

Unraveling the Challenges of Unemployment in Africa

By Nwokebu Goodness Chizorom

General Overview

Unemployment remains one of the most daunting challenges facing African nations today. It is a multifaceted problem with deep roots in socio-economic, educational, and policy-related factors. Analysing this case study would uncover insights and strategies that could assist in mitigating the unemployment crisis in Africa.

```
In [1]: #importing important libraries
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
import numpy as np
import squarify
import plotly.graph_objects as go
```

```
In [2]: #getting the data
df1 = pd.read_csv("1. unemployment-rate-men-vs-women.csv")
df2 = pd.read_csv("2. national-strategy-for-youth-employment.csv")
df3 = pd.read_csv("3. share-of-education-in-government-expenditure.csv")
df4 = pd.read_csv("4. share-of-the-population-with-access-to-electricity.csv")
df5 = pd.read_excel("5. Total_firms_Historical_data.xlsx")
df6 = pd.read_csv("6. Country Codes.csv")
```

- I did a majority of the cleaning using Power Query in excel. It was easier to holistically look at the data in excel.

```
In [3]: #analysing data 1
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6231 entries, 0 to 6230
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Entity            6231 non-null   object  
 1   Code              5828 non-null   object  
 2   Year              6231 non-null   int64  
 3   Unemployed female (%) 6231 non-null   float64
 4   Unemployed male (%) 6231 non-null   float64
 5   Population (historical estimates) 5952 non-null   float64
 6   Continent          187 non-null    object  
 7   Income             6231 non-null   object  
dtypes: float64(3), int64(1), object(4)
memory usage: 389.6+ KB
```

```
In [4]: Ab = df1[df1["Population (historical estimates)"].isna()] #found out that it was channel island and sub-saharan Africa have
Ab["Entity"].unique()
```

```
Out[4]: array(['Channel Islands', 'East Asia and Pacific (WB)', 'Europe and Central Asia (WB)', 'Latin America and Caribbean (WB)', 'Middle East and North Africa (WB)', 'Middle-income countries', 'North America (WB)', 'South Asia (WB)', 'Sub-Saharan Africa (WB)'],
   dtype=object)
```

Most of the column here had no null set except for population and continent. For population but it was not for Entity(Country) but a collection of some countries. I deleted the Continent cos most of it was empty

```
In [5]: df1["Unemployed Persons"] = df1["Unemployed female (%)"] + df1["Unemployed male (%)]#containing total unemployed persons
```

```
In [6]: #extracting the countries segregated by incomes
df1_income = df1[df1["Income"]=="Yes"]
df1_income
```

Out[6]:

	Entity	Code	Year	Unemployed female (%)	Unemployed male (%)	Population (historical estimates)	Continent	Income	Unemployed Persons
2356	High-income countries	NaN	1991	7.385763	6.296432	1.007966e+09	NaN	Yes	13.682195
2357	High-income countries	NaN	1992	7.874202	6.909115	1.016302e+09	NaN	Yes	14.783318
2358	High-income countries	NaN	1993	8.370617	7.453398	1.024111e+09	NaN	Yes	15.824015
2359	High-income countries	NaN	1994	8.481362	7.370254	1.031536e+09	NaN	Yes	15.851616
2360	High-income countries	NaN	1995	8.185502	6.899377	1.038644e+09	NaN	Yes	15.084879
...
5947	Upper-middle-income countries	NaN	2017	5.887670	5.972475	2.500617e+09	NaN	Yes	11.860145
5948	Upper-middle-income countries	NaN	2018	5.746158	5.803769	2.516786e+09	NaN	Yes	11.549926
5949	Upper-middle-income countries	NaN	2019	5.955110	6.046194	2.530546e+09	NaN	Yes	12.001304
5950	Upper-middle-income countries	NaN	2020	6.615547	6.869691	2.541464e+09	NaN	Yes	13.485238
5951	Upper-middle-income countries	NaN	2021	6.858722	6.683779	2.548718e+09	NaN	Yes	13.542501

155 rows × 9 columns

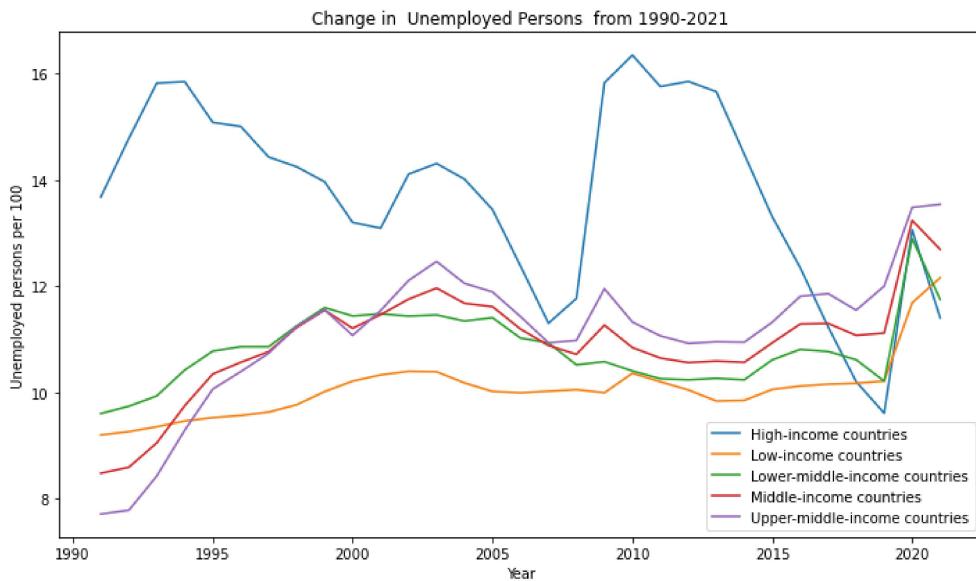
In [7]:

```
# Group data by Country and Year and calculate the mean unemployed person for each group
grouped_data = df1_income.groupby(['Year','Entity'])[['Unemployed Persons']].mean().unstack()

# Plotting
fig, ax = plt.subplots(figsize=(10, 6))

for Entity in grouped_data.columns:
    ax.plot( grouped_data.index, grouped_data[Entity], label=Entity)

ax.set_xlabel('Year')
ax.set_ylabel('Unemployed persons per 100')
ax.set_title('Change in Unemployed Persons from 1990-2021')
ax.legend()
plt.xticks()
plt.tight_layout()
plt.show()
```



