

หัวข้อ : World population Analysis

Data source : open source จากinternet

Data Journey : [Load Data](#) นำเข้าข้อมูลก่อนต้องการใช้

```
#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 หลังจากนี้ต้องการ unpivot data ด้วยคำสั่งmelt เพื่อกำdata frame ที่เราต้องการ และใช้คำสั่ง merge เพื่อ map region และ continent เข้าเพื่อกำหนดเวลาไปทำ visual กับ insightเพิ่มเติม

1.1 ด้วย 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 เพื่อจัด group ข้อมูล

```
#เพิ่มค่าอธิบาย 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 วายมาต่อท้าย

```
# Life นำปีมาต่อหลัง
year_list = list(Life.iloc[:, 4:].columns)
Life1 = pd.melt(Life,['Country Name','Country Code','Indicator
Name'],value_vars=year_list,value_name='Life')
Life1.rename({'variable': 'Year'}, axis=1, inplace=True)
#ตัดมาเฉพาะ column Land มาต่อท้าย
Life2=Life1[['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 ตรวจสอบ data และ แทนค่าและจัดการ 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 คอลัมที่ไม่ต้องการทิ้ง

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

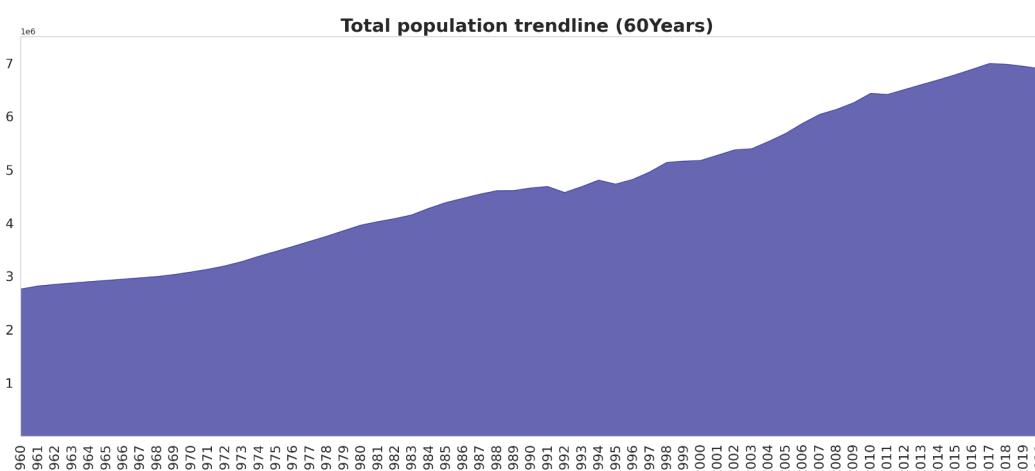
Data Journey : Step2 [EDA](#) โดยนำ data frame จากการทำ cleaning มาทำ analysis แต่ละประเด็นที่สนใจ

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("cubebehelix")
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] #population 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})
```



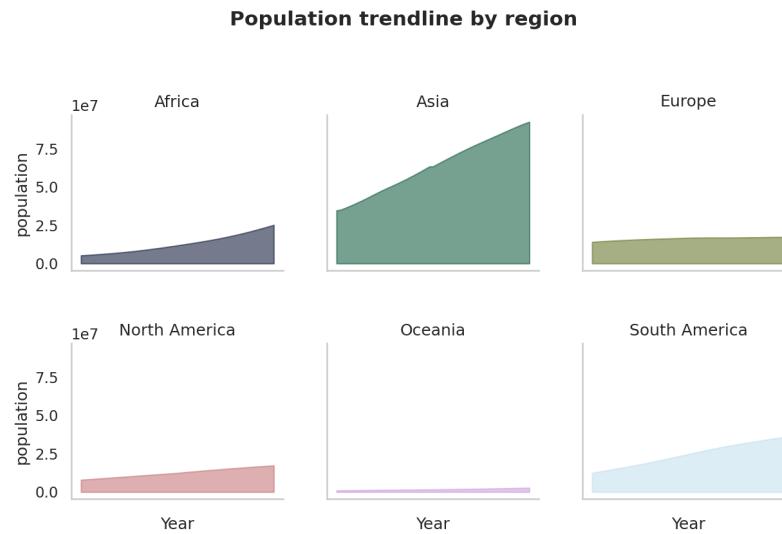
2.2 population trendline by region

```
## 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'] # សំណើនាំ series data

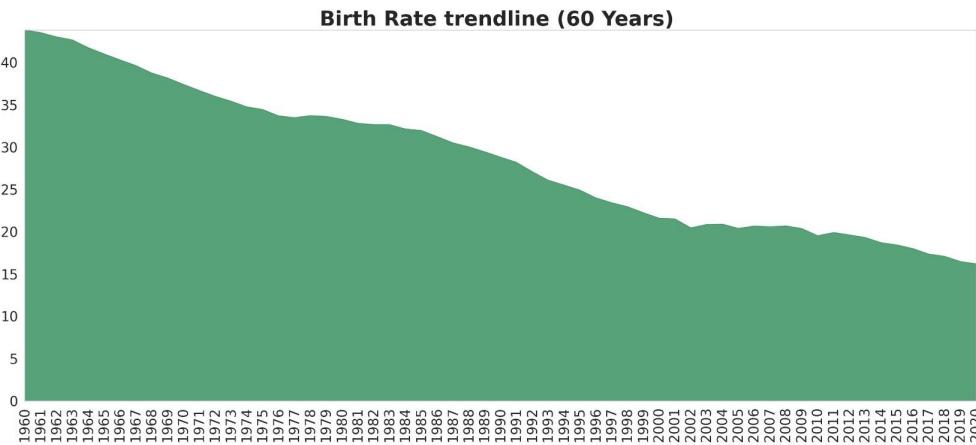
sns.set_style("whitegrid", {'axes.grid' : False})
g = sns.FacetGrid(df2,
col='Continent', hue='Continent', col_wrap=3, palette=sns.color_palette("cubehelix"))
g.set(xticklabels[])
g.tick_params(bottom=False)

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



2.3 birth rate trendline

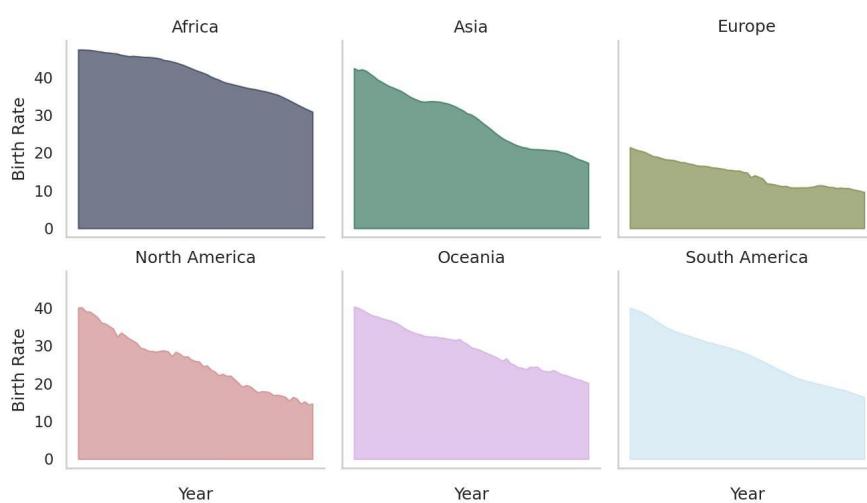
```
### 4 . Birth rate Trendline ##
btotal = df.pivot_table(index='Year', columns=None, values=['Birth Rate'], aggfunc=['median'])
type(btotal)
c=palette=sns.color_palette("cubehelix")
plt.figure(figsize=(15, 5))
plt.xticks(fontsize=10, rotation=90)
plt.yticks(fontsize=10)
tyear = btotal.index
btotal =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

