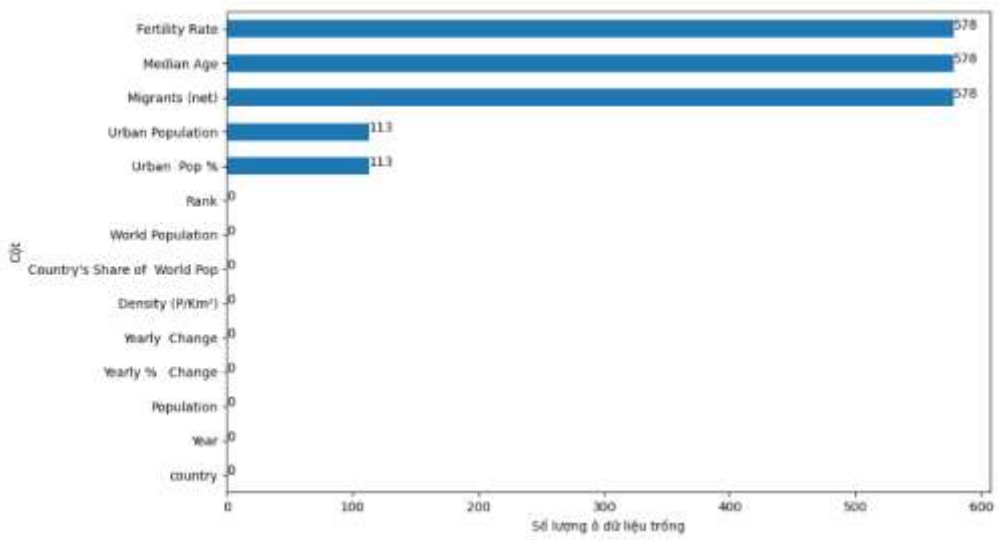
Chương 1. TỔNG QUAN VỀ DỮ LIỆU

1.1. Nội dung:

Bộ dữ liệu world population and forecast là bộ dữ liệu chứa thông tin về dân số của các quốc gia trên toàn thế giới được lấy từ năm 1955 đến năm 2020, kèm theo đó là bộ dữ liệu dự đoán dân số các quốc gia trên thế giới từ năm 2020 đến năm 2025.Bộ dữ liệu được đăng tải tại trang web https://www.worldometers.info/population/ vào ngày 11 tháng 6 năm 2023.

1.2. Thuộc tính:

Dữ liệu bao gồm 14 cột và 4196 dòng dữ liệu. Trong đó có tổng cộng 1960 hàng dữ liệu trống được biểu diễn bằng đồ thị sau:



1.3. Chuẩn hóa dữ liệu:

Dữ liệu đã có thể thể hiện được ý nghĩa của mình nhưng vẫn còn thô nên phải tiến hành chuẩn hóa lại dữ liệu. Việc chuẩn hóa bộ dữ liệu này bao gồm những công việc sau:

- Đặt lại tên của các cột dữ liệu.
- Thay thế các ký tự biểu thị giá trị trống thành các giá trị trống (để dễ dàng xử lý các bước xử lý sau).
- Thay đổi kiểu dữ liệu và định dạng lại 1 số dữ liệu.

```
path = "/content/drive/MyDrive/Colab Notebooks/world-population.csv"
wp_df = pd.read_csv(path)
wp_df.replace('N.A', pd.NA, inplace = True)
wp_df.replace('N.A.', pd.NA, inplace = True)
wp_df['Yearly %  Change'] = wp_df['Yearly %  Change'].str.rstrip('%')
wp_df['Urban  Pop %'] = wp_df['Urban  Pop %'].str.rstrip('%')
wp_df['Country\'s Share of  World Pop'] = wp_df['Country\'s Share of  World Pop'].str.rstrip('%')
wp_df['Population'] = wp_df['Population'].astype(float)
wp_df['World Population'] = wp_df['World Population'].astype(float)
wp_df['Yearly %  Change'] = pd.to_numeric(wp_df['Yearly %  Change'])
wp_df['Urban  Pop %'] = pd.to_numeric(wp_df['Urban  Pop %'])
wp_df['Urban Population'] = pd.to_numeric(wp_df['Urban Population'])
wp_df['Country\'s Share of  World Pop'] = pd.to_numeric(wp_df['Country\'s Share of  World Pop'])
new_colums_name = ['country', 'year', 'population', 'year percent change', 'year change', 'migrants(net)',
                   'median age', 'fertility rate', 'density (P/Km²)', 'urban pop percent', 'urban population',
                   'country share of  world pop', 'world population', 'rank']
new_wp_df = wp_df
new_wp_df.columns = new_colums_name
```

Hình 1.2 đoạn code tiến hành chuẩn hóa dữ liệu

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4196 entries, 0 to 4195
Data columns (total 14 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   country                     4196 non-null   object
 1   year                        4196 non-null   int64
 2   population                  4196 non-null   float64
 3   year percent change         4196 non-null   float64
 4   year change                 4196 non-null   int64
 5   migrants(net)               3618 non-null   float64
 6   median age                  3618 non-null   float64
 7   fertility rate              3618 non-null   float64
 8   density (P/Km²)             4196 non-null   int64
 9   urban pop percent           4083 non-null   float64
 10  urban population            4083 non-null   float64
 11  country share of  world pop  4196 non-null   float64
 12  world population            4196 non-null   float64
 13  rank                        4196 non-null   int64
dtypes: float64(9), int64(4), object(1)
memory usage: 459.1+ KB
```

Hình 1.3 kết quả thu được sau khi thực hiện đoạn code

Xử lý những dữ liệu trống: ở đây chúng tôi sử dụng lớp KNNImputer của thư viện sklearn để điền vào các dữ liệu bị trống (KNNImputer cung cấp phép gán để điền vào các giá trị thiếu bằng cách sử dụng phương pháp K-Nearest Neighbors).

```
from sklearn.impute import KNNImputer
knn_imputer = KNNImputer(n_neighbors= 15)
temp_df = wp_df.drop(columns = ['country'])
wp_imputer = pd.DataFrame(knn_imputer.fit_transform(temp_df), columns = temp_df.columns)
wp_df[wp_imputer.columns] = wp_imputer
new_path = "/content/drive/MyDrive/Colab Notebooks/temp_df.csv"
wp_df.to_csv(new_path, index = False)
wp_df.info()
```

Hình 1.4 đoạn code xử lý giá trị trống bằng lớp KNNImputer

- Ta thu được kết quả sau:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4196 entries, 0 to 4195
Data columns (total 14 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   country                     4196 non-null   object
 1   year                        4196 non-null   float64
 2   population                  4196 non-null   float64
 3   year percent change         4196 non-null   float64
 4   year change                 4196 non-null   float64
 5   migrants(net)               4196 non-null   float64
 6   median age                  4196 non-null   float64
 7   fertility rate              4196 non-null   float64
 8   density (P/Km²)             4196 non-null   float64
 9   urban pop percent           4196 non-null   float64
 10  urban population            4196 non-null   float64
 11  country share of  world pop  4196 non-null   float64
 12  world population            4196 non-null   float64
 13  rank                        4196 non-null   float64
dtypes: float64(13), object(1)
memory usage: 459.1+ KB
```

Hình 1.5 kết quả sau khi code xử lý giá trị trống bằng lớp KNNImputer

Chương 2. PHÂN TÍCH CƠ BẢN VỀ DỮ LIỆU

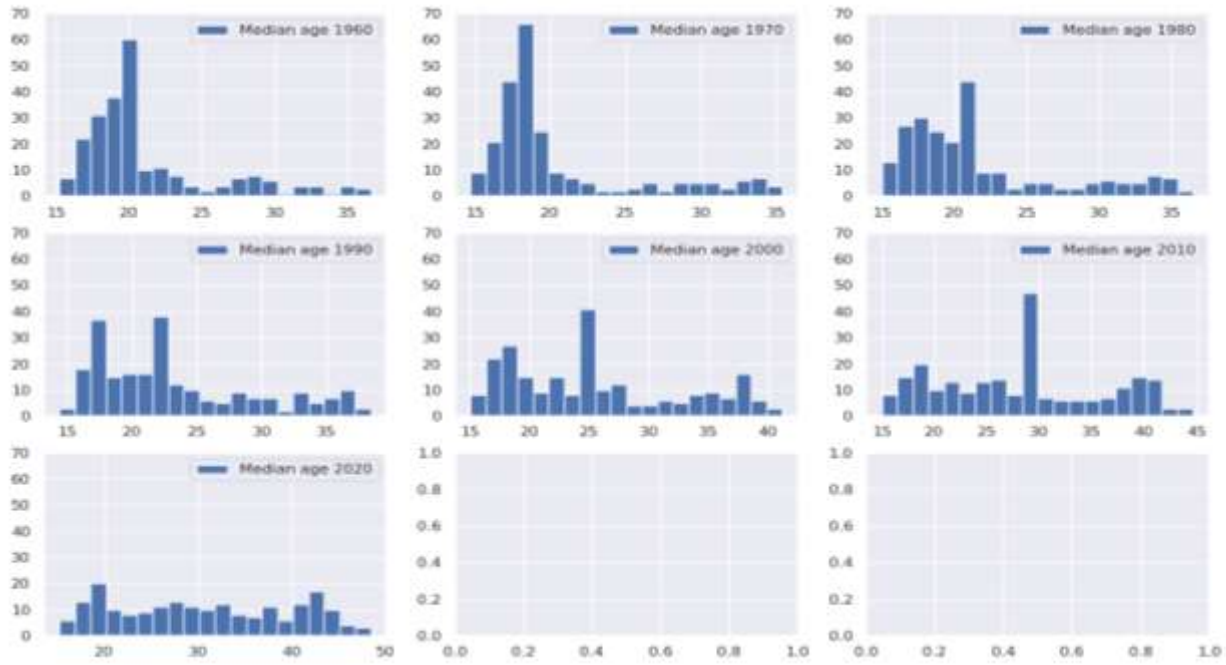
2.1. Phân tích cơ bản về các số liệu của dữ liệu:

Tuổi trung vị của các nước trên thế giới từ 1960 đến 2020 qua mỗi 10 năm:

- Nhóm độ tuổi trung vị từ 17 đến 20 tuổi giảm dần qua nhiều năm.
- Đến năm 2020 nhóm độ tuổi trung vị từ 40 đến 45 chiếm 1 phần đáng kể.

➔ Tuổi trung vị của các quốc gia trên toàn thế giới đang có xu hướng chuyển dịch về phía phải và cân bằng nhau, tuổi trung vị của con người tăng đáng kể.

