spatial_data

April 5, 2023

[ ]:
```
# import libraries
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
```

Worldwide Choropleth Map

[ ]:
```
# read internet data usage
df = pd.read_csv ('internet_usage.csv')
```

[ ]:
```
# sanity checks
df.shape
```

[ ]:
```
df.head(5)
```

[ ]:
```
df.tail(5)
```

[ ]:
```
df.rename (columns = {'Individuals using the Internet (% of population)' :␣
↪'Internet_Usage'}, inplace = True)
```

[ ]:
```
df.describe()
```

[ ]:
```
df.info()
```

[ ]:
```
# get data only for the year 2016
df_2016 = df.query("Year==2016")
```

[ ]:
```
df_2016.shape
```

[ ]:
```
df_2016.head(5)
```

[ ]:
```
df_2016.tail(5)
```

[ ]:
```
# create the choropleth in Plotly
fig = px.choropleth (
    df_2016,
    locations = 'Code',
    color = 'Internet_Usage',
```

```
    hover_name = 'Country',
    color_continuous_scale = px.colors.sequential.Jet,
    width = 1000,
    height = 1000,
)
fig.show()
```

https://plotly.com/python/builtin-colorscales/

[ ]:
```
# adding features to the Worldwide Choropleth Map
fig = px.choropleth (
    df_2016,
    locations = 'Code',
    color = 'Internet_Usage',
    hover_name = 'Country',
    color_continuous_scale = px.colors.sequential.Jet,
)

fig.update_layout (
    title_text = 'Internet Usage across the World in 2016',
    width = 1000,
    height = 1000,
    geo_scope = 'asia' #[north america | africa | australia | europe]
)

fig.show()
```

[ ]:
```
# set the projection type
fig = px.choropleth (
    df_2016,
    locations = 'Code',
    color = 'Internet_Usage',
    hover_name = 'Country',
    color_continuous_scale = px.colors.sequential.Jet,
)

fig.update_layout (
    title_text = 'Internet Usage across the World in 2016',
    width = 1000,
    height = 1000,
    geo = dict (projection = {'type' : 'hammer'})
)

fig.show()
```

https://plotly.com/python/map-configuration/

[ ]:
```
# adding animation to a Choropleth Map
fig = px.choropleth (
    df,
    locations = 'Code',
    color = 'Internet_Usage',
    hover_name = 'Country',
    animation_frame = 'Year',
    color_continuous_scale = px.colors.sequential.Jet,
)

fig.update_layout (
    title_text = 'Internet Usage across the World',
    width = 600,
    height = 600,
    geo = dict (projection = {'type' : 'hammer'})
)

fig.show()
```

[ ]:
```
# sort the dataset by Year
df.sort_values (by = ['Year'], inplace = True)
```

[ ]:
```
df.head()
```

[ ]:
```
# adding animation to a Choropleth Map using sorted Data
fig = px.choropleth (
    df,
    locations = 'Code',
    color = 'Internet_Usage',
    hover_name = 'Country',
    animation_frame = 'Year',
    color_continuous_scale = px.colors.sequential.Jet,
)

fig.update_layout (
    title_text = 'Internet Usage across the World',
    width = 600,
    height = 600,
    geo = dict (projection = {'type' : 'hammer'})
)

fig.show()
```

[ ]:
```
# read country code data
df_code = pd.read_csv ('country_codes.tsv', sep = '\t')
```

[ ]:
```
df_code.shape
```

[ ]:
```
df_code.head()
```

[ ]:
```
# create us state choropleth map
df_state = pd.read_csv ('us_state_population.tsv', sep = '\t')
```

[ ]:
```
df_state.shape
```

[ ]:
```
df_state.head()
```

[ ]:
```
df_state.tail()
```

[ ]:
```
df_state.describe()
```

[ ]:
```
df_state.info()
```

[ ]:
```
# use melt function to convert data to desired format
df_state = pd.melt (
    df_state,
    id_vars = ['State', 'Code'],
    var_name = 'Year',
    value_name = 'Population'
)
```

