|  |  |
| --- | --- |
| Covid-19 (November 2023)  Data visualisation | Abstract  Summarized analysis with pictures, charts, and a dashboard about Coronavirus data from WHO. Only with 6 illustrations, it is clearly and fast understandable the effects and progress by the novel virus in the planet Earth.  YIT 19488399 (Tony)  Data Visualization - CMP020X302A Module Tutor: Mohammad Ahmad BSc Computer science, year 3rd (autumn term). |

**Task 1: questions that a visualization will answer.**

* From “vaccination data” dataset:

1. Number of different vaccines used by region and ALL countries that used 11 and 12 different types of vaccines.
2. Double bars plot about the speed and amount of vaccines administration to people in different countries (11 countries in each bar plot).

* From “latest reported counts of cases and deaths” dataset:

1. Total number of deaths by country with top100 countries. Looks like BUBLES but code is SCATTER.
2. Cases cumulative death by continent (region). PIE.
3. Confirmed cases per country and region by days (24 hours timeframe). MAP/INTERACTIVE MAP.

**Task 2: TABLEAU or POWER BI and follow instructions (one dashboard with five dynamic visualizations to analyze data).**

A dashboard with 8 worksheets, 1 picture and short text (2 references) to explain coronavirus data from WHO. Additional dataset used from official WHO website, plus dataset provided in Moodle. Datasets are recent, but not live updated.

All dashboard is interactive by region and country selection in the “Highlight” area. Analysis and data visualization implemented in Tableau Public software and online.

First question: Number of different vaccines used by country or region

Chart, histogram

Description automatically generated

Output from running python code from VSC and screenshot. Both images are same data visualization, but different formats.

Chart

Description automatically generated

Python code follows with screenshot included. In this occasion, code output gives some information on terminal for familiarize with data (.csv):

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

data = pd.read\_csv("vaccination-data.csv") #reading the database

print(data.head(), data.columns)

integer\_vaccines\_type = data["NUMBER\_VACCINES\_TYPES\_USED"]

integer\_vaccines\_type.fillna(0)

print(integer\_vaccines\_type, len(integer\_vaccines\_type), integer\_vaccines\_type.unique())

region = data["WHO\_REGION"]

print(region, len(region), region.unique())

# print(max(data["NUMBER\_VACCINES\_TYPES\_USED"]))

print(data['NUMBER\_VACCINES\_TYPES\_USED'].nlargest(n=12)) #untill 11 inclusive

print(data.loc[data['NUMBER\_VACCINES\_TYPES\_USED']>10, 'COUNTRY'])

top\_variety = data.loc[data['NUMBER\_VACCINES\_TYPES\_USED']>10, 'COUNTRY']

print(top\_variety, len(top\_variety))

top\_varietylist = np.array(top\_variety)

fig, axe = plt.subplots(figsize=(14, 7))

plt.bar(data["WHO\_REGION"], data["NUMBER\_VACCINES\_TYPES\_USED"], color = "blue")

plt.xlabel('Regional Offices (EMRO = Eastern Mediterranean, WPRO = Western Pacific, AMRO = Americas, SEARO = South-East Asia, Other = Liechtenstein)')

plt.ylabel('Number of different vaccines used')

plt.title("How many types of vaccines by region", color = "#091b82", fontsize = 22)

plt.axhline(y = 6, color = 'r', linestyle = '-')

axe.text(4.8, 8.8, f" Countries whith highest variety of vaccines (12 & 11)  \n{top\_variety}", bbox=dict(facecolor='red', alpha=0.5),fontsize = 9)

plt.tight\_layout(pad=1.1, w\_pad=0.5, h\_pad=0.2)

plt.show()

fig.savefig("output1.png")

plt.close(fig)

Graphical user interface, text

Description automatically generated

Second question: Which countries started vaccinations first (speed) and which countries has the highest amount of vaccinated people?

Excel, bar chart

Description automatically generated with medium confidence

Timeline, Excel

Description automatically generated

# https://towardsdatascience.com/covid-19-vaccination-progress-analysis-around-the-world-736d7e57f198

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

sns.set\_style('whitegrid')

df = pd.read\_csv("vaccination-data.csv")

plt.figure(1, figsize=(15, 7),facecolor="#5b92e5")

plt.subplot(211)

cols = ['COUNTRY', 'TOTAL\_VACCINATIONS', 'ISO3', 'VACCINES\_USED','TOTAL\_VACCINATIONS\_PER100']

vacc\_amount = df[cols].groupby('COUNTRY').max().sort\_values('TOTAL\_VACCINATIONS', ascending=False).dropna(subset=['TOTAL\_VACCINATIONS'])

vacc\_amount = vacc\_amount.iloc[:11]

vacc\_amount = vacc\_amount.sort\_values('TOTAL\_VACCINATIONS\_PER100', ascending=False)

plt.bar(vacc\_amount.index, vacc\_amount.TOTAL\_VACCINATIONS\_PER100, color = 'r')

plt.xticks(rotation = 10)

