

หัวข้อ : World population Analysis

Data source : Kaggle :

Data Journey : [Load Data](#) นำเข้าข้อมูลที่ต้องการใช้

```
#ข้อมูล Total Population
Tpop = pd.read_csv('/content/sample_data/Population.total.csv')
#ข้อมูล Male Population
Mpop = pd.read_csv('/content/sample_data/Population.male.csv')
#ข้อมูล Female Population
Fpop = pd.read_csv('/content/sample_data/Population.female.csv')
#ข้อมูล Birth Rate
Brate = pd.read_csv('/content/sample_data/Birth.rate.csv')
#ข้อมูล Death Rate
Drate = pd.read_csv('/content/sample_data/Death.rate.csv')
#ข้อมูล Land
Land = pd.read_csv('/content/sample_data/Land.area.csv')
Life = pd.read_csv('/content/sample_data/Life.expectancy.csv')
Ccode = pd.read_csv('/content/sample_data/country.csv')
```

Data Journey : [Collect data](#) Step 1 : Data inspect หลังจากนั้นต้องการ unpivot data ด้วยคำสั่ง melt เพื่อทำ data frame ที่เราต้องการ และใช้คำสั่ง merge เพื่อ map region และ continent เข้าเพื่อที่จำเอาไปทำ visual กับหา insight เพิ่มเติม

1.1 ตั้งด้วย total population file

```
#Total Population นำปีมาต่อหลัง
year_list = list(Tpop.iloc[:, 4:].columns)
df = pd.melt(Tpop, ['Country Name', 'Country Code', 'Indicator Name'], value_vars=year_list, value_name='population')
df.rename({'variable': 'Year'}, axis=1, inplace=True)
```

1.2 นำข้อมูล region และ continent มา map เพื่อ group ข้อมูล

```
#เพิ่มคำอธิบาย region
df=pd.merge(df,Ccode,left_on='Country Name',right_on='Country Name',how='left')
df.rename({'Country Code_x': 'Country Code'}, axis=1, inplace=True)
df=df.drop(columns=['Country Code_y'])
```

1.3 นำข้อมูล male population มาต่อท้าย

```
#Male Population นำปีมาต่อหลัง
year_list = list(Mpop.iloc[:, 4:].columns)
Mpop1 = pd.melt(Mpop, ['Country Name', 'Country Code', 'Indicator Name'], value_vars=year_list, value_name='Male')
Mpop1.rename({'variable': 'Year'}, axis=1, inplace=True)
#ตัดมาเฉพาะ column Male มาต่อท้าย
Mpop2=Mpop1[['Male']]
df = pd.concat([df,Mpop2],ignore_index=False,axis=1)
```

1.4 นำข้อมูล female population มาต่อท้าย

```
#Female Population นำปีมาต่อหลัง
year_list = list(Fpop.iloc[:, 4:].columns)
Fpop1 = pd.melt(Fpop, ['Country Name', 'Country Code', 'Indicator
Name'], value_vars=year_list, value_name='Female')
Fpop1.rename({'variable': 'Year'}, axis=1, inplace=True)
#ตัดมาเฉพาะ column Female มาต่อท้าย
Fpop2=Fpop1[['Female']]
df = pd.concat([df,Fpop2],ignore_index=False,axis=1)
```

1.5 นำข้อมูล birth rate มาต่อท้าย

```
#Birth Rate นำปีมาต่อหลัง
year_list = list(Brate.iloc[:, 4:].columns)
Brate1 = pd.melt(Brate, ['Country Name', 'Country Code', 'Indicator
Name'], value_vars=year_list, value_name='Birth Rate')
Brate1.rename({'variable': 'Year'}, axis=1, inplace=True)
#ตัดมาเฉพาะ column Birth Rate มาต่อท้าย
Brate2=Brate1[['Birth Rate']]
df = pd.concat([df,Brate2],ignore_index=False,axis=1)
```

1.6 นำข้อมูล Death rate มาต่อท้าย

```
#Death Rate นำปีมาต่อหลัง
year_list = list(Drate.iloc[:, 4:].columns)
Drate1 = pd.melt(Drate, ['Country Name', 'Country Code', 'Indicator
Name'], value_vars=year_list, value_name='Death Rate')
Drate1.rename({'variable': 'Year'}, axis=1, inplace=True)
#ตัดมาเฉพาะ column Death Rate มาต่อท้าย
Drate2=Drate1[['Death Rate']]
df = pd.concat([df,Drate2],ignore_index=False,axis=1)
```

1.7 นำข้อมูล อายุมาต่อท้าย

```
# Life นำปีมาต่อหลัง
year_list = list(Life.iloc[:, 4:].columns)
Lif1 = pd.melt(Life, ['Country Name', 'Country Code', 'Indicator
Name'], value_vars=year_list, value_name='Life')
Lif1.rename({'variable': 'Year'}, axis=1, inplace=True)
#ตัดมาเฉพาะ column Land มาต่อท้าย
Life2=Lif1[['Life']]
df = pd.concat([df,Life2],ignore_index=False,axis=1)
```

1.8 reindexเพื่อreset ค่าของindex

```
from pandas.core.internals.array_manager import new_block
df[df.isna().any(axis=1)]
new_cols = ["Region", "Continent", "Country Code", "Country Name", "Indicator
Name", "Year", "population", "Male", "Female", "Birth Rate", "Death Rate", "Land", "Life"]
df=df.reindex(columns=new_cols)
```

1.9 ตรวจสอบdata และ แทนค่าและจัดการการ nan values

```
# Handling with null data
df['Death Rate'].fillna(df['Death Rate'].median(),inplace=True)
df['Birth Rate'].fillna(method='ffill',inplace=True)
df['Land'].fillna(method = 'ffill', inplace = True)
df['Land'].fillna(df.groupby('Country Code')['Land'].transform('median'), inplace = True)
df['Life'].fillna(df.groupby('Country Code')['Life'].transform('median'), inplace = True)
df.drop(['Indicator Name'],axis=1)
df=df[df["Year"].str.contains('2021')== False]
df['area_cap'] =df['population']/df['Land']
df.head(2)
```

1.10 drop คอลัมน์ที่ไม่ต้องการทิ้ง

```
# การdrop column ที่ไม่ต้องการทิ้ง
df=df.dropna()
```

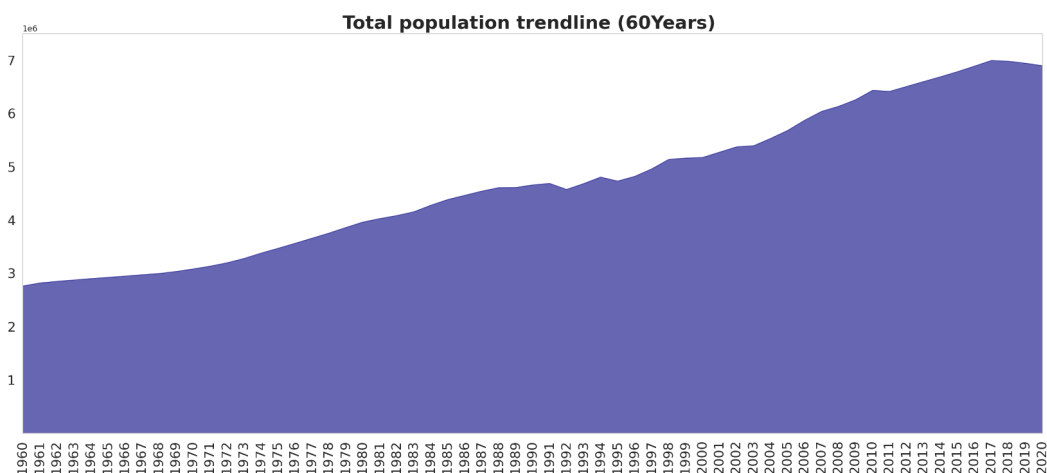
Data Journey : Step2 [EDA](#) โดยนำdata frameจากการทำcleaning มาทำanalysisแต่ละประเด็นที่สนใจ

2.1 population Trendline

```
### 1. population Trendline ####

totalp = df.pivot_table(index='Year',columns=None,values=['population'],aggfunc=['median'])
type(totalp)
c=palette=sns.color_palette("cubehelix")
plt.figure(figsize=(15, 5))
plt.xticks(fontsize = 10,rotation=90)
plt.yticks(fontsize = 10,rotation=90)

tyear = totalp.index # index x
tpop =totalp.iloc[:,0] #popluation y
plt.fill_between(tyear,tpop,color='slateblue')
plt.title("Total population trendline (60 Years) ",fontsize = 20,fontweight="bold")
plt.margins(0,0)
plt.ylim(1, 10000000)
sns.set_style("whitegrid", {'axes.grid' : False})
```



