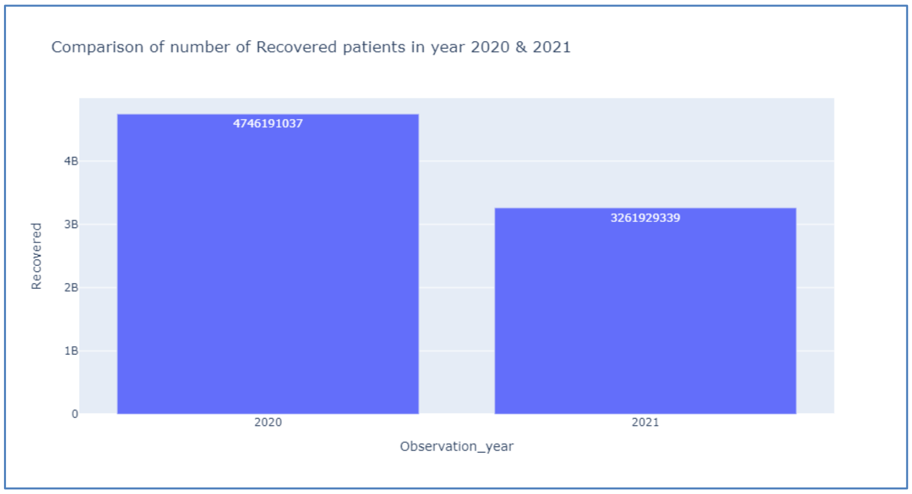
- What was the difference between the confirmed, recovered and death numbers for different countries?



- What was the difference in number of deaths in year 2020 and year 2021?

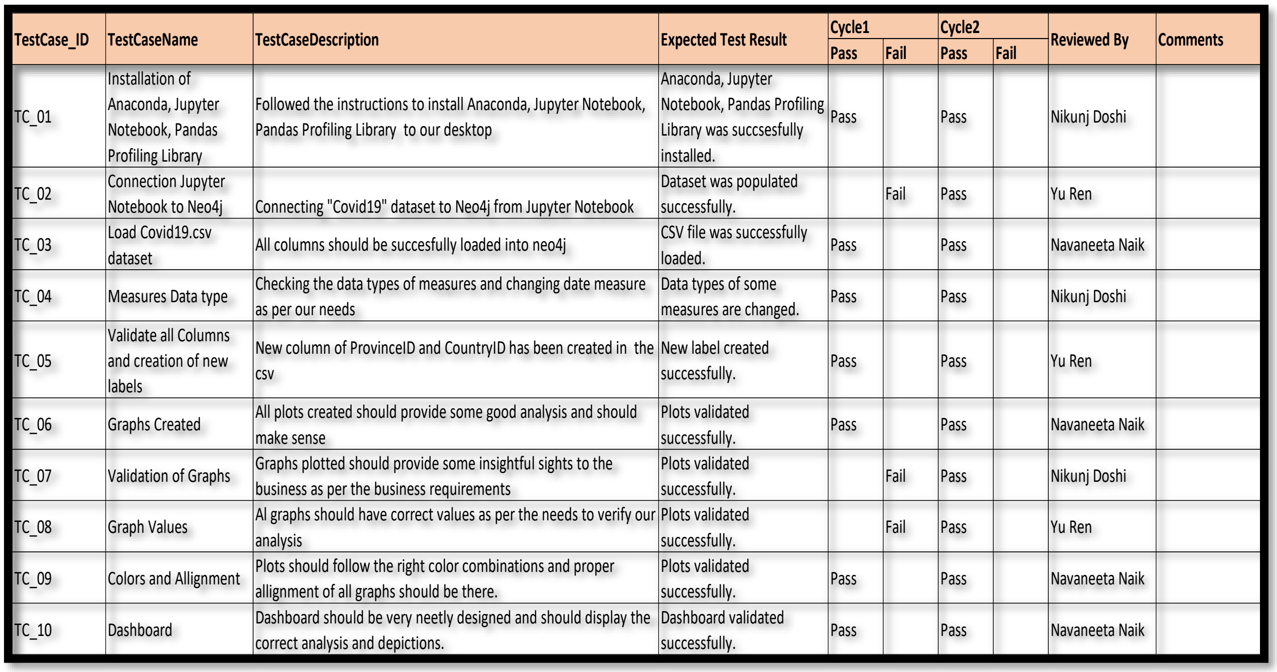


- What was the difference in number of patients that recovered from covid in the year 2020 and year 2021?