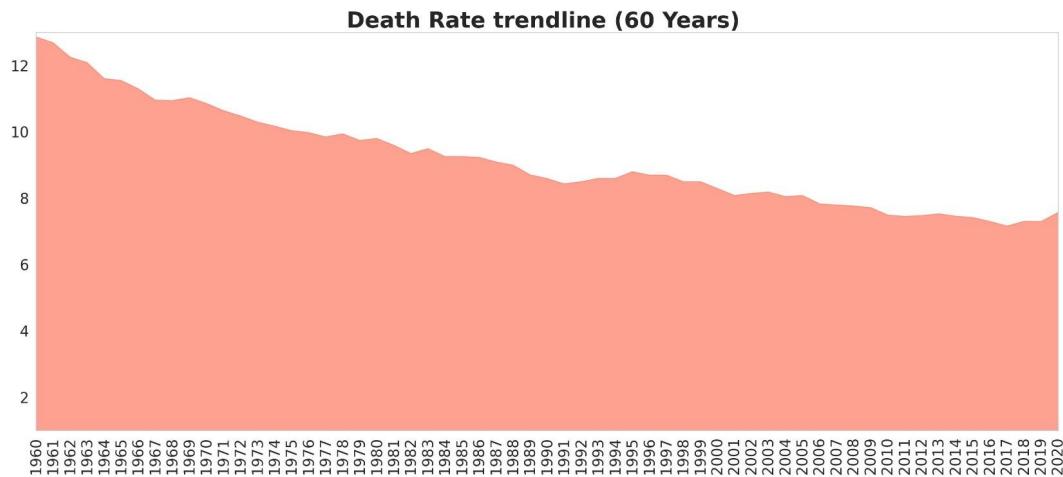
```
from seaborn._core.properties import Color
df_pivot19 = df.pivot_table(index=['Year', 'Continent'], columns=None, values=['Birth Rate'], aggfunc=['mean'])
type(df_pivot19) #pivot data
df2 = df_pivot19.reset_index() # make series
df2.columns = ['Year', 'Continent', 'Birth Rate']
sns.set_style("whitegrid", {'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
```

Birth Rate trendline by region



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)

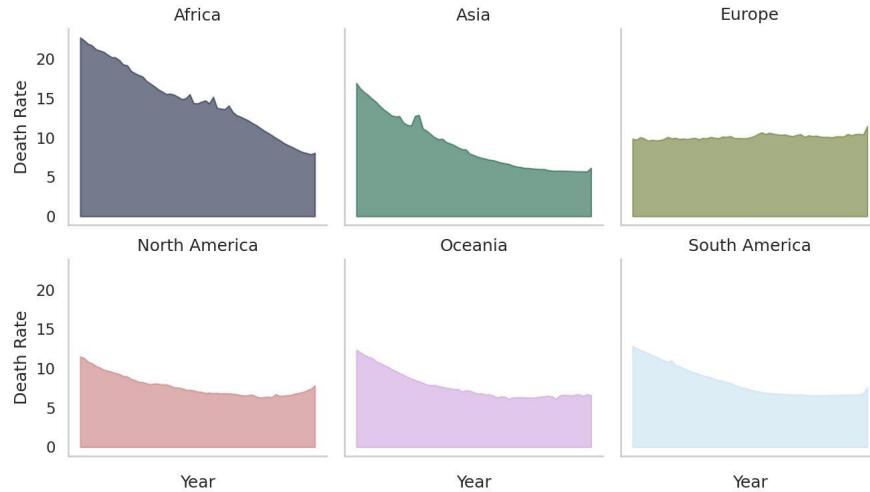
df2 = df_pivot8.reset_index()
df2.columns = ['Year','Continent','Death Rate']

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



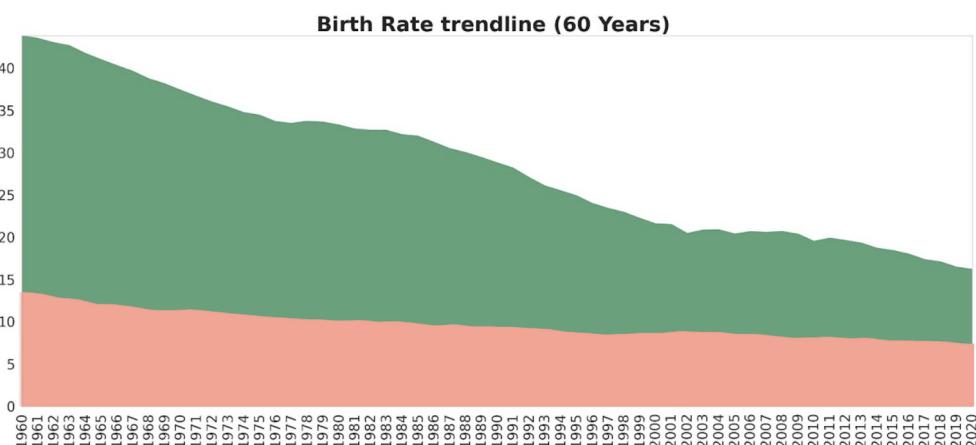
2.7 death rate VS birth rate

```

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

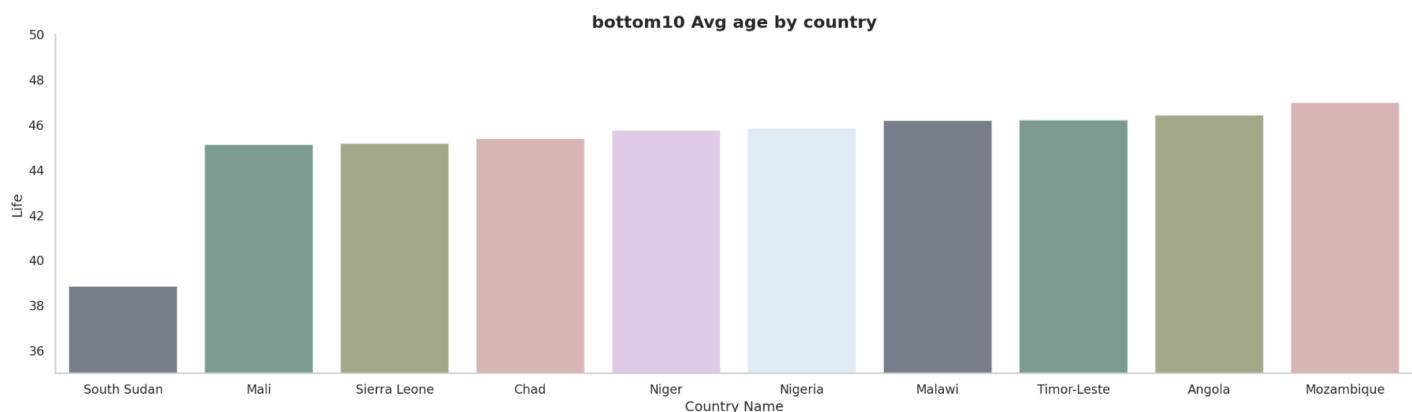
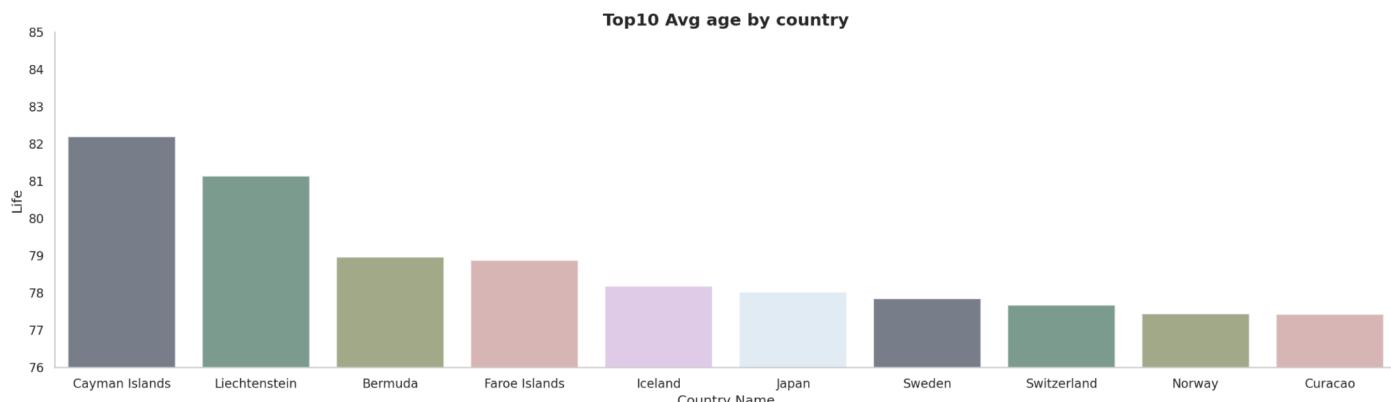
plt.figure(figsize=(20 ,10))
plt.xticks(fontsize=10,rotation=90)
plt.yticks(fontsize=10)
plt.margins(0,0)
pal = ["tomato", "seagreen",]
nal = ["alpha=0.5","alpha=0.70"]
plt.stackplot(xyear_asia,b,d, labels=['Death Rate','Birth Rate '], colors=pal, alpha=0.6 )
plt.legend(loc='upper right')
plt.title("Death 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)