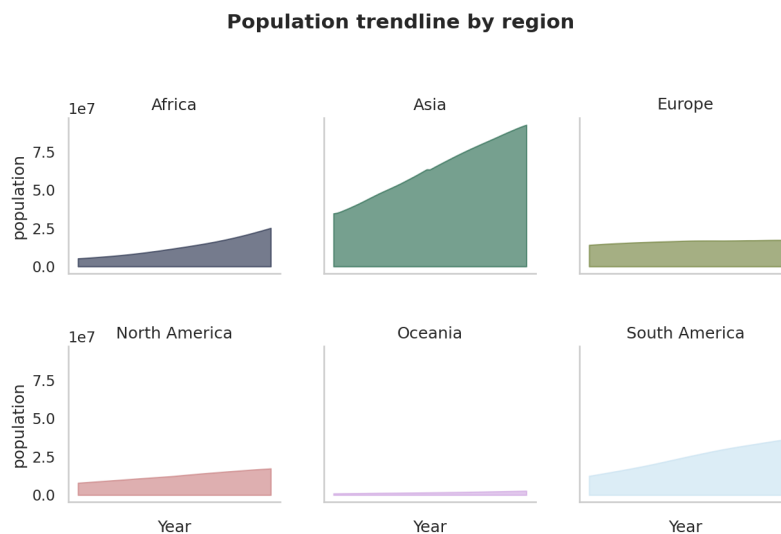
2.2 population trendline by region

```
## 2. population trendline by region #####
from seaborn._core.properties import Color
df_pivot8 =
df.pivot_table(index=['Year', 'Continent'], columns=None, values=['population'], aggfunc=['mean'])
type(df_pivot8) #pivot data

df2 = df_pivot8.reset_index()
df2.columns = ['Year', 'Continent', 'population'] # สร้าง series data

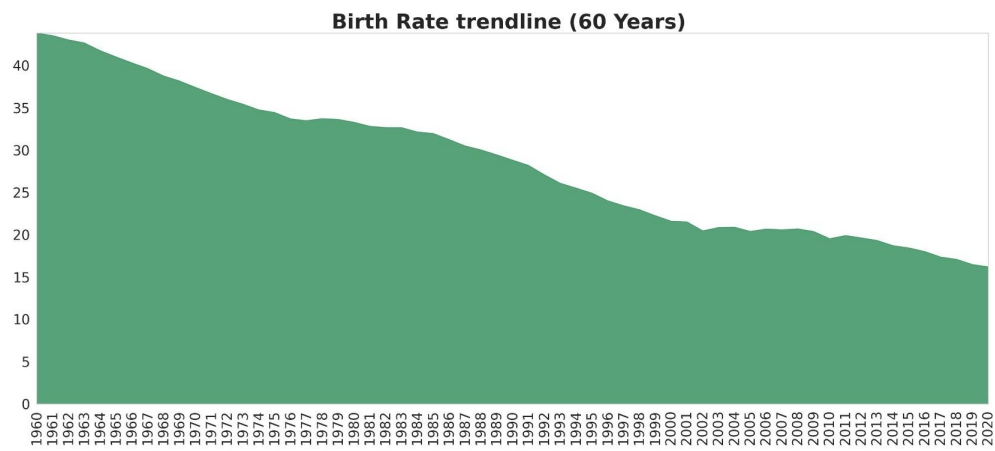
sns.set_style("whitegrid", {'axes.grid' : False})
g = sns.FacetGrid(df2,
col='Continent', hue='Continent', col_wrap=3, palette=sns.color_palette("cubehelix"))
g.set(xticklabels=[])
g.tick_params(bottom=False)

g = g.map(plt.fill_between, 'Year', 'population', alpha=0.6).set_titles("{col_name} Continent")
g = g.set_titles("{col_name}")
plt.subplots_adjust(top=0.80)
g = g.fig.suptitle('Population trendline by region', fontsize = 20, fontweight="bold")
```



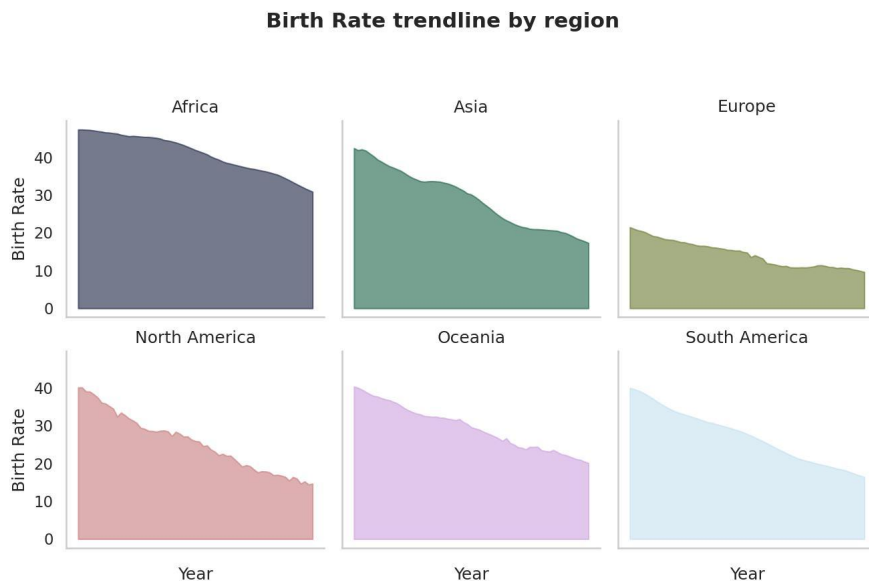
2.3 birth rate trendline

```
### 4 . Birth rate Trendline ##
btotal = df.pivot_table(index='Year', columns=None, values=['Birth Rate'], aggfunc=['median'])
type(btotal)
c=palette=sns.color_palette("cubehelix")
plt.figure(figsize=(15, 5))
plt.xticks(fontsize=10, rotation=90)
plt.yticks(fontsize=10)
tyear = btotal.index
btotal =btotal.iloc[:,0] #popluation
plt.fill_between(tyear,btotal,color='seagreen',alpha=0.8)
plt.title("Birth Rate trendline (60 Years) ", fontsize = 20, fontweight="bold")
plt.margins(0,0)
plt.savefig("Birth Rate.jpg", dpi=150)
```



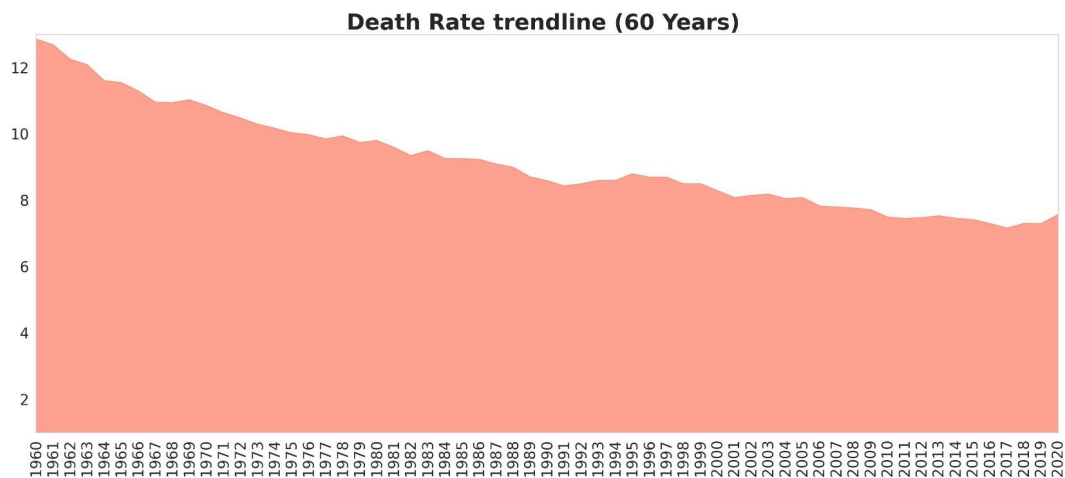
2.4 birth rate by region

```
from seaborn._core.properties import Color
df_pivot19 = df.pivot_table(index=['Year', 'Continent'], columns=None, values=['Birth
Rate'], aggfunc=['mean'])
type(df_pivot19) #pivot data
df2 = df_pivot19.reset_index() # make series
df2.columns = ['Year', 'Continent', 'Birth Rate']
sns.set_style("whitegrid", {'axes.grid' : False}) #styling them
g = sns.FacetGrid(df2,
col='Continent', hue='Continent', col_wrap=3, palette=sns.color_palette("cubehelix")) #seaborn
port
g.set(xticklabels=[])
g.tick_params(bottom=False)
g = g.map(plt.fill_between, 'Year', 'Birth Rate', alpha=0.6).set_titles("{col_name} Continent",)
g = g.set_titles("{col_name}")
plt.subplots_adjust(top=0.80)
g = g.fig.suptitle('Birth Rate trendline by region', fontsize=20, fontweight="bold")
plt.savefig("Birth Rate by region.jpg", dpi=150) #Save image
```



2.5 death rate trendline

```
dtotal = df.pivot_table(index='Year',columns=None,values=['Death Rate'],aggfunc=['median'])
type(dtotal) #pivot data
c=palette=sns.color_palette("cubehelix")
plt.figure(figsize=(15, 5)) # figure sizing
plt.xticks(fontsize=10,rotation=90) #axisx
plt.yticks(fontsize=10) ## axisy
tyear = dtotal.index
dtotal =dtotal.iloc[:,0] #popluation
plt.fill_between(tyear,dtotal,color='tomato',alpha=0.6)
plt.title("Death Rate trendline (60 Years) ",fontsize = 20,fontweight="bold")
plt.margins(0,0)
plt.ylim(1, 17)
plt.savefig("Death Rate.jpg",dpi=150)
```