plt.title("Which country is vaccinating its citizens the fastest?", color = "r", fontweight="bold")

plt.ylabel('Amount of vaccinated people per hundred')

plt.subplot(212)

cols = ['COUNTRY', 'TOTAL\_VACCINATIONS', 'ISO3', 'VACCINES\_USED']

vacc\_amount = df[cols].groupby('COUNTRY').max().sort\_values('TOTAL\_VACCINATIONS', ascending=False).dropna(subset=['TOTAL\_VACCINATIONS'])

vacc\_amount = vacc\_amount.iloc[:11]

plt.bar(vacc\_amount.index, vacc\_amount.TOTAL\_VACCINATIONS, color = 'g')

plt.title('Which country has the highest number of vaccinated people?', color ="#025839", fontweight="bold")

plt.ylabel('Amount of vaccinated citizens')

plt.xlabel('NOTE: Countries are not the same for both figures. Total number of countries in each figure = 11')

plt.xticks(rotation = 10)

plt.tight\_layout()

plt.show()

plt.savefig("output2.png")

Text

Description automatically generated

Third question: Cumulative deaths grouped by countries and showing only top 100 countries (total countries are 237):

Chart, bubble chart

Description automatically generated

First image is the output that generates python code from VSC editor and saved using plotly buttons (top right). Second and third are screenshots of same picture.

A picture containing graphical user interface

Description automatically generated

Chart, bubble chart

Description automatically generated

#https://www.geeksforgeeks.org/covid-19-analysis-and-visualization-using-plotly-express/

import plotly.express as px

import pandas as pd

import plotly.io as pio

df = pd.read\_csv("WHO-COVID-19-global-table-data.csv")

figure3 = px.scatter(df.head(100), x='Name', y= 'Deaths - cumulative total',

        hover\_data=['WHO Region'],

        color='Name', size= 'Deaths - cumulative total', size\_max=80, hover\_name="Name",

        title=("100 Countries gruped by continent showing number of cumulative deaths"))

figure3.update\_xaxes(title\_font\_family="Helvetica")

figure3.show()

# export as static image

pio.write\_image(figure3, "output3.png")

Graphical user interface, text, application

Description automatically generated

Fourth question: Total cumulative deaths grouped by continent using ALL countries. Pie chart.

Chart, pie chart

Description automatically generated

First picture is the output saved using top right button implemented in plotly. This picture is from VSC editor, using python programming language.

Chart, pie chart

Description automatically generated

Both images are screenshots from previous initial image. Browser is used (Firefox). The third and last image is a full screen on the image on the browser (2nd).

Chart, pie chart

Description automatically generated

# https://www.kaggle.com/code/siddheshmahajan/covid-19-data-visualization

import pandas as pd

import plotly.io as pio

import plotly.express as px

df = pd.read\_csv("WHO-COVID-19-global-table-data.csv")

df\_countries = df.groupby(['WHO Region']).max().reset\_index()

df\_countries = df\_countries.drop\_duplicates(subset = ['WHO Region'])

#the classic pie chart

fig = px.pie(df\_countries, values = 'Deaths - cumulative total', names='Name', height=800, title='Comulative death by continent')

fig.update\_traces(textposition='inside', textinfo='percent+label')

fig.update\_layout(

    title\_x = 0.5,

    autosize = True,

    geo=dict(

        showframe = False,

        showcoastlines = False,

    ))

fig.update\_layout(legend=dict(

    title="WHO Regions"