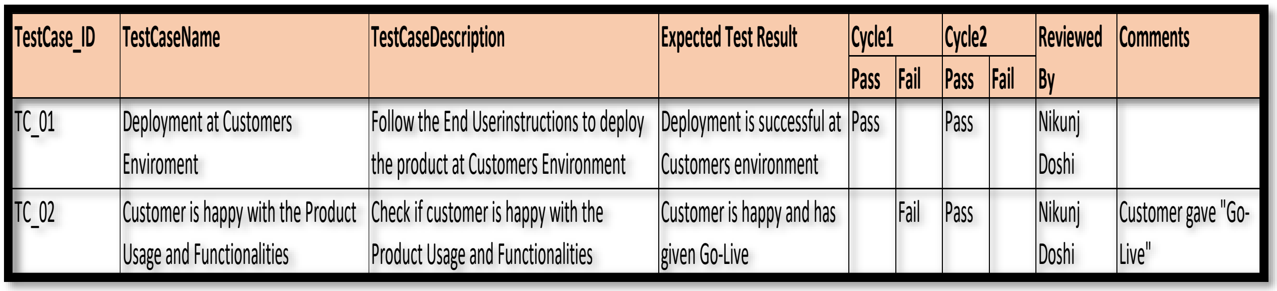
3.6. System Integration and User Acceptance Testing

System Integration Test Cases



TestCase_ID	TestCaseName	TestCaseDescription	Expected Test Result	Cycle1		Cycle2		Reviewed By	Comments
				Pass	Fail	Pass	Fail		
TC_01	Installation of Anaconda, Jupyter Notebook, Pandas Profiling Library	Followed the instructions to install Anaconda, Jupyter Notebook, Pandas Profiling Library to our desktop	Anaconda, Jupyter Notebook, Pandas Profiling Library was succsesfully installed.	Pass		Pass		Nikunj Doshi	
TC_02	Connection Jupyter Notebook to Neo4j	Connecting "Covid19" dataset to Neo4j from Jupyter Notebook	Dataset was populated successfully.		Fail	Pass		Yu Ren	
TC_03	Load Covid19.csv dataset	All columns should be succesfully loaded into neo4j	CSV file was successfully loaded.	Pass		Pass		Navaneeta Naik	
TC_04	Measures Data type	Checking the data types of measures and changing date measure as per our needs	Data types of some measures are changed.	Pass		Pass		Nikunj Doshi	
TC_05	Validate all Columns and creation of new labels	New column of ProvinceID and CountryID has been created in the csv	New label created successfully.	Pass		Pass		Yu Ren	
TC_06	Graphs Created	All plots created should provide some good analysis and should make sense	Plots validated successfully.	Pass		Pass		Navaneeta Naik	
TC_07	Validation of Graphs	Graphs plotted should provide some insightful sights to the business as per the business requirements	Plots validated successfully.		Fail	Pass		Nikunj Doshi	
TC_08	Graph Values	Al graphs should have correct values as per the needs to verify our analysis	Plots validated successfully.		Fail	Pass		Yu Ren	
TC_09	Colors and Allignment	Plots should follow the right color combinations and proper allignment of all graphs should be there.	Plots validated successfully.	Pass		Pass		Navaneeta Naik	
TC_10	Dashboard	Dashboard should be very neetly designed and should display the correct analysis and depictions.	Dashboard validated successfully.	Pass		Pass		Navaneeta Naik	

UAT Test Cases



TestCase_ID	TestCaseName	TestCaseDescription	Expected Test Result	Cycle1		Cycle2		Reviewed By	Comments
				Pass	Fail	Pass	Fail		
TC_01	Deployment at Customers Enviroment	Follow the End Userinstructions to deploy the product at Customers Environment	Deployment is successful at Customers environment	Pass		Pass		Nikunj Doshi	
TC_02	Customer is happy with the Product Usage and Functionalities	Check if customer is happy with the Product Usage and Functionalities	Customer is happy and has given Go-Live		Fail	Pass		Nikunj Doshi	Customer gave "Go-Live"

3.7. Challenges Encountered

1. Python:
    - Null Values: The dataset had around 62000 missing values to be handled. Since dropping the columns would reduce the dataset and have an impact on the visualizations, we decided to fill the null values with “not available” so that even this data could be used for our further analysis
    - Generating ProvinceID: A new column ProvinceID was created to better help in viewing data by grouping them as different provinces. It was a challenging execution, but with references from python documents we were able to do it
2. NEO4J:
    - Forget Password and Reset Error: When we want to connect with the database, we forgot the password and we try to reset the password but it fails. Finally we create a new one so that we can use.
    - Exporting Metadata Error: There are some NaN values when we want to export our metadata which make things wrong. We filter the error value in the loop of processing the data.

3.8. End User Instructions

Python:

1. Install python libraries: pandas, pandas-profiling, matplotlib, seaborn.
2. Download the dataset: Novel Corona Virus 2019 Dataset | Kaggle
3. Run the csye7250_project. ipynb in Jupyter Lab or Jupyter Notebook
4. Visualizations can be used to compare and understand how different countries have been affected by COVID-19