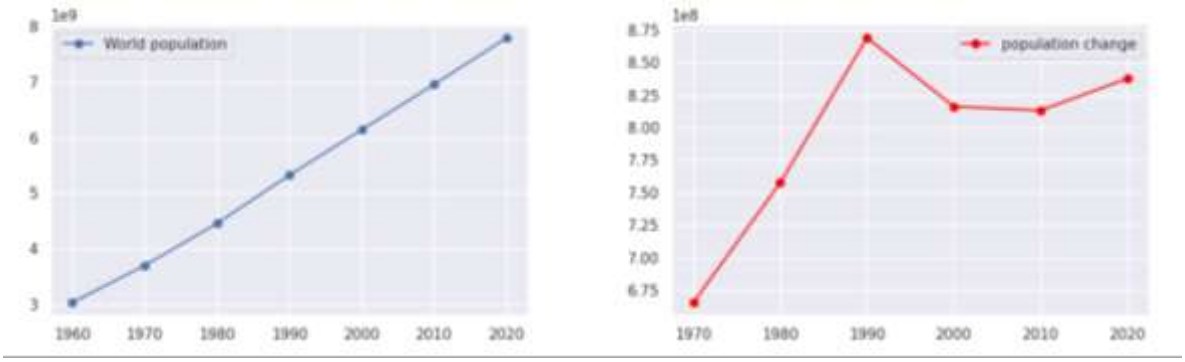
Hình 2.1 Biểu đồ Tuổi trung vị của các nước trên thế giới từ 1960 đến 2020 qua mỗi 10 năm



Sự biến động dân số thế giới:

- Tốc độ gia tăng dân số thế giới trong khoảng thời gian từ năm 1960 đến năm 1990 tăng nhanh, đạt đỉnh điểm vào năm 1990.
- Tốc độ gia tăng dân số thế giới giảm từ năm 1990 đến năm 2010 có xu hướng giảm.

➔ Nhìn chung dân số thế giới sau năm trước 2000 phát triển rất nhanh, biến động mạnh. Dân số thế giới sau năm 2000 dần được kiểm soát, giảm mức độ biến động và ổn định hơn.

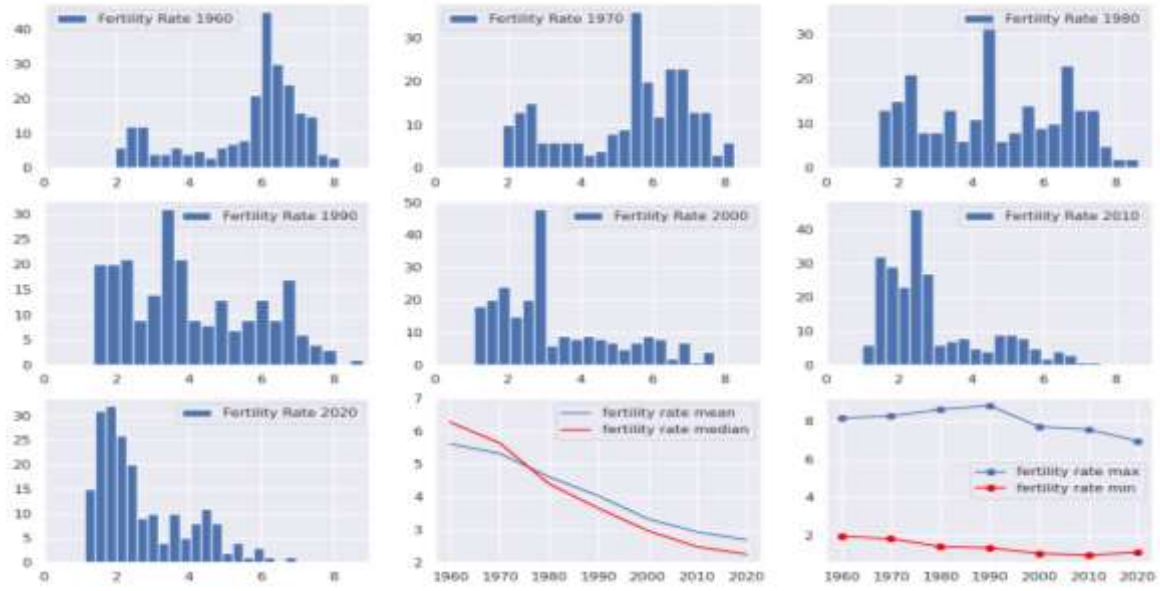


Hình 2.2 Biểu đồ dân số thế giới từ năm 1955 đến 2022

Tỷ suất sinh của các quốc gia trên thế giới từ năm 1960 đến năm 2020:

- Tỷ suất sinh trước năm 1980 lệch nhiều về phía bên phải. Có nhiều quốc gia có tỷ suất sinh cao
- Tỷ suất sinh giữa các quốc gia dần cân bằng
- Tỷ suất sinh từ năm 1990 đến nay có xu hướng lệch mạnh về phía bên trái. Tỷ suất sinh của các quốc gia trên thế giới đang ngày càng giảm
- Giá trị trung bình và trung vị của tỷ suất sinh từ năm 1970 đến năm 1980 có lúc gần bằng nhau.

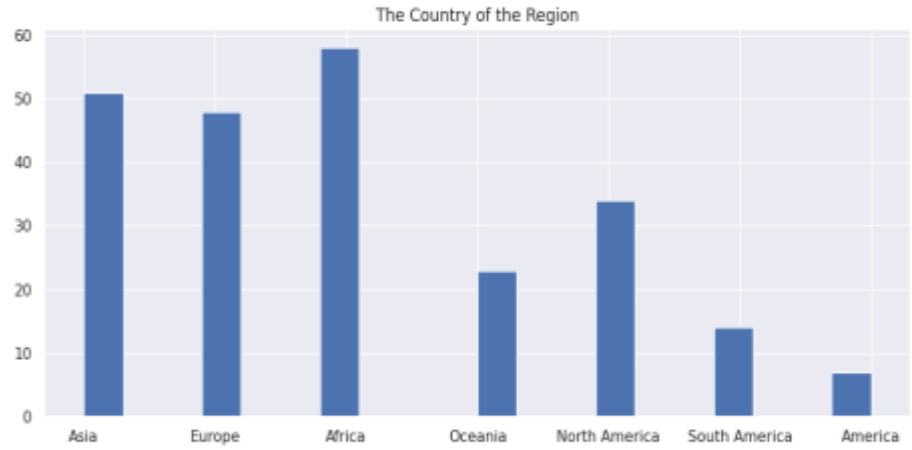
➔ Tỷ suất sinh của các quốc gia đang có xu hướng chuyển dịch từ năm 1980 đến năm 2020. Có thời điểm cân bằng tỷ suất sinh vào khoảng thời gian từ năm 1970 đến năm 1980.



Hình 2.3 Biểu đồ biến động tỷ suất sinh của các quốc gia trên thế giới từ năm 1960 đến 2020

Để dễ dàng phần tích dữ liệu, chúng em thêm vào dữ liệu cột region. Dữ liệu của region được lấy từ trang github: https://github.com/lukes/ISO-3166-Countries-with-Regional-

Codes/blob/master/all/all.csv, từ dữ liệu continent được lấy từ thư viện geopandas và một vài dữ liệu tự thu thập được để lấy được dữ liệu khu vực của các quốc gia.

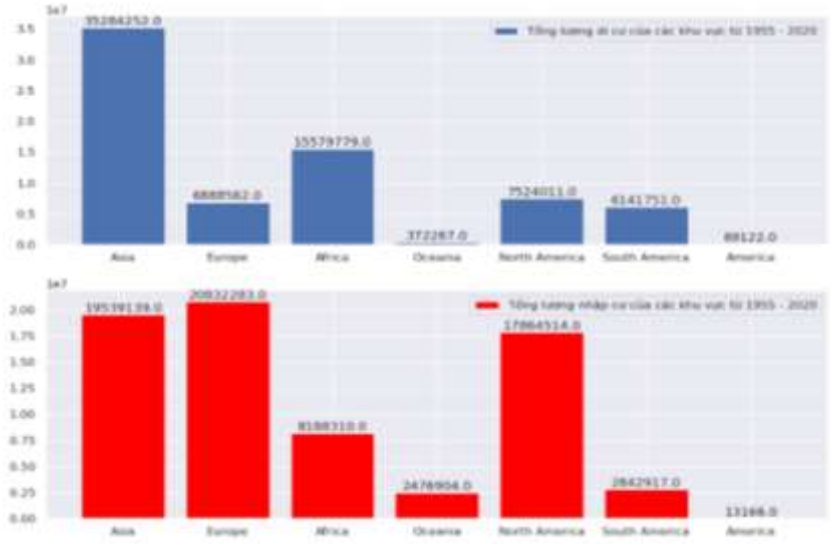


Hình 2.4 Biểu đồ phân bố các quốc gia theo khu vực

Số lượng dân số nhập cư và di cư:

- Số lượng người di cư chiếm nhiều nhất tại khu vực Asia và thấp nhất ở America
- Số lượng người nhập cư chiếm số lượng lớn ở Europe, North America, Asia.

➔ Người ở khu vực Asia chiếm số lượng lớn số người di cư và nhập cư. Các khu vực North America, Europe, Oceania là nơi có số lượng người nhập cư khá cao. Khu vực America không biến động nhiều.

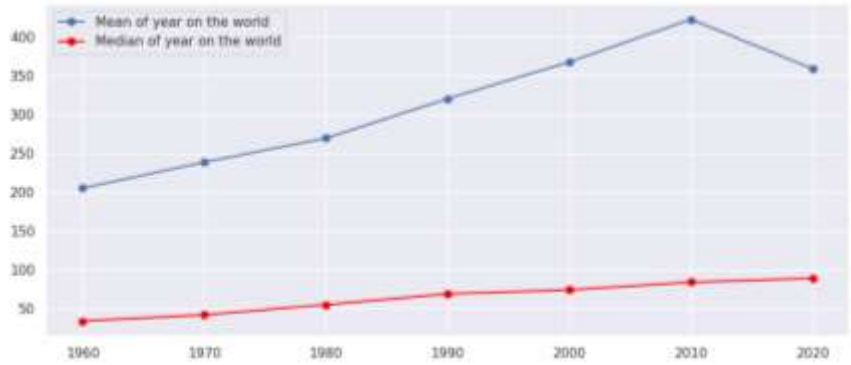


Hình 2.5 Biểu đồ phân bố di nhập cư theo khu vực

Mật độ dân số trên thế giới từ năm 1960 – 2020:

- Giá trị bình vị của số liệu mật độ dân số thế giới khá cao, tăng từ 1960 đến 2010 nhưng đến năm 2020 có giảm nhưng vẫn cao.
- Giá trị trung vị của số liệu mật độ dân số thế giới thấp hơn rất nhiều, tăng trưởng ổn định qua từng khoảng thời gian.

➔ Mật độ dân số thế giới phân bố không đồng đều, những nơi có mật độ dân số rất dày đặt, một số nơi dân cư rất thưa thớt.

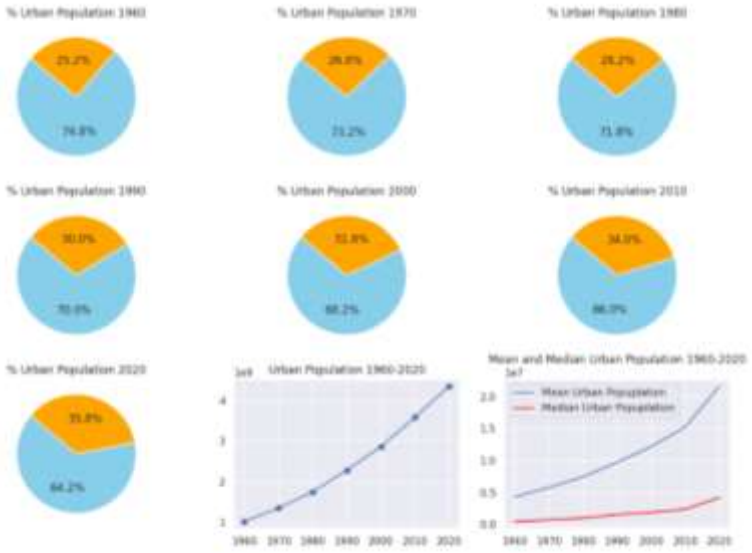


Hình 2.6: Biểu đồ phân bố mật độ dân số thế giới từ 1960 – 2020

Số lượng dân cư thành thị trên thế giới từ 1960 – 2020:

- Dân cư thành thị chiếm phần trăm đáng kể qua từng mốc thời gian (từ 25,2% - 35,8%).
- Giá trị trung bình và trung vị ngày càng phân tách nhau.