In this graph, there seems to be a correlation between income countries and unemployment, evident in the juxtaposition of high income and low income. Furthermore, the year 2019 witnessed a historic milestone as high-income nations achieved their lowest recorded levels of unemployment.

```
In [8]: df6_processed = df6.drop(["alpha-2","alpha-3","country-code"],axis = 1)
df1_merge = df1.merge(df6_processed, on = "Entity", how = "left")
```

```
In [9]: #bargraph
df1_merge_processed = df1_merge.groupby('region').agg({'Unemployed female (%)': 'mean', 'Unemployed male (%)': 'mean'}).sort_values("Unemployed male (%)", ascending=False)
df1_merge_processed1 = df1_merge.groupby('region').agg({'Unemployed Persons': 'mean'}).sort_values("Unemployed Persons", ascending=False)
```

```
Out[9]:
```

	region	Unemployed Persons
0	Oceania	12.467375
1	Asia	12.529939
2	Americas	18.299768
3	Europe	20.639765
4	Africa	20.737999

```
In [10]: colors = ['grey', 'grey', 'grey', 'green', 'green', ]
plt.figure(figsize=(12, 5))
bars = plt.barh(df1_merge_processed1['region'], df1_merge_processed1['Unemployed Persons'], color=colors, alpha=0.7)
for bar in bars:
    width = bar.get_width()
    plt.text(width, bar.get_y() + bar.get_height()/2, f'{width:.2f}', ha='center', va='center', weight='bold')

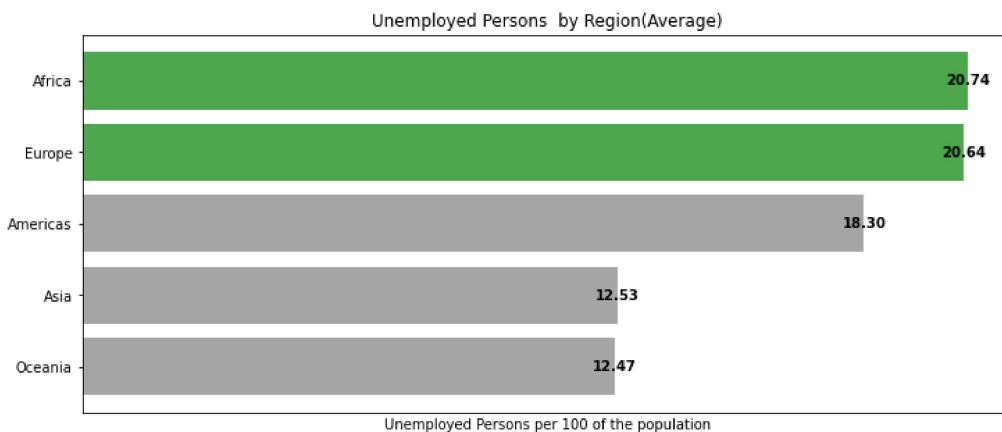
plt.gca().set_xticklabels([]) # Remove y-axis labels
plt.xticks([])

plt.xlabel('Unemployed Persons per 100 of the population')
plt.title('Unemployed Persons by Region(Average)')

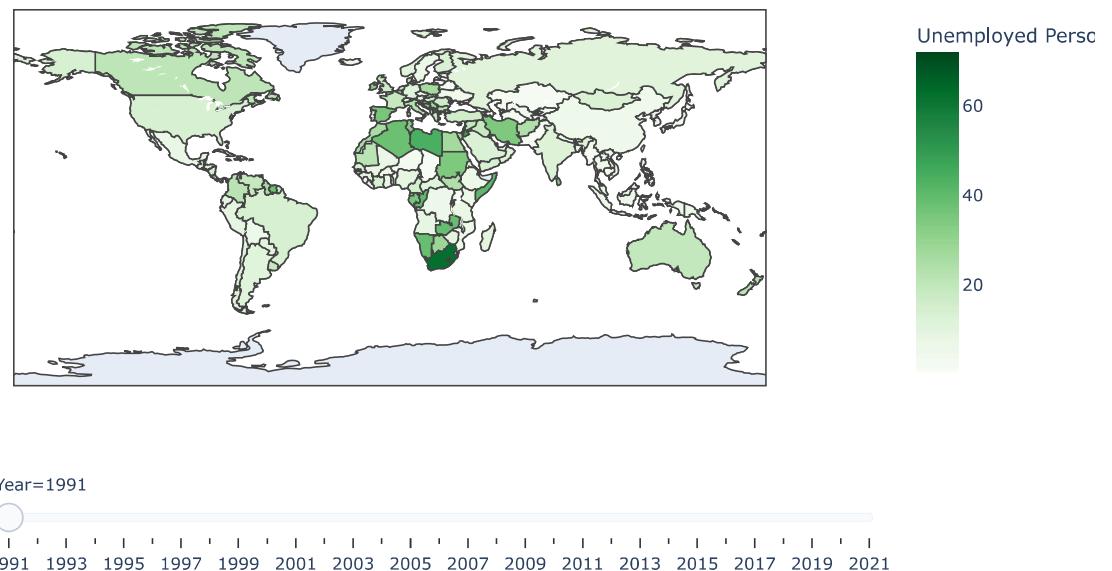
plt.show()

fig = px.choropleth(df1,
                     locations='Entity',
                     locationmode='country names',
                     color='Unemployed Persons',
                     hover_name='Entity',
                     color_continuous_scale='Greens',
                     title='A map of Unemployed Persons per 100 of the country',
                     animation_frame = "Year"
                    )

fig.show()
```



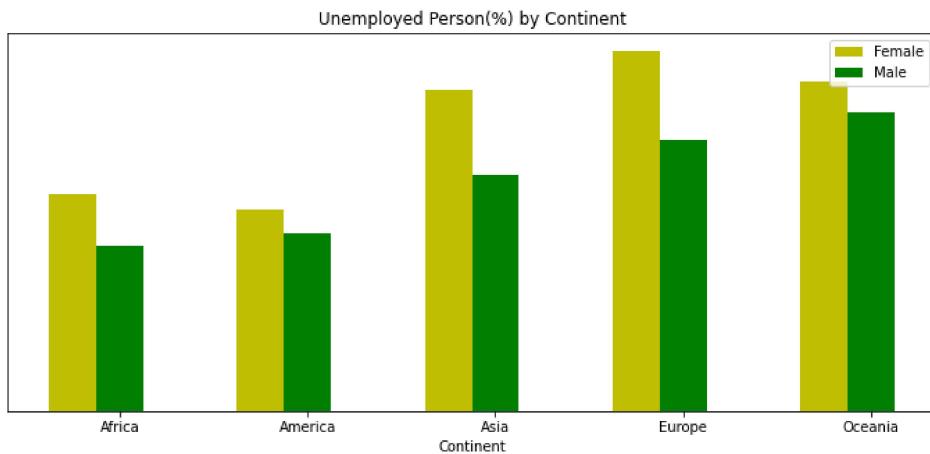
A map of Unemployed Persons per 100 of the country



- Undoubtedly, Africa has the highest number of unemployed individuals, presenting a significant and pressing challenge for the continent. The next continent to look out for is Europe.