2.6 death rate by region

```
from seaborn._core.properties import Color
df_pivot8 = df.pivot_table(index=['Year','Continent'],columns=None,values=['Death
Rate'],aggfunc=['mean'])
type(df_pivot8)

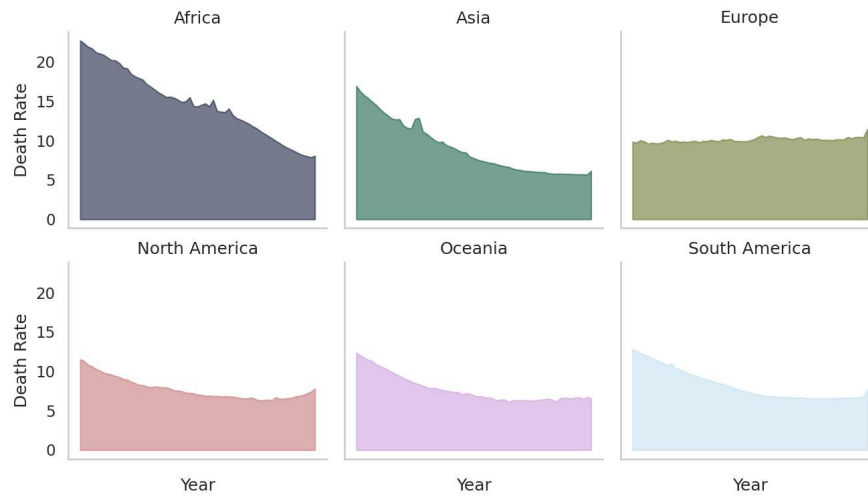
df2 = df_pivot8.reset_index()
df2.columns = ['Year','Continent','Death Rate']

g = sns.FacetGrid(df2,
col='Continent',hue='Continent',col_wrap=3,palette=sns.color_palette("cubehelix"))
g.set(xticklabels=[])
g.tick_params(bottom=False)

g = g.map(plt.fill_between, 'Year', 'Death Rate',alpha=0.6).set_titles("{col_name} Continent")
g = g.set_titles("{col_name}")
plt.subplots_adjust(top=0.80)
g = g.fig.suptitle('Death Rate trendline by region',fontsize=15,fontweight="bold")

sns.set_style("whitegrid", {'axes.grid' : False})
plt.savefig("Death Rate by region.jpg",dpi=150)
```

Death Rate trendline by region

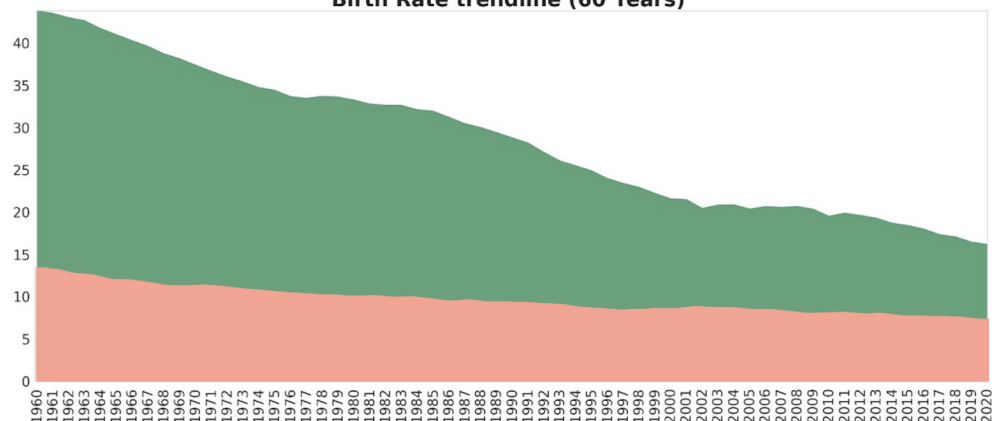


2.7 death rate VS birth rate

```
df_pivot13 = df.pivot_table(index='Year',columns=None,values=['Death Rate','Birth Rate'],aggfunc=['median'])
type(df_pivot13)
xyear_asia = df_pivot13.index
xyear_asia
b=df_pivot13.iloc[:,0] # Deathrate
d=df_pivot13.iloc[:,1] # Brithrate

plt.figure(figsize=(20 ,10))
plt.xticks(fontsize=10,rotation=90)
plt.yticks(fontsize=10)
plt.margins(0,0)
pal = ["tomato", "seagreen",]
nal = ["alpha=0.5","alpha=0.70"]
plt.stackplot(xyear_asia,b,d, labels=['Death Rate','Birth Rate '], colors=pal, alpha=0.6 )
plt.legend(loc='upper right')
plt.title("Death Rate VS Birth Rate trendline (60 Years) ",fontsize = 30,fontweight="bold")
plt.savefig("Death Rate VS Birth Rate.jpg",dpi=300)
plt.ylim(1, 100)
```

Birth Rate trendline (60 Years)

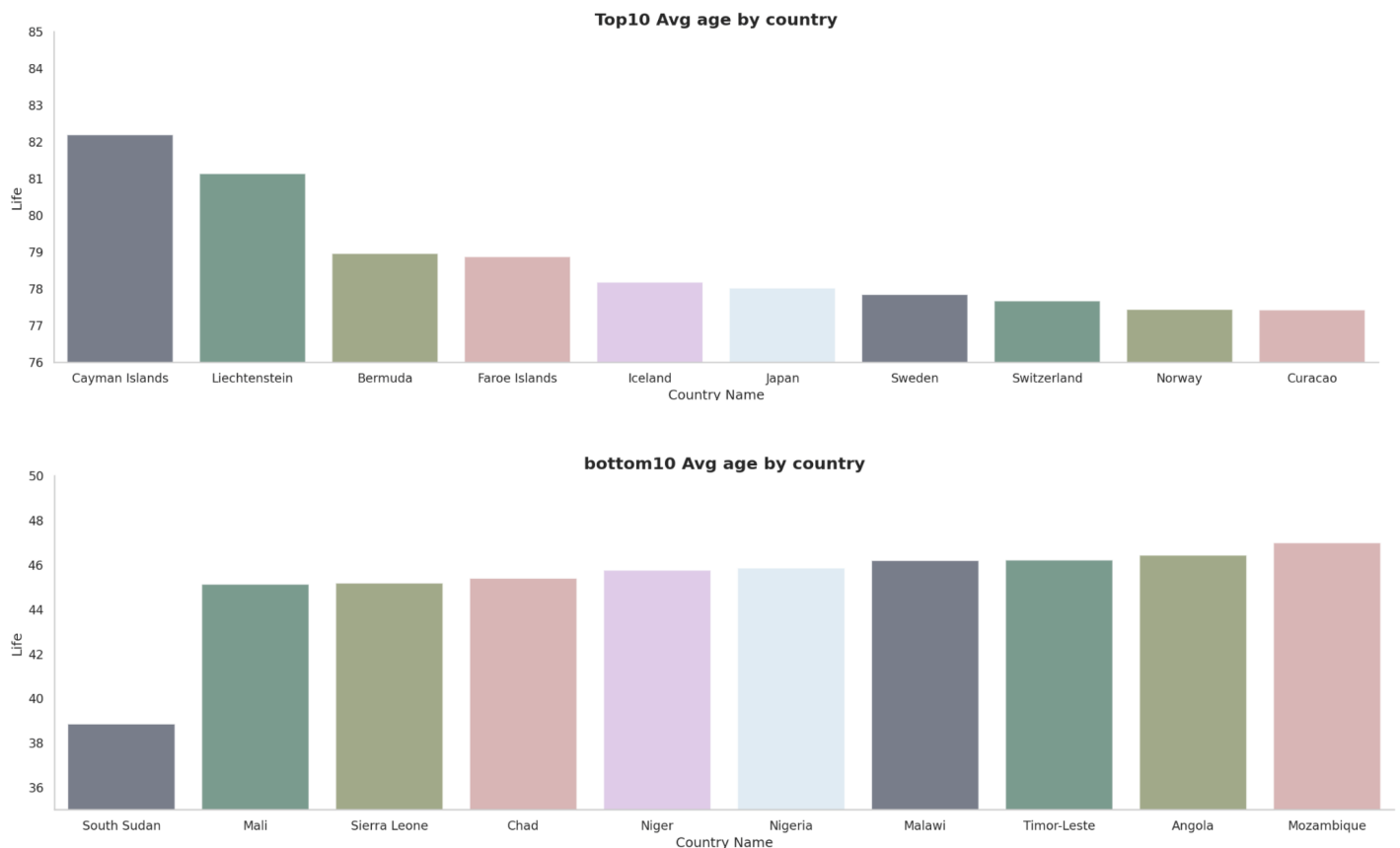


2.8 Top and bottom (rank10) by country

```
df2=df.pivot_table(index='Country Name',columns=None,values=['Birth
Rate','Life'],aggfunc='mean').sort_values('Life',ascending=False).reset_index()
df22 = df2.iloc[0:10]
sns.set(rc={'figure.dpi':150})
sns.set_style("whitegrid", {'axes.grid' : False})
sns.catplot( kind='bar',
             data=df22,
             x='Country Name',          # Axis to show the categories
             y='Life',                  # Quantitative values
             height=5, aspect=3.5,
             alpha=0.6,palette=sns.color_palette("cubehelix"))

plt.ylim(76, 85)
plt.title("Top10 Avg age by country",fontsize=15,fontweight="bold")
plt.savefig("avg age.jpg",dpi=300)
##bottom 10 ##
df3=df.pivot_table(index='Country Name',columns=None,values=['Birth
Rate','Life'],aggfunc='mean').sort_values('Life',ascending=True).reset_index()
df33 = df3.iloc[0:10]
sns.set(rc={'figure.dpi':150})
sns.set_style("whitegrid", {'axes.grid' : False})
sns.catplot( kind='bar',
             data=df33,
             x='Country Name',          # Axis to show the categories
             y='Life',                  # Quantitative values
             height=5, aspect=3.5,
             alpha=0.6,palette=sns.color_palette("cubehelix"))

plt.ylim(35,50 )
plt.title("bottom10 Avg age by country",fontsize=15,fontweight="bold")
plt.savefig("avg age low.jpg",dpi=300)
```