➔ Dân cư thành thị ngày càng tăng, tốc độ đô thị hóa ngày càng lớn, số lượng dân cư giữa các đô thị tại các quốc gia có sự không đồng đều, khả năng có thể dẫn đến bùng nổ dân số tại các đô thị lớn.

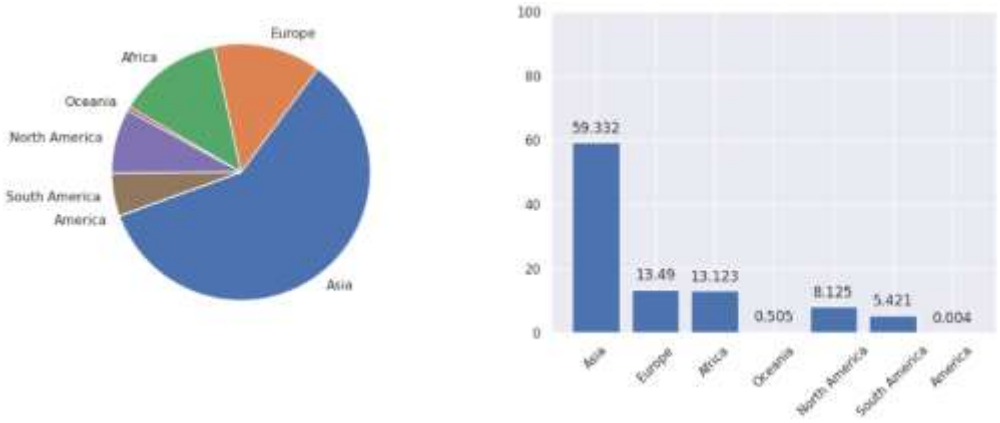


Hình 2.7 biểu đồ dân cư thành thị trong khoảng thời gian từ 1960 - 2020

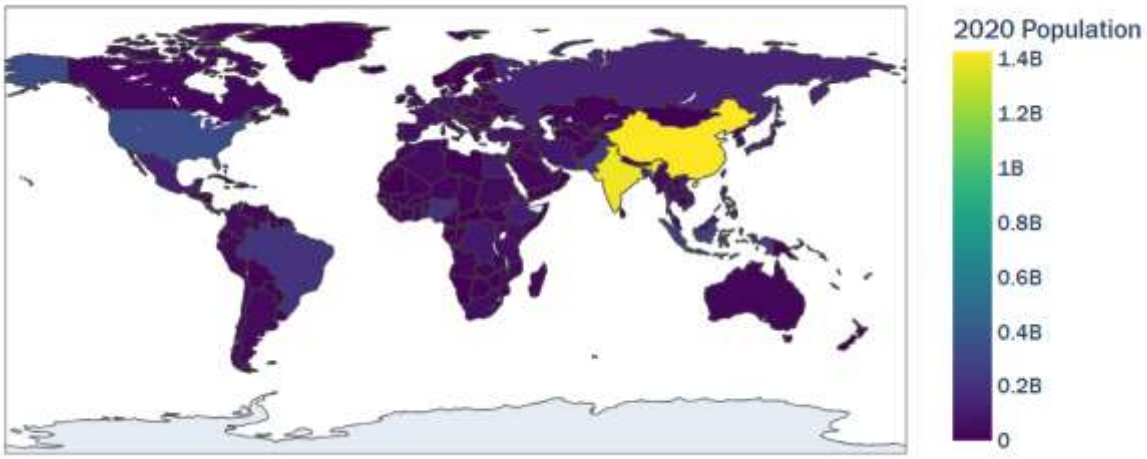
Sự ảnh hưởng về dân số từ các khu vực từ năm 1955 – 2020:

- Khu vực Asia có tỷ lệ phần trăm đóng góp dân số cao nhất thế giới.
- Khu vực America có tỷ lệ phần trăm đóng góp dân số rất thấp.

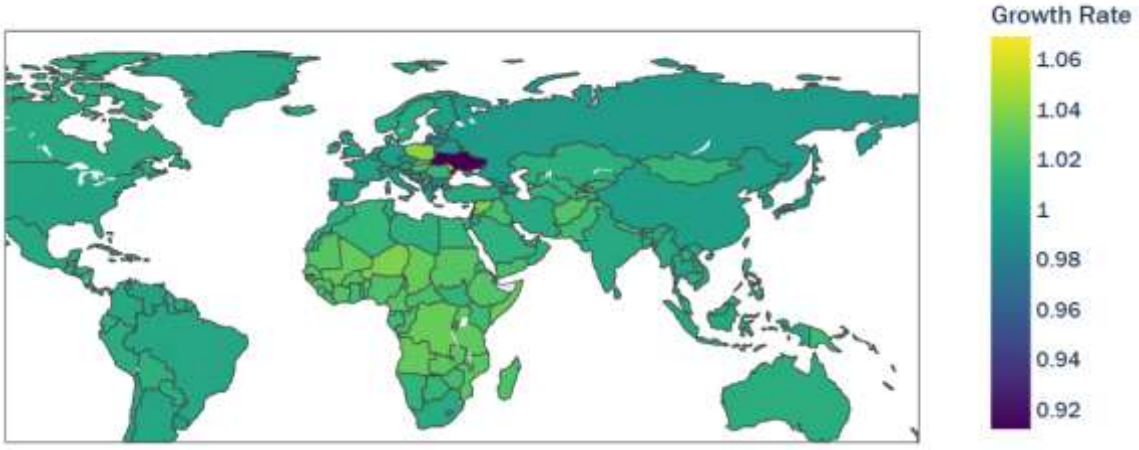
➔ Các nước khu vực Asia cao nhất trong tỷ lệ phần trăm đóng góp dân số trong khi các khu vực còn lại có tỷ lệ chỉ tầm 15%, cá biệt khu vực America có tỷ lệ xê xích 0.



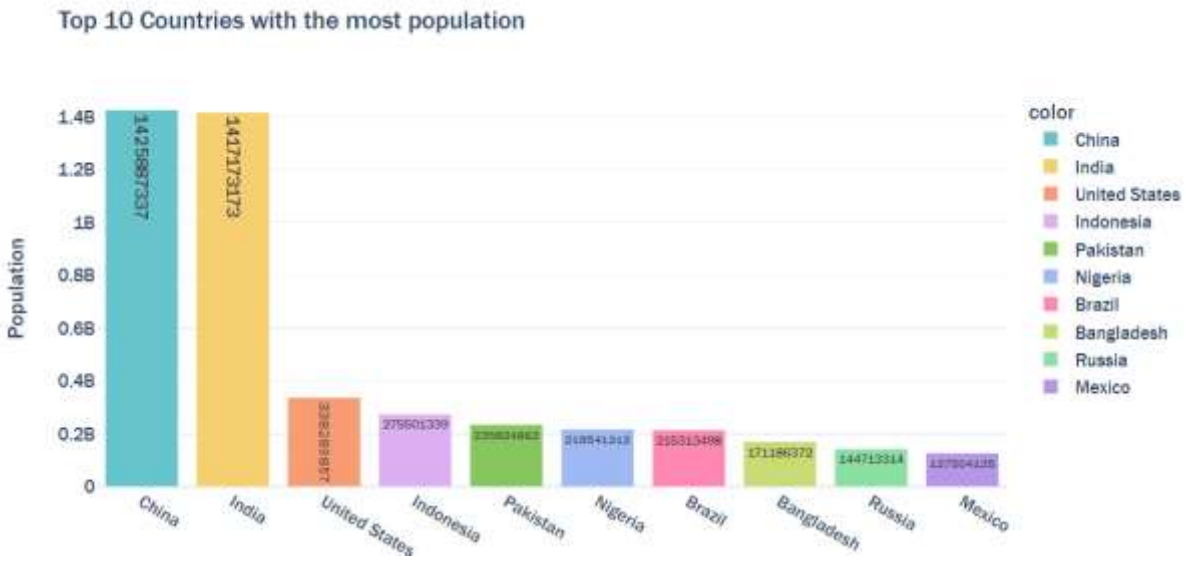
Hình 2.8 Biểu đồ đóng góp số dân của các khu vực trên thế giới.



Biểu đồ dân số thế giới năm 2020



Biểu đồ mức độ tăng trưởng dân số theo từng khu vực



Top 10 những quốc gia có số dân nhiều nhất thế giới

2.2. Kết luận:

Về dân số thế giới từ năm 1960 – 2020 có những nhận định chính như sau:

- Thông qua các biểu đồ chúng ta có thể thấy được mức độ tăng trưởng dân số thế giới chia làm hai giai đoạn đó là trước năm 1990 và từ năm 1990 đến 2020. Giai đoạn trước 1990 dân số thế giới tăng nhanh, sau giai đoạn đó dân số thế giới chững lại và có phần giảm và có phần điều hòa ổn định hơn nhưng nhìn chung vẫn cao.
- Dân số của China, India phân hóa rõ rệt so với các quốc gia còn lại.
- Dân số thế giới ở khu vực Asia là đông nhất đồng, là khu vực có số dân di cư cũng như nhập cư đông nhất đồng thời là khu vực có mức độ đóng góp dân số thế giới chiếm hơn 50% dân số (do có sự hậu thuẫn từ dân số của 2 quốc gia đông dân nhất thế giới đó là China và India). Có thể thấy phần lớn dân cư khu vực này di nhập trong chính khu vực của mình khá cao.
- Có sự phân hóa khá cao và rõ rệt giữa các quốc gia trong các năm trên thế giới.
- Dân cư thành thị ngày càng cao cho thấy mức độ đô thị hóa ngày càng tăng.
- Tỷ lệ sinh có sự chuyển dịch từ khá thấp đến cao dần, khoảng thời gian ổn định nhất là từ năm 1970 - 1980.
- Biểu đồ mức độ tăng trưởng dân số theo từng khu vực cho thấy các quốc gia châu Âu có mức độ tăng trưởng chậm. Ngược lại các vùng quốc gia ở châu Phi lại cao hơn. Cho thấy các nước phát triển có mức độ tăng trưởng dân ít ngược lại các nước đang phát triển lại có mức độ tăng trưởng cao.
- Tuổi trung vị của các quốc gia phát triển từ lệch trái dần sang lệch phải và ngày càng cân bằng, điểm tích cực ở đây có thể thấy là tuổi trung vị dưới 20 tuổi đang dần ít đi, xuất hiện nhóm độ tuổi trung vị cao cho thấy tuổi thọ của con người ngày càng tăng. Điểm tiêu cực cũng chính do có xuất hiện nhóm độ tuổi trung vị cao cho thấy đa số dân cư có tuổi cao, trẻ em và thanh niên của các quốc gia đó thấp hơn tạo ra nhiều vấn đề về xã hội.