In [11]: `#plotting the graph`

```
ind = np.arange(len(df1_merge_processed["region"]))
Width = 0.25
plt.figure(figsize=(12, 5))#setting the figure
plt.bar(ind,df1_merge_processed["Unemployed female (%)"],width = Width, color = 'y', label = "Female")
plt.bar(ind+Width,df1_merge_processed["Unemployed male (%)"],width = Width, color = 'g', label = "Male")
plt.xlabel("Continent")
plt.ylabel("Unemployed persons per 100 of the Population")
plt.title("Unemployed Person(%) by Continent")
plt.xticks(ind+Width,['Africa','America','Asia','Europe','Oceania'])#limitation was not shared into the diff 2 america contin
plt.legend(loc='upper right')
plt.gca().axes.get_yaxis().set_visible(False)
plt.show()
```

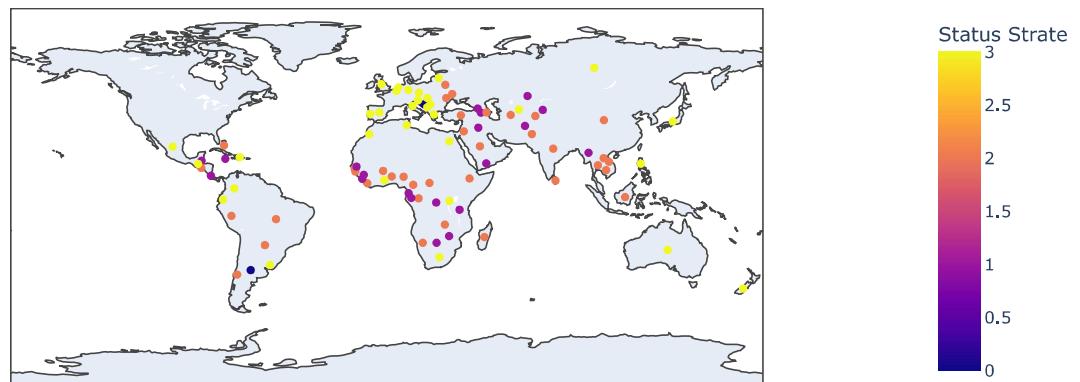


- On an average Females are more employed than males

In [12]:

```
fig = px.scatter_geo(df2, locations='Entity', locationmode="country names", color="Status Strategy",
                     hover_name="Entity", projection="equirectangular",
                     size_max = 7,animation_frame = "Year", title='A map of the different national strategy over time')
fig.show()
```

A map of the different national strategy over time



- Asia and Oceania had the lowest unemployed percentage and a good number of the countries in the continent have gone beyond adopting to using

```
In [13]: df1.Africa = df1_merge[df1_merge["region"]=="Africa"];
```

C:\Users\User\AppData\Local\Temp\ipykernel_3388\1319210366.py:1: UserWarning:

Pandas doesn't allow columns to be created via a new attribute name - see <https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access> (<https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access>)

```
In [14]:
```

```
# Grouping data by 'Entity' (country) and summing the Unemployed Persons for each country for each year
df1.Africa_data = df1.Africa.groupby(['Entity', 'Year'])['Unemployed Persons'].sum().reset_index()

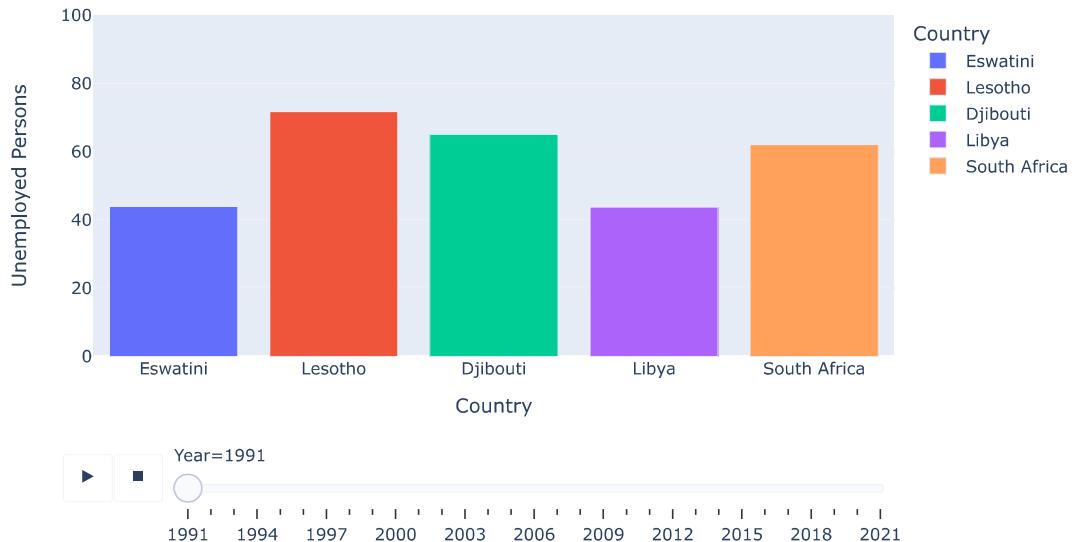
df1.Africa_data = df1.Africa_data.groupby('Year').apply(lambda x: x.nlargest(5, 'Unemployed Persons')).sort_values('Unemployed Persons', ascending=False)
df1.Africa_data = df1.Africa_data.sort_values('Year')
# Creating an animated bar plot with Plotly Express
fig = px.bar(df1.Africa_data, x='Entity', y='Unemployed Persons', color='Entity',
              animation_frame='Year',
              labels={'Entity': 'Country', 'Unemployed Persons': 'Unemployed Persons', 'Year': 'Year'},
              title='Top five Countries with Unemployed Person per year', width=800, height=500)

fig.update_layout(xaxis_title='Country')
fig.update_yaxes(range=[0, 100])
fig.show()
```

C:\Users\User\AppData\Local\Temp\ipykernel_3388\871158467.py:2: UserWarning:

Pandas doesn't allow columns to be created via a new attribute name - see <https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access> (<https://pandas.pydata.org/pandas-docs/stable/indexing.html#attribute-access>)

Top five Countries with Unemployed Person per year



- Eswatini, Lesotho, Djibouti, South Africa and Algeria are the five countries that have the highest unemployment rate from 1991 - 2021
-