```



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



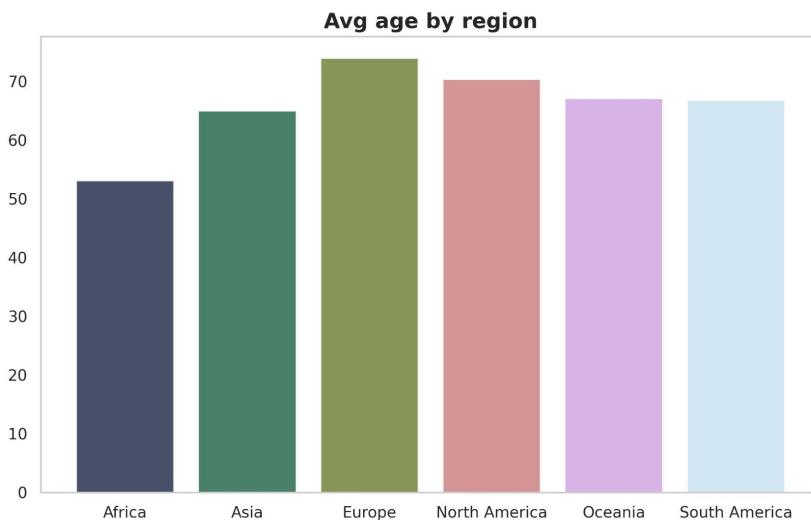
2.9 avg age by region

```
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("cubebehelix")

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

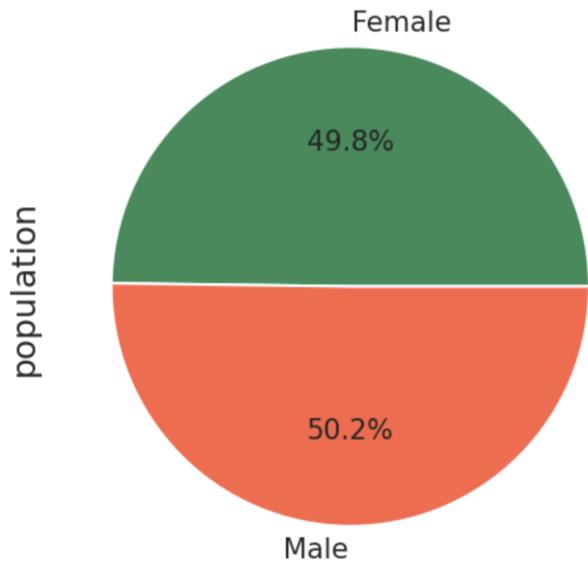
plt.figure(figsize=(10,6))
plt.bar(Continent,total_pop,color=c, alpha=0.8)
plt.title("Avg age by region", fontsize=15, fontweight="bold")
plt.savefig(" Bar Age Rate by region.jpg", dpi=300)
```



2.10 gender ratio

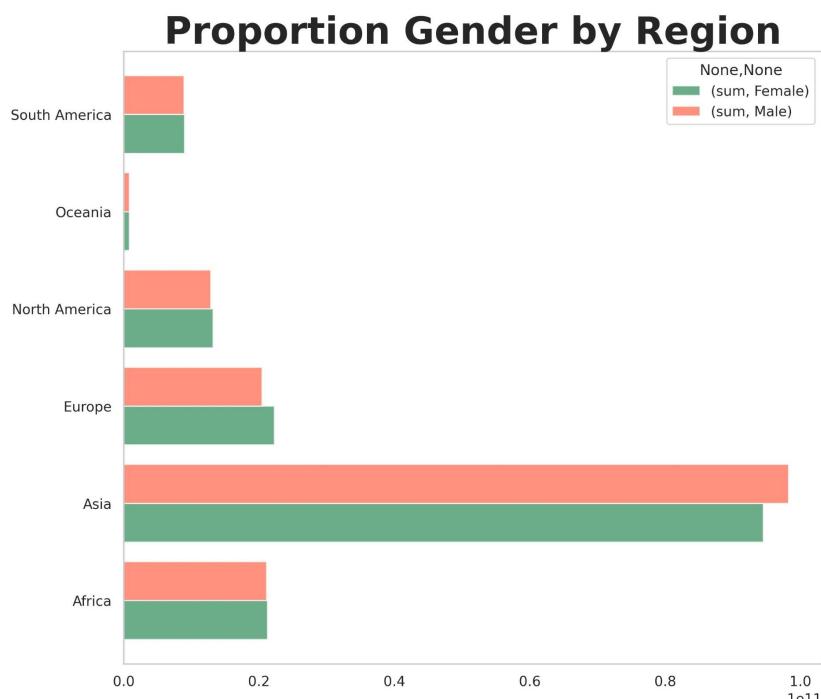
```
male =df['Male'].sum()
female = df['Female'].sum()
df4 = pd.DataFrame([['Male',male], ['Female',female]],columns=('Gender', 'population'))
type(df4)
colors1 = ['tomato', 'seagreen']
plt.figure(figsize=(10, 10))
palette = {"Female": "tab:pink",
           "male": "tab:royalblue"}
data = df4.groupby("Gender") ["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

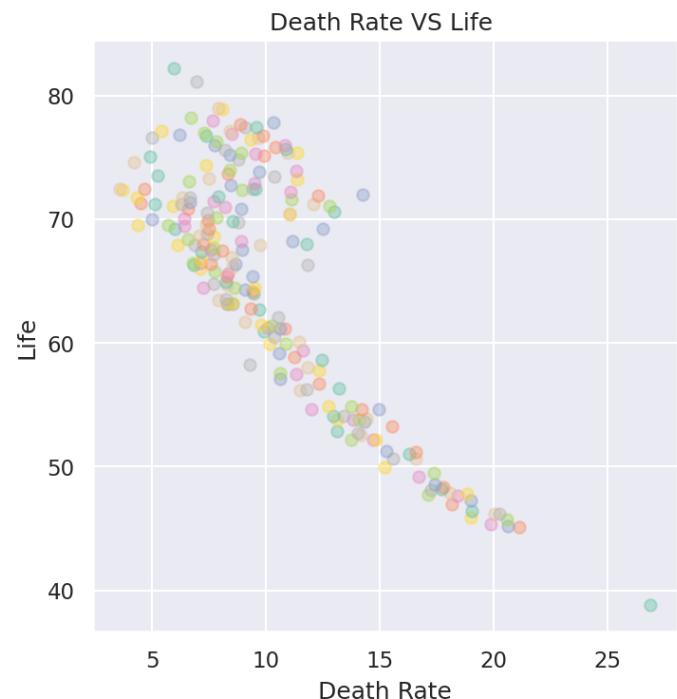
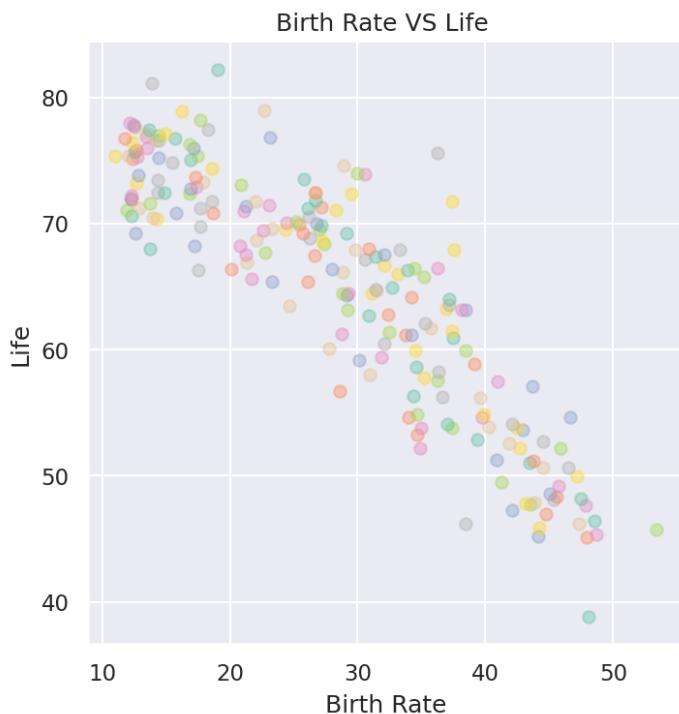
```
sex = df.pivot_table(index='Continent',columns=None,values=['Male','Female'],aggfunc=['sum'])  
sex  
type(sex)  
pal = ["seagreen", "tomato",]  
c=palette=sns.color_palette("cubehelix")  
sex.plot.barh(figsize=(10,9),color=pal,title = "Proportion Gender by Region",alpha=0.7,width =  
0.8,)  
plt.title("Proportion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