Chương 3. PHÂN TÍCH DỮ LIỆU CHUYÊN SÂU

3.1. Gom cụm với K-Mean

Thư viện sử dụng:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

Chọn các cột cần gom cụm từ kho dữ liệu:

```
wp_df = wp_df[['Year', 'Population', 'Yearly %  Change', 'Yearly  Change',
       'Migrants (net)', 'Median Age', 'Fertility Rate', 'Density (P/Km²)',
       'Urban  Pop %', 'Urban Population', "Country's Share of  World Pop",
       'World Population', 'Rank']]
```

Sử dụng lớp StandardScaler để chuẩn hóa dữ liệu về mức trung bình bằng 0 và độ lệch chuẩn bằng 1:

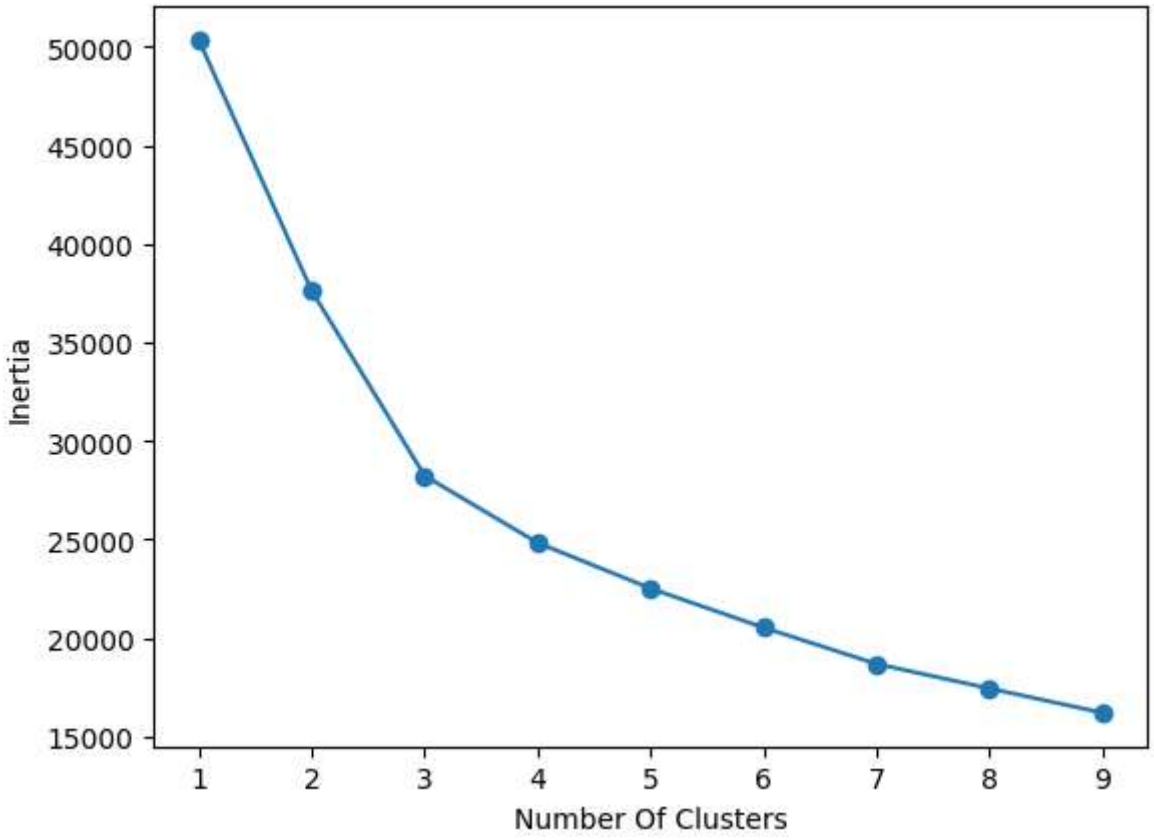
```
scaler = StandardScaler()
wp_df[['Population_T', 'Yearly %  Change_T', 'Yearly  Change_T',
       'Migrants (net)_T', 'Median Age_T', 'Fertility Rate_T', 'Density (P/Km²)_T',
       'Urban  Pop %_T', 'Urban Population_T', "Country's Share of  World Pop_T",
       'World Population_T', 'Rank_T']] = scaler.fit_transform(wp_df[['Population', 'Yearly %  Change', 'Yearly  Change',
       'Migrants (net)', 'Median Age', 'Fertility Rate', 'Density (P/Km²)',
       'Urban  Pop %', 'Urban Population', "Country's Share of  World Pop",
       'World Population', 'Rank']])
```

Sử dụng một thuật toán tối ưu hóa KMeans. Thuật toán này được sử dụng để tìm số lượng cụm tối ưu cho dữ liệu. Kết quả ta nhận được một biểu đồ Elbow. Biểu đồ này sẽ giúp ta xác định số lượng cụm tối ưu cho dữ liệu:

```
def optimise_k_means(data, max_k):
  means = []
  inertias = []
  for k in range(1, max_k):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(data)
    means.append(k)
    inertias.append(kmeans.inertia_)
  plt.plot(means, inertias, 'o-')
  plt.xlabel('Number Of Clusters')
  plt.ylabel('Inertia')

optimise_k_means(wp_df[['Population_T', 'Yearly %  Change_T', 'Yearly  Change_T',
       'Migrants (net)_T', 'Median Age_T', 'Fertility Rate_T', 'Density (P/Km²)_T',
       'Urban  Pop %_T', 'Urban Population_T', "Country's Share of  World Pop_T",
       'World Population_T', 'Rank_T']], 10)
```

Với max_k = 10, ta thu được biểu đồ số lượng cụm của các năm:



Dựa vào biểu đồ trên ta có thể phân dữ liệu từng năm thành 3 cụm là tối ưu nhất.

Tạo một mô hình KMeans với số lượng cụm K là 3 và huấn luyện mô hình trên dữ liệu.

Tạo thêm 1 cột chứa nhãn của mỗi điểm dữ liệu trong cụm:

```
kmeans = KMeans(n_clusters=3)

kmeans.fit(wp_df[['Population_T', 'Yearly %   Change_T', 'Yearly  Change_T',
       'Migrants (net)_T', 'Median Age_T', 'Fertility Rate_T', 'Density (P/Km²)_T',
       'Urban  Pop %_T', 'Urban Population_T', "Country's Share of  World Pop_T",
       'World Population_T', 'Rank_T']])

wp_df['kmeans'] = kmeans.labels_
```

Tạo mảng lưu dữ liệu của từng năm (1955-2020):

```
array = []
for i in wp_df[['Year']].drop_duplicates()["Year"].iteritems():
  array.append(i[1])

data = []
for i in array:
  data.append(wp_df[wp_df['Year'] == i])
```

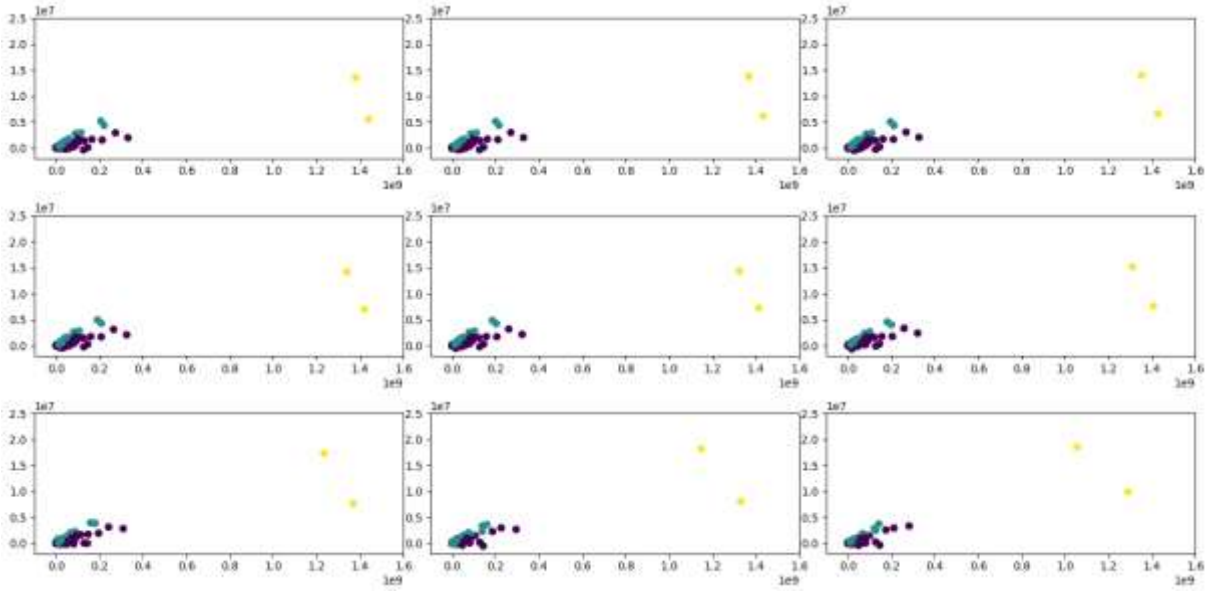
3.1.1. Tốc độ tăng trưởng dân số:

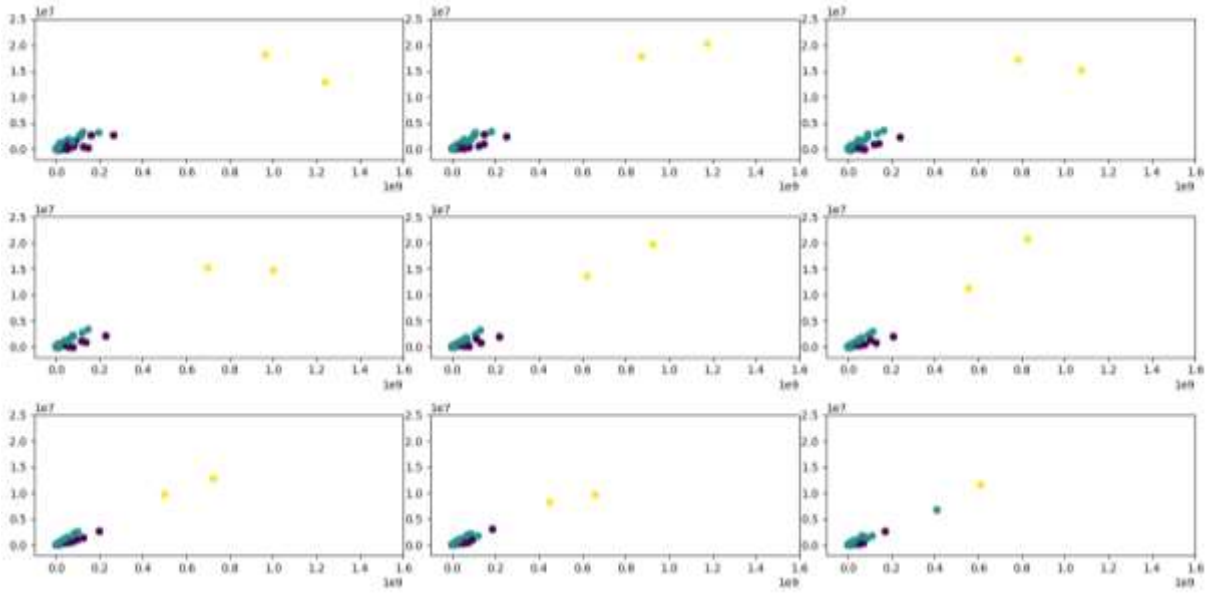
Sử dụng matplotlib.pyplot để tạo biểu đồ phân tán của dữ liệu sau khi thực hiện phân cụm KMeans:

```
fig, axes = plt.subplots(nrows=6, ncols=3, figsize = (15, 15))

for i, d in enumerate(data):
  axes[i // 3, i % 3].scatter(x=d['Population'], y=d['Yearly  Change'], c=d['kmeans'])
  axes[i // 3, i % 3].set_xlim(-100000000, 1600000000)
  axes[i // 3, i % 3].set_ylim(-2000000, 25000000)
plt.subplots_adjust(hspace=0.5, wspace=0.5)
plt.tight_layout()
plt.show()
```

Kết quả của thuật toán ta thu được biểu đồ gom cụm tốc độ tăng trưởng dân số của từng năm:





Nhận xét:

Nhìn vào biểu đồ phân cụm sau ta thấy ba cụm rõ ràng.