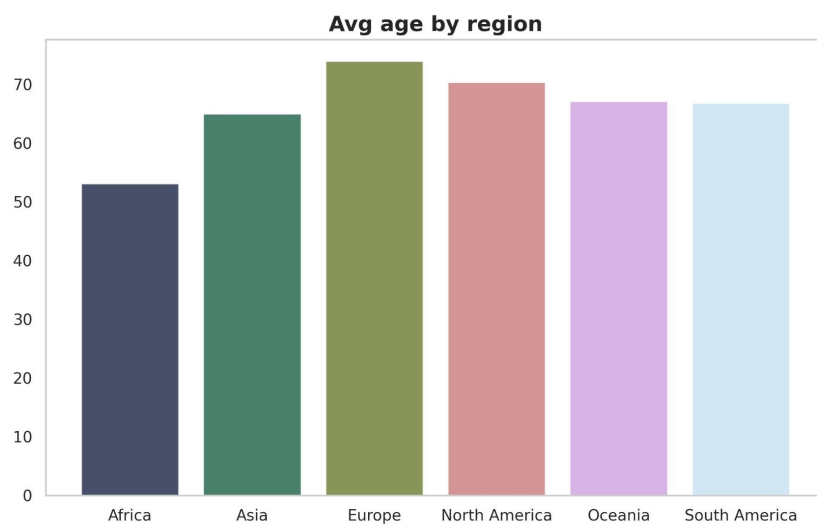
2.9 avg age by region

```
df_pivot =df.pivot_table(index='Continent',columns=None,values=['Life'],aggfunc=['mean'])

type(df_pivot)
Continent= df_pivot.index
c=palette=sns.color_palette("cubehelix")

total_pop=df_pivot.iloc[:,0] # population

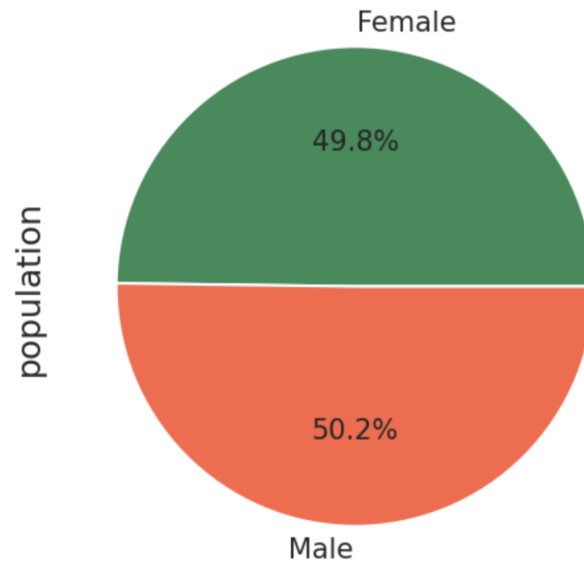
plt.figure(figsize=(10,6))
plt.bar(Continent,total_pop,color=c,alpha=0.8)
plt.title("Avg age by region",fontsize=15,fontweight="bold")
plt.savefig(" Bar Age Rate by region.jpg",dpi=300)
```



2.10 gender ratio

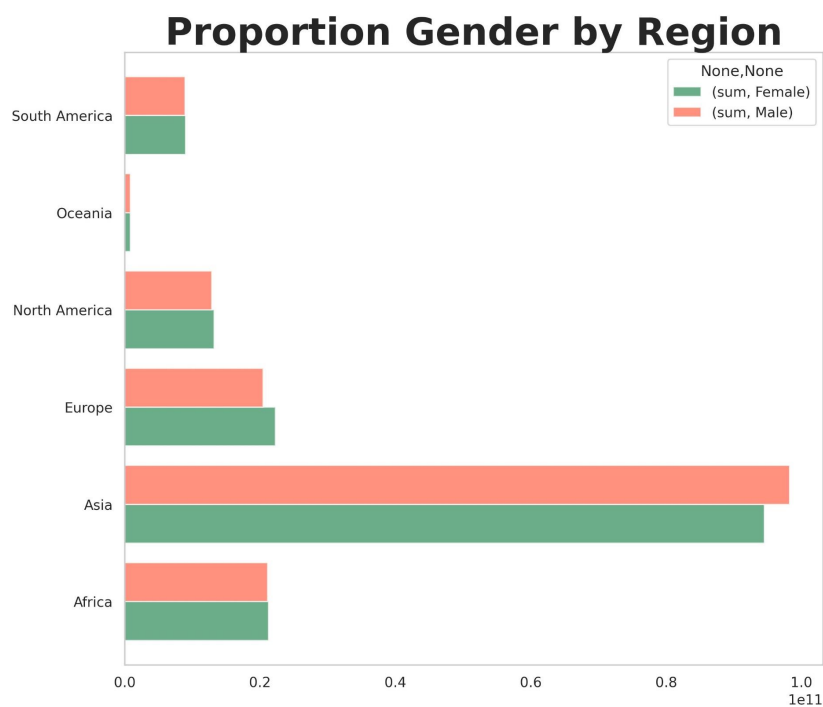
```
male =df['Male'].sum()
female = df['Female'].sum()
df4 = pd.DataFrame([['Male',male],['Female',female]],columns=('Gender', 'population'))
type(df4)
colors1 = ['tomato','seagreen']
plt.figure(figsize=(10, 10))
palette = {"Female":"tab:pink",
           "male":"tab:royalblue"}
data = df4.groupby("Gender")["population"].sum()
data.plot.pie(autopct="%.1f%%",colors=colors1,fontsize=20)
plt.title("Gender ratio",fontsize=30,fontweight="bold")
plt.figure(figsize=(25, 10))
plt.savefig("Pie Gender.jpg",dpi=300)
plt.show()
```

Gender ratio



2.11 proportion gender by region

```
sex = df.pivot_table(index='Continent',columns=None,values=['Male','Female'],aggfunc=['sum'])
sex
type(sex)
pal = ["seagreen", "tomato",]
c=palette=sns.color_palette("cubehelix")
sex.plot.barh(figsize=(10,9),color=pal,title = "Proportion Gender by Region",alpha=0.7,width =
0.8,)
plt.title("Proportion Gender by Region",fontsize=30,fontweight="bold")
#fontsize=15,fontweight="bold"
plt.savefig(" mf Bar agender Rate by region.jpg",dpi=300)
```



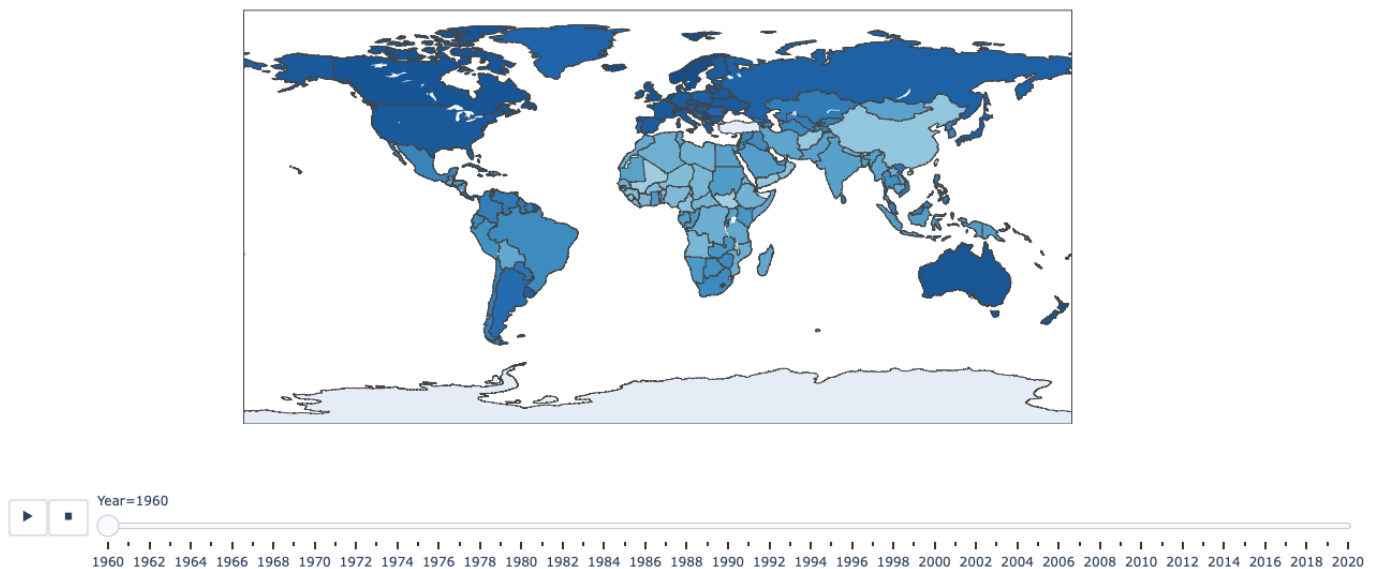
2.13 Animation map

```
import plotly.express as px
import plotly.graph_objects as go
import plotly.figure_factory as ff
import matplotlib.animation as animation
from matplotlib.animation import FuncAnimation
from plotly import tools
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot
from plotly.subplots import make_subplots

def animated_graph(df,column_name):
    fig=px.choropleth(df.sort_values('Year'),locations='Country Name',color='Life',
                      locationmode='country names',
                      animation_frame='Year',
                      color_continuous_midpoint=0.8,
                      color_continuous_scale= px.colors.sequential.RdBu,
                      width=1500,height=600)
    fig.update_layout(title=' Compare Population Country ( Age )',autosize=True)
    color_continuous_scale=[[0, 'red'], [0.5, 'yellow'], [1, 'blue']]
    fig.show()

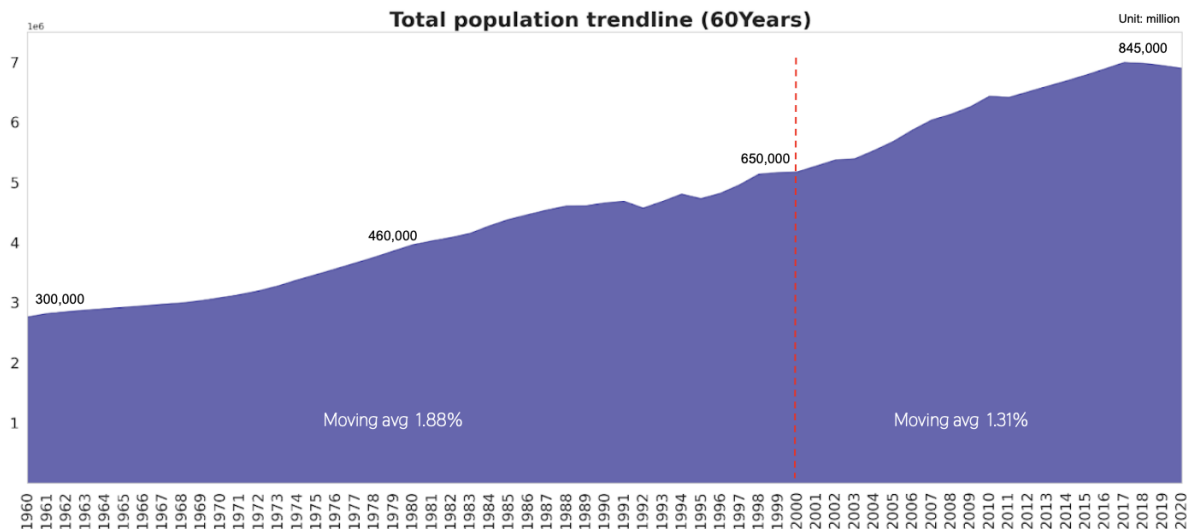
animated_graph(df,'Age by country ')
```