2.12 correlation

```
df33=df.pivot_table(index='Country Name',columns=None,values=['Birth Rate','Death Rate','Life'],aggfunc='mean').sort_values('Life',ascending=True).reset_index()
sns.lmplot( data=df33,
            x='Birth Rate', y='Life',
            hue='Country Name',
            markers="o",
            palette="Set2",
            scatter_kws=dict(alpha=0.4),
            line_kws=dict(alpha=0.6),
            legend=False
        ).set(title='Birth Rate VS Life ')
```

```
df33=df.pivot_table(index='Country Name',columns=None,values=['Birth Rate','Death Rate','Life'],aggfunc='mean').sort_values('Life',ascending=True).reset_index()
sns.lmplot( data=df33,
            x='Death Rate', y='Life',
            hue='Country Name',
            markers="o",    # Set markers for each hue color
            palette="Set2",      # Colors to use for the different levels of the hue variable
            scatter_kws=dict(alpha=0.4),  # Set the scatter plot
            line_kws=dict(alpha=0.6),      # Set the regression line
            legend=False
        ).set(title='Death Rate VS Life ')
```



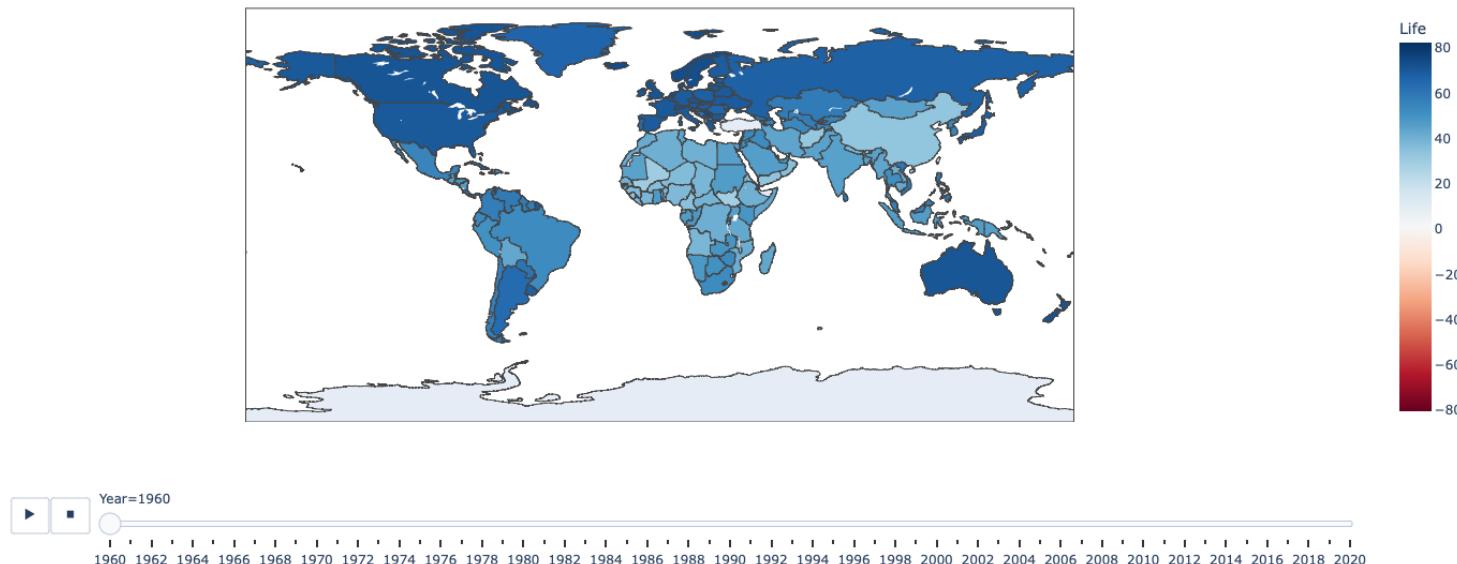
2.13 Animation map

```
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()

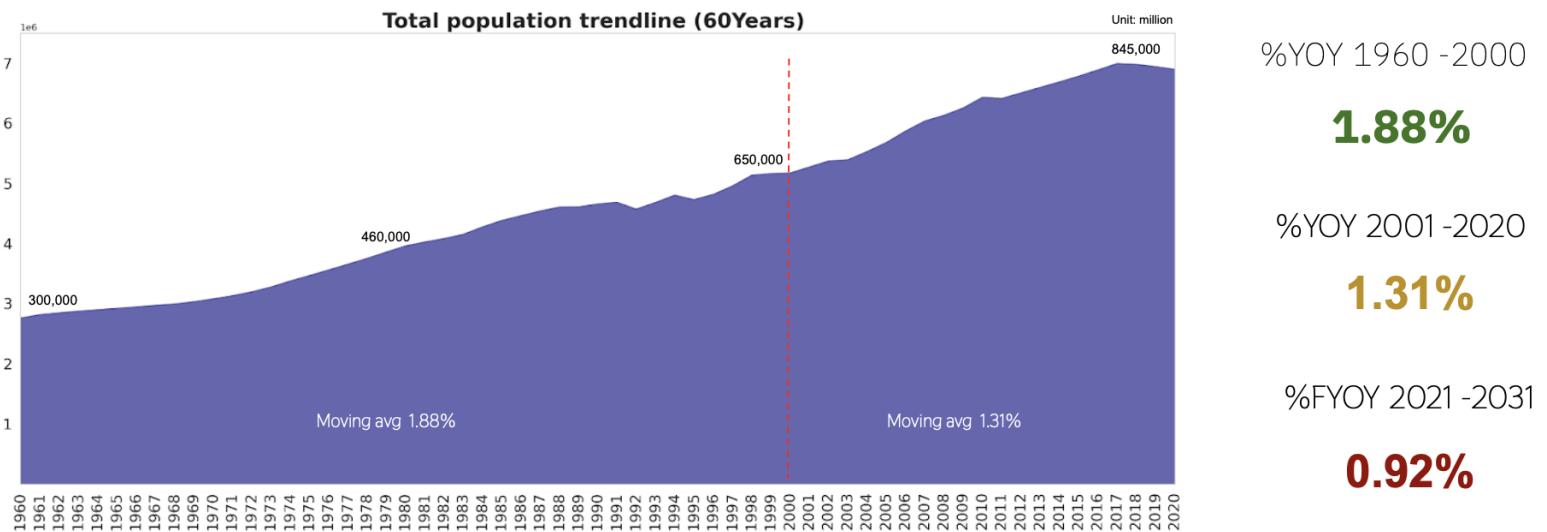
animated_graph(df, 'Age by country ')
```

Compare Population Country (Age)