Sự gia tăng dân số không đồng đều giữa các quốc gia. Một số quốc gia có dân số rất đông, chẳng hạn như Trung Quốc và Ấn Độ, trong khi một số quốc gia khác có dân số rất ít, chẳng hạn như Liechtenstein và Tuvalu.

Tốc độ tăng trưởng dân số cũng không đồng đều giữa các quốc gia. Một số quốc gia có có dân số lớn và tốc độ tăng trưởng dân số rất cao, trong khi một số quốc gia khác có dân số nhỏ và tốc độ tăng trưởng dân số rất thấp.

Cụm 1 là các quốc gia có dân số nhỏ và tăng trưởng chậm. Các quốc gia này thường là các quốc gia phát triển với nền kinh tế đã phát triển.

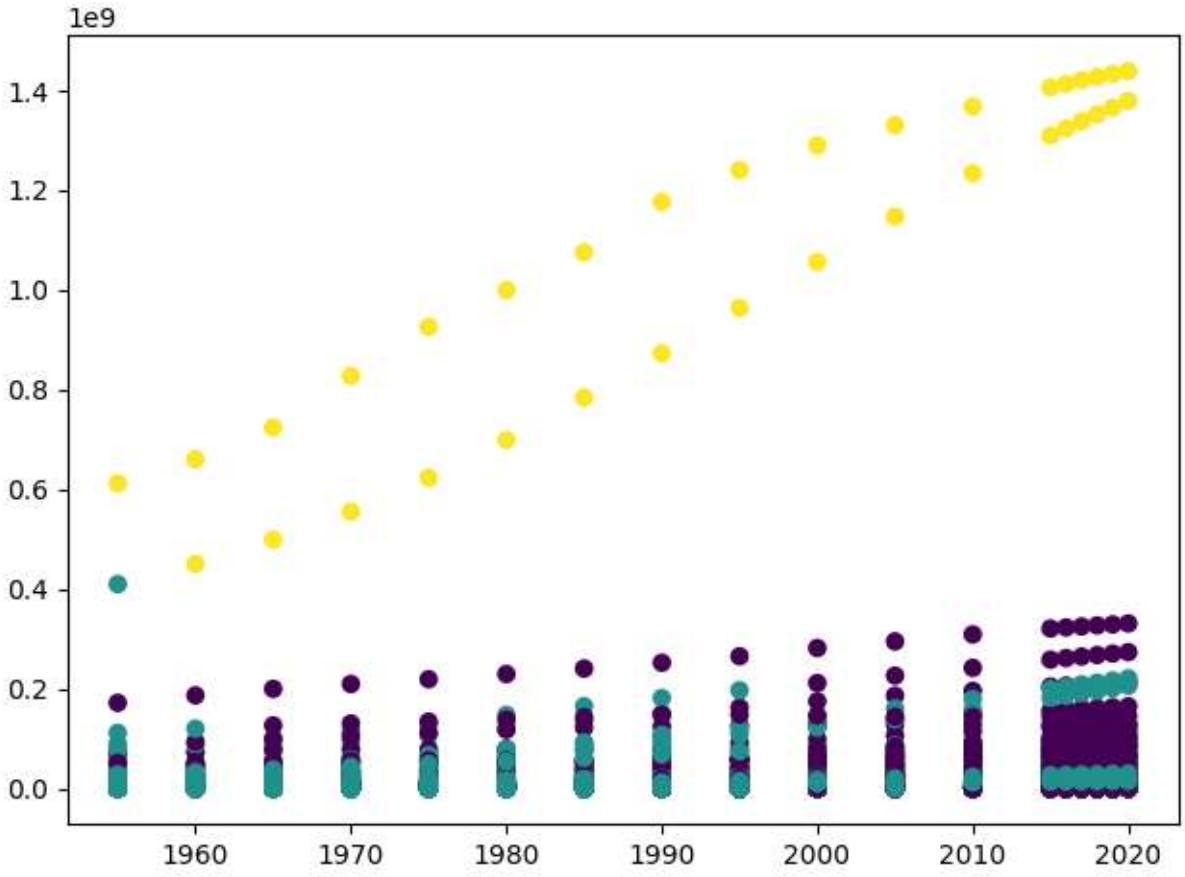
Cụm 2 là các quốc gia có dân số lớn và tăng trưởng nhanh. Các quốc gia này thường là các quốc gia đang phát triển với nền kinh tế đang phát triển nhanh chóng.

Cụm 3 là các quốc gia có dân số trung bình và tăng trưởng trung bình. Các quốc gia này thường là các quốc gia đang phát triển với nền kinh tế đang phát triển vừa phải.

3.1.2. Biểu đồ phân cụm toàn cục:

Sử dụng matplotlib.pyplot để tạo biểu đồ phân tán của dữ liệu sau khi thực hiện phân cụm KMeans:

```
plt.scatter(x=wp_df["Year"], y=wp_df['Population'], c=wp_df['kmeans'])
plt.tight_layout()
plt.show()
```



Một số bảng dữ liệu mẫu sau khi phân cụm:

	country	Year	Population	Yearly % Change	Yearly Change	Migrants (net)	Median Age	Fertility Rate	Density (P/Km²)	Urban Pop %	Urban Population	Share of W	World Population	Rank	kmeans
18	Albania	2020	2877797	-0,11	-3130	-14000	38,4	1,62	105	63,5	1827363	0,04	7794798739	140	1
36	Algeria	2020	43851044	1,85	797990	-10000	28,5	3,05	18	72,9	31950910	0,56	7794798739	33	1
123	Antigua And Barbuda	2020	97928	0,84	811	0	34	2	223	26,2	25682	0	7794798739	201	1
159	Armenia	2020	2963243	0,19	5551	-4958	35,4	1,76	104	62,8	1860554	0,04	7794798739	137	1
3927	Ukraine	2020	43733762	-0,58	-259878	10000	41,2	1,44	75	69,4	30334832	0,56	7794798739	35	1
3965	United Kingdom	2020	67886011	0,53	355839	260650	40,5	1,75	281	83,2	56495180	0,87	7794798739	21	1
3981	United States	2020	331002651	0,59	1937734	954806	38,3	1,78	36	82,8	273675139	4,25	7794798739	3	1
4017	Uruguay	2020	3473730	0,35	11986	-3000	35,8	1,88	20	95,1	3337571	0,04	7794798739	134	1
4035	Uzbekistan	2020	33469203	1,48	487487	-8861	27,8	2,43	79	50,1	16756329	0,43	7794798739	42	1
4071	Venezuela	2020	28435940	-0,28	-79889	-653249	29,6	2,28	32	45,18	11367346,8	0,36	7794798739	50	1
4089	Vietnam	2020	97338579	0,91	876473	-80000	32,5	2,06	314	37,7	36727248	1,25	7794798739	15	1
4124	Western Sahara	2020	597339	2,55	14874	5582	28,4	2,42	2	86,8	518863	0,01	7794798739	170	1
0	Afghanistan	2020	38928346	2,33	886592	-62920	18,9	4,56	60	25,4	9904337	0,5	7794798739	37	2
88	Angola	2020	32866272	3,27	1040877	6411	16,7	5,55	26	66,7	21936953	0,42	7794798739	44	2
3400	South Sudan	2020	11193725	1,19	131932	-174200	19	4,74	18	24,6	2749081	0,14	7794798739	84	2
3678	Tajikistan	2020	9537645	2,32	216627	-20000	22,4	3,61	68	27,3	2606273	0,12	7794798739	95	2
3696	Tanzania	2020	59734218	2,98	1728755	-40076	18	4,82	67	37	22113353	0,77	7794798739	24	2
3732	Timor Leste	2020	1318445	1,96	25328	-5385	20,8	4,1	89	32,8	432649	0,02	7794798739	156	2
3909	Uganda	2020	45741007	3,32	1471453	168694	16,7	5,01	229	25,7	11775012	0,59	7794798739	31	2
4142	Yemen	2020	29825964	2,28	664042	-30000	20,2	3,84	56	38,4	11465414	0,38	7794798739	48	2
4160	Zambia	2020	18383955	2,93	522923	-8000	17,6	4,66	25	45,3	8336381	0,24	7794798739	65	2
4178	Zimbabwe	2020	14862924	1,48	217456	-116856	18,7	3,63	38	38,4	5700468	0,19	7794798739	74	2
767	China	2020	1439323776	0,39	5540090	-348399	38,4	1,69	153	60,8	875075419	18,47	7794798739	1	3
1696	India	2020	1380004385	0,99	13586631	-532687	28,4	2,24	464	35	483008840	17,7	7794798739	2	3

Bảng dữ liệu năm 2010 sau khi phân cụm

	country	Year	Population	Yearly % Change	Yearly Change	Migrants (net)	Median Age	Fertility Rate	Density (P/Km²)	Urban Pop %	Urban Population	% Share of Wo	World Population	Rank	kmeans
28	Albania	2010	2948023	-0,92	-27757	-43672	32,2	1,64	108	52	1533860	0,04	6956823603	138	1
1666	Hungary	2010	9937370	-0,32	-31713	5030	40,1	1,33	110	68,9	6841343	0,14	6956823603	85	1
1684	Iceland	2010	320328	1,66	5070	1401	34,8	2,13	3	93,6	299745	0	6956823603	180	1
1720	Indonesia	2010	241834215	1,34	3108949	-266727	27,2	2,5	133	50,1	121052980	3,48	6956823603	4	1
2096	Liechtenstein	2010	35904	0,73	259	-502,6	30,24	1,478	225	14,5	5208	0	6956823603	214	1
2114	Lithuania	2010	3523816	-1,35	-44090	-40186	40,6	1,42	50	66,8	2085346	0,04	6956823603	135	1
2132	Luxembourg	2010	507880	2,1	10009	6456	38,9	1,62	196	88,5	449723	0,01	6956823603	170	1
4077	Venezuela	2010	28439940	1,47	401499	-40048	25,8	2,55	32	89,9	25568720	0,41	6956823603	43	1
4095	Vietnam	2010	87967851	0,97	826008	-159994	28,5	1,93	284	30,6	26910886	1,26	6956823603	13	1
4112	Wallis And Futuna Isls	2010	12669	-1,21	-450	-502,6	30,24	2,478	91	0	0	0	6956823603	225	1
4130	Western Sahara	2010	480274	1,88	8552	1058	25,5	2,55	2	86,3	414125	0,01	6956823603	172	1
6	Afghanistan	2010	29185507	2,61	706246	-109272	15,9	6,48	45	23,4	6836980	0,42	6956823603	40	2
94	Angola	2010	23356246	3,75	784529	71460	16,4	6,25	10	59,8	13970854	0,34	6956823603	50	2
381	Benin	2010	9199150	2,88	243407	-9755	17,9	5,49	82	43,1	3964272	0,13	6956823603	80	2
559	Burkina Faso	2010	15605217	3,06	416657	-25000	16,7	6,08	57	24,6	3844025	0,22	6956823603	64	2
577	Burundi	2010	8875602	3,33	282148	6001	17	6,39	338	10,8	932971	0,12	6956823603	93	2
619	Cambodia	2010	14312213	1,52	207772	-59197	22,7	3,08	81	20,3	2903806	0,21	6956823603	69	2
631	Cameroon	2010	20341241	2,78	521566	-10800	17,9	5,25	43	50,8	10296808	0,29	6956823603	58	2
701	Central African Republ	2010	4386768	1,67	69677	-25123	17,7	5,3	7	38,5	1730652	0,06	6956823603	121	2
809	Comoros	2010	689602	2,43	15611	-2000	19,1	4,9	371	28	192925	0,01	6956823603	184	2
4166	Zambia	2010	13665984	2,79	349947	-35000	16,2	5,6	18	40,1	5450867	0,2	6956823603	70	2
4184	Zimbabwe	2010	12887723	1,01	124205	-102535	18,6	3,89	33	36,8	4676106	0,18	6956823603	71	2
773	China	2010	1368810615	0,57	7606847	-435677	35	1,62	146	48,9	669353357	19,68	6956823603	1	3
1702	India	2010	1234281170	1,47	17334249	-531169	25,1	2,8	415	30,8	380744554	17,74	6956823603	2	3