Compare Population Country (Age)



4.1 แนวโน้มประชากรโลกเป็นอย่างไร ในอดีต ปัจจุบันและ trend ในอนาคต

จำนวนประชากรโลกย้อนหลัง 60 ปี



%YOY 1960 -2000

1.88%

%YOY 2001 -2020

1.31%

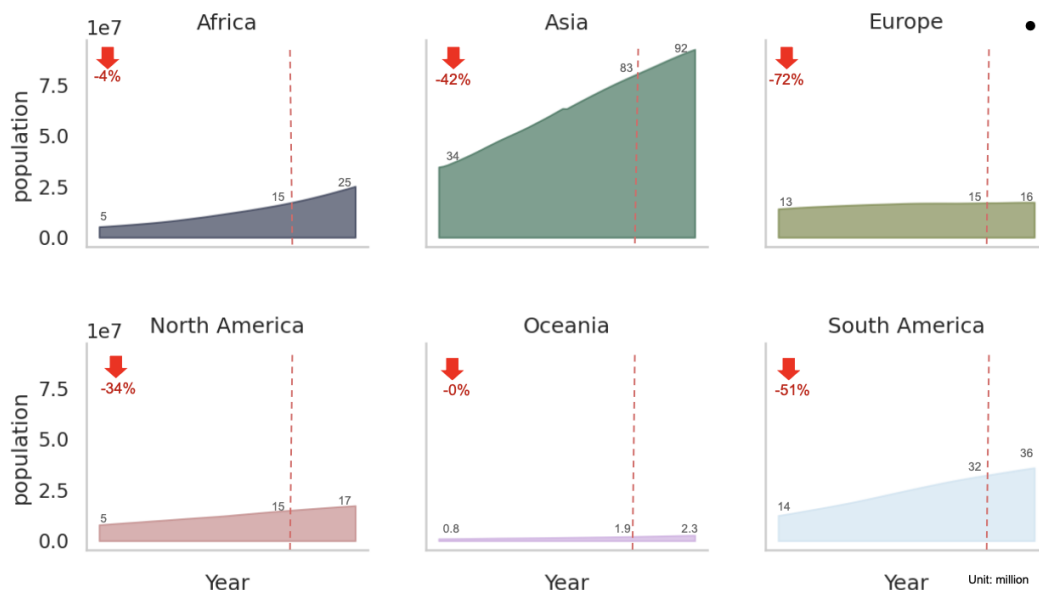
%FY0Y 2021 -2031

0.92%

- จากการศึกษาพบว่าจำนวนประชากรโลกเพิ่มขึ้นในทุกๆ ปี อย่างมีนัยยะสำคัญ เมื่อเปรียบเทียบ YOYพบว่าการเติบโตสูงถึงปีละ **1.88%** ช่วง40ปีแรก หลังจากนั้นตั้งแต่ปี 2010 การขยายตัวของ ประชากรลดลงเหลือเพียง **1.31%** ซึ่งจากการคาดการณ์ในอีก 10ข้างหน้าการเติบโตจะลดลงเหลือเพียง **0.92%**

จำนวนประชากรโลกย้อนหลัง 60 ปี รายRegion

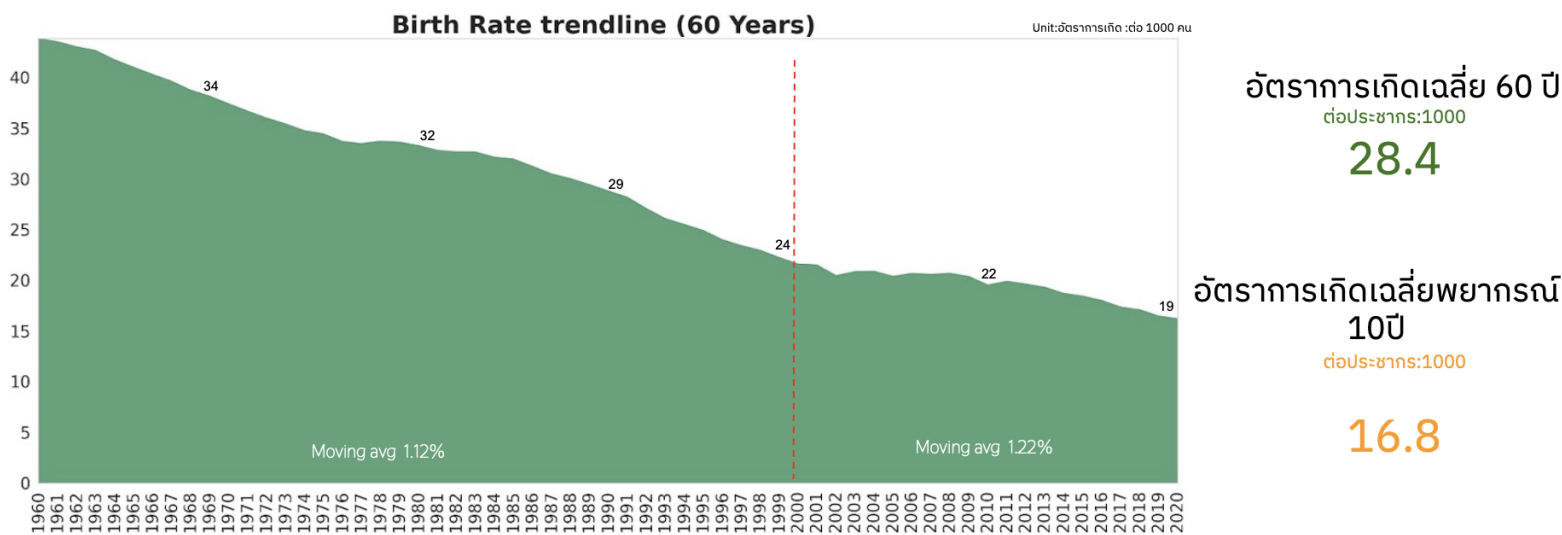
Population trendline by region



- เมื่อแบ่งข้อมูลดูรายภูมิภาคพบว่าAsiaเป็นภูมิภาคที่มีประชากรมากที่สุดและเติบโตอย่างต่อเนื่องในขณะที่ภูมิภาคOceaniaมีประชากรน้อยที่สุดและเมื่อมองในมุมการเติบโตของประชากรEuropeเป็นภูมิภาคที่มีการเติบโตเฉลี่ยต่ำสุด

4.2 แนวโน้มการเกิดของประชากรโลก

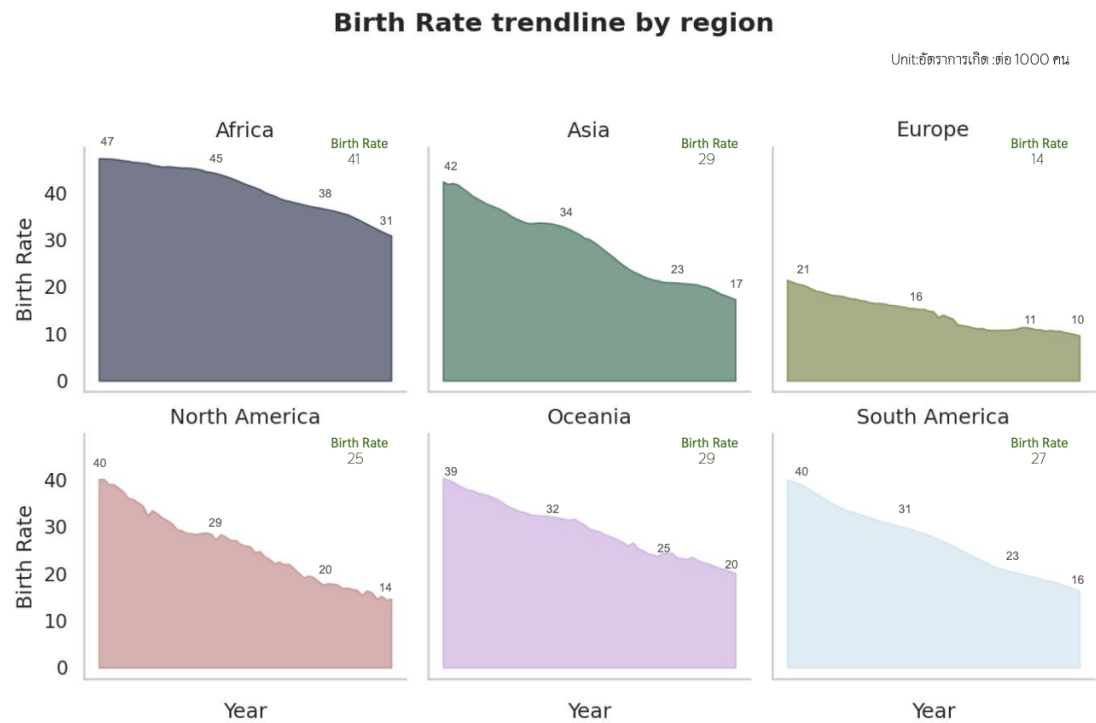
อัตราการเกิดของประชากรย้อนหลัง 60 ปี



- เมื่อพิจารณาพบว่า อัตราการเกิดมีแนวโน้มลดลงอย่างต่อเนื่อง มีค่าเฉลี่ย 60 ปีย้อนหลังอยู่ที่ **28.4** คน ต่อประชากร 1,000 คน และอัตราการเกิดจะลดลงเหลือ **16.8** คน ในอีก 10 ปีข้างหน้า ด้วยเหตุการณ์มีคู่น้อยลง, เพศทางเลือก , สภาพเศรษฐกิจ ที่มีผลต่อการมีลูกมากขึ้นด้วยความพร้อมหลายประการ

