

**Nama : Ghina Khoerunnisa**  
**Program : Data Scientist Intern**

### Advanced Visualization

- **Waffle Chart** merupakan Pie Chart dengan versi persegi. Fungsinya sama seperti pie chart yaitu untuk menampilkan proporsi suatu data. Di python sendiri belum ada library yang mendukung, tetapi dapat dibuat dari scratch. Langkah-langkahnya adalah sebagai berikut:
  - Step 1: Menghitung proporsi untuk setiap kategori yang akan ditampilkan. Untuk menghitung proporsi dilakukan dengan membagi value dari suatu kategori dengan total value untuk seluruh kategori.
  - Step 2 : Mendefinisikan lebar dan tinggi waffle serta luasnya (ukuran keseluruhan).
  - Step 3 : Setelah mendapatkan proporsi dan mendefinisikan ukuran, maka selanjutnya adalah menentukan jumlah ubin/kotak untuk setiap kategori sesuai dengan proporsinya.
  - Step 4 : Membuat matriks yang shapenya sesuai dengan lebar dan tinggi yang telah didefinisikan. Lebar sebagai kolom dan tinggi sebagai baris pada .shape. Kemudian mengisi matriksnya
  - Step 5 : Membuat plotnya dari matriks yang telah dibuat sebelumnya dengan cara plt.matshow(matriks).
  - Step 6 : Styling plotnya dengan mengeset xticks, yticks, dan memberi grid pada plotnya
  - Step 7 : Menambahkan legend seperti pada gambar di bawah,

```
values_cumsum = np.cumsum(df_waffle['Total'])
legend_handles = []
for i, category in enumerate(df_waffle.index.values):
    label_str = category + ' (' + str(df_waffle['Total'][i]) + ')'
    color_val = colormap(float(values_cumsum[i])/total)
    legend_handles.append(mpatches.Patch(color=color_val, label=label_str))

plt.legend(handles=legend_handles, loc='lower center', ncol=len(df_waffle.index.values), bbox_to_anchor=(0., -0.5),
plt.show())
```

- **Word Cloud**, merupakan visualisasi yang menampilkan data teks dalam bentuk visual. Semakin banyak suatu kata muncul dalam sumber data teks kita, maka semakin besar juga ukuran dan semakin tebal warna dari kata tersebut di cloudnya. Untuk menampilkannya membutuhkan package wordcloud `from wordcloud import WordCloud, STOPWORDS`. Langkah yang pertama setelah import adalah membuat object word cloud, kemudian men-generate word cloud dari data teks yang ingin ditampilkan. Lalu untuk menampilkannya menggunakan `plt.imshow()`.

```
alice_wc = WordCloud(background_color='white', max_words=2000, stopwords=stopwords)

alice_wc.generate(alice_novel)

plt.imshow(alice_wc, interpolation='bilinear')
```

- **Regression Plot** merupakan visualisasi yang menunjukkan scatter plot dengan garis regresinya. Untuk menampilkan regresi dapat menggunakan library seaborn. `import`

`seaborn as sns`. Langkah pertama setelah import adalah membuat dataframe baru yang menampung data x dan y yang akan divisualisasikan. Kemudian membuat plot regresi dengan `sns.regplot(x=data_x, y=data_y, data=dataframe)`.

- **Folium** merupakan library visualisasi di python seperti matplotlib hanya saja folium digunakan untuk memvisualisasikan data geospasial. Untuk menampilkannya kita perlu untuk meng-install (dengan pip atau conda) dan meng-import librarynya dengan `import folium`. Kita dapat menampilkan peta dunia yang interaktif dengan `folium.Map(location=[latitude, longitude], zoom_start=10)`. `Zoom start` merupakan atribut yang digunakan untuk menentukan tingkat zoom pada saat awal peta ditampilkan. Semakin tinggi nilainya maka semakin besar zoomnya.

Di Folium terdapat beberapa gaya peta yang berbeda, yaitu Stamen Toner dan Stamen Terrain. Stamen Toner merupakan gaya peta yang mana peta berwarna hitam putih. Stamen Toner cocok untuk data mashup dan untuk menjelajahi dan memvisualisasikan liku-liku sungai dan zona pesisir. Untuk menampilkannya hanya perlu menambahkan atribut `tiles="Stamen Toner"`. Sedangkan Stamen Terrain merupakan gaya peta yang mana menampilkan bayangan bukit dan warna vegetasi alami. Stamen Terrain menampilkan pelabelan lanjutan dan generalisasi garis dari jalan dengan dua jalur. Untuk menampilkannya hanya perlu menambahkan atribut `tiles="Stamen Terrain"`.

Pada Folium juga kita dapat menampilkan peta dengan marker. Langkah awal sama seperti pada saat membuat peta sebelumnya. Setelah itu membuat marker dengan cara membuat feature group kemudian looping untuk menambah setiap marker ke feature groupnya. Lalu feature group tersebut ditambahkan ke peta yang telah dibuat sebelumnya.

- **Choropleth** merupakan peta tematik di mana area yang diarsir atau berpola sebanding dengan pengukuran variabel statistik yang ditampilkan di peta, seperti kepadatan penduduk atau pendapatan per kapita. Untuk membuat Choropleth, kita membutuhkan file GeoJSON yang mendefinisikan area negara bagian, kabupaten, atau negara yang dibutuhkan. Langkah pertama untuk menampilkan peta Choropleth adalah membuat peta dengan cara yang sebelumnya `folium.Map(location=[latitude, longitude], zoom_start=10)`. Kemudian men-generate peta Choroplethnya misal contoh kasusnya adalah total imigrasi masing-masing negara ke Kanada dari tahun 1980 hingga 2013 yang disimpan dalam dataframe `citizen`.

```
world_geo = r'world_countries.json'

world_map = folium.Map(location=[0, 0], zoom_start=2)

world_map.choropleth(
    geo_data=world_geo,
    data=citizen,
    columns=['Country', 'Total'],
    key_on='feature.properties.name',
    fill_color='YlOrRd', #Yellow, Orange, Red
    fill_opacity=0.7,
    line_opacity=0.2,
    legend_name='Immigration to Canada'
)
```