4.1 แนวโน้มประชากรโลกเป็นอย่างไรในอดีต ปัจจุบันและ trend ในอนาคต

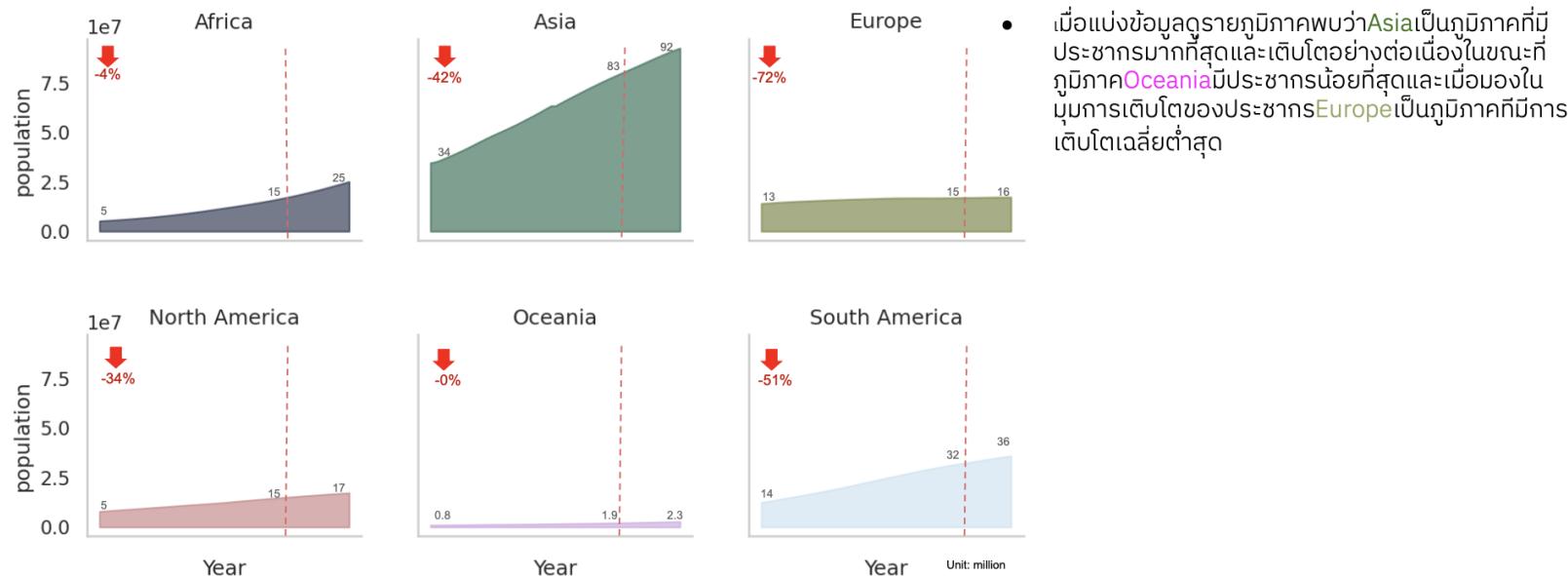
จำนวนประชากรโลกย้อนหลัง 60 ปี



- จากการศึกษาพบว่าจำนวนประชากรโลกเพิ่มขึ้นในทุกๆ ปี อย่างมั่นคงสำคัญ เมื่อเปรียบเทียบ YOY พบว่าการเติบโตสูงถึงปีละ **1.88%** ช่วง 40 ปีแรก หลังจากนั้นตั้งแต่ปี 2010 การขยายตัวของประชากรลดลงเหลือเพียง **1.31%** ซึ่งจากการคาดการณ์ในอีก 10 ปีข้างหน้าการเติบโตจะลดลงอีกเหลือเพียง **0.92%**

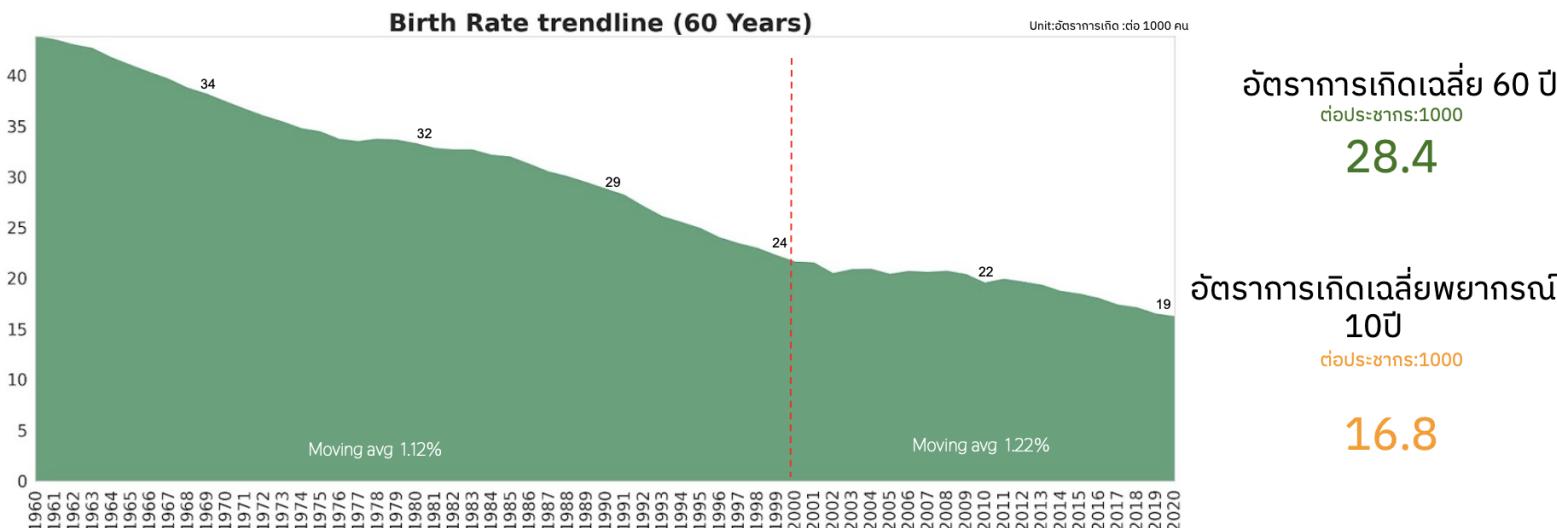
จำนวนประชากรโลกย้อนหลัง 60 ปี ราย Region

Population trendline by region



4.2 แนวโน้มการเกิดของประชากรโลก

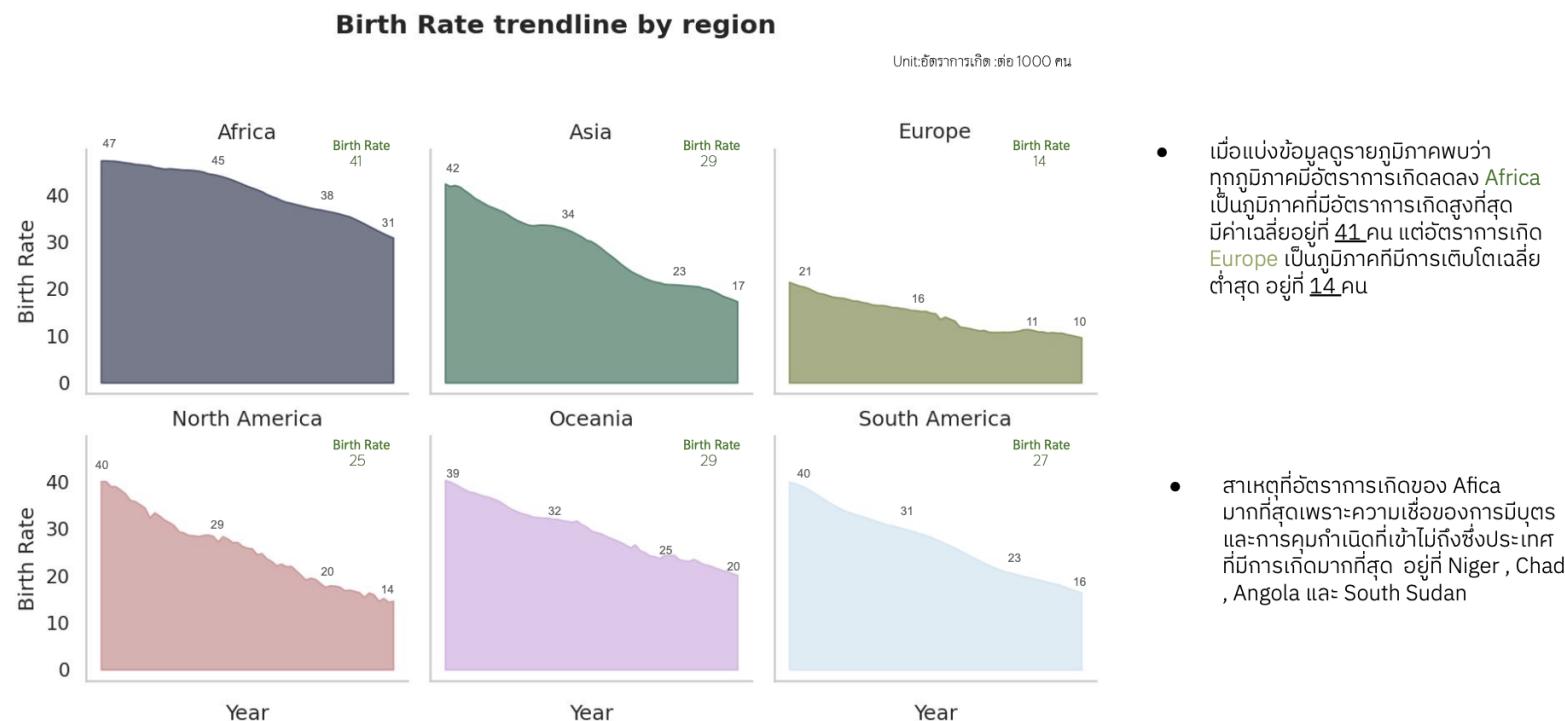
อัตราการเกิดของประชากรย้อนหลัง 60 ปี



- เมื่อพิจารณาพบว่า อัตราการเกิดมีแนวลดลงอย่างต่อเนื่อง มีค่าเฉลี่ย 60 ปีย้อนหลังอยู่ที่ **28.4** คน ต่อประชากร 1,000 คน และอัตราการเกิดจะลดลงเหลือ **16.8** คน ในอีก 10 ปีข้างหน้า ด้วยเหตุผลการมีครรภ์น้อยลง, เพศถูกเลือก, สภาพเศรษฐกิจ ที่มีผลต่อการมีลูกมากขึ้นด้วยความพร้อมทางการเงิน