cr. <https://www.thunkhaotoday.com/news/up-to-date>

อัตราการเกิดของประชากรย้อนหลัง 60 ปี รายภูมิภาค

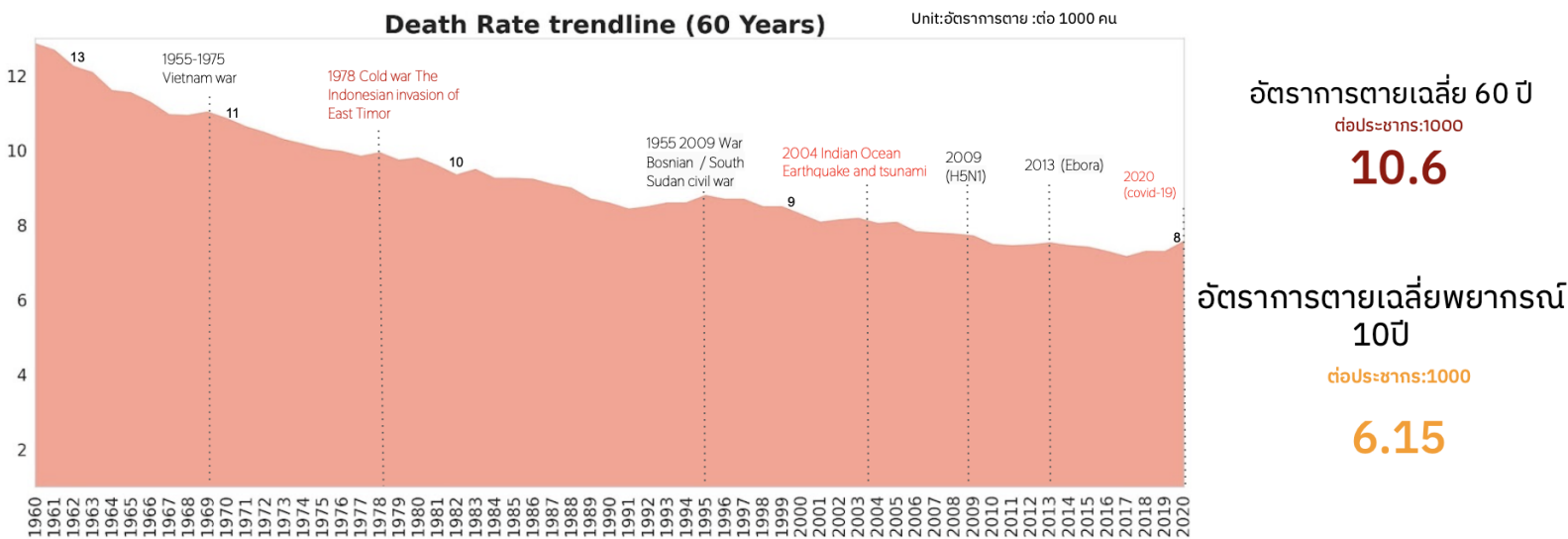


- เมื่อแบ่งข้อมูลดูรายภูมิภาคพบว่า ทุกภูมิภาคมีอัตราการเกิดลดลง **Africa** เป็นภูมิภาคที่มีอัตราการเกิดสูงที่สุด มีค่าเฉลี่ยอยู่ที่ **41** คน แต่อัตราการเกิด **Europe** เป็นภูมิภาคที่มีการเติบโตเฉลี่ยต่ำสุด อยู่ที่ **14** คน
- สาเหตุที่อัตราการเกิดของ Africa มากที่สุดเพราะความเชื่อของการมีบุตร และการคุมกำเนิดที่เข้าไม่ถึงซึ่งประเทศที่มีการเกิดมากที่สุด อยู่ที่ Niger , Chad , Angola และ South Sudan

Cr : <https://www.brandthink.me/content/africa>

4.3 แนวโน้มการตายของประชากรโลก

อัตราการตายของประชากรย้อนหลัง 60 ปี

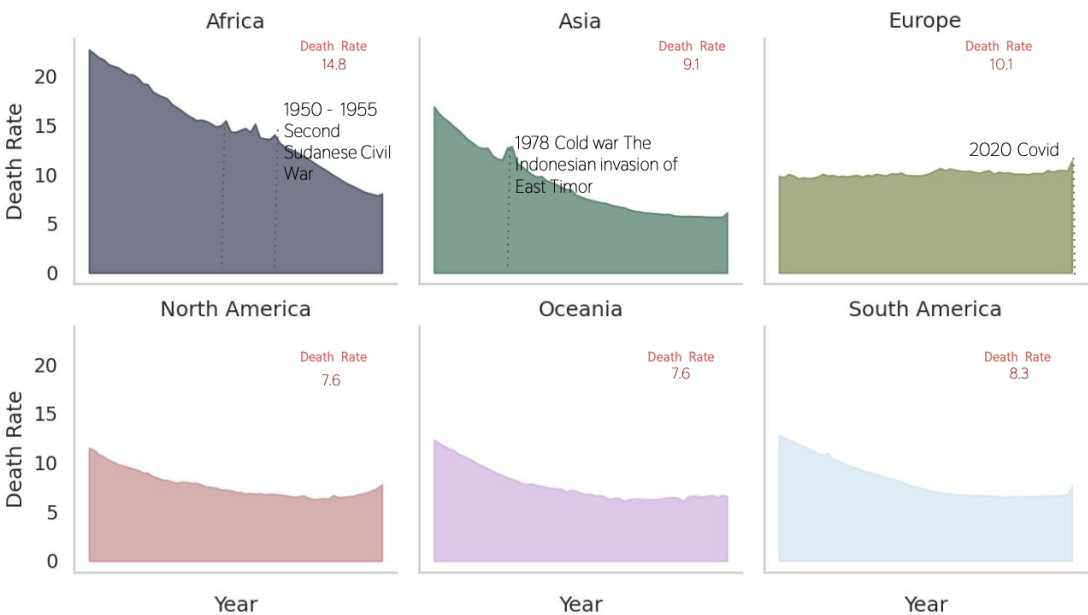


- จากการศึกษาพบว่า อัตราการตายมีแนวโน้มลดลงอย่างต่อเนื่อง มีค่าเฉลี่ยอยู่ที่ 10.6 คน ต่อประชากร 1,000 คน และอัตราการตายจะลดลงเหลือ 6.15 คน ในอีก 10 ปีข้างหน้า

cr: https://en.wikipedia.org/wiki/Indonesian_invasion_of_East_Timor
<https://www.cdc.gov/flu/pandemic-resources/1968-pandemic.html>
https://en.wikipedia.org/wiki/Vietnam_War

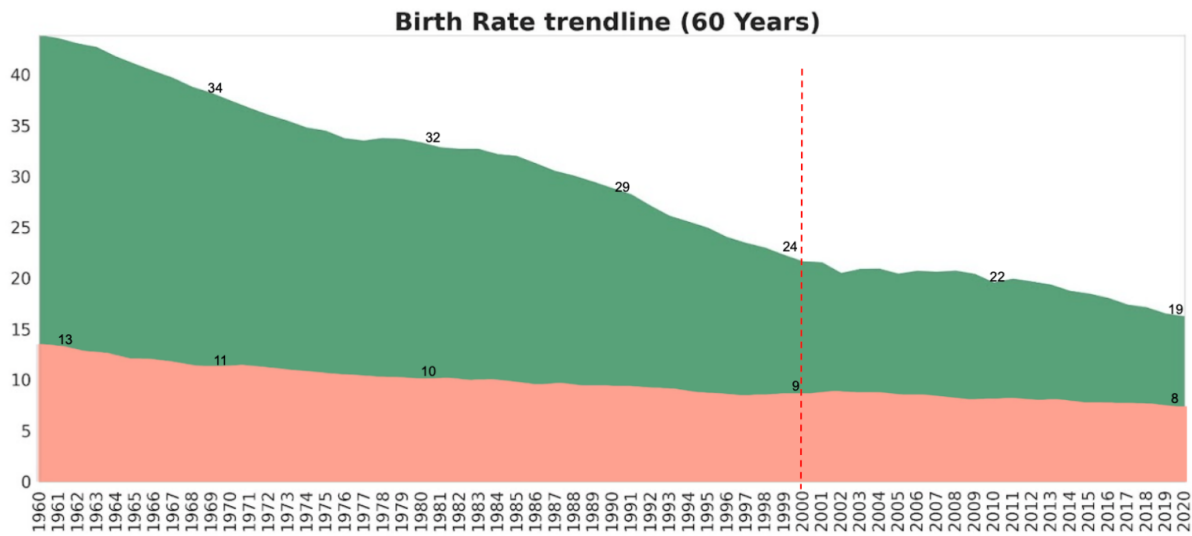
อัตราการตายของประชากรย้อนหลัง 60 ปี รายภูมิภาค