[ ]:
```
df_state.head()
```

[ ]:
```
# initialize the figure
fig = go.Figure (
    data = go.Choropleth (
        locations = df_state['Code'],
        z = df_state ['Population'].astype(int),
        locationmode = 'USA-states',
        colorscale = 'Blues',
        colorbar_title = 'Population'
    )
)
```

[ ]:
```
# update layout
fig.update_layout (
    title_text = 'US Population across States',
    geo_scope = 'usa'
)

fig.show()
```

[ ]:
```
# Scatter Plot on a Geographical Map
walmart = pd.read_csv ('walmart_store_openings.csv')
```

[ ]:
```
walmart.shape
```

[ ]:
```
walmart.head(5)
```

[ ]:
```
walmart.tail()
```

[ ]:
```
walmart.describe()
```

[ ]:
```
walmart.info()
```

[ ]:
```
# create a scatter plot of walmart stores in the us
fig = go.Figure (
    data = go.Scattergeo(
        lon = walmart['LON'],
        lat = walmart['LAT'],
        text = walmart['STREETADDR'],
        mode = 'markers'
    )
)

fig.update_layout (
    title = 'Walmart Stores in the US',
    geo_scope = 'usa'
)

fig.show()
```

Bubble Plots

[ ]:
```
# get number of walmart stores by state
walmart_state = walmart.groupby('STRSTATE').count()
```

[ ]:
```
walmart_state.head()
```

[ ]:
```
walmart_state = walmart_state['storenum'].reset_index()
```

[ ]:
```
walmart_state.rename (columns = {'storenum' : 'num_stores'}, inplace = True)
```

[ ]:
```
# generate bubble plot
fig = px.scatter_geo (
    walmart_state,
    locations = 'STRSTATE',
    size = 'num_stores',
    locationmode = 'USA-states',
    hover_name = 'STRSTATE',
    size_max = 20
)
```

```
fig.update_layout (
    title_text = 'Walmart Stores in the US',
    geo_scope = 'usa'
)

fig.show()
```

[ ]:
```
# read the number of internet users by country
internet_users = pd.read_csv ('internet_users.csv')
```

[ ]:
```
internet_users.shape
```

[ ]:
```
internet_users.head(5)
```

[ ]:
```
internet_users.tail(5)
```

[ ]:
```
internet_users.describe()
```

[ ]:
```
internet_users.info()
```

[ ]:
```
internet_users.rename (columns = {'Number of internet users (users)' :␣
↪'users'}, inplace = True)
```

[ ]:
```
internet_users.head()
```

[ ]:
```
# sort by Year
internet_users.sort_values (by = ['Year'], inplace = True)
```

[ ]:
```
# animate the internet usage
fig = px.scatter_geo (
    internet_users,
    locations = 'Code',
    size = 'users',
    hover_name = 'Country',
    size_max = 40,
    animation_frame = 'Year'
)

fig.update_layout (
    title_text = 'Internet Users across the World',
    geo = dict (projection = {'type' : 'aitoff'})

)

fig.show()
```

[ ]:
```
# plot all the airports in the US
airports = pd.read_csv ('airports.csv')
```

[ ]:
```
airports.shape
```

[ ]:
```
airports.head(5)
```

[ ]:
```
airports.tail(5)
```

[ ]:
```
airports.describe()
```

[ ]:
```
airports.info()
```

[ ]:
```
# plot all airports
fig = go.Figure()

fig.add_trace (
    go.Scattergeo (
        locationmode = 'USA-states',
        lon = airports['LONGITUDE'],
        lat = airports['LATITUDE'],
        hoverinfo = 'text',
        text = airports['AIRPORT'],
        mode = 'markers',
        marker = dict (size = 2, color = 'black')
    )
)

fig.update_layout (
    title_text = 'Airports in the US',
    showlegend = False,
    geo = go.layout.Geo (
        scope = 'usa'
    )
)

fig.show()
```