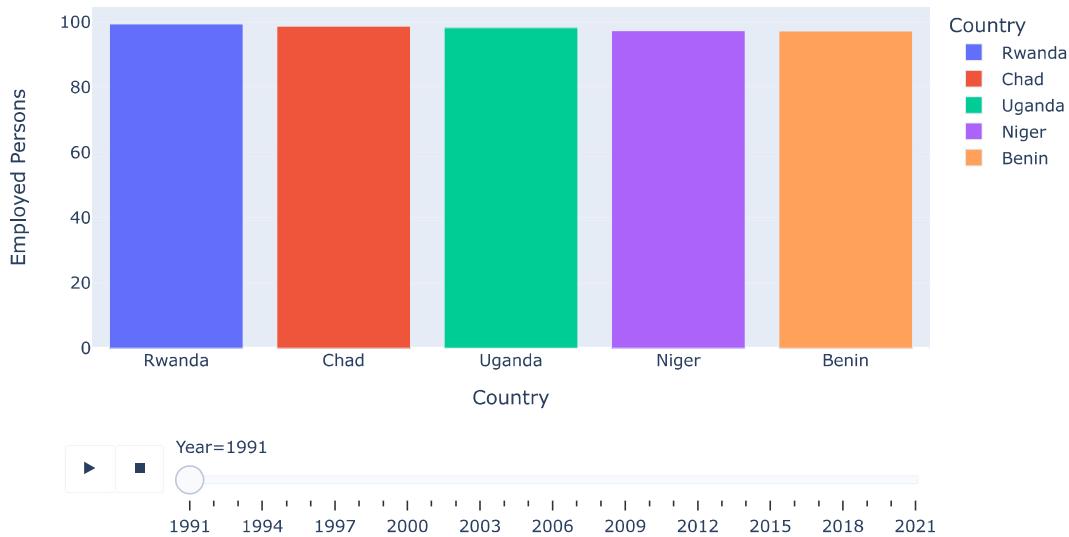
```
In [15]: # Grouping data by 'Entity' (country) and summing the Unemployed Persons for each country for each year
df1.Africa_data = df1.Africa.groupby(['Entity', 'Year'])[['Unemployed Persons']].sum().reset_index()
df1.Africa_data[['Employed Persons']] = 100 - df1.Africa_data[['Unemployed Persons']]

df1.Africa_data = df1.Africa_data.groupby('Year').apply(lambda x: x.nlargest(5, 'Employed Persons')).reset_index(drop=True)

# Creating an animated bar plot with Plotly Express
fig = px.bar(df1.Africa_data, x='Entity', y='Employed Persons', color='Entity',
              animation_frame= 'Year',
              labels={'Entity': 'Country', 'Unemployed Persons': 'Unemployed Persons', 'Year': 'Year'},
              title='Top 3 Countries with greater percentage of employed Person over Years',
              width=800, height=500)

fig.update_layout(xaxis_title='Country')
fig.show()
```

Top 3 Countries with greater percentage of employed Person over Years



- Rwanda, Chad, Uganda, Niger, Benin and Burundi and Liberia are the top countries in africa with high percent of employed person with not less than 95%.
- Rwanda and Chad have consistently as one of the top for the 20 years analysed
-

```
In [16]: df2_merge = df2.merge(df6_processed, on = "Entity", how = "left")
df2_merge
```

Out[16]:

	Entity	Code	Year	Status	Strategy	region	sub-region
0	Afghanistan	AFG	2019			1	Asia
1	Afghanistan	AFG	2020			1	Asia
2	Argentina	ARG	2019			0	Americas
3	Argentina	ARG	2020			0	Americas
4	Armenia	ARM	2019			1	Asia
...
204	Yemen	YEM	2020			1	Asia
205	Zambia	ZMB	2019			2	Africa
206	Zambia	ZMB	2020			2	Africa
207	Zimbabwe	ZWE	2019			1	Africa
208	Zimbabwe	ZWE	2020			1	Africa

209 rows × 6 columns

#limitation. The were less data for recent years

In [17]:

```
df3_merge = df3.merge(df6, on = "Entity", how = "left")
df3_merge_Africa = df3_merge[df3_merge["region"] == "Africa"]
df3_merge_processed = df3_merge_Africa.groupby('Entity')[["Education expenditure(%)"]].mean().reset_index()
df3_merge_processed1 = df3_merge_processed.sort_values("Education expenditure(%)", ascending = False).head()
```

In [18]:

```
df3_merge_Asia = df3_merge[df3_merge["region"] == "Asia"]
df3_merge_processed = df3_merge_Asia.groupby('Entity')[["Education expenditure(%)"]].mean().reset_index()
df3_merge_processed4 = df3_merge_processed.sort_values("Education expenditure(%)", ascending = False).head()
```

```
In [19]: plt.figure(figsize=(10, 6))
#renaming some countries to match those in df2
plt.bar(df3_merge_processed1['Entity'], df3_merge_processed1['Education expenditure(% )'], color="green", alpha=0.7)

for i, (entity, EE) in enumerate(zip(df3_merge_processed1['Entity'], df3_merge_processed1['Education expenditure(% )'])):
    plt.text(entity, EE, f'{EE:.2f}', ha='center', va='bottom', weight='bold')

# Set Labels and title

plt.gca().set_yticklabels([])# Hide y-axis label
plt.xlabel('Country') # Adding x-axis label
plt.title('Top countries with high education')

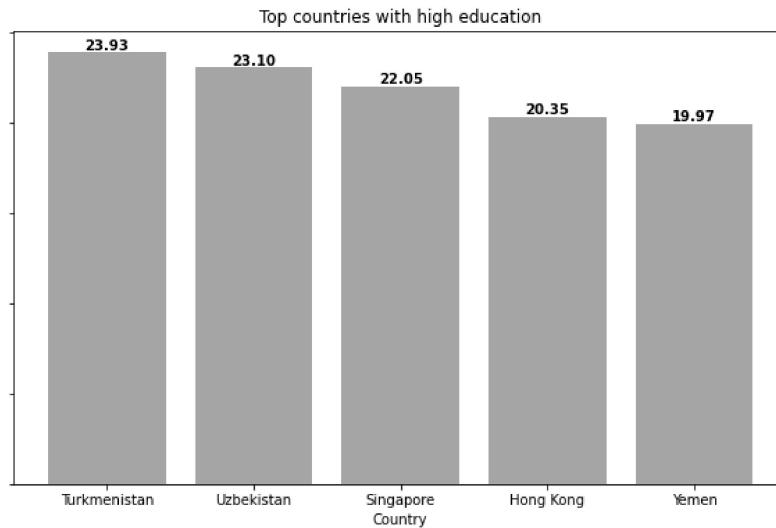
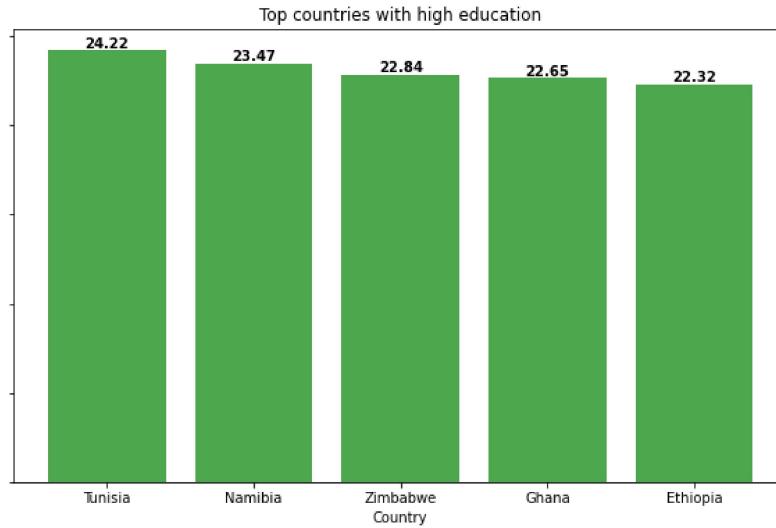
plt.show()
plt.figure(figsize=(10, 6))
#renaming some countries to match those in df2
plt.bar(df3_merge_processed4['Entity'], df3_merge_processed4['Education expenditure(% )'], color="grey", alpha=0.7)

for i, (entity, EE) in enumerate(zip(df3_merge_processed4['Entity'], df3_merge_processed4['Education expenditure(% )'])):
    plt.text(entity, EE, f'{EE:.2f}', ha='center', va='bottom', weight='bold')

# Set Labels and title

plt.gca().set_yticklabels([])# Hide y-axis label
plt.xlabel('Country') # Adding x-axis label
plt.title('Top countries with high education')

plt.show()
```