Death Rate trendline by region



- จากการสำรวจข้อมูลพบว่าการตายของประชากรแต่ละภูมิภาคลดต่ำลงด้วยเทคโนโลยีแล้ว ความทันสมัยของการแพทย์ Africa ยังตายมากที่สุด จากการ ขาดแคลนอาหาร และความยากจนภายในประเทศ
- นอกจากนี้อัตราการตายมีผลมาจากหลายปัจจัยทั้งสงครามโรคระบาดและภัยพิบัติจากธรรมชาติ จากธรรมชาติ

อัตราการเกิดเปรียบเทียบกับอัตราการตายของประชากรย้อนหลัง 60 ปี



ส่วนต่างระหว่างเกิด - ตาย

3 เท่า

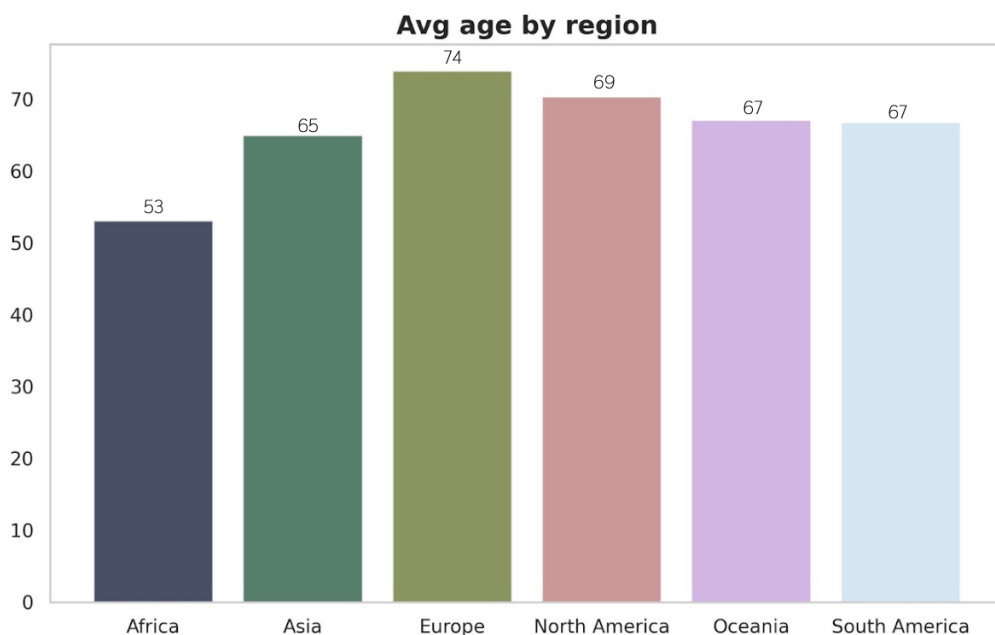
ส่วนต่างระหว่างเกิด - ตาย

2 เท่า

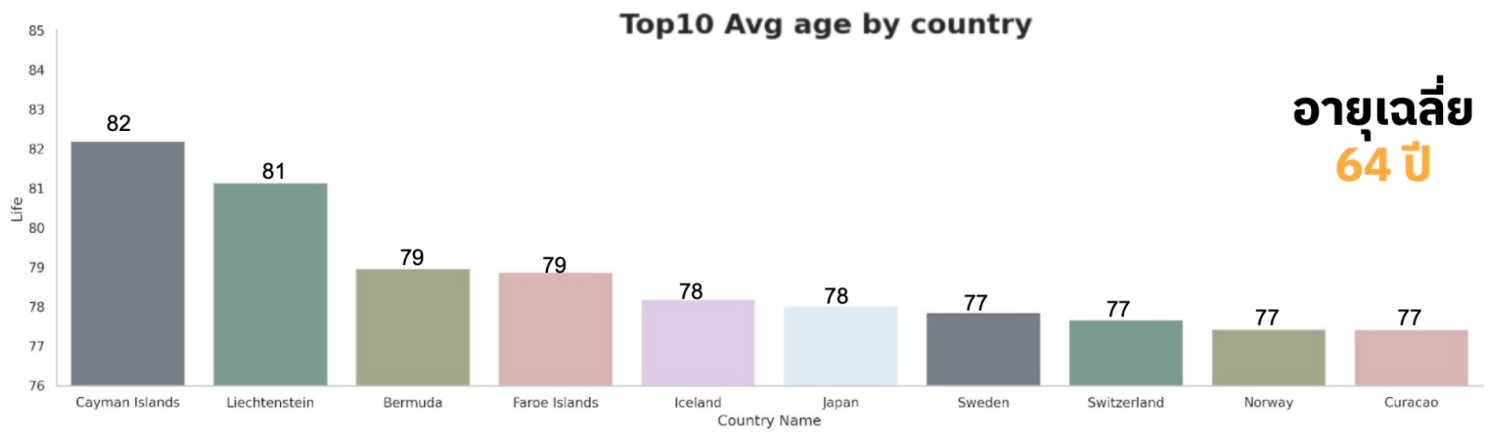
- จากข้อมูลในอดีตสะท้อนให้เห็นอัตราการเกิดและการตายเปรียบเทียบกับกันจาก 40 ปีก่อนห่างกันถึง 3 เท่า หลังจากปี 2020 ด้วยหลายประการเช่น คนโสดกันมากขึ้น , ปัญหาเศรษฐกิจ , ค่าใช้จ่ายที่เปลี่ยนไป , การคุมกำเนิด และ เพศทางเลือก ส่งผลให้อัตราการเกิดลดลงและการตายเพิ่มขึ้นจากโรคระบาด และ ภัยธรรมชาติ

4.4 อายุเฉลี่ยประชากรโลก

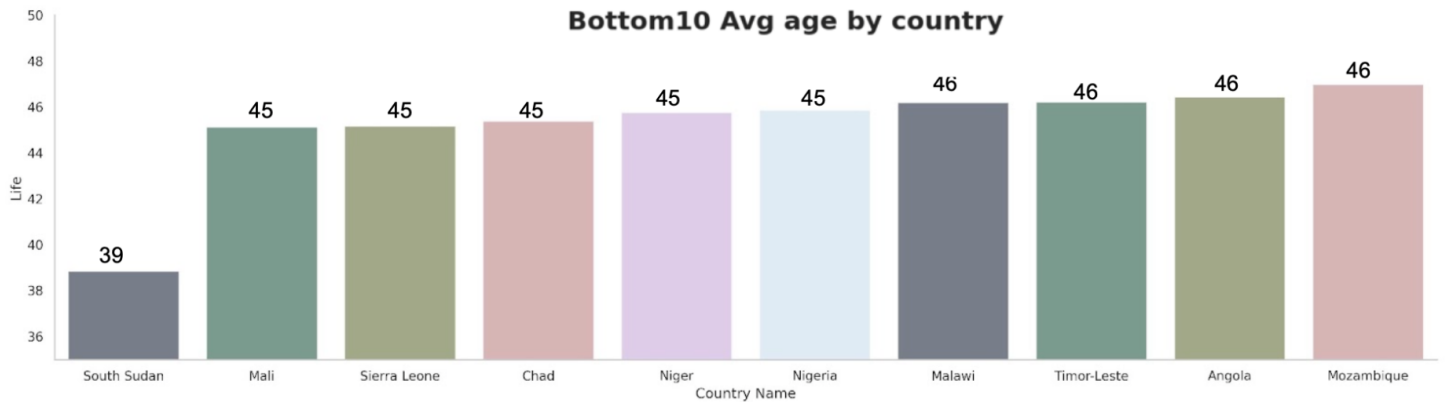
อายุเฉลี่ยของประชากรย้อนหลัง 60 ปี รายภูมิภาค



- อายุเฉลี่ยของ Africa ต่ำที่สุดเพราะระบบสาธารณสุขที่ยังไม่ครอบคลุม อีกทั้งยังเป็นแหล่งระบาดของโรค HIV วัณโรคปอดมาลาเลีย, รวมถึงปัญหาความยากจนส่งผลต่ออายุเฉลี่ยในขณะชาติตะวันตกอย่างยุโรปสามารถมีอายุได้ยืนกว่าเพราะมีการเข้าถึงระบบสาธารณสุขและการเป็นอยู่ที่ดีกว่า