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

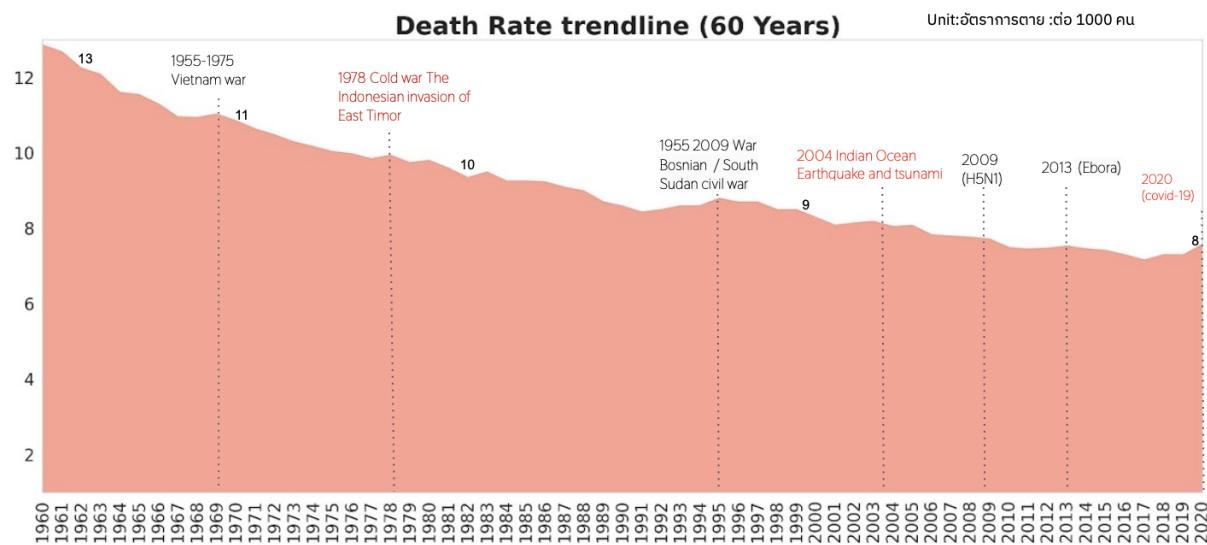
อัตราการเกิดของประชากรย้อนหลัง 60 ปี รายภูมิภาค



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

4.3 แนวโน้มการตายของประชากรโลก

ວັດທະນາການຕາຍຂອງປະຊາກົມຢ້ວນຮລັງ 60 ປີ



อัตราการตายเฉลี่ย 60 ปี

ព្រះរាជាណាចក្រកម្ពុជា

10.6

ວັດທະນາການຕາຍເຈລ້ຽພຢາກຮນ 10ປີ

ຕົວປະຈາກຮຽນ:1000

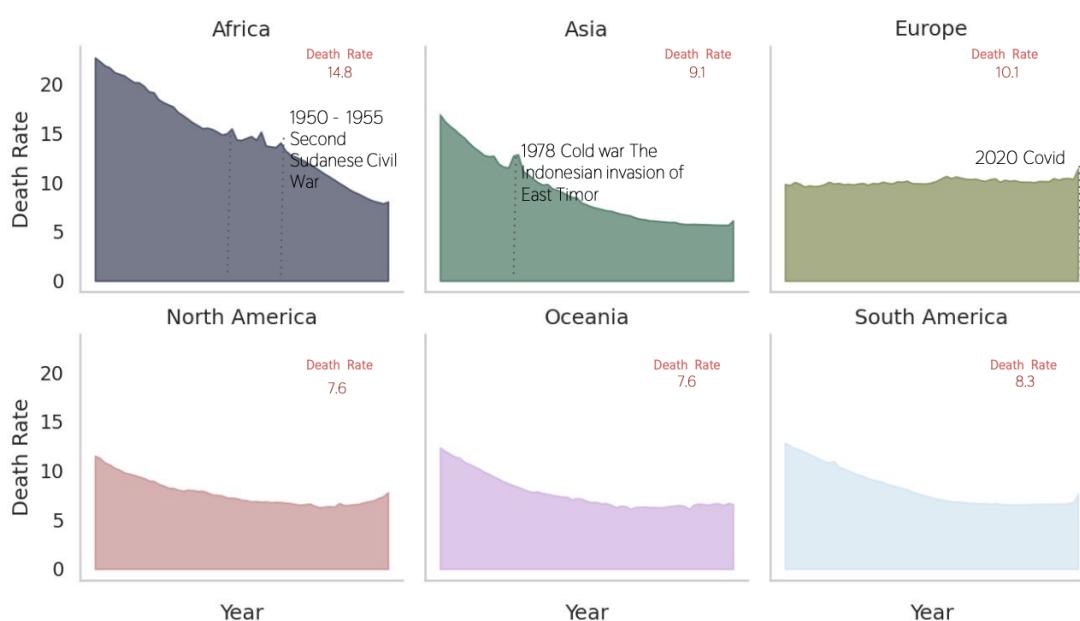
6.15

- 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

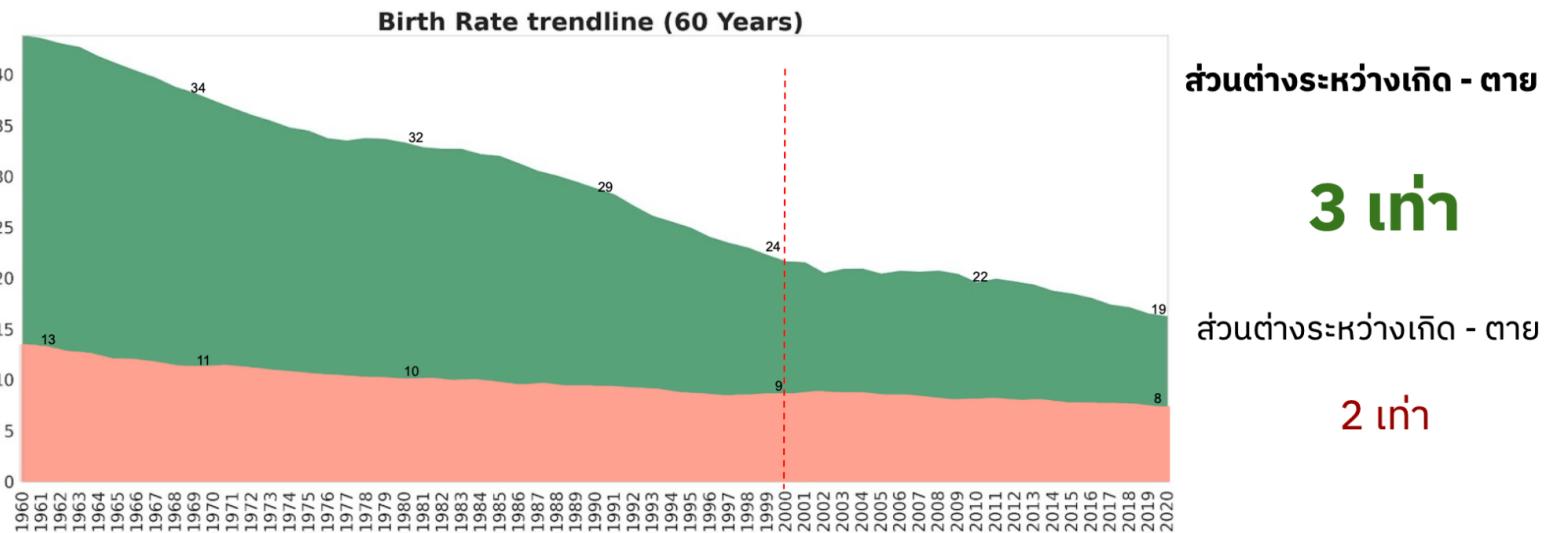
ວັດທະນາການຕາຍຂອງປະຊາກຍ້ວນແລ້ງ 60 ປີ ຮາຍກມືການ

Death Rate trendline by region



- จากการสำรวจข้อมูลพบว่าการต่ายของประชากรแต่ละภูมิภาคคล้ายกันอย่างมากในโลกี้ แล้ว ความทันสมัยของการแพทท์ Africa ยังต่ายมากที่สุด จากการ ขาดแคลนอาหาร และความยากจนภายในประเทศ
 - นอกจากนี้อัตราการต่ายมีผลมาจากการปัจจัยกั้งสัมภาระโรคระบาดและภัยพิบัติจากธรรมชาติ จากรัฐบาลชาติ