- In Africa has the Tunisia highest total government expenditure that is allocated to education

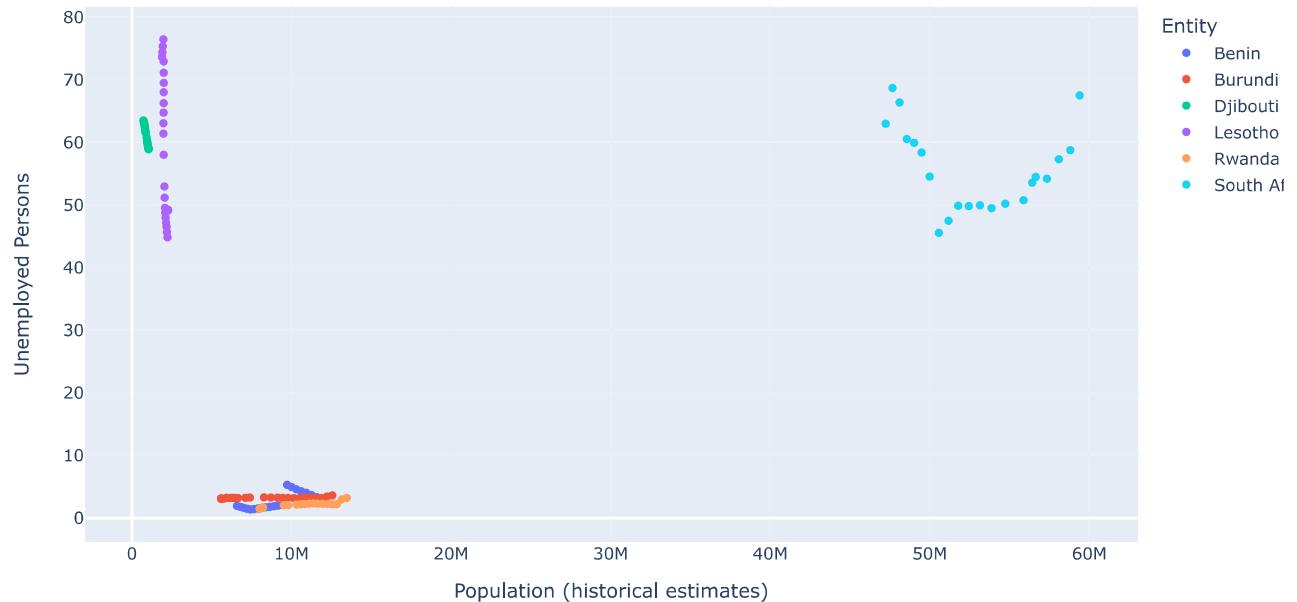
```
In [20]: df1_merge_Africa = df1_merge[df1_merge["region"] == "Africa"]
df1_merge_Africa_processed = df1_merge_Africa[["Entity", "Unemployed Persons", "Population (historical estimates)", "Year", "region"]]
df1_merge_Africa_processed
```

Out[20]:

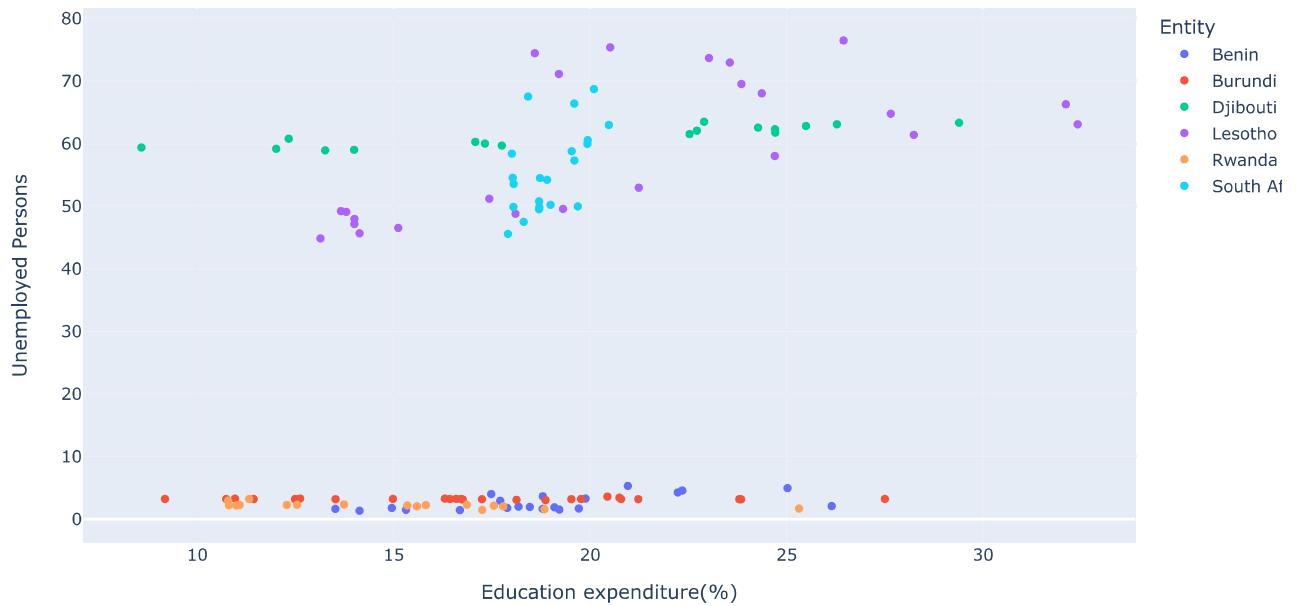
	Entity	Unemployed Persons	Population (historical estimates)	Year	region
62	Algeria	37.755	26133906.0	1991	Africa
63	Algeria	45.624	26748302.0	1992	Africa
64	Algeria	49.957	27354326.0	1993	Africa
65	Algeria	53.752	27937006.0	1994	Africa
66	Algeria	62.741	28478024.0	1995	Africa
...
6226	Zimbabwe	9.545	14751101.0	2017	Africa
6227	Zimbabwe	9.566	15052191.0	2018	Africa
6228	Zimbabwe	9.641	15354606.0	2019	Africa
6229	Zimbabwe	10.681	15669663.0	2020	Africa
6230	Zimbabwe	10.324	15993525.0	2021	Africa