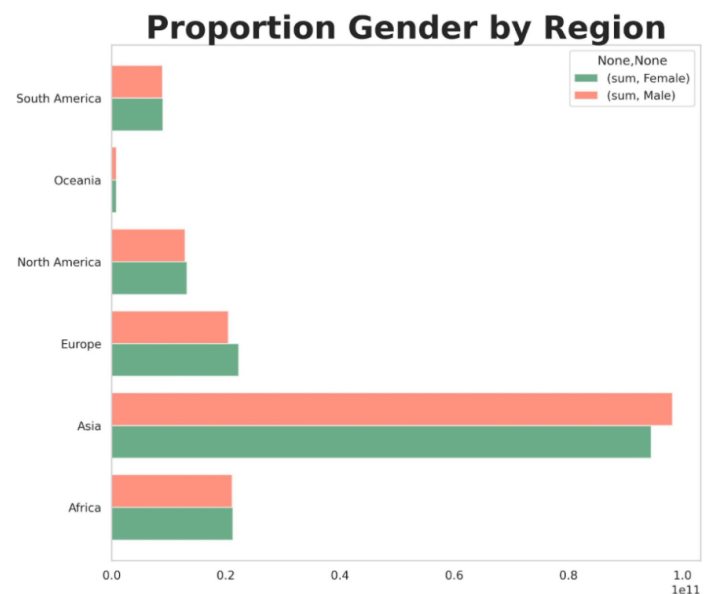
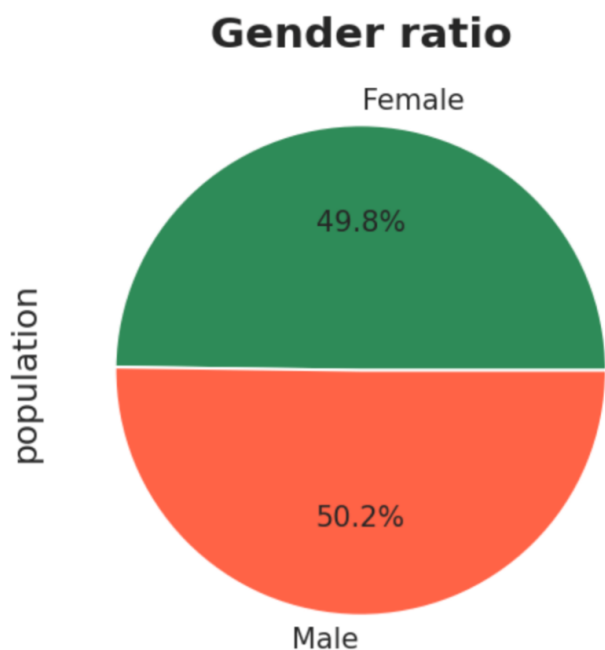
อายุเฉลี่ย
64 ปี



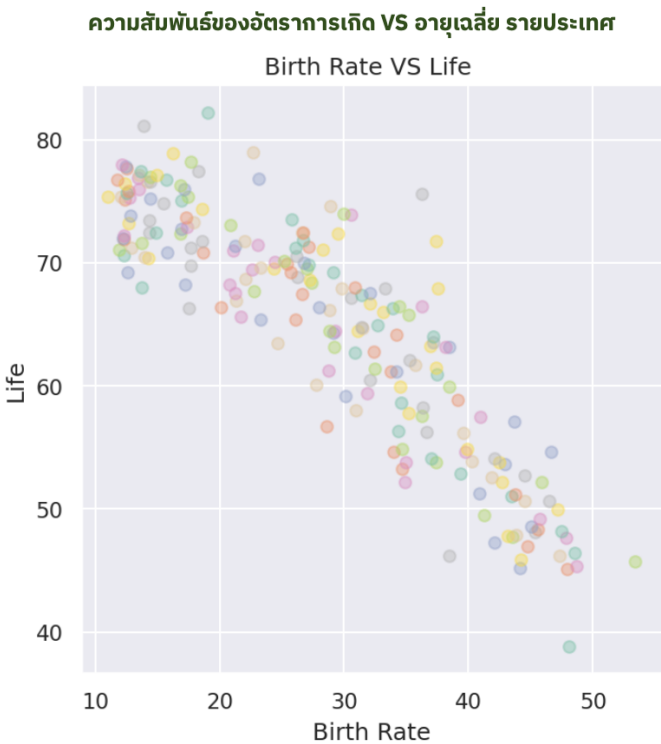
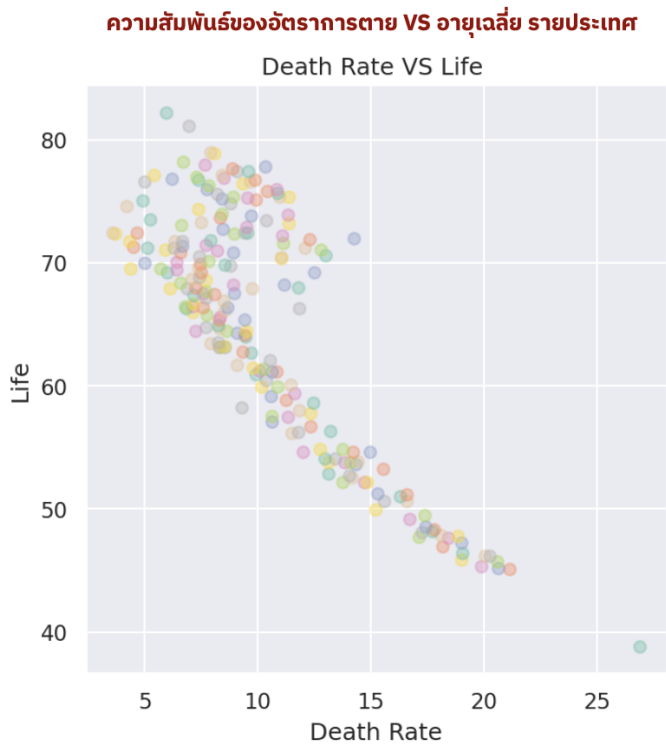
4.5 สัดส่วนประชากรโลกตามเพศสภาพ

สัดส่วนประชากรหญิงชาย 60 ปี้อยหลัง

จากจำนวนประชากรพบว่า ประชากรหญิงมากกว่าประชากรชาย และพบมากในภูมิภาคEurope และ north America แต่ในทางกลับกัน asiaยังมีประชากรชายมากกว่าประชากรหญิง

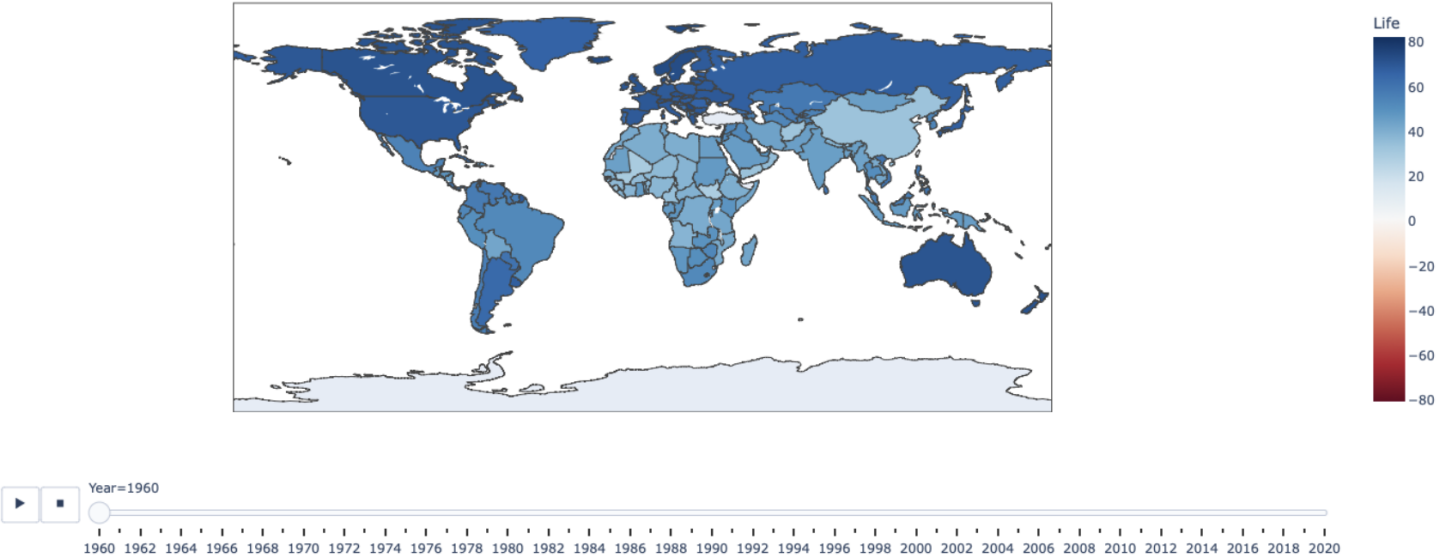


ความสัมพันธ์ของข้อมูล (Correlation)



ประชากรย้อนหลัง60 รายประเทศ

Compare Population Country (Age)



Q&A

- **Q : .แนวโน้มสถานการณ์ประชากรโลกในปัจจุบัน และอนาคต ?**
 - A : ลดลงอย่างมีนัยสำคัญจากสถานแวดล้อมหลายๆอย่างและใน10ปีข้างหน้าก็ยังลดต่ำอย่างต่อเนื่องเหลือเพียง 0.92%
- **Q : .อายุเฉลี่ยประชากรโลก ?**
 - A : 64 ปี จากข้อมูล 60 ปีย้อนหลัง
- **Q : .สัดส่วนประชากรชายต่อประชากรหญิง?**
 - A : 50.2% ต่อ 49.8 %
- **Q : .สัดส่วนประชากรชายต่อประชากรหญิง?**
 - A : 50.2% ต่อ 49.8 %
- **Q : แนวโน้มการเกิดของโลก**
 - A : 28.4 คน ต่อ ประชากร 1000 คน
- **Q : แนวโน้มการตายของโลก**
 - A : 10.6 คน ต่อ ประชากร 1000 คน

Challenge :

- 1 . การ unpivot data เพราะ ปกติdataที่ทำงานส่วนใหญ่เป็น sales transaction จะมี values แค่ช่องเดียวตั้งนี้ต้องการเตรียมdata frame นี้ประมาณ 40% ของการทำโปรเจค
- 2 . การทำvisual ตามที่เราต้องการ ค่อนข้างยากที่จะทำchart ออกมาให้ตรงใจ เช่นการกำหนด labels สี และ Front และใช้เวลาศึกษานาน ในการจะgenerate chart และstep การเตรียมdata frameในการทำกราฟ