Bảng dữ liệu năm 2020 sau khi phân cụm

Nhận xét:

Các quốc gia ở Châu Âu và Bắc Mỹ chủ yếu nằm trong Cụm 1, các quốc gia ở Châu Á, Châu Phi và Nam Mỹ chủ yếu nằm trong Cụm 2.

Các quốc gia ở cụm 1 có dân số tăng trưởng chậm chủ yếu là các quốc gia phát triển, có dân số già, tuổi trung bình cao và tỷ lệ sinh thấp. Một số quốc gia trong cụm này bao gồm Nhật Bản, Đức và Ý.

Các quốc gia ở cụm 2 có dân số tăng trưởng nhanh chủ yếu là các quốc gia đang phát triển, có dân số trẻ, tuổi trung bình thấp và tỷ lệ sinh cao. Một số quốc gia trong cụm này bao gồm Trung Quốc, Ấn Độ.

Các quốc gia ở cụm 3 có dân số tăng trưởng ổn định có dân số tăng trưởng với tốc độ vừa phải, với tỷ lệ sinh và tử vong tương đương, độ tuổi trung bình dân số ở mức trung bình. Một số quốc gia trong cụm này bao gồm Hoa Kỳ, Brazil và Nga.

Kết quả phân cụm cho thấy dân số thế giới đang trải qua những thay đổi lớn. Các quốc gia ở Châu Âu và Bắc Mỹ đang già đi và dân số giảm, trong khi các quốc gia ở Châu Á, Châu Phi và Nam Mỹ đang trẻ ra và dân số tăng nhanh. Những thay đổi này sẽ có tác động sâu sắc đến nền kinh tế, chính trị và xã hội toàn cầu trong những thập kỷ tới.

3.2. Luật kết hợp

3.2.1. Tuổi trung vị - mật độ dân số - tỷ suất sinh:

Sử dụng thuật toán fp-growth để tìm những mối liên hệ của dữ liệu

Phân tích dữ liệu dựa trên tuổi trung vị - mật độ dân số - tỷ suất sinh:

Xử lý dữ liệu:

- Đối với tuổi trung vị của các quốc gia chúng tôi chia thành 3 khoảng dữ liệu bao gồm: dưới 35 tuổi là Young, từ độ 35 đến 50 là Middle, từ 50 trở lên là Old
- Mật độ dân số của các quốc gia được chia như sau: dưới 100 là Small, từ 100 đến 500 là Middle và trên 500 là Large.
- Tỷ suất sinh của các quốc gia được chia như sau: dưới 3.5 là Low, từ trên 3.5 đến 5 là Mid và trên 5 là High.

Sau khi thực hiện các xử lý chia mức độ dữ liệu, chúng tôi tiến hành xử dụng thuật toán fp-growth từ thư viện mlxtend.

```
from mlxtend.frequent_patterns import fpgrowth
# Luật kết hợp giữa tuổi trung vị - dân số - tỷ lệ sinh
rule_df = wp_df[['Median Age','Fertility Rate','Density (P/Km²)']]
rule_df['Median Age'] = rule_df['Median Age'].apply(categorize_age)
rule_df['Density (P/Km²)'] = rule_df['Density (P/Km²)'].apply(categorize_density)
rule_df['Fertility Rate'] = rule_df['Fertility Rate'].apply(categorize_fertility)
df_encoded = pd.get_dummies(rule_df)
# frequent_itemsets = apriori(df_encoded, min_support=0.01, use_colnames=True)
frequent_itemsets_fpgrowth = fpgrowth(df_encoded, min_support=0.2, use_colnames=True)
# Tạo các luật kết hợp từ các tập phổ biến
rules = association_rules(frequent_itemsets_fpgrowth, metric="lift", min_threshold=1)
```

Kết quả thu được:

ANTECEDENTS	CONSEQUENTS	SUPPORT	CONFIDENCE
Median Age_Young	Density P/Km²_Small	0.482364156	0.687967369
Density P/Km²_Small	Median Age_Young	0.482364156	0.794660385
Median Age_Young	Fertility Rate_High	0.320781697	0.457511897
Fertility Rate_High	Median Age_Young	0.320781697	1
Density P/Km²_Small	Fertility Rate_High	0.251429933	0.414212799
Fertility Rate_High	Density P/Km²_Small	0.251429933	0.783803863
Median Age_Young, Density P/Km²_Small	Fertility Rate_High	0.251429933	0.521245059
Median Age_Young, Fertility Rate_High	Density P/Km²_Small	0.251429933	0.783803863
Density P/Km²_Small, Fertility Rate_High	Median Age_Young	0.251429933	1
Median Age_Young	Density P/Km²_Small, Fertility Rate_High	0.251429933	0.358599592
Density P/Km²_Small	Median Age_Young, Fertility Rate_High	0.251429933	0.414212799
Fertility Rate_High	Median Age_Young, Density P/Km²_Small	0.251429933	0.783803863
Density P/Km²_Middle	Fertility Rate_Low	0.207340324	0.655614167
Fertility Rate_Low	Density P/Km²_Middle	0.207340324	0.397987191
Fertility Rate_Low	Median Age_Middle	0.298856053	0.573650503
Median Age_Middle	Fertility Rate_Low	0.298856053	1

Dựa vào bảng kết quả của luật kết hợp được tạo ra từ thuật toán Fp-growth trên chúng ta có những đánh giá sau:

- Với những nước có tuổi trung vị trẻ (Median_Young) thì mật độ dân số thấp (Density P/Km²_Small), tỷ lệ sinh ở mức cao (Fertility Rate_High).

- Tỷ lệ xuất ở mức thấp (Fertility Rate_Low) diễn ra ở các nước có mật độ dân số trung bình (Density P/Km²_Middle) và tuổi trung vị ở mức trung niên (Median Age_Middle).

Nhận xét: các nước có dân cư thưa thớt thường là những nước có dân cư trong độ tuổi khá trẻ nhưng lại có tỷ lệ sinh khá cao. Mật độ dân số càng tăng thì mức sinh càng giảm trong khi tuổi của đa số dân cư tăng theo.

Nhận xét chủ quan:

- Tại những nơi dân cư thưa thớt đã có điều kiện sống khó khăn (do tuổi trung vị khá thấp), nhưng vẫn sinh nở nhiều (tỷ xuất sinh khá cao) có thể đây là những đất nước còn kém phát triển.
- Khi mật độ dân số tăng cao, cho thấy những nơi này có điều kiện sống lý tưởng hơn thì con người dễ tồn tại hơn (do tuổi trung vị tăng lên ở mức trung niên), lúc này nhận thức của con người thay đổi kèm theo điều kiện sống thích hợp có thể thấy đây có thể là những nước phát triển hoặc đang phát triển.

3.2.2. Độ tương quan giữa tỷ lệ dân số thành thị - tỷ suất sinh:

```
from mlxtend.frequent_patterns import fpgrowth
rule_df = wp_df[['Fertility Rate', 'Urban  Pop %']]
rule_df['Urban  Pop %'] = rule_df['Urban  Pop %'].apply(categorize_urban)
rule_df['Fertility Rate'] = rule_df['Fertility Rate'].apply(categorize_fertility)
df_encoded = pd.get_dummies(rule_df)
# frequent_itemsets = apriori(df_encoded, min_support=0.01, use_colnames=True)
frequent_itemsets_fpgrowth = fpgrowth(df_encoded, min_support=0.1, use_colnames=True)
# Tạo các luật kết hợp từ các tập phổ biến
rules = association_rules(frequent_itemsets_fpgrowth, metric="lift", min_threshold=1)
rules.to_csv('/content/drive/MyDrive/Colab Notebooks/csv/rule_2.csv', index=False)
```

Áp dụng thuật toán trên đối với dữ 2 dạng dữ liệu này.

Dữ liệu về Urban Pop % được chia như sau: dưới 30% là Low, từ 30% đến 55% là Mid, trên 55% là High.

Ta thu được kết quả:

ANTECEDENTS	CONSEQUENTS	SUPPORT	CONFIDENCE
Urban_Pop %_Low	Fertility Rate_High	0.158960915	0.671701913
Fertility Rate_High	Urban_Pop %_Low	0.158960915	0.495542348
Urban_Pop %_High	Fertility Rate_Low	0.355100095	0.78338591
Fertility Rate_Low	Urban_Pop %_High	0.355100095	0.681610247
Urban_Pop %_Mid	Fertility Rate_High	0.120352717	0.388162952
Fertility Rate_High	Urban_Pop %_Mid	0.120352717	0.375185736

Dựa vào kết quả ta có những nhận định sau:

- Những nước có tỷ suất sinh cao (Fertility Rate_High) thường tập trung ở các quốc gia có tỷ lệ dân thành thị vừa (Urban Pop %_Mid) và thấp (Urban Pop %_Low).
- Những nước có tỷ lệ dân số sống ở thành thị cao (Urban Pop %_High) thì có tỷ suất sinh ở mức thấp (Fertility Rate_Low)

Nhận xét: tỷ lệ sinh ở các nước có mức độ đô thị hóa cao giảm thấp. Tỷ lệ sinh ở các quốc gia có mức độ đô thị hóa trung bình và thấp nằm ở mức cao

Nhận xét chủ quan:

- Kết hợp với những dữ liệu ở phần phân tích tuổi trung vị - mật độ dân số - tỷ suất sinh và phần này ta có thể thấy tại những nơi có dân cư đông đúc và tỷ lệ dân số thành thị cao thì nhận thức và xu hướng của con người có sự thay đổi rõ rệt so với những nơi có dân cư thưa thớt.
- Những nơi có tiềm năng để đầu tư sẽ nằm ở những quốc gia có mức độ đô thị hóa ở mức trung bình (Urban Pop %_Mid) bởi vì: nơi đây vẫn còn chỉ số tỷ suất sinh cao trong khi mức độ đô thị hóa không thấp, dó đó mức sống những nơi này có thể khá ổn và nguồn lao động tri thức dồi dào, dễ phát triển kinh tế, kinh doanh.
- Những nơi có tiềm năng để sinh sống: là những nơi có mức độ đô thị hóa cao bởi vì những nơi này có độ tuổi trung vị trung niên do đó có thể tạm nhận xét nơi đây an toàn, điều kiện sống dễ dàng, và nền y tế phát triển. Nhưng mức sống những nơi đây có thể là cao nhất vì có tỷ suất sinh khá thấp.