1519 rows × 5 columns

```
In [21]: df_scatter = df3_merge_Africa.merge(df1_merge_Africa_processed, on = ["Entity", "Year"], how = 'left')
df_scatter1 = df_scatter[df_scatter["Entity"].isin(["Benin", "Burundi", "Rwanda", "Lesotho", "South Africa", "Djibouti"])]
fig = px.scatter(df_scatter1, x="Population (historical estimates)", y="Unemployed Persons", color = "Entity", hover_name="Entity")
fig.show()
```



```
In [22]: df_scatter = df3_merge_Africa.merge(df1_merge_Africa_processed, on = ["Entity", "Year"], how = 'left')
df_scatter1 = df_scatter[df_scatter["Entity"].isin(["Benin", "Burundi", "Rwanda", "Lesotho", "South Africa", "Djibouti"])]
fig = px.scatter(df_scatter1, x="Education expenditure(%)", y="Unemployed Persons", color = "Entity", hover_name="Entity")
fig.show()
```



```
In [23]: df1_merge_processed2 = df1_merge[["Entity", "Unemployed Persons", "Year", "region"]]
```

```
In [24]: df5_merge = df5.merge(df1_merge_processed2, on = ["Entity", "Year"], how = "left")
```

```
In [25]: df5_merge_Africa = df5_merge[df5_merge["region"] == "Africa"]
df5_merge_Africa
```

Out[25]:

	Entity	Adult population	Year	TOTAL Number of \nLimited Liability Companies	Total business density rate	OFC	BIN LLC	Unemployed Persons	region
25	Algeria	22354242	2006	45055	2.015501	No	1	26.921	Africa
26	Algeria	22845303	2007	51909	2.272196	No	1	31.814	Africa
27	Algeria	23315424	2008	53825	2.308558	No	1	27.711	Africa
28	Algeria	23766657	2009	59346	2.497028	No	1	26.659	Africa
29	Algeria	24198798	2010	61674	2.548639	No	1	27.189	Africa
...
1517	Zimbabwe	7642765	2016	158627	20.755185	No	1	9.549	Africa
1518	Zimbabwe	7759792	2017	168145	21.668751	No	1	9.545	Africa
1519	Zimbabwe	7892169	2018	184955	23.435256	No	1	9.566	Africa
1520	Zimbabwe	8034692	2019	209983	26.134543	No	1	9.641	Africa
1521	Zimbabwe	8185716	2020	230256	28.128999	No	1	10.681	Africa

193 rows × 9 columns

```
In [26]: df5_merge = df5_merge.dropna( )
```

```
In [27]: #removing outliers to show the colors clarity well
# Calculate the Lower and upper quartiles
Q1 = df5_merge['TOTAL Number of \nLimited Liability Companies'].quantile(0.25)
Q3 = df5_merge['TOTAL Number of \nLimited Liability Companies'].quantile(0.75)

# Calculate the IQR
IQR = Q3 - Q1

# Define the lower and upper bounds to filter outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter the DataFrame to remove outliers based on the IQR
filtered_df = df5_merge[(df5_merge["TOTAL Number of \nLimited Liability Companies"] >= lower_bound) & (df5_merge["TOTAL Number of \nLimited Liability Companies"] <= upper_bound)]
```

```
In [28]: filtered_df['Employed Persons'] = 100 - filtered_df['Unemployed Persons']
```

C:\Users\User\AppData\Local\Temp\ipykernel_3388\1840072258.py:1: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
In [29]: fig = px.scatter(filtered_df, x='TOTAL Number of \nLimited Liability Companies', y="Employed Persons",
                     size="Total business density rate", size_max=40, color="region", hover_name="Entity",
                     animation_frame="Year")
                     #range_y=[min(df5_merge_cleaned['Unemployed Persons']), max(df5_merge_cleaned['Unemployed Persons'])])

fig.show()
```



```
In [30]: Q1 = df5_merge['Adult population'].quantile(0.25)
Q3 = df5_merge['Adult population'].quantile(0.75)

# Calculate the IQR
IQR = Q3 - Q1

# Define the Lower and upper bounds to filter outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter the DataFrame to remove outliers based on the IQR
filtered_df = df5_merge[(df5_merge["Adult population"] >= lower_bound) & (df5_merge["Adult population"] <= upper_bound)]
```