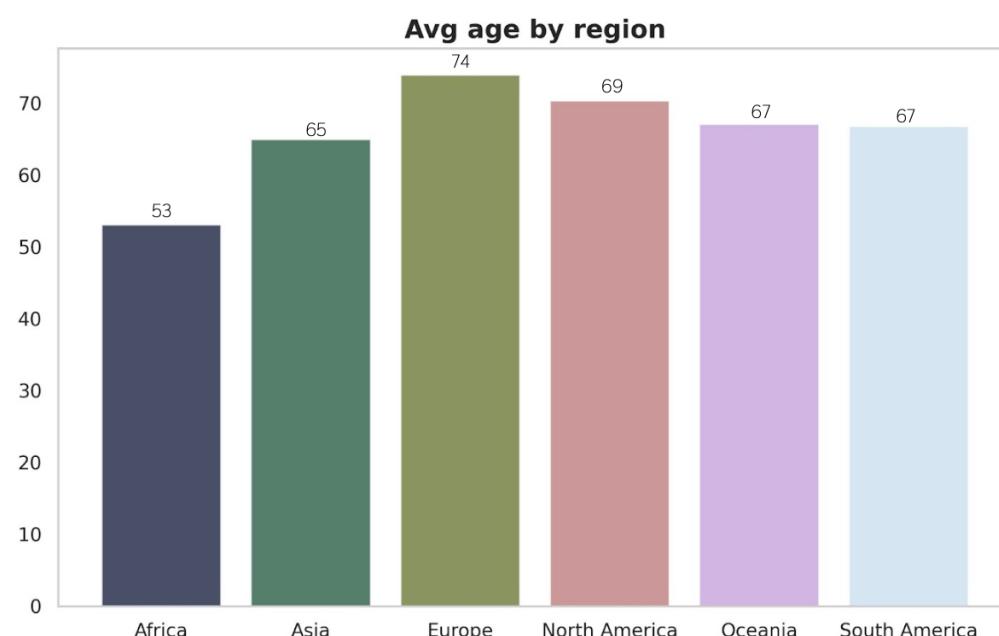
อัตราการเกิดเปรียบเทียบอัตราการตายของประชากรยังหลัง 60 ปี



- จากข้อมูลในเดือนก่อนให้เห็นอัตราการเกิดและการตายเปรียบเทียบกันจาก 40 ปีก่อนห่างกันถึง 3 เท่า หลังจากปี 2020 ด้วยผลของการเช่น คนโสดกันมากขึ้น , ปัญหาเศรษฐกิจ , ค่านิยมที่เปลี่ยนไป , การคุณกำเนิด และ เพศทางเลือก ส่งผลให้อัตราการเกิดลดลงและการตายเพิ่มขึ้นจากโรคระบาด และ กัยธรรมชาติ

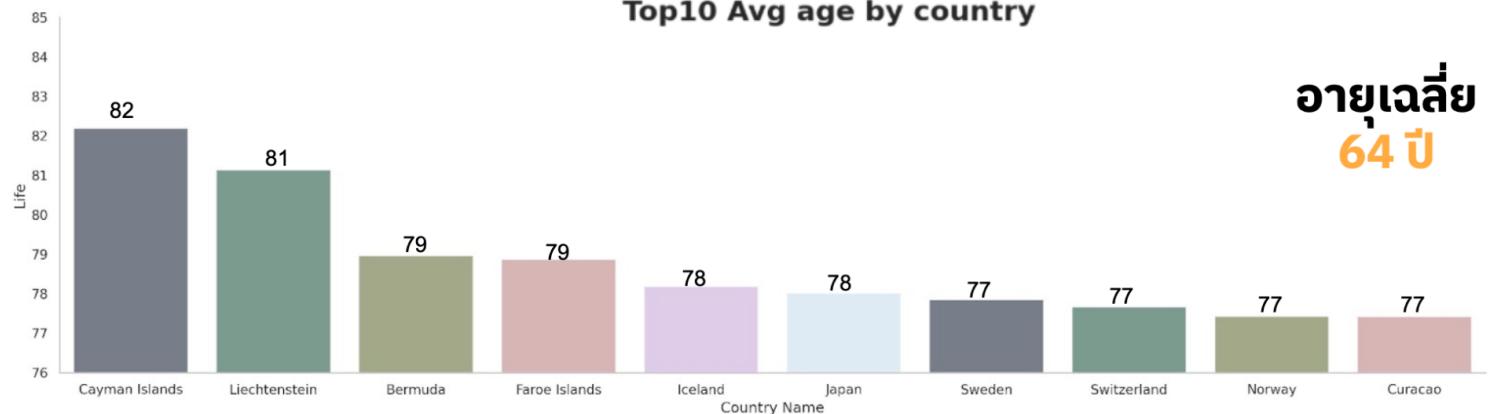
4.4 อายุเฉลี่ยประชากรโลก

อายุเฉลี่ยของประชากรยังหลัง 60 ปี รายภูมิภาค



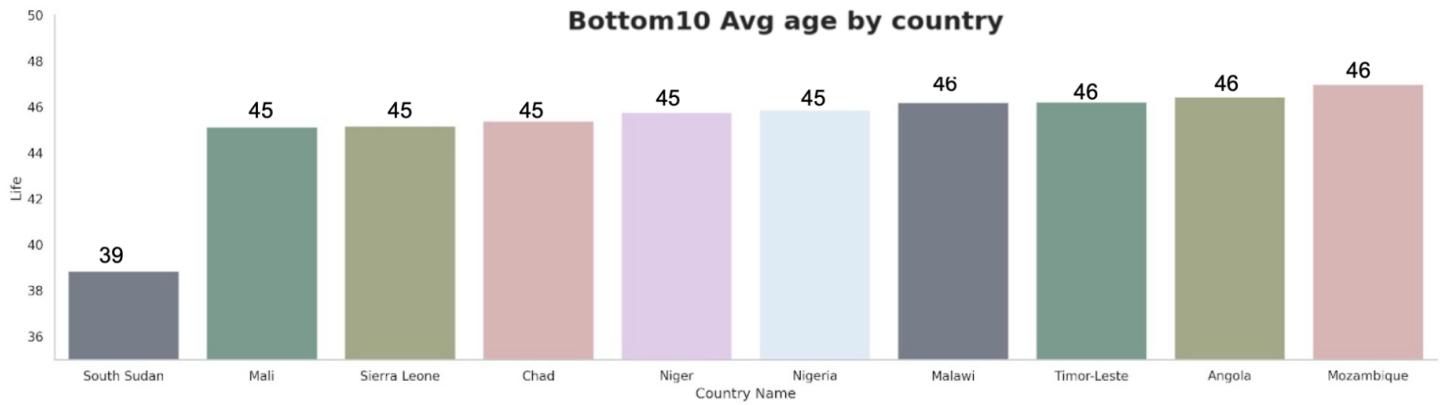
- อายุเฉลี่ยของ Africa ต่ำที่สุด เพราะระบบสาธารณสุขที่ยังไม่ครอบคลุมอีก ทั้งยังเป็นแหล่งระบาดของโรค HIV วันโรคปอดมาลาเรีย, รวมถึง ปัญหาความอยากรจนส่งผลต่ออายุ เฉลี่ยในขณะชาติตะวันตกอย่าง ยุโรปสามารถมีอายุได้ยืนกว่า เพราะมีการเข้าถึงระบบสาธารณสุขและการเป็นอยู่ที่ดีกว่า

Top10 Avg age by country



อายุเฉลี่ย
64 ปี

Bottom10 Avg age by country



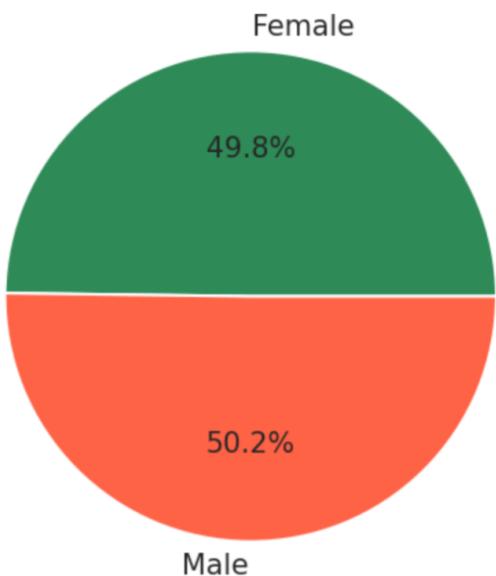
4.5 สัดส่วนประชากรโลกตามเพศสภาพ

สัดส่วนประชากรหญิงชาย 60 ปีขึ้นไป

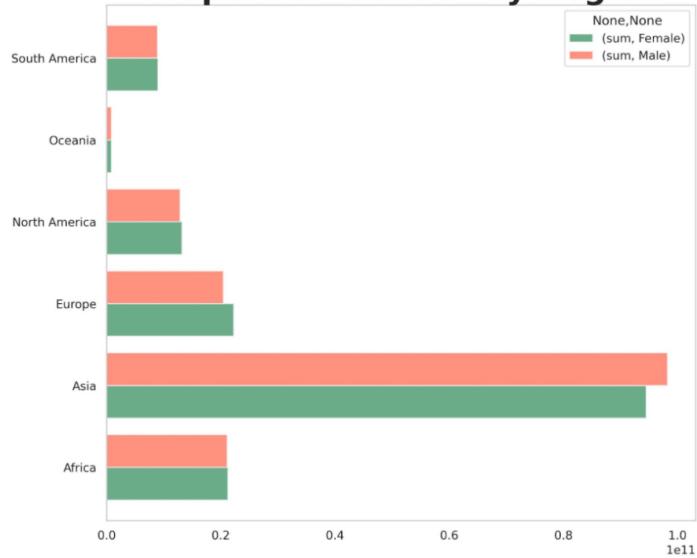
จากจำนวนประชากรพบว่า ประชากรหญิงมากกว่าประชากรชาย และพบมาในภูมิภาค Europe และ north America แต่ในทางกลับกัน asia เอเชียมีประชากรชายมากกว่าประชากรหญิง

