Data Visualization 1

INSTALL geopandas

UPLOAD UID.XLSX FILE

UPLOAD newcause_of_deaths-1.CSV FILE

Double-click (or enter) to edit

```
!pip install geopandas
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: geopandas in /usr/local/lib/python3.8/dist-packages (0.12.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-packages (from geopandas) (21.3)
Requirement already satisfied: fiona>=1.8 in /usr/local/lib/python3.8/dist-packages (from geopandas) (1.8.22)
Requirement already satisfied: pyproj>=2.6.1.post1 in /usr/local/lib/python3.8/dist-packages (from geopandas) (3.4.1)
Requirement already satisfied: shapely>=1.7 in /usr/local/lib/python3.8/dist-packages (from geopandas) (1.8.5.post1)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.8/dist-packages (from geopandas) (1.3.5)
Requirement already satisfied: munch in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (2.5.0)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (0.7.2)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (22.1.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (2022.9.24
Requirement already satisfied: setuptools in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (57.4.0
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (1.15.0)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.8/dist-packages (from fiona>=1.8->geopandas) (7.1.2)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.0.0->geopandas) (1
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.0.0->geop
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-packages (from pandas>=1.0.0->geopandas) (20
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packaging->geopar
```

```
import altair as alt
from vega_datasets import data
from altair import datum
import geopandas as gpd
import pandas as pd

# Load UID and Main data set
UID = pd.read_excel('UID.xlsx')
data = pd.read_csv('cause_of_deaths-with-total.csv')
df = pd.merge(UID, data, on='Code')
alt.data_transformers.disable_max_rows()
```

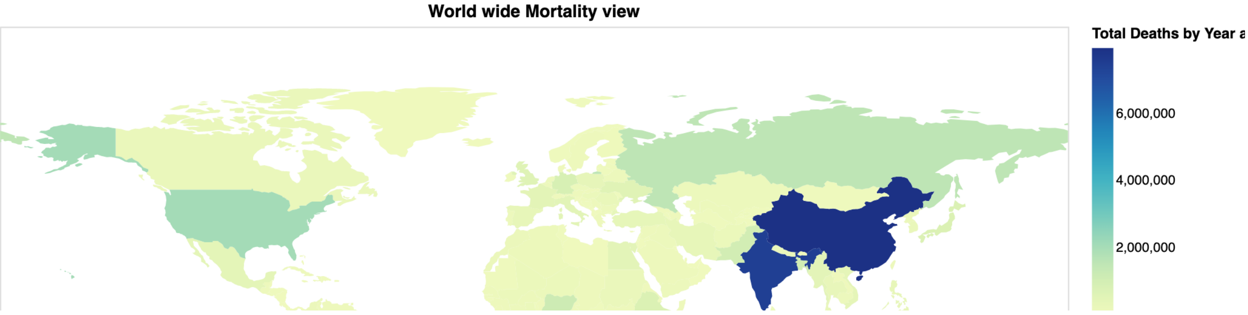
```
DataTransformerRegistry.enable('default')
```

```
#Clean data set
cols = df.columns.tolist()
all_disease = cols[7:-1]
all_disease[1]= 'Alzheimers Disease and Other Dementias'
all_disease[2]='Parkinsons Disease'
```

Data Visualisation 1: Interactive Coordinated view of the world map with a stacked Bar chart