```
In [31]: filtered_df['Employed Persons'] = 100 - filtered_df['Unemployed Persons']
```

C:\Users\User\AppData\Local\Temp\ipykernel_3388\1840072258.py:1: SettingWithCopyWarning:

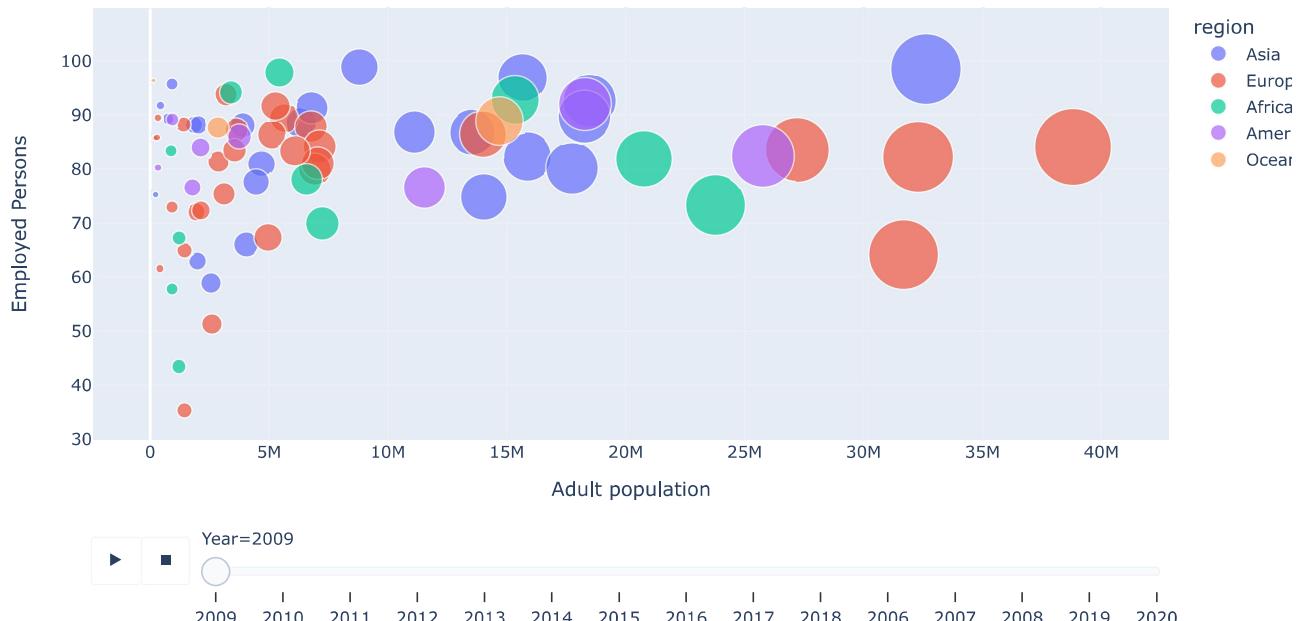
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
In [32]:
```

```
fig = px.scatter(filtered_df, x='Adult population', y="Employed Persons",
                  size="Adult population", size_max=40, color="region", hover_name="Entity",
                  animation_frame="Year")
#range_y=[min(df5_merge_cleaned['Unemployed Persons']), max(df5_merge_cleaned['Unemployed Persons'])]

fig.show()
```

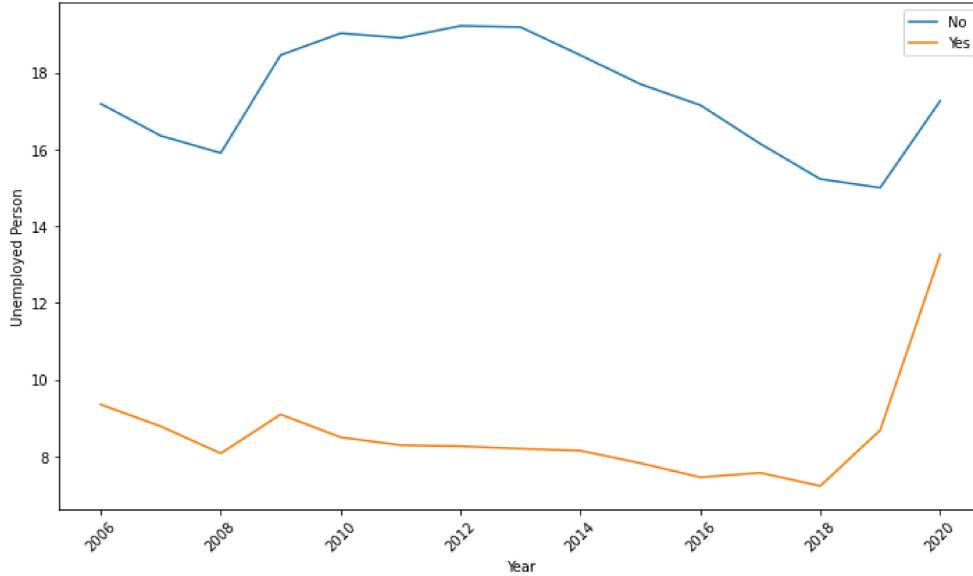


```
In [33]: grouped_data = df5_merge.groupby(['Year', 'OFC', 'Entity'])['Unemployed Persons'].mean().unstack()

# Plotting
fig, ax = plt.subplots(figsize=(10, 6))

for things in grouped_data.columns:
    ax.plot(grouped_data.index, grouped_data[things], label=things)

ax.set_xlabel('Year')
ax.set_ylabel('Unemployed Person')
#ax.set_title('Change in PR over Years by Continent')
ax.legend()
plt.xticks(rotation = 45) # Ensure all years are displayed on x-axis
plt.tight_layout()
plt.show()
```



OFC Column was extracted from Entity with Asterisks

```
In [34]: df4_merge = df4.merge(df6, on = "Entity", how = "left")
df4_merge_Africa = df4_merge[df4_merge["region"] == "Africa"]
df4_merge_processed = df4_merge_Africa.groupby('Entity')[["Access to electricity (% of population)"]].mean().reset_index()
df4_merge_processed1 = df4_merge_processed.sort_values("Access to electricity (% of population)", ascending = False).head()
```

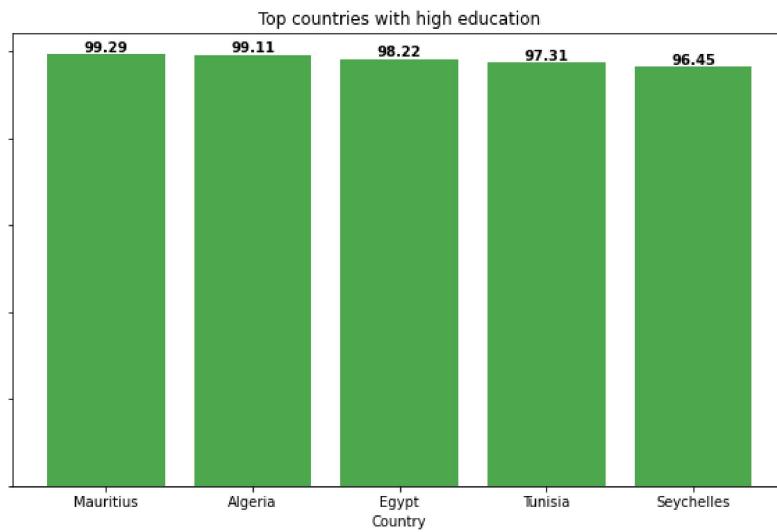
```
In [35]: plt.figure(figsize=(10, 6))
#renaming some countries to match those in df2
plt.bar(df4_merge_processed1['Entity'], df4_merge_processed1['Access to electricity (% of population)'], color="green", alpha=0.8)

for i, (entity, EE) in enumerate(zip(df4_merge_processed1['Entity'], df4_merge_processed1['Access to electricity (% of population)'])):
    plt.text(entity, EE, f'{EE:.2f}', ha='center', va='bottom', weight='bold')

# Set Labels and title

plt.gca().set_yticklabels([])# Hide y-axis label
plt.xlabel('Country') # Adding x-axis Label
plt.title('Top countries with high education')

plt.show()
```