)) # write a title for the legend on the right

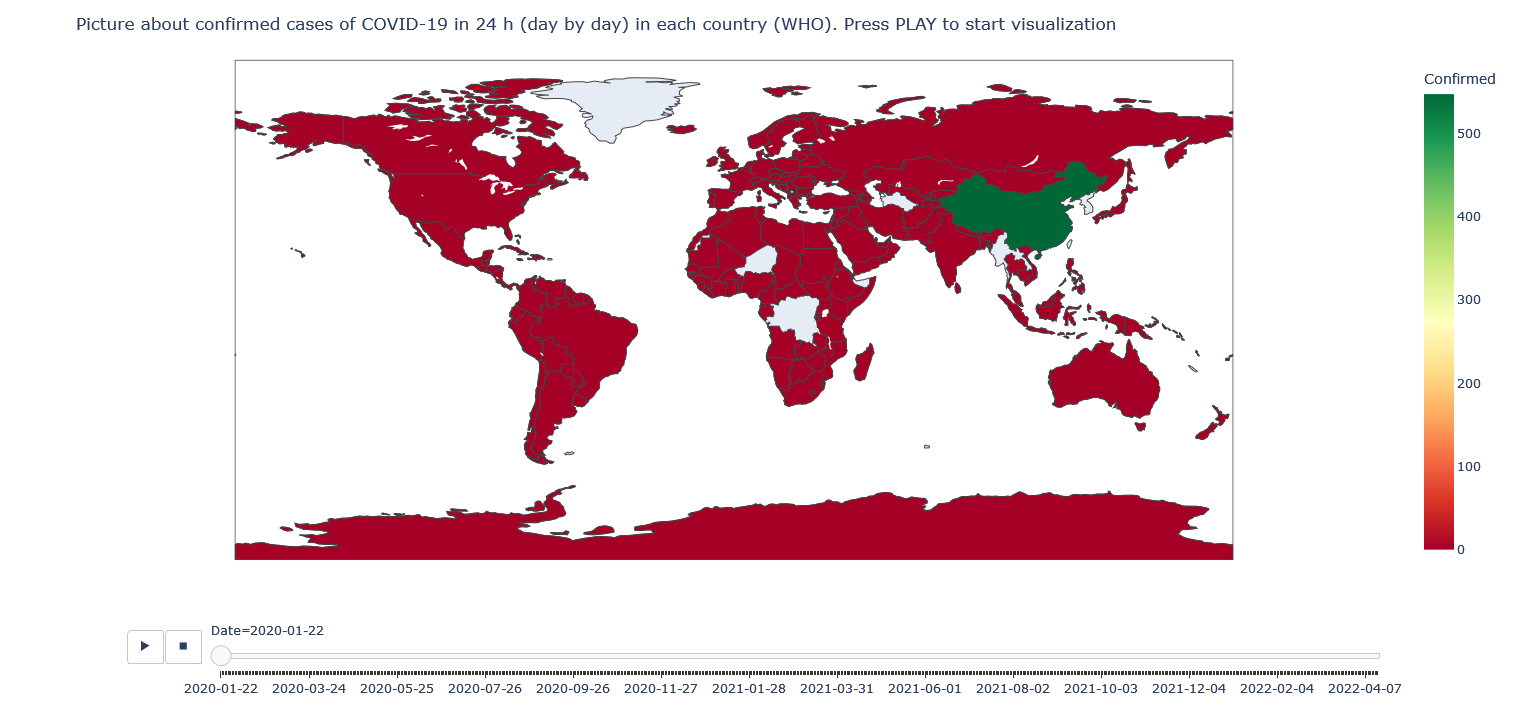
fig.show()

pio.write\_image(fig, "output4.png")

Graphical user interface, text

Description automatically generated

Fifth question: Animated visualization of the spread of the coronavirus. Updates by day (24 hours).



Graphical user interface

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Map

Description automatically generated

# https://www.kaggle.com/code/lokeshrth4617/covid-19-world-wise-visualization

# https://opensource.com/article/20/4/python-map-covid-19

import pycountry

import plotly.express as px

import pandas as pd

# ----------- Step 1 ------------

URL\_DATASET = r'https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv'

df1 = pd.read\_csv(URL\_DATASET)

# print(df1.head) # Uncomment to see what the dataframe is like

# ----------- Step 2 ------------

list\_countries = df1['Country'].unique().tolist()

# print(list\_countries) # Uncomment to see list of countries

d\_country\_code = {}  # To hold the country names and their ISO

for country in list\_countries:

    try:

        country\_data = pycountry.countries.search\_fuzzy(country)

        # country\_data is a list of objects of class pycountry.db.Country

        # The first item  ie at index 0 of list is best fit object of class Country have an alpha\_3 attribute

        country\_code = country\_data[0].alpha\_3

        d\_country\_code.update({country: country\_code})

    except:

        print('could not add ISO 3 code for ->', country)

        d\_country\_code.update({country: ' '}) # If could not find country, make ISO code ' '

# print(d\_country\_code) # Uncomment to check dictionary

# create a new column iso\_alpha in the df and fill it with appropriate iso 3 code

for k, v in d\_country\_code.items():

    df1.loc[(df1.Country == k), 'iso\_alpha'] = v

# print(df1.head)  # Uncomment to confirm that ISO codes added

# ----------- Step 3 ------------

fig = px.choropleth(data\_frame = df1,

                    locations= "iso\_alpha",

                    color= "Confirmed",  # value in column 'Confirmed' determines color

                    hover\_name= "Country",

                    color\_continuous\_scale= 'RdYlGn',  #  color scale red, yellow green

                    animation\_frame= "Date")

fig.update\_layout(title="Picture about confirmed cases of COVID-19 in 24 h (day by day) in each country (WHO). Press PLAY to start visualization")

fig.show()

Text

Description automatically generated

Text

Description automatically generated

**Code on my GitHub:** [**https://github.com/ittonycash/data\_visualisation/tree/main/coursework**](https://github.com/ittonycash/data_visualisation/tree/main/coursework)

Graphical user interface, application

Description automatically generated

**Task 2 screenshots. Individual sheets and final dashboard:**

**Tableau Public own workspace:** [**https://public.tableau.com/app/profile/tony6278**](https://public.tableau.com/app/profile/tony6278)

**Tableau Public dashboard direct link for task 2:** [**https://public.tableau.com/app/profile/tony6278/viz/Courseworkpublic/dashboard**](https://public.tableau.com/app/profile/tony6278/viz/Courseworkpublic/dashboard)

**#END#**