


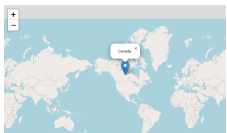

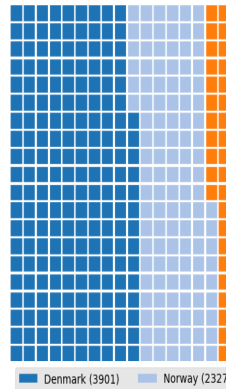


Data Visualization with Python

Cheat Sheet : Maps, Waffles, WordCloud and Seaborn

Function	Description	Syntax	Example	Visual
Folium				
Map	Create a map object with specified center coordinates and zoom level.	<code>folium.Map(location=[lat, lon], zoom_start=n)</code>	<pre>world_map = folium.Map() canada = folium.Map(location=[56.130, -106.35], zoom_start=4)</pre>	
Marker	Add a marker to the map with custom icon, popup, and tiles Tiles as Stamen Toner	<code>folium.Marker(location=[lat, lon], popup='Marker Popup', tiles='Stamen Toner').add_to(map)</code>	<code>folium.Marker(location=[556.130, -106.35], tooltip='Marker', tiles='Stamen Toner').add_to(world_map)</code>	
	Tiles as Stamen Terrain	<code>folium.Marker(location=[lat, lon], popup='Marker Popup', tiles='Stamen Terrain').add_to(map)</code>	<code>folium.Marker(location=[556.130, -106.35], tooltip='Marker', tiles='Stamen Terrain').add_to(world_map)</code>	
Circle	Add a circle to the map with specified radius, color, and fill opacity.	<code>folium.features.CircleMarker(location=[lat, lon], radius=n, color='red', fill_opacity=n).add_to(map)</code>	<code>folium.features.CircleMarker(location=[56.130, -106.35], radius=1000, color='red', fill_opacity=0.5).add_to(world_map)</code>	
Choropleth	Create a choropleth map based on a GeoJSON file and a specified data column.	<code>folium.Choropleth(geo_data='path/to/geojson_file', data=df, columns=['region', 'value_column'], key_on='feature.properties.id', radius=n, color='red', fill_opacity=0.7, line_opacity=0.2, legend_name='Legend').add_to(map)</code>	<code>world_map.choropleth(geo_data=world_geo, data=df_can, columns=['Country', 'Total'], key_on='feature.properties.name', fill_color='YlGnBu', fill_opacity=0.7, line_opacity=0.2, legend_name='Immigration to Canada')</code>	
PyWaffle				
Waffle	Create a waffle chart based on values and categories.	<pre>plt.figure(FigureClass = Waffle, rows = 20, columns = 30, values = values) waffle_chart = waffle.Waffle(values=[value1, value2, ...], rows=n, columns=n)</pre>	<pre>plt.figure(FigureClass = Waffle, rows = 20, columns = 30, values = df_dsn['Total'], cmap_name = 'tab20', legend = {'labels': label, 'loc': 'lower left', 'bbox_to_anchor': (0, -0.1, 'ncol': 3)})</pre>	
Legend	Add a legend to the waffle chart.	<code>waffle_chart.legend(loc='upper left', bbox_to_anchor=(1, 1))</code>		
Title	Add a title to the waffle chart.	<code>waffle_chart.set_title('Waffle Chart Title')</code>		

Changelog

Date	Version	Changed by	Change Description
2023-06-18	0.1	Dr. Pooja	Initial version created