In [36]:

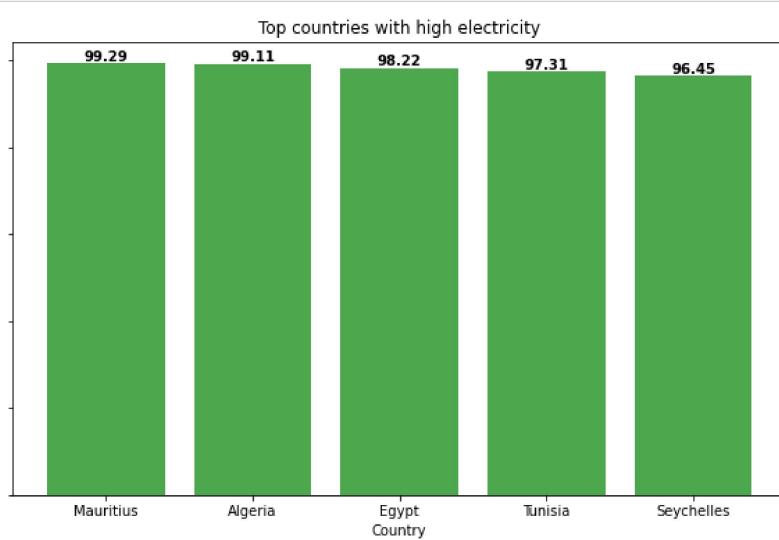
```
plt.figure(figsize=(10, 6))
#renaming some countries to match those in df2
plt.bar(df4_merge_processed1['Entity'], df4_merge_processed1['Access to electricity (% of population)'], color="green", alpha=0.8)

for i, (entity, EE) in enumerate(zip(df4_merge_processed1['Entity'], df4_merge_processed1['Access to electricity (% of population)'])):
    plt.text(entity, EE, f'{EE:.2f}', ha='center', va='bottom', weight='bold')

# Set Labels and title

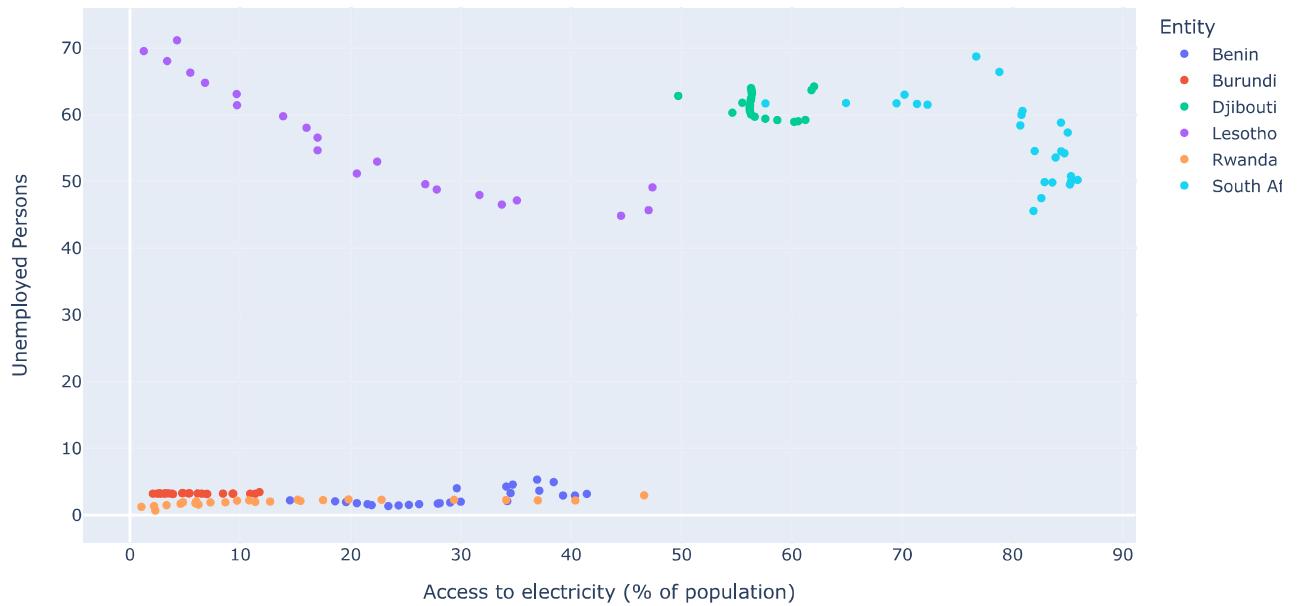
plt.gca().set_yticklabels([])# Hide y-axis Label
plt.xlabel('Country') # Adding x-axis Label
plt.title('Top countries with high education')

plt.show()
```



```
In [37]: df_scatter3 = df4_merge_Africa.merge(df1_merge_Africa_processed, on = ["Entity", "Year"], how = 'left')
df_scatter4 = df_scatter3[df_scatter3["Entity"].isin(["Benin", "Burundi", "Rwanda", "Lesotho", "South Africa", "Djibouti"])] 

fig = px.scatter(df_scatter4, x="Access to electricity (% of population)", y="Unemployed Persons", color = "Entity", hover_name="Entity")
fig.show()
```



RECOMMENDATIONS

- An offshore financial centre (OFC) is defined as a "country or jurisdiction that provides financial services to nonresidents on a scale that is incommensurate with the size and the financing of its domestic economy". Countries should adopt this
- More investment should be done to improve electricit especially with countries greater than 40 percent umemployed perrson
- An increase in LLC would assuredly increase employment rate
- Ensure that more firms are registered
- Africa should look into developing and operationalizing their strategy and not just adopting
- Start with Lesotho, South Africa and Dijoubi as the are the top countries with high unemployment rate

Limitations

- Most of the countries were missing like Canana in 5. Total_firms_Historical_data
- For "2.national-strategy-for-youth-employment dataset" most countries status strategy was not given

```
In [ ]:
```