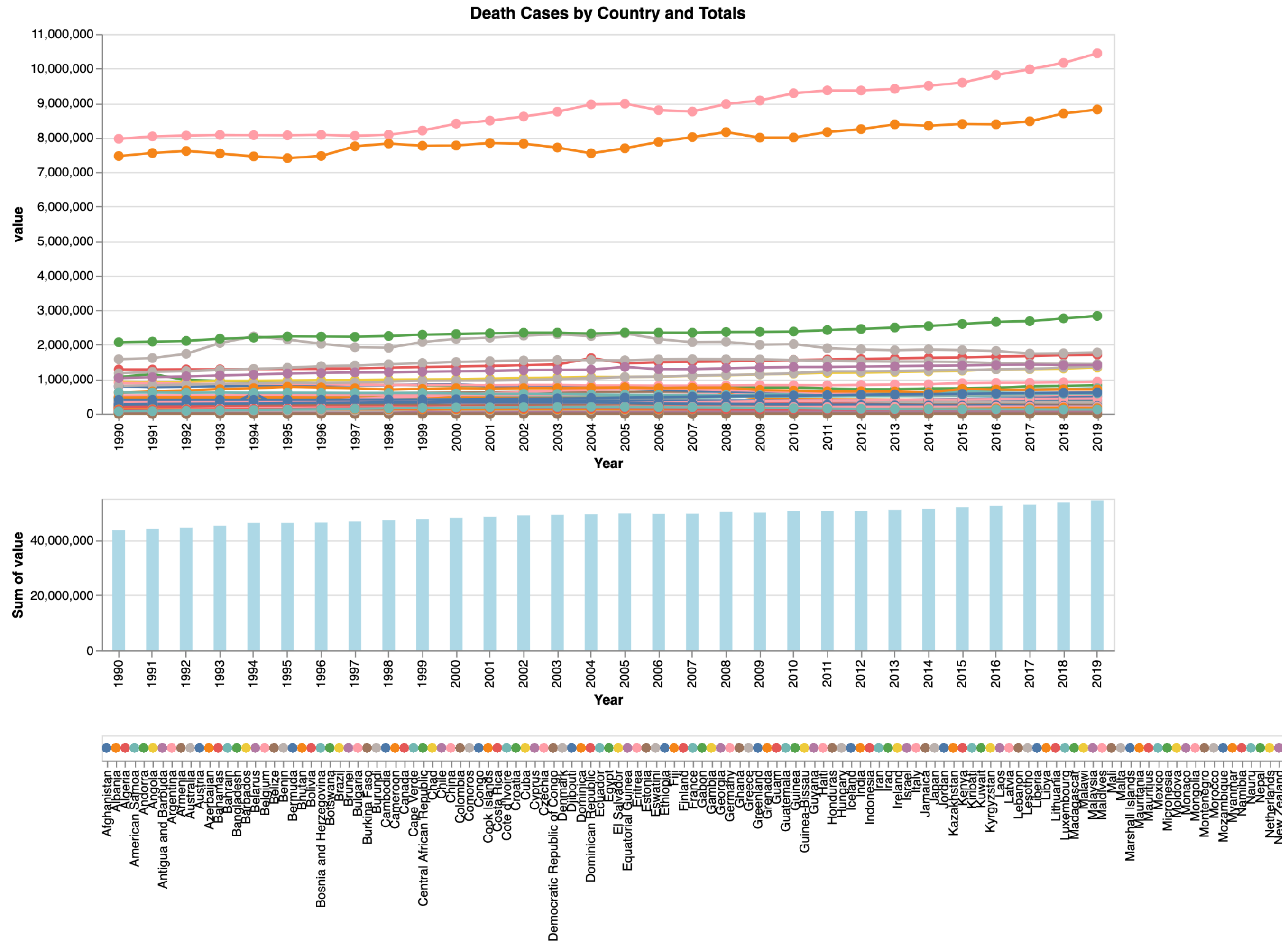
```
import altair as alt
from vega_datasets import data
from altair import datum
import geopandas as gpd
click = alt.selection_multi(encodings=['color'])

# Get data for countries geo location
countries = alt.topo_feature(data.world_110m.url, "countries")

slider = alt.binding_range(
    step=1,
    min=1990,
    max=2019
)

select_year = alt.selection_single(
    name="slider",
    fields=['Year'],
    bind=slider, init={'Year': 1990}
)

# World Map visualisation
```

```
world_map = alt.Chart(df).mark_geoshape()\
    .encode(color='Total Deaths by Year and country:Q',tooltip = ('Country','Total Deaths by Year and country:Q'))\
    .add_selection(select_year)\
    .transform_filter(select_year)\
    .transform_lookup(
      lookup='id',
      from_=alt.LookupData(countries, key='id', fields=["type", "properties", "geometry"])
      )\
    .project('equirectangular')\
    .transform_filter(click)\
    .add_selection(click)\
    .properties(
        width=800,
        height=400,
        title='World wide Mortality view'
    )
# Bar chart visualisation
bar = alt.Chart(df).transform_fold(
    all_disease,as_=['Disease','value']
).mark_bar(point=alt.MarkConfig(shape='circle', size=60)).encode(
    x='Disease:N',
    #y=alt.Y('Total Deaths by Year and country:Q', title=None),
    y=alt.Y('value:Q'),
    #color=alt.Color('Country/Territory:N', legend=None),
    color=alt.Color('Country/Territory:N',
        # optional: make color order in legend match stack order
        sort=alt.EncodingSortField('value:Q', order='descending')
    ),
    tooltip=['Year','Country/Territory','value:Q','Disease:N']
).transform_filter(
    click
).add_selection(select_year
).transform_filter(select_year
).properties(
    width=800,
    title='Death Cases by Country and Totals',
    selection=click
).add_selection(
    click
)

world_map & bar
```



Data Visualization 2 Interactive Coordinated view of the time series Line chart with a Bar chart

```
df = pd.read_csv('cause_of_deaths-with-total.csv')
cols = df.columns.tolist()
#print(len(cols))
all_disease = cols[7:]
all_disease
df.head()
countries = df['Country/Territory'].unique()
countries
alt.data_transformers.disable_max_rows()


click = alt.selection_multi(encodings=['color'])
brush = alt.selection_interval(encodings=['x'])

input_dropdown = alt.binding_select(options=all_disease, name='Disease')
sel = alt.selection_single(fields=['Disease'], bind=input_dropdown,init={'Disease' :'Total Deaths by Year and country'})

#Line chart visualisation

line = alt.Chart(df).transform_fold(
    all_disease,as_=['Disease','value']
).mark_line(point=alt.MarkConfig(shape='circle', size=60)).encode(
    x='Year:N',
    #y=alt.Y('Total Deaths by Year and country:Q', title=None),
    y=alt.Y('value:Q'),
    color=alt.Color('Country/Territory:N', legend=None),
    tooltip=['Year','Country/Territory','value:Q']
).transform_filter(
    sel
).transform_filter(
    brush
).transform_filter(
    click
).properties(
    width=800,
    title='Death Cases by Country and Totals',
    selection=click
).add_selection(
    sel
)

## Historical Bar char visualisation

hist = alt.Chart(df).transform_fold(
    all_disease,as_=['Disease','value']
).mark_bar(size=10).encode(
    x=alt.X('Year:N'),
    #y=alt.Y('sum(Total Deaths by Year and country):Q', title=None),
    y=alt.Y('sum(value):Q'),
    color=alt.value('lightblue'),
    tooltip=['Year','sum(value):Q']
).transform_filter(
    sel
).transform_filter(
    click
).properties(
    width=800,
    height=120,
).add_selection(
    brush
).add_selection(
    sel
)

legend1 = alt.Chart(df).mark_circle(size=50).encode(
    x=alt.X('Country/Territory', title=None),
    color=alt.condition(click, alt.Color('Country/Territory:N', legend=None), alt.value('lightgray'))
).properties(
    selection=click,
    width=1500
)

(line) & (hist)  & legend1
```



Disease Total Deaths by Year and country