3.2.3. Phân tích dữ liệu dân số di cư – tỷ lệ dân số thành thị:

Độ tương quan giữa dân số di cư – tỷ lệ dân số thành thị.

Dữ liệu về số dân di cư được chia như sau:

- Di cư dưới 1000 là Low_negative, di cư từ 1000 đến 50000 là Mid_negative, di cư trên 50000 là High_negative.
- Nhập cư dưới 1000 là Low_positive, nhập cư từ 1000 đến 50000 là Mid_positive, nhập cư trên 50000 là High_positive.

Ta thu được kết quả:

ANTECEDENTS	CONSEQUENTS	SUPPORT	CONFIDENCE
Urban_Pop %_Mid	Migrants (net)_Mid_negative	0.132983794	0.428900846
Migrants (net)_Mid_negative	Urban_Pop %_Mid	0.132983794	0.383768913
Migrants (net)_Mid_positive	Urban_Pop %_High	0.135367016	0.641083521
Urban_Pop %_High	Migrants (net)_Mid_positive	0.135367016	0.298633018

Nhận xét: Các nước có tỷ lệ di cư ở mức trung bình (Migrants (net)_Mid_negative) thường từ các nước có tỷ lệ dân thành thị mức trung bình. Các nước có tỷ lệ dân thành thị cao (Urban Pop %_High) thường thu hút các người nhập cư hơn (Migrants (net)_Mid_positive).

Nhận xét chủ quan: người di cư có xu hướng di cư từ nơi có mức độ đô thị hóa thấp sang những nơi có mức độ đô thị hóa cao hơn. Những nơi có số lượng di cư nhiều có thể do

điều kiện sống còn thấp, những nơi có tỷ lệ người nhập cư cao có điều kiện sống tốt hơn, đa dạng bản sắc văn hóa hơn nhưng vẫn có thể có nhiều nguy cơ khác như quá tải dân số, khan hiếm việc làm, tội phạm quốc tế…

3.3. Mức độ tương quan của bộ dữ liệu

Để đánh giá độ tương quan của dữ liệu, ta sử dụng các bộ thư viện sklearn

Trước đó dữ liệu đã được chuẩn hóa và làm sạch, ở đầu.

```
from sklearn.impute import KNNImputer
knn_imputer = KNNImputer(n_neighbors= 5)
temp_df = wp_df.drop(columns = ['country'])
wp_imputer = pd.DataFrame(knn_imputer.fit_transform(temp_df), columns = temp_df.columns)
wp_df[wp_imputer.columns] = wp_imputer
new_path = "/content/drive/MyDrive/Colab Notebooks/Bài Tập Lớn/population/clean_wp_df.csv"
wp_df.to_csv(new_path, index = False)
wp_df.info()
```

Kết quả thu được sau khi chuẩn hóa- làm sạch:

	country	Year	Population	Yearly % Change	Yearly Change	Migrants (net)	Median Age	Fertility Rate	Density (P/Km²)	Urban Pop %	Urban Population	Country's Share of World Pop	World Population	Rank
0	Afghanistan	2020.0	38928346.0	2.33	886592.0	-62920.0	18.4	4.58	60.0	25.4	9904337.0	0.50	7.794798e+09	37.0
1	Afghanistan	2019.0	38041754.0	2.34	869933.0	-62920.0	17.4	5.26	59.0	25.2	9582625.0	0.49	7.713468e+09	37.0
2	Afghanistan	2018.0	37171921.0	2.41	875806.0	-62920.0	17.4	5.26	57.0	24.9	9273302.0	0.48	7.631091e+09	36.0
3	Afghanistan	2017.0	36296113.0	2.56	913081.0	-62920.0	17.4	5.26	56.0	24.7	8971472.0	0.48	7.547858e+09	39.0
4	Afghanistan	2016.0	35383032.0	2.82	969429.0	-62920.0	17.4	5.26	54.0	24.5	8670939.0	0.47	7.464022e+09	39.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4191	Zimbabwe	1975.0	6230875.0	3.54	200914.0	-9109.0	15.4	7.40	16.0	19.3	1215331.0	0.15	4.079481e+09	79.0
4192	Zimbabwe	1970.0	6289300.0	3.42	163625.0	-8400.0	15.6	7.40	14.0	17.0	898584.0	0.14	3.700437e+09	79.0
4193	Zimbabwe	1965.0	4471177.0	3.43	138886.0	-3002.0	16.0	7.30	12.0	14.4	644767.0	0.13	3.339684e+09	91.0
4194	Zimbabwe	1960.0	3779681.0	3.28	112679.0	-1501.0	17.2	7.00	10.0	12.5	472478.0	0.12	3.034950e+09	87.0
4195	Zimbabwe	1955.0	3213268.0	3.19	93287.0	-901.0	18.1	6.80	8.0	11.5	371106.0	0.12	2.773020e+09	91.0

4196 rows × 14 columns

Vì mục đích muốn hiểu được những giá trị cốt lõi tương quan của dữ liệu, nên chúng ta cần phải loại bỏ bớt những cột dữ liệu mang tính chất là những cột kết quả của những cột dữ liệu khác.

Ở đây chúng ta sẽ bỏ 3 cột cuối là:

- “Country's Share of World Pop” : tỉ lệ dân số thế giới quốc gia (thể hiện số tiền mà một quốc gia nhất định đã đóng góp vào tổng dân số Trái đất)
- World Population: số đại diện cho dân số thế giới theo năm
- Rank: Số đại diện cho thứ hạng mà một quốc gia nhất định trong năm

```
fields = ["Country's Share of  World Pop", "World Population", "Rank"]
wd_df_new = wp_df.drop(fields, axis=1)
wd_df_new.head()
```

Kết quả sau khi lược bỏ:

	country	Year	Population	Yearly % Change	Yearly Change	Migrants (net)	Median Age	Fertility Rate	Density (P/Km²)	Urban Pop %	Urban Population
0	Afghanistan	2020.0	38928346.0	2.33	886592.0	-62920.0	18.4	4.66	60.0	25.4	9904337.0
1	Afghanistan	2019.0	38041754.0	2.34	869633.0	-62920.0	17.4	5.26	58.0	25.2	9582625.0
2	Afghanistan	2018.0	37171921.0	2.41	875808.0	-62920.0	17.4	5.26	57.0	24.9	9273302.0
3	Afghanistan	2017.0	36296113.0	2.50	913001.0	-62920.0	17.4	5.26	56.0	24.7	8971472.0
4	Afghanistan	2016.0	35383032.0	2.62	909429.0	-62920.0	17.4	5.26	54.0	24.5	8670939.0

Dùng hàm corr() trong thư viện pandas để tính toán hệ số tương quan giữa các cột trong một DataFrame hoặc Series. Hệ số tương quan thường được sử dụng để đo lường mức độ tương quan giữa hai biến số.

```
#Tương quan giữa các thành phần
wd_df = wd_df_new.corr() #Get correlation data
wd_df
```

Sau thuật toán trên ta có được ma trận tương quan như sau:

Year	1.000000	0.072056	-0.237644	0.031480	-0.000266	0.453712	-0.574304	0.052965	0.318824	0.114177
Population	0.072056	1.000000	-0.034499	0.870684	-0.142698	0.058530	-0.082885	-0.020228	-0.028356	0.920387
Yearly % Change	-0.237644	-0.034499	1.000000	0.057351	0.099192	-0.525030	0.539049	-0.038428	-0.149209	-0.072118
Yearly Change	0.031480	0.870684	0.057351	1.000000	-0.119300	-0.073455	0.029668	-0.023764	-0.102385	0.660356
Migrants (net)	-0.000266	-0.142698	0.099192	-0.119300	1.000000	0.170307	-0.082727	0.052866	0.159794	-0.024300
Median Age	0.453712	0.058530	-0.525030	-0.073455	0.170307	1.000000	-0.854967	0.105591	0.533182	0.142194
Fertility Rate	-0.574304	-0.082885	0.539049	0.029668	-0.082727	-0.854967	1.000000	-0.108121	-0.563485	-0.152210
Density (P/Km²)	0.052965	-0.020228	-0.038428	-0.023764	0.052866	0.105591	-0.108121	1.000000	0.123213	-0.024175
Urban Pop %	0.318824	-0.028356	-0.149209	-0.102385	0.159794	0.533182	-0.563485	0.123213	1.000000	0.070998
Urban Population	0.114177	0.920387	-0.072118	0.660356	-0.024300	0.142194	-0.152210	-0.024175	0.070998	1.000000

Tạo một ma trận cùng cấp với ma trận tương quan để hỗ trợ cho việc trực quan:

```
#cắt tam giác ma trận
ones_corr = np.ones_like(wd_df, dtype=bool)
ones_corr
```

```
array([[ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True]])
```

```
ones_corr.shape, wd_df.shape
```

```
((10, 10), (10, 10))
```

Tạo một ma trận mặt nạ (mask) chỉ giữ lại các giá trị tương quan nằm trên hoặc trên đường chéo chính của ones_corr, trong khi các giá trị nằm dưới đường chéo chính đã được thay thế bằng 0.