Gender ratio

population



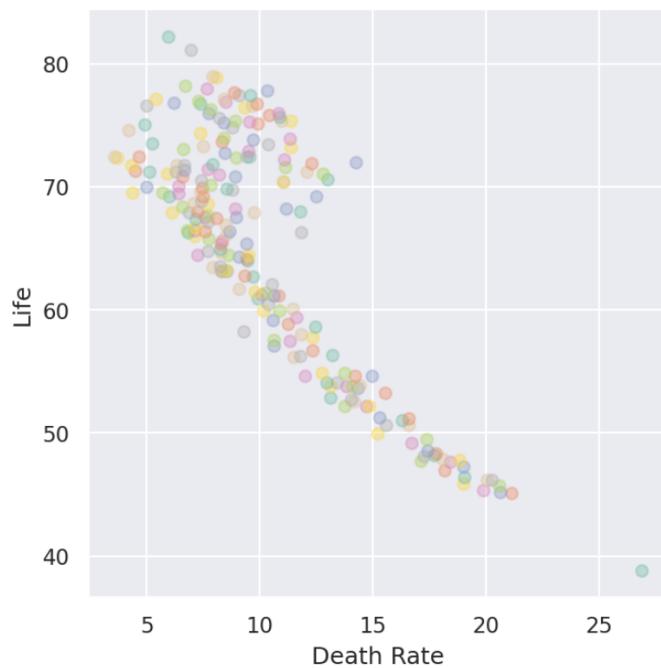
Proportion Gender by Region



ความสัมพันธ์ของข้อมูล (Correlation)

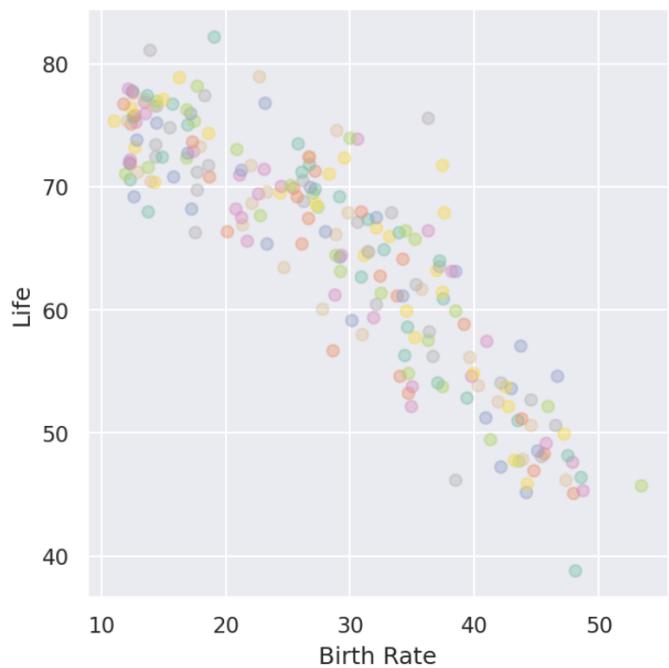
ความสัมพันธ์ของอัตราการตาย VS อายุเฉลี่ย รายประเทศ

Death Rate VS Life



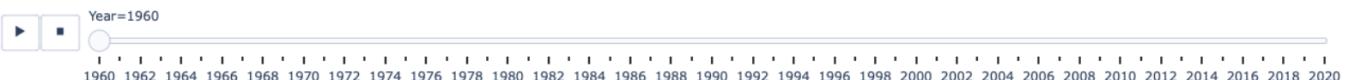
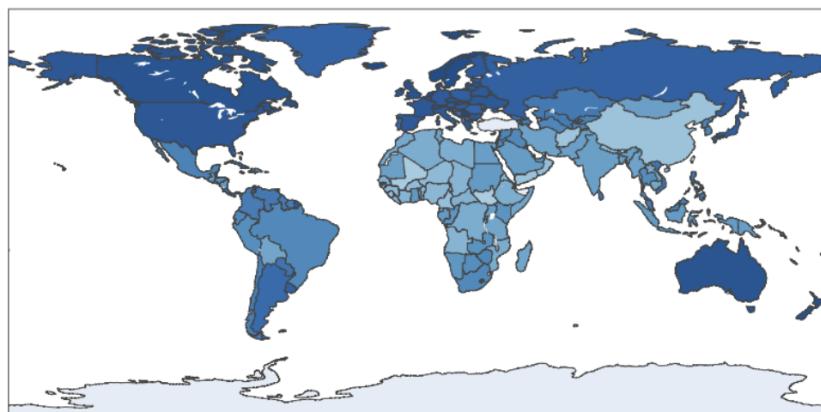
ความสัมพันธ์ของอัตราการเกิด VS อายุเฉลี่ย รายประเทศ

Birth Rate VS Life



ประชากรย้อนหลัง60 รายประเทศ

Compare Population Country (Age)



Q&A

- **Q : .ແວໂນ້ນສະານກາຣນີປະຊາກຣໂລກໃນປັຈຈຸບັນ ແລະອາຄາຕ ?**
 - A : ລດລອຍ່າງມືນຍໍາຄັນຈາກສະານແວດໍລ້ວນຫລາຍໆອຍ່າງແລະໃນ10ປີຂ້າງໜ້າກີຍັງລດຕໍ່ວ່າຍ່າງຕ່ອງເນື່ອງເໜືອງເໜືອງ 0.92%
- **Q : .ອາຍຸເຈລ່ຽນປະຊາກຣໂລກ ?**
 - A : 64 ປີ ຈາກຂ້ອມູລ 60 ປີຢ້ອນຫລັງ
- **Q : .ສັດສວນປະຊາກຮາຍຕ່ອປະຊາກຮ່າງ?**
 - A : 50.2% ຕ່ອ 49.8 %
- **Q : .ສັດສວນປະຊາກຮາຍຕ່ອປະຊາກຮ່າງ?**
 - A : 50.2% ຕ່ອ 49.8 %
- **Q : ແວໂນ້ນກາຣເກີດຂອງໂລກ**
 - A : 28.4 ຄນ ຕ່ອ ປະຊາກ 1000 ຄນ
- **Q : ແວໂນ້ນກາຣຕາຍຂອງໂລກ**
 - A : 10.6 ຄນ ຕ່ອ ປະຊາກ 1000 ຄນ

Challenge :

- 1 . ກາຣ unpivot data ເພຣະ ປກຕິdataທີ່ກຳຈານສ່ວນໃໝ່ເປັນ sales transaction ຈະມີ values ແລະ ຊ່ວຍເລີ່ມຕົ້ນຕໍ່ວ່າຍ່າງຕໍ່ວ່າມີການ 40% ຂອງກາຣກົດເປັນ
- 2 . ກາຣກົດເປັນຕາມທີ່ເຮົາຕ້ອງກາຣ ຄ່ວມຂ້າງຍາກທີ່ຈະກຳchart ອອກມາໃຫ້ຕຽນໃຈ ເຊັ່ນກາຣກຳຫຼັດ labels ສັງລະອຸງການ ແລະ Front ແລະໃຫ້ເວລາສັກເຫັນ ໃນກາຣຈະgenerate chart ແລະ step ກາຣເຕັມdata frameໃນກາຣກົດເປັນ

ໜ້າທີ່ຄວາມຮັບຜົດຜວບ

ກັກຮກ : ຮັບຜົດຜວບຮັວໜ້ວ EDA 2.1 - 2.8 ແລະ 2.10- 2.11 insight : 4.1 - 4.3

ວົກສາ : ຮັບຜົດຜວບ Collect data 1.1 -1.9 EDA 2.8. -2.9 ແລະ 2.12 - 2.13 insight 4.4 - 4.5

Link Github : <https://github.com/o-joe-v/World-Population.git>