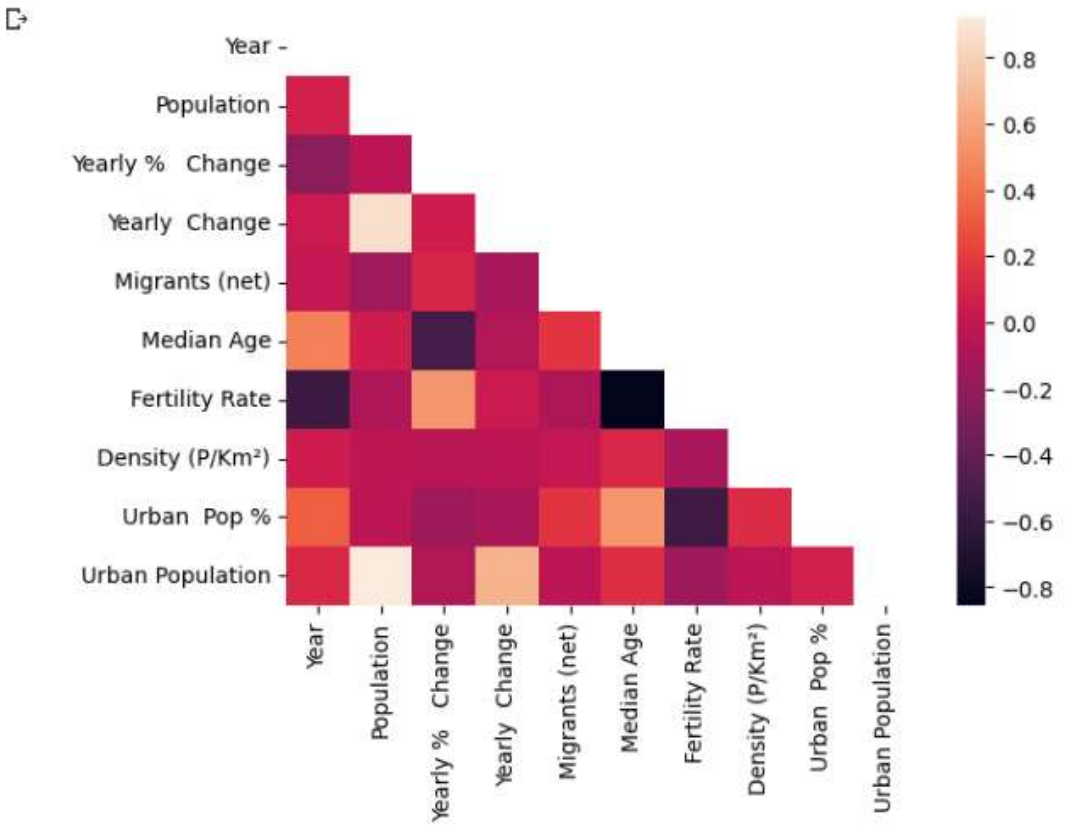
Mục đích của việc tạo ma trận mặt nạ như vậy có thể liên quan đến việc tránh tính toán lặp lại cho các phần tử không cần thiết của ma trận hoặc để tạo ra một biểu đồ hiển thị các tương quan quan trọng hơn.

```
# np's triu: return only upper triangle matrix (trả về một nữa ma trận là true, còn lại là false)
mask = np.triu(ones_corr)
mask
```

```
array([[ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [False,  True,  True,  True,  True,  True,  True,  True,  True,
         True],
       [False, False,  True,  True,  True,  True,  True,  True,  True,
         True],
       [False, False, False,  True,  True,  True,  True,  True,  True,
         True],
       [False, False, False, False,  True,  True,  True,  True,  True,
         True],
       [False, False, False, False, False,  True,  True,  True,  True,
         True],
       [False, False, False, False, False, False,  True,  True,  True,
         True],
       [False, False, False, False, False, False, False,  True,  True,
         True],
       [False, False, False, False, False, False, False, False,  True,
         True],
       [False, False, False, False, False, False, False, False, False,
         True]])
```

Trong thư viện Seaborn dùng thuật toán để vẽ một biểu đồ heatmap (biểu đồ nhiệt) từ một DataFrame trước đó ‘wb_df’ (tên biến đã lưu kết quả) và một ma trận mặt nạ (mask)

```
sns.heatmap(data=wd_df, mask=mask);
```



Vì cột đầu và cột cuối mức tương quan sẽ là bằng 1 nên chúng ta có thể bỏ 2 miền giá trị này.

```
#bỏ cột cuối và hàng đầu

adjusted_mask = mask[1:, :-1] #(xuất phát từ cột đầu tiên, kế thúc ở cột cuối cùng)
adjusted_mask
```

```
array([[False,  True,  True,  True,  True,  True,  True,  True,  True],
       [False, False,  True,  True,  True,  True,  True,  True,  True],
       [False, False, False,  True,  True,  True,  True,  True,  True],
       [False, False, False, False,  True,  True,  True,  True,  True],
       [False, False, False, False, False,  True,  True,  True,  True],
       [False, False, False, False, False, False,  True,  True,  True],
       [False, False, False, False, False, False, False,  True,  True],
       [False, False, False, False, False, False, False, False,  True],
       [False, False, False, False, False, False, False, False, False]])
```

Bước cuối cùng là hiện thực hóa và tạo thêm màu sắc cho tiện quan sát.

```
adjusted_wd_df = wd_df.iloc[1:, :-1]
fig, ax = plt.subplots(figsize=(10,8))

cmap = sns.diverging_palette(0, 230, 90, 60, as_cmap=True)#thang màu

sns.heatmap(data=adjusted_wd_df, mask=adjusted_mask,
            annot=True , annot_kws={"fontsize":13}, fmt=".2f", cmap=cmap,
            vmin=-1, vmax=1,
            linecolor='white', linewidths=0.5);

yticks = [i.upper() for i in adjusted_wd_df.index] #In Hoa
xticks = [i.upper() for i in adjusted_wd_df.columns] #In Hoa

ax.set_yticklabels(yticks, rotation=0, fontsize=13);
ax.set_xticklabels(xticks, rotation=90, fontsize=13);
title = 'MA TRẬN TƯƠNG QUAN\nTHÀNH PHẦN BỘ DỮ LIỆU ĐÃ LẤY MẪU\n'
ax.set_title(title, loc='left', fontsize=18);
```

Kết quả:

Tương quan và hệ số tương quan: Mức độ tương quan thể hiện mức độ liên quan giữa hai biến số. Hệ số tương quan nằm trong khoảng từ -1 đến 1, với giá trị gần 1 thể hiện mối tương quan thuận mạnh, giá trị gần -1 thể hiện mối tương quan nghịch mạnh, và giá trị gần 0 thể hiện không có tương quan tuyến tính.

Kết quả tương quan: Kết quả mức độ tương quan của từng cặp biến số có thể cung cấp thông tin quan trọng về mối quan hệ giữa chúng. Có thể thấy rằng có những cặp biến có mức độ tương quan mạnh (ví dụ: "Population" và "Urban Population") và cặp biến có mức độ tương quan yếu hơn (ví dụ: "Yearly % Change" và "Yearly Change").

Nhận thức về mối quan hệ: Kết quả này có thể giúp ta hiểu rõ hơn về mối quan hệ giữa các biến số trong bộ dữ liệu. Ví dụ, nhận thấy rằng "Median Age" và "Fertility Rate" có mức độ tương quan mạnh với giá trị -0.854967 có thể cho thấy sự tương quan nghịch giữa độ tuổi trung bình và tỷ lệ sinh.

Mặc dù có thể thấy mối tương quan giữa các biến số, nhưng kết quả này chưa chứng tỏ một biến số là nguyên nhân của biến số khác. Sự tương quan chỉ thể hiện mối liên quan thống kê giữa chúng.

Tóm lại, việc tìm hiểu mức độ tương quan giữa các yếu tố trong bộ dữ liệu có thể cung cấp cái nhìn sâu hơn về sự tương quan giữa chúng, giúp bạn hiểu rõ hơn về mô hình dữ liệu và có thể hướng dẫn các quyết định và phân tích dựa trên thông tin này.

3.4. Xây dựng model dự đoán số dân trong tương lai:

Thư viện sử dụng:

```
] import pandas as pd
  from sklearn.model_selection import train_test_split
```

Import dữ liệu và chọn 2 cột cho model:

```
# Đọc dữ liệu từ tệp CSV hoặc DataFrame
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks

# Chọn cột 'Year' và 'Population' cho mô hình dự đoán
features = data[['Year']]
target = data['World Population']

# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(
    features, target, test_size=0.2, random_state=42)
```

Ở đây sử chúng ta sẽ chia ra làm hai bộ “train” và “test”

80% dữ liệu sẽ được đưa vào model để huấn luyện, 20% còn lại được dùng để kiểm tra lại, nhằm mang lại độ tin cậy cho thuật toán

trong python cho phép bạn tạo, đào tạo và sử dụng mô hình hồi quy tuyến tính trong Python bằng cách sử dụng lớp LinearRegression trong thư viện sklearn.

Hàm mean_squared_error. được sử dụng để tính toán giá trị trung bình của bình phương sai số (mean squared error, MSE) giữa các giá trị dự đoán và giá trị thực tế. MSE thường được sử dụng để đánh giá hiệu suất của mô hình dự đoán.

```
#Xây dựng mô hình tuyến tính
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Khởi tạo mô hình
model = LinearRegression()

# Huấn luyện mô hình trên tập huấn luyện
model.fit(X_train, y_train)

# Dự đoán dân số trên tập kiểm tra
y_pred = model.predict(X_test)

# Đánh giá hiệu suất mô hình
mse = mean_squared_error(y_test, y_pred)
print('Mean Squared Error:', mse)
```

```
Mean Squared Error: 4782841856210144.0
```

Ở đây chỉ có thể dự đoán một khung thời gian không quá xa so với dữ liệu đã được cung cấp cho model. Chọn năm 2030 dự đoán số dân trên thế giới sẽ là bao nhiêu dân.

```
#Dự đoán tương lai
future_year = [[2030]]
future_population = model.predict(future_year)
print('Dự đoán dân số vào năm 2030:', future_population)
```

```
Dự đoán dân số vào năm 2030: [8.55614159e+09]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not
  warnings.warn(
```

Kết quả số thế giới vào năm 2030 là khoản 8 tỷ 500 triệu người. Điều này cho thấy mức độ gia tăng dân số theo các năm là tỷ lệ thuận.

Tuy nhiên thuật toán chỉ mang tính chất tham khảo vì có thể sẽ có những nguyên nhân tác động đến như:

- Khả năng dự đoán hạn chế: Khi xây dựng mô hình dự đoán trong tương lai, bạn dựa vào dữ liệu có sẵn để tạo mô hình. Nếu các điều kiện hoặc biến thay đổi một cách đáng kể trong tương lai mà không được phản ánh trong dữ liệu hiện tại, thì mô hình có thể dự đoán sai.

- Nhiễu và biến động: Dữ liệu trong tương lai thường có mức độ nhiễu và biến động cao hơn so với dữ liệu lịch sử. Điều này có thể dẫn đến khó khăn trong việc xác định xu hướng thực sự và dự đoán chính xác.
- Sự không chắc chắn: Dự đoán trong tương lai thường đi kèm với mức độ không chắc chắn cao. Ngay cả khi bạn có mô hình tốt, không thể tránh khỏi yếu tố không chắc chắn như biến đổi kinh tế, sự kiện thế giới, thiên tai, chính trị, v.v. Điều này có thể ảnh hưởng đáng kể đến kết quả dự đoán.
- Khó khăn trong việc thu thập dữ liệu tương lai: Điều kiện để thu thập dữ liệu trong tương lai có thể thay đổi, làm cho việc thu thập dữ liệu tương lai trở nên khó khăn hơn. Nếu dữ liệu không có tính liên tục hoặc khả dụng đầy đủ, mô hình dự đoán có thể bị hạn chế.
- Tác động của thông tin mới: Trong tương lai, thông tin mới có thể xuất hiện và ảnh hưởng đến mô hình dự đoán. Mô hình dự đoán có thể không thể đáp ứng nhanh chóng và linh hoạt với thông tin mới, đặc biệt khi mô hình cần được cập nhật thường xuyên để phản ánh các thay đổi này.
- Overfitting và underfitting: Khi xây dựng mô hình dự đoán trong tương lai, có nguy cơ mô hình bị overfitting (mô hình quá phức tạp và hiệu suất trên dữ liệu mới kém) hoặc underfitting (mô hình quá đơn giản để thể hiện dữ liệu mới).

Tóm lại, việc xây dựng mô hình dự đoán trong tương lai có thể rất hữu ích, nhưng cần phải nhận thức về các nhược điểm và thách thức mà nó mang lại. Việc sử dụng mô hình cùng với sự hiểu biết về bản chất của dữ liệu và các yếu tố tác động trong tương lai là cách tốt để đối mặt